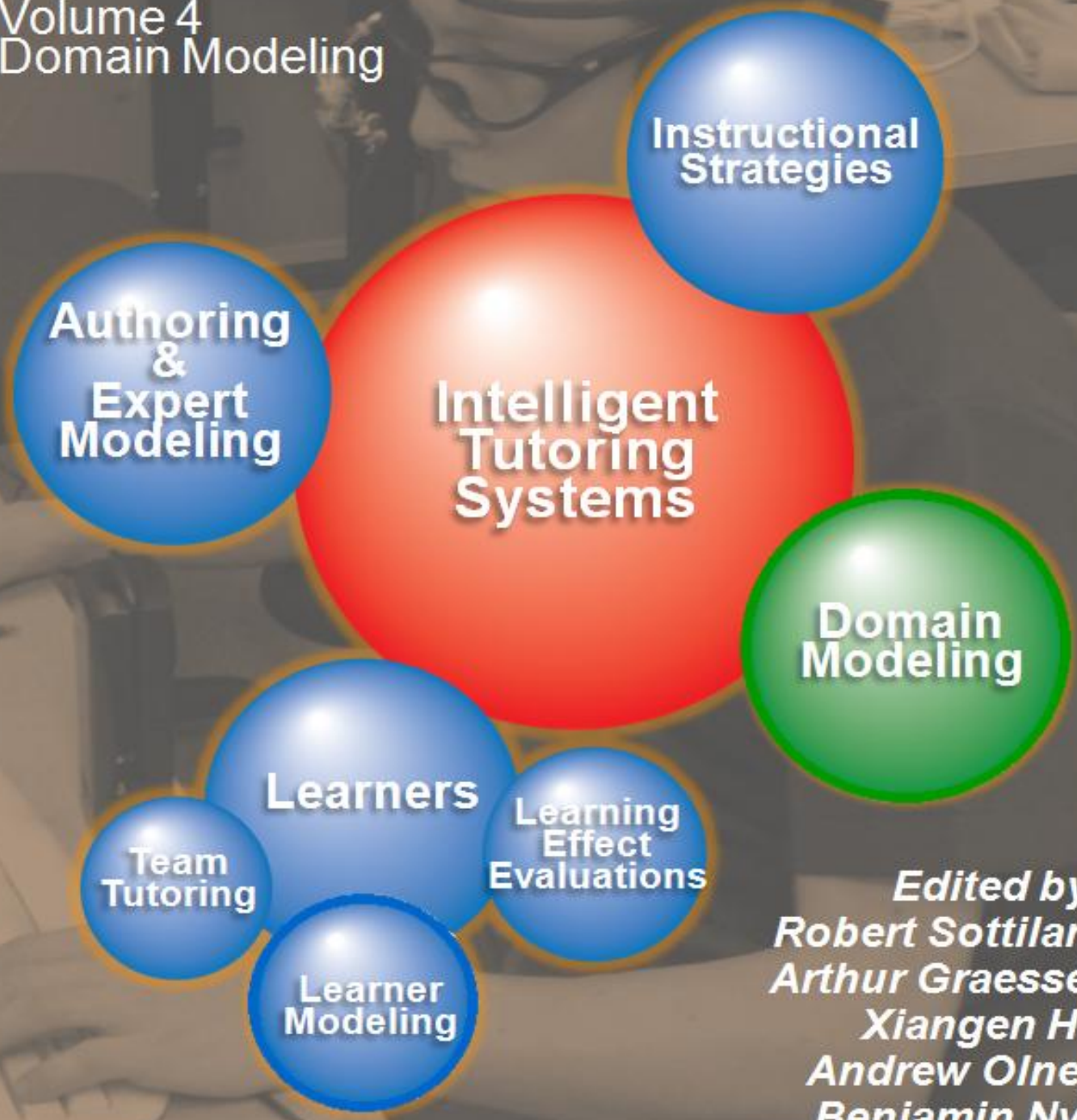
Design Recommendations for Intelligent Tutoring Systems

Volume 4
Domain Modeling



Edited by:
Robert Sottilare
Arthur Graesser
Xiangen Hu
Andrew Olney
Benjamin Nye
Anne Sinatra

A Book in the Adaptive Tutoring Series

Design Recommendations for Intelligent Tutoring Systems

Volume 4
Domain Modeling

Edited by:
Robert A. Sottilare
Arthur C. Graesser
Xiangen Hu
Andrew M. Olney
Benjamin D. Nye
Anne M. Sinatra

A Book in the Adaptive Tutoring Series

Printed in the United States of America
First Printing, July 2016

US Army Research Laboratory
Human Research & Engineering Directorate
Orlando, Florida

International Standard Book Number: 978-0-9893923-9-6

We wish to acknowledge the editing and formatting contributions of Carol Johnson and Deeja Cruz, ARL

Special thanks to Jody Cockroft, University of Memphis, for her efforts in coordinating the workshops that led to this volume.

Dedicated to current and future scientists and developers of adaptive learning technologies

CONTENTS

INTRODUCTION 1

Robert A. Sottilare, Arthur C. Graesser, Xiangen Hu, Andrew Olney, Benjamin D. Nye and Anne M. Sinatra

Section I: Fundamentals of Domain Modeling 13

CHAPTER 1 – Conceptualizing and Representing Domains to Guide Tutoring 15

Benjamin D. Nye and Xiangen Hu

CHAPTER 2 – Defining the Ill-Defined: From Abstract Principles to Applied Pedagogy 19

Benjamin D. Nye, Michael W. Boyce, and Robert A. Sottilare

CHAPTER 3 – Methods to Refine the Mapping of Items to Skills 39

Michel C. Desmarais and Peng Xu

CHAPTER 4 – Ontology Alignment of Learner Models and Domain Models 49

Douglas B. Lenat

CHAPTER 5 – Qualitative Representations for Education 57

Bert Bredeweg and Kenneth D. Forbus

CHAPTER 6 – A Work Practice Simulation Approach to Modeling Socio-Technical Domains 69

Benjamin Bell, William J. Clancey, and Winston Bennett, Jr.

Section II: Methods of Domain Modeling 91

CHAPTER 7 – Design and Construction of Domain Models 93

Andrew M. Olney and Arthur Graesser

CHAPTER 8 – Scaling Across Domains and the Implications for GIFT **97**
Keith W. Brawner, Gregory Goodwin, and Damon Regan

CHAPTER 9 – A Review of Self-Reference and Context Personalization in Different Computer-Based Educational Domains **107**
Anne M. Sinatra

CHAPTER 10 – Discovering Domain Models in Learning Curve Data **115**
Ilya Goldin, Philip I. Pavlik, Jr., and Steven Ritter

CHAPTER 11 – Making Static Lessons Adaptive through Crowdsourcing and Machine Learning **127**
Joseph Jay Williams, Juho Kim, Elena Glassman, Anna Rafferty, and Walter S. Lasecki

CHAPTER 12 – Data-Driven Domain Models for Problem Solving **137**
Tiffany Barnes, Behrooz Mostafavi, and Michael J. Eagle

CHAPTER 13 – Mining Expertise: Learning New Tricks from an Old Dog **147**
Brandt Dargue and Elizabeth Biddle

Section III: Applications of Domain Modeling **159**

CHAPTER 14 – Exploring the Diversity of Domain Modeling for Training and Educational Applications **161**
Anne M. Sinatra and Robert A. Sottilare

CHAPTER 15 – Domain Modeling for Personalized Guidance **165**
Peter Brusilovsky

CHAPTER 16 – A Process for Adaptive Instruction of Tasks in the Psychomotor Domain **185**
Robert A. Sottilare and Joseph LaViola

CHAPTER 17 – Domain Modeling in a Psychomotor World: A Marksmanship Use Case **195**
Benjamin Goldberg and Charles Amburn

CHAPTER 18 – Domain Modeling in AutoTutor **205**

Zhiqiang Cai, Arthur Graesser, and Xiangen Hu

CHAPTER 19 – Modeling Mathematical Reasoning as Trained Perception-Action Procedures **213**

Robert L. Goldstone, Erik Weitnauer, Erin R. Ottmar, Tyler Marghetis, and David H. Landy

CHAPTER 20 – Sketch Understanding for Education **225**

Kenneth D. Forbus

Biographies **237**

Introduction to Domain Modeling & GIFT

Robert A. Sottilare[1], Arthur C. Graesser[2], Xiangen Hu[2], Andrew Olney[2], Benjamin Nye[3], and Anne M. Sinatra[1] , Eds.

U.S. Army Research Laboratory - Human Research and Engineering Directorate[1]
University of Memphis Institute for Intelligent Systems[2]
University of Southern California Institute for Creative Technologies[3]

This book is the fourth in a planned series of books that examine key topics (e.g., learner modeling, instructional strategies, authoring, domain modeling, assessment, impact on learning, and team tutoring) in intelligent tutoring system (ITS) design through the lens of the Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare, Brawner, Goldberg & Holden, 2012; Sottilare, Holden, Goldberg & Brawner, 2013). GIFT is a modular, service-oriented architecture created to reduce the cost and skill required to author ITSs, manage instruction within ITSs, and evaluate the effect of ITS technologies on learning, performance, retention, and transfer.

Along with this volume, the first three books in this series, Learner Modeling (ISBN 978-0-9893923-0-3), Instructional Management (ISBN 978-0-9893923-2-7), and Authoring Tools (ISBN 978-0-9893923-6-5) are freely available at www.GIFTtutoring.org and on Google Play.

This introduction begins with a description of tutoring functions, provides a glimpse of domain modeling best practices, and examines the motivation for standards in the design, authoring, instruction, and evaluation of ITS tools and methods. We introduce GIFT design principles and discuss how readers might use this book as a design tool. We begin by examining the major components of ITSs.

Components and Functions of Intelligent Tutoring Systems

It is generally accepted that an ITS has four major components (Elson-Cook, 1993; Nkambou, Mizoguchi & Bourdeau, 2010; Graesser, Conley & Olney, 2012; Psotka & Mutter, 2008; Sleeman & Brown, 1982; VanLehn, 2006; Woolf, 2009): the domain model, the student model, the tutoring model, and the user-interface model. GIFT similarly adopts this four-part distinction, but with slightly different corresponding labels (domain module, learner module, pedagogical module, and tutor-user interface) and the addition of the sensor module, which can be viewed as an expansion of the user interface.

(1) The **domain model** contains the set of skills, knowledge, and strategies/tactics of the topic being tutored. It normally contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit.

(2) The **learner model** consists of the cognitive, affective, motivational, and other psychological states that evolve during the course of learning. Since learner performance is primarily tracked in the domain model, the learner model is often viewed as an overlay (subset) of the domain model, which changes over the course of tutoring. For example, "knowledge tracing" tracks the learner's progress from problem to problem and builds a profile of strengths and weaknesses relative to the domain model (Anderson, Corbett, Koedinger & Pelletier, 1995). An ITS may also consider psychological states outside of the domain model that need to be considered as parameters to guide tutoring.

(3) The **tutor model** (also known as the pedagogical model or the instructional model) takes the domain and learner models as input and selects tutoring strategies, steps, and actions on what the tutor should do next in the exchange. In mixed-initiative systems, the learners may also take actions, ask questions, or request help (Aleven, McClaren, Roll & Koedinger, 2006; Rus & Graesser, 2009), but the ITS always needs to be ready to decide "what to do next" at any point and this is determined by a tutoring model that captures the researchers' pedagogical theories.

(4) The **user interface** interprets the learner's contributions through various input media (speech, typing, clicking) and produces output in different media (text, diagrams, animations, agents). In addition to the conventional human-computer interface features, some recent systems have incorporated natural language interaction (Graesser et al., 2012; Johnson & Valente, 2008), speech

recognition (D'Mello, Graesser & King, 2010; Litman, 2013), and the sensing of learner emotions (Baker, D'Mello, Rodrigo & Graesser, 2010; D'Mello & Graesser, 2010; Goldberg, Sottilare, Brawner, Holden, 2011).

The designers of a tutor model must make decisions on each of the various major components in order to create an enhanced learning experience through well-grounded pedagogical strategies (optimal plans for action by the tutor) that are selected based on learner states and traits and that are delivered to the learner as instructional tactics (optimal actions by the tutor). Next, tactics are chosen based on the previously selected strategies and instructional context (the conditions of the training at the time of the instructional decision). This is part of the learning effect model (Sottilare, 2012; Fletcher & Sottilare, 2013; Sottilare, 2013; Sottilare, Ragusa, Hoffman & Goldberg, 2013), which has been updated and described below in more detail in the section titled "Motivations for Intelligent Tutoring System Standards" in this introductory chapter.

Principles of Learning and Instructional Techniques, Strategies, and Tactics

Instructional techniques, strategies, and tactics play a central role in the design of GIFT. Instructional techniques represent instructional best practices and principles from the literature, many of which have yet to be implemented within GIFT at the writing of this volume. Examples of instructional techniques include, but are not limited to, error-sensitive feedback, mastery learning, adaptive spacing and repetition, and fading worked examples. Others are represented in the next section of this introduction. It is anticipated that techniques within GIFT will be implemented as software-based agents where the agent will monitor learner progress and instructional context to determine if best practices (agent policies) have been adhered to or violated. Over time, the agent will learn to enforce agent policies in a manner that optimizes learning and performance.

Some of the best instructional practices (techniques) have yet to be implemented in GIFT, but many instructional strategies and tactics have been implemented. Instructional strategies (plans for action by the tutor) are selected based on changes to the learner's state (cognitive, affective, physical). If a sufficient change in any learner's state occurs, this triggers GIFT to select a generic strategy (e.g., provide feedback). The instructional context along with the instructional strategy then triggers the specific selection of an instructional tactic (an action to be taken by the tutor). If the strategy is to "provide feedback," then the tactic might be to "provide feedback on the error committed during the presentation of instructional concept 'B' in the chat window during the next turn." Tactics detail what is to be done, why, when, and how.

An adaptive, intelligent learning environment needs to select the right instructional strategies at the right time, based on its model of the learner in specific conditions and the learning process in general. Such selections should be taken to maximize deep learning and motivation while minimizing training time and costs.

Motivations for Intelligent Tutoring System Standards

An emphasis on self-regulated learning has highlighted a requirement for point-of-need training in environments where human tutors are either unavailable or impractical. ITSs have been shown to be as effective as expert human tutors (VanLehn, 2011) in one-to-one tutoring in well-defined domains (e.g., mathematics or physics) and significantly better than traditional classroom training environments. ITSs have demonstrated significant promise, but 50 years of research have been unsuccessful in making ITSs ubiquitous in military training or the tool of choice in our educational system. This begs the question: "Why?"

Part of the answer lies in the fact that the availability and use of ITSs have been constrained by their high development costs, their limited reuse, a lack of standards, and their inadequate adaptability to the needs of learners. Educational and training technologies like ITSs are primarily researched and developed in a few key environments: industry, academia, and government including military domains. Each of these environments has its own challenges and design constraints. The application of ITSs to military domains is further hampered by the complex and often ill-defined environments in which the US military operates today. ITSs are often built as domain-specific, unique, one-of-a-kind, largely domain-dependent solutions focused on a single pedagogical strategy (e.g., model tracing or constraint-based approaches) when complex learning domains may require novel or hybrid approaches. Therefore, a modular ITS framework and standards are needed to enhance reuse, support authoring, optimize instructional strategies, and lower the cost and skillset needed for users to adopt ITS solutions for training and education. It was out of this need that the idea for GIFT arose.

GIFT has three primary functions: authoring, instructional management, and evaluation. First, it is a framework for authoring new ITS components, methods, strategies, and whole tutoring systems. Second, GIFT is an instructional manager that integrates selected instructional theory, principles, and strategies for use in ITSs. Finally, GIFT is an experimental testbed used to evaluate the effectiveness and impact of ITS components, tools, and methods. GIFT is based on a learner-centric approach with the goal of improving linkages in the updated adaptive tutoring learning effect model (Figure 1; Sottilare, 2012; Fletcher & Sottilare, 2013; Sottilare, 2013; Sottilare, Ragusa, Hoffman & Goldberg, 2013).

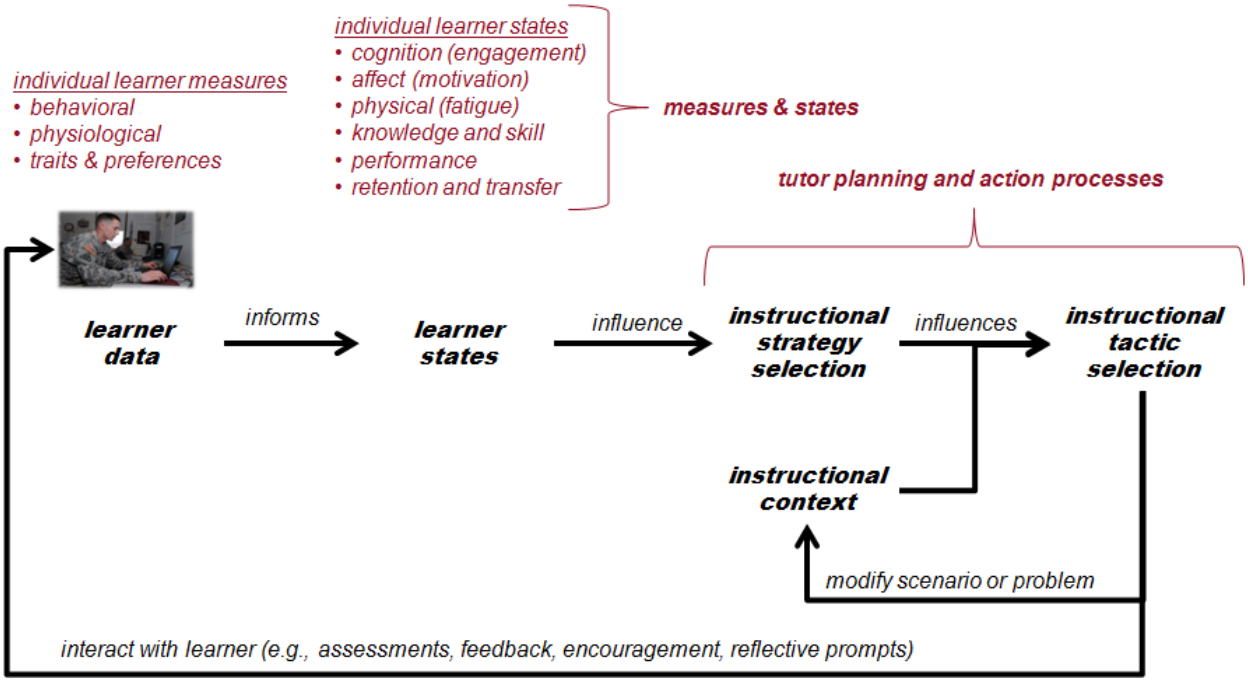


Figure 1. Updated adaptive tutoring learning effect model

A deeper understanding of the learner's behaviors, traits, and preferences (learner data) collected through performance, physiological and behavioral sensors, and surveys will allow for more accurate evaluation of the learner's states (e.g., engagement level, confusion, frustration). This will result in a better and more persistent model of the learner. To enhance the adaptability of the ITS, methods are needed to accurately classify learner states (e.g., cognitive, affective, psychomotor, social) and select optimal instructional strategies given the learner's existing states. A more comprehensive learner model will allow the ITS to adapt more appropriately to address the learner's needs by changing the instructional strategy (e.g., con-

tent, flow, or feedback). An instructional strategy better aligned to the learner's needs is more likely to positively influence their learning gains. It is with the goal of optimized learning gains in mind that the design principles for GIFT were formulated.

This version of the learning effect model has been updated to gain understanding of the effect of optimal instructional tactics and instructional context (both part of the domain model) on specific desired outcomes including knowledge and skill acquisition, performance, retention, and transfer of skills from training or tutoring environments to operational contexts (e.g., from practice to application). The feedback loops in Figure 1 have been added to identify tactics as either a change in instructional context or interaction with the learner. This allows the ITS to adapt to the need of the learner. Consequently, the ITS changes over time by reinforcing learning mechanisms.

GIFT Design Principles

The GIFT methodology for developing a modular, computer-based tutoring framework for training and education considered major design goals, anticipated uses, and applications. The design process also considered enhancing one-to-one (individual) and one-to-many (collective or team) tutoring experiences beyond the state of practice for ITSs today. A significant focus of the GIFT design was on domain-dependent elements in the domain module only. This is a design tradeoff to foster reuse and allows ITS decisions and actions to be made across any/all domains of instruction.

One design principle adopted in GIFT is that each module should be capable of gathering information from other modules according to the design specification. Designing to this principle resulted in standard message sets and message transmission rules (i.e., request-driven, event-driven, or periodic transmissions). For instance, the pedagogical module is capable of receiving information from the learner module to develop courses of action for future instructional content to be displayed, manage flow and challenge level, and select appropriate feedback. Changes to the learner's state (e.g., engagement, motivation, or affect) trigger messages to the pedagogical module, which then recommends general courses of action (e.g., ask a question or prompt the learner for more information) to the domain module, which provides a domain-specific intervention (e.g., what is the next step?).

Another design principle adopted within GIFT is the separation of content from the executable code (Patil & Abraham, 2010). Data and data structures are placed within models and libraries, while software processes are programmed into interoperable modules. Efficiency and effectiveness goals (e.g., accelerated learning and enhanced retention) were considered to address the time available for military training and the renewed emphasis on self-regulated learning. An outgrowth of this emphasis on efficiency and effectiveness led Dr. Sottilare to seek external collaboration and guidance. In 2012, ARL with the University of Memphis developed expert workshops of senior tutoring system scientists from academia and government to influence the GIFT design goals moving forward. Expert workshops have been held each year since 2012 resulting in volumes in the Design Recommendations for Intelligent Tutoring Systems series the following year. The learner modeling expert workshop was completed in September 2012 and Volume 1 followed in July 2013. An expert workshop on instructional management was completed in July 2013 and Volume 2 followed in June 2014. The authoring tools expert workshop was completed in June of 2014 and Volume 3 was published in June 2015.The domain modeling expert workshop was held in June 2015, and the assessment expert workshop was held in May 2016. Future expert workshops are planned for team training, and learning effect evaluations.

Design Goals and Anticipated Uses

GIFT may be used for a number of purposes, with the primary ones enumerated below:

1. An architectural framework with modular, interchangeable elements and defined relationships to support stand-alone tutoring or guided training if integrated with a training system

2. A set of specifications to guide ITS development

3. A set of exemplars or use cases for GIFT to support authoring, reuse, and ease-of-use

4. A technical platform or testbed for guiding the evaluation, development/refinement of concrete systems

These use cases have been distilled down into the three primary functional areas, or *constructs*: authoring, instructional management, and the recently renamed evaluation construct. Discussed below are the purposes, associated design goals, and anticipated uses for each of the GIFT constructs.

GIFT Authoring Construct

The purpose of the GIFT authoring construct is to provide technology (tools and methods) to make it affordable and easier to build ITSs and ITS components. Toward this end, a set of authoring interfaces with backend XML configuration tools continues to be developed to allow for data-driven changes to the design and implementation of GIFT-generated ITSs. The design goals for the GIFT authoring construct have been adapted from Murray (1999, 2003) and Sottilare and Gilbert (2011). The GIFT authoring design goals are as follow:

- Decrease the effort (time, cost, and/or other resources) for authoring and analyzing ITSs by automating authoring processes, developing authoring tools and methods, and developing standards to promote reuse.

- Decrease the skill threshold by tailoring tools for specific disciplines (e.g., instructional designers, training developers, and trainers) to author, analyze, and employ ITS technologies.

- Provide tools to aid designers/authors/trainers/researchers in organizing their knowledge.

- Support (structure, recommend, or enforce) good design principles in pedagogy through user interfaces and other interactions.

- Enable rapid prototyping of ITSs to allow for rapid design/evaluation cycles of prototype capabilities.

- Employ standards to support rapid integration of external training/tutoring environments (e.g., simulators, serious games, slide presentations, transmedia narratives, and other interactive multimedia).

- Develop/exploit common tools and user interfaces to adapt ITS design through data-driven means.

- Promote reuse through domain-independent modules and data structures.

- Leverage open-source solutions to reduce ITS development and sustainment costs.

- Develop interfaces/gateways to widely-used commercial and academic tools (e.g., games, sensors, toolkits, virtual humans).

As a user-centric architecture, anticipated uses for GIFT authoring tools are driven largely by the anticipated users, which include learners, domain experts, instructional system designers, training and tutoring system developers, trainers and teachers, and researchers. In addition to user models and GUIs, GIFT authoring tools include domain-specific knowledge configuration tools, instructional strategy development tools, and a compiler to generate executable ITSs from GIFT components in a variety of formats (e.g., PC, Android, and iPad).

Within GIFT, domain-specific knowledge configuration tools permit authoring of new knowledge elements or reusing existing (stored) knowledge elements. Domain knowledge elements include learning objectives, media, task descriptions, task conditions, standards and measures of success, common misconceptions, feedback library, and a question library, which are informed by instructional system design principles that, in turn, inform concept maps for lessons and whole courses. The task descriptions, task conditions, standards and measures of success, and common misconceptions may be informed by an expert or ideal learner model derived through a task analysis of the behaviors of a highly skilled user. ARL is investigating techniques to automate this expert model development process to reduce the time and cost of developing ITSs. In addition to feedback and questions, supplementary tools are anticipated to author explanations, summaries, examples, analogies, hints, and prompts in support of GIFT's instructional management construct.

GIFT Instructional Management Construct

The purpose of the GIFT instructional management construct is to integrate pedagogical best practices in GIFT-generated ITSs. The modularity of GIFT will also allow GIFT users to extract pedagogical models for use in tutoring/training systems that are not GIFT-generated. GIFT users may also integrate pedagogical models, instructional strategies, or instructional tactics from other tutoring systems into GIFT. The design goals for the GIFT instructional management construct are the following:

- Support ITS instruction for individuals and small teams in local and geographically distributed training environments (e.g., mobile training), and in both well-defined and ill-defined learning domains.

- Provide for comprehensive learner models that incorporate learner states, traits, demographics, and historical data (e.g., performance) to inform ITS decisions to adapt training/tutoring.

- Support low-cost, unobtrusive (passive) methods to sense learner behaviors and physiological measures and use these data along with instructional context to inform models to classify (in near real time) the learner's states (e.g., cognitive and affective).

- Support both macro-adaptive strategies (adaptation based on pre-training learner traits) and micro-adaptive instructional strategies and tactics (adaptation based learner states and state changes during training).

- Support the consideration of individual differences where they have empirically been documented to be significant influencers of learning outcomes (e.g., knowledge or skill acquisition, retention, and performance).

- Support adaptation (e.g., pace, flow, and challenge level) of the instruction based the domain and learning class (e.g., cognitive learning, affective learning, psychomotor learning, social learning).

- Model appropriate instructional strategies and tactics of expert human tutors to develop a comprehensive pedagogical model.

To support the development of optimized instructional strategies and tactics, GIFT is heavily grounded in learning theory, tutoring theory, and motivational theory. Learning theory applied in GIFT includes conditions of learning and theory of instruction (Gagne, 1985), component display theory (Merrill, Reiser, Ranney & Trafton, 1992), cognitive learning (Anderson & Krathwohl, 2001), affective learning (Krathwohl, Bloom & Masia, 1964; Goleman, 1995), psychomotor learning (Simpson, 1972), and social learning (Sottilare, Holden, Brawner, & Goldberg, 2011; Soller, 2001). Aligning with our goal to model expert human tutors, GIFT considers the intelligent, nurturant, Socratic, progressive, indirect, reflective, and encouraging (INSPIRE) model of tutoring success (Lepper, Drake, & O'Donnell-Johnson, 1997) and the tutoring process defined by Person, Kreuz, Zwaan, and Graesser (1995) in the development of GIFT instructional strategies and tactics.

Human tutoring strategies have been documented by observing tutors with varying levels of expertise. For example, Lepper's INSPIRE model is an acronym that highlights the seven critical characteristics of successful tutors. Graesser and Person's (1994) 5-step tutoring frame is a common pattern of the tutor-learner interchange in which the tutor asks a question, the learner answers the question, the tutor gives short feedback on the answer, then the tutor and learner collaboratively improve the quality of (or embellish) the answer, and finally, the tutor evaluates whether the learner understands the answer. Cade, Copeland, Person, and D'Mello (2008) identified a number of tutoring modes used by expert tutors, which hopefully could be integrated with ITS.

As a learner-centric architecture, anticipated uses for GIFT instructional management capabilities include both automated instruction and blended instruction, where human tutors/teachers/trainers use GIFT to support their curriculum objectives. If its design goals are realized, it is anticipated that GIFT will be widely used beyond military training contexts as GIFT users expand the number and type of learning domains and resulting ITS generated using GIFT.

GIFT Evaluation Construct

The GIFT Analysis Construct has recently migrated to become the GIFT Evaluation Construct with an emphasis on the evaluation of effect on learning, performance, retention and transfer. The purpose of the GIFT evaluation construct is to allow ITS researchers to experimentally assess and evaluate ITS technologies (ITS components, tools, and methods). The design goals for the GIFT evaluation construct are the following:

- Support the conduct of formative assessments to improve learning.
- Support summative evaluations to gauge the effect of technologies on learning.
- Support assessment of ITS processes to understand how learning is progressing throughout the tutoring process.
- Support evaluation of resulting learning versus stated learning objectives.
- Provide diagnostics to identify areas for improvement within ITS processes.
- Support the ability to comparatively evaluate ITS technologies against traditional tutoring or classroom teaching methods.

- Develop a testbed methodology to support assessments and evaluations (Figure 2).

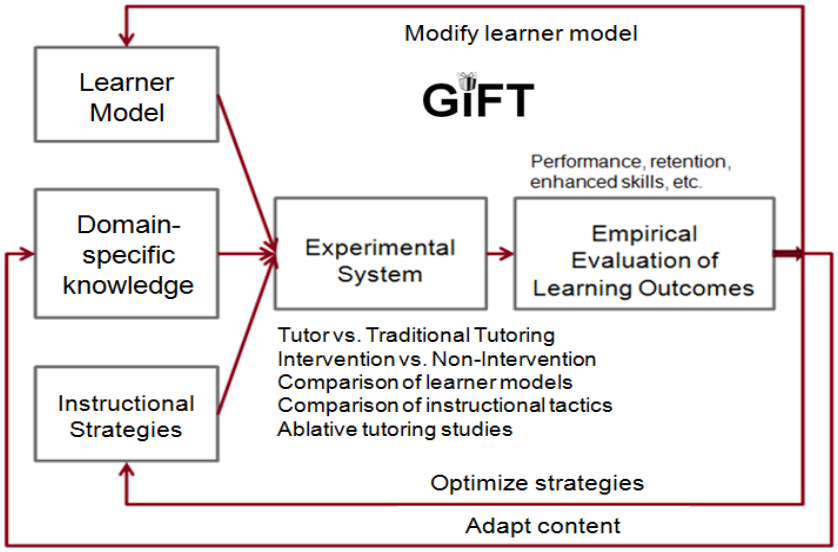


Figure 2. GIFT evaluation testbed methodology

Figure 2 illustrates an analysis testbed methodology being implemented in GIFT. This methodology was derived from Hanks, Pollack, and Cohen (1993). It supports manipulation of the learner model, instructional strategies, and domain-specific knowledge within GIFT, and may be used to evaluate variables in the adaptive tutoring learning effect model (Sottilare, 2012; Sottilare, Ragusa, Hoffman, & Goldberg, 2013). In developing their testbed methodology, Hanks et al. reviewed four testbed implementations (Tileworld, the Michigan Intelligent Coordination Experiment [MICE], the Phoenix testbed, and Truckworld) for evaluating the performance of artificially intelligent agents. Although agents have changed substantially in complexity during the past 20–25 years, the methods to evaluate their performance have remained markedly similar.

The ARL adaptive training team designed the GIFT analysis testbed based upon Cohen's assertion (Hanks et al., 1993) that testbeds have three critical roles related to the three phases of research. During the exploratory phase, agent behaviors need to be observed and classified in broad categories. This can be performed in an experimental environment. During the confirmatory phase, the testbed is needed to allow more strict characterizations of agent behavior to test specific hypotheses and compare methodologies. Finally, in order to generalize results, measurement and replication of conditions must be possible. Similarly, the GIFT evaluation methodology (Figure 2) enables the comparison/contrast of ITS elements and assessment of their effect on learning outcomes (e.g., knowledge acquisition, skill acquisition, and retention).

How to Use This Book

This book is organized into three sections:

I. Fundamentals of Domain Modeling

II. Methods of Domain Modeling

III. Applications of Domain Modeling

Section I, *Fundamentals of Domain Modeling*, describes a variety of approaches to representing the complexity and definition of domains with ITSs. This includes discussion about ontologies, qualitative and ill-defined representations, and methods to map content to skills. Section II, *Methods of Domain Modeling*, examines methods for understanding and developing domain models including methods to turn static lessons (one size fits all) into adaptive representations (individually tailored domains). Finally, Section III, *Applications of Domain Modeling*, discusses the variety of tutoring domains and their attributes.

Chapter authors in each section were carefully selected for participation in this project based on their expertise in the field as ITS scientists, developers, and practitioners. *Design Recommendations for Intelligent Tutoring Systems: Volume 4 – Domain Modeling* is intended to be a design resource as well as community research resource. Volume 4 can also be of significant benefit as an educational guide for developing ITS scientists, as a roadmap for ITS research opportunities.

References

Aleven, V., McLaren, B., Roll, I. & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education, 16*, 101-128.

Anderson, J. R., Corbett, A. T., Koedinger, K. R. & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences, 4*, 167-207.

Anderson, L. W. & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of Educational Objectives: Complete edition*. New York : Longman.

Baker, R.S., D'Mello, S.K., Rodrigo, M.T. & Graesser, A.C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies, 68*, 223-241.

Cade, W., Copeland, J. Person, N., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. In B. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (Eds.), Proceedings of the Ninth International Conference on Intelligent Tutoring Systems (pp. 470-479). Berlin, Heidelberg: Springer-Verlag.

D'Mello, S. & Graesser, A.C. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-adapted Interaction, 20*, 147-187.

D'Mello, S. K., Graesser, A. C. & King, B. (2010). Toward spoken human-computer tutorial dialogues. *Human Computer Interaction, 25*, 289-323.

Elson-Cook, M. (1993). Student modeling in intelligent tutoring systems. *Artificial Intelligence Review, 7*, 227-240.

Fletcher, J.D. & Sottilare, R. (2013). Shared Mental Models and Intelligent Tutoring for Teams. In R. Sottilare, A. Graesser, X. Hu, and H. Holden (Eds.) Design *Recommendations for Intelligent Tutoring Systems: Volume I - Learner Modeling*. Army Research Laboratory, Orlando, Florida. ISBN 978-0-9893923-0-3.

Gagne, R. M. (1985). *The conditions of learning and theory of instruction* (4th ed.). New York: Holt, Rinehart & Winston.

Goldberg, B.S., Sottilare, R.A., Brawner, K.W. & Holden, H.K. (2011). Predicting Learner Engagement during Well-Defined and Ill-Defined Computer-Based Intercultural Interactions. In S. D'Mello, A. Graesser, , B. Schuller & J.-C. Martin (Eds.), *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction (ACII 2011) (Part 1: LNCS 6974)* (pp. 538-547). Berlin Heidelberg: Springer.

Goleman, D. (1995). Emotional intelligence. Bantam Books, New York (1995).

Graesser, A.C., Conley, M. & Olney, A. (2012). Intelligent tutoring systems. In K.R. Harris, S. Graham & T. Urdan (Eds.), *APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching* (pp. 451-473). Washington, DC: American Psychological Association.

Graesser, A. C. & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal, 31*, 104–137.

Hanks, S., Pollack, M.E. & Cohen, P.R. (1993). Benchmarks, test beds, controlled experimentation, and the design of agent architectures. *AI Magazine, 14* (4), 17-42.

Johnson, L. W. & Valente, A. (2008). Tactical language and culture training systems: Using artificial intelligence to teach foreign languages and cultures. In M. Goker & K. Haigh (Eds.), Proceedings of the Twentieth Conference on Innovative Applications of Artificial Intelligence (pp. 1632-1639). Menlo Park, CA: AAAI Press.

Krathwohl, D.R., Bloom, B.S. & Masia, B.B. (1964). *Taxonomy of Educational Objectives: Handbook II: Affective Domain*. New York: David McKay Co.

Lepper, M. R., Drake, M. & O'Donnell-Johnson, T. M. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds), *Scaffolding learner learning: Instructional approaches and issues* (pp. 108-144). New York: Brookline Books.

Litman, D. (2013). Speech and language processing for adaptive training. In P. Durlach & A. Lesgold (Eds.), *Adaptive technologies for training and education*. Cambridge, MA: Cambridge University Press.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, *10*(1), 98–129.

Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In Murray, T.; Blessing, S.; Ainsworth, S. (Eds.), *Authoring tools for advanced technology learning environments* (pp. 491-545). Berlin: Springer.

Merrill, D., Reiser, B., Ranney, M., & Trafton, J. (1992). Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems. The Journal of the Learning Sciences, 2(3), 277-305

Nkambou, R., Mizoguchi, R. & Bourdeau, J. (2010). *Advances in intelligent tutoring systems*. Heidelberg: Springer.

Patil, A. S. & Abraham, A. (2010). Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development. In F. Xhafa, S. Caballe, A. Abraham, T. Daradoumis & A. Juan Perez (Eds.), *Computational Intelligence for Technology Enhanced Learning. Studies in Computational Intelligence* (Vol 273, pp. 79-97). Berlin: Springer-Verlag.

Person, N. K., Kreuz, R. J., Zwaan, R. A. & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. *Cognition and Instruction, 13*(2), 161–188.

Psotka, J. & Mutter, S.A. (1988). *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rus, V. & Graesser, A.C. (Eds.) (2009). The Question Generation Shared Task and Evaluation Challenge. Retrieved from http://www.questiongeneration.org/.

Simpson, E. (1972). The classification of educational objectives in the psychomotor domain: *The psychomotor domain*. Vol. 3. Washington, DC: Gryphon House.

Sleeman D. & J. S. Brown (Eds.) (1982). *Intelligent Tutoring Systems*. Orlando, Florida: Academic Press, Inc.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, *12*(1), 40-62.

Sottilare, R. & Gilbert, S. (2011). Considerations for tutoring, cognitive modeling, authoring and interaction design in serious games. *Authoring Simulation and Game-based Intelligent Tutoring workshop at the Artificial Intelligence in Education Conference (AIED) 2011*, Auckland, New Zealand, June 2011.

Sottilare, R., Holden, H., Brawner, K. & Goldberg, B. (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training. *Interservice/Industry Training Systems & Education Conference*, Orlando, Florida, December 2011.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*. Orlando, FL: U.S. Army Research Laboratory Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R. (2012). Considerations in the development of an ontology for a Generalized Intelligent Framework for Tutoring. *International Defense & Homeland Security Simulation Workshop* in Proceedings of the I3M Conference. Vienna, Austria, September 2012.

Sottilare, R., Holden, H., Goldberg, B., & Brawner, K. (2013). Chapter 20: The Generalized Intelligent Framework for Tutoring (GIFT). In C. Best, G. Galanis, J. Kerry, & R. Sottilare (Eds.) *Fundamental Issues in Defence Simulation & Training*. Ashgate Publishing.

Sottilare, R., Ragusa, C., Hoffman, M. & Goldberg, B. (2013). Characterizing an adaptive tutoring learning effect chain for individual and team tutoring. In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2013.

Sottilare, R. (2013). Special Report: Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model- Research Outline. *Army Research Laboratory* (ARL-SR-0284), December 2013.

VanLehn, K. (2006) The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education. 16*(3), 227-265.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist*, *46*(4), 197-221.

Woolf, B.P. (2009). *Building intelligent interactive tutors*. Burlington, MA: Morgan Kaufmann Publishers.

SECTION I

FUNDAMENTALS OF DOMAIN MODELING

Benjamin D. Nye and Xiangen Hu, Ed.

CHAPTER 1 – Conceptualizing and Representing Domains to Guide Tutoring

Benjamin D. Nye[1] and Xiangen Hu[2]
[1] USC Institute for Creative Technologies, [2]University of Memphis

Introduction

Any discussion about intelligent tutoring system (ITS) domains must begin with considering how ITSs conceptualize and represent domains. This process requires building formal, mathematically specifiable operationalization of the often implicit knowledge about learning domains and their pedagogy. Across different domains and pedagogical approaches, a wide variety of methods have been taken: a scope that would be better-covered by an encyclopedia rather than a single book. Since this section could not possibly cover every possible approach to domain modeling, the chapters within this section have instead been chosen to cover a representative range of fundamentally different approaches to domain modeling.

This section focuses primarily on the characteristics of different domain-modeling frameworks that are effective for modeling different aspects of a domain. Each chapter offers insight into a specific fundamental problem for domain modeling:

- What makes a domain ill-defined and how to we model the ill-defined parts of a domain?
- How can skills be mapped onto tasks, starting from guesses and then leveraging empirical data?
- How can knowledge in terms of one system's domain model be shared with another system?
- How can systems be modeled at the right level of detail to tutor learners?
- How can complex systems with socio-technological interactions be modeled?

These may be thought of as three distinct themes: (1) ill-defined domains, (2) modeling skills and mastery, and (3) models of systems. These cover areas of domain-modeling fundamentals that are both non-trivial (i.e., actively being researched) and occur across a variety of learning domains. That said, gaining a broad understanding of the fundamentals of domain modeling across different ITSs would require not just this volume, but also a number of others that address core concepts. Some recommended supporting resources include Woolf's (2010) textbook on ITSs for a primer; Aleven, Ashley, Lynch, and Pinkwart's (2008) workshop proceedings on ill-defined domains, Nkambou, Mizoguchi, and Bourdeau's (2010) book on ITS designs; and also the three prior volumes in the Generalized Intelligent Framework for Tutoring (GIFT) series (*Learner Modeling*, *Pedagogical Strategies*, and *Authoring Tools*).

Defining a Learning Domain

Before attempting to build any domain model for an ITS, the domain should first be carefully evaluated for ill-defined components. Chapter 2 introduces the principle that no learning domain is wholly well defined or ill defined, but that there are instead a variety of components that can all be independently well- or ill-defined in specific ways. This chapter also introduces the issue that domain pedagogy (teaching) may be well defined, even when domain expertise (doing) may not be well understood (or vice-versa). After outlining a checklist of components to evaluate to determine what is ill defined (if anything), this chapter presents strategies and solutions to these issues. These strategies include approaches to simplify

the domain, split the domain and model certain parts differently (hybrid tutoring), transform the domain by changing to an alternate representation, and leverage alternative assessments when no objective or criterion-based assessments are available. One core recommendation of this chapter is that GIFT and other generalized ITSs should support alternative assessments (e.g., norm-based measures) that explicitly map to pedagogical interventions that are valid even when no traditional assessment is possible.

Modeling Skills and Mastery

After a domain is defined, a key step for any system that recommends tasks is to model the relationship between tasks and skills. Chapter 3 focuses on domain models for student mastery (i.e., how domain content relates to skills and knowledge). Perhaps the most common format for such domain models is the Q-matrix representation that denotes to presence or absence of a particular skill or knowledge component (KC) in a particular item or problem. Rather than being completely data driven from student responses, these techniques start with an expert "guess" of the Q-matrix and refine that guess with student interaction data. This chapter presents an excellent overview of current research on Q-matrices and current data-driven techniques to refine Q-matrices with student data. In addition, the chapter describes cutting-edge work on ensemble methods that achieve state of the art performance by combining multiple Q-matrix refinement algorithms.

Chapter 4 expands the discussion of KCs into a broader consideration about knowledge ontologies, which represent relationships between KCs and also with other entities (e.g., roles, contexts, tasks, etc.). The focus of this chapter is on approaches to align ontologies and map knowledge in terms one ontology into another. As students use more ITSs, sharing knowledge between systems is increasingly important. Types of data that can be used to map between different domain ontologies are discussed, which include ontology labels (string or semantic similarity between the labels in the ontologies), topographical structure (similarity in the links and relationships between entities in ontologies), task outcome distributions (comparing how summative performance on tasks are interpreted in terms of each ontology), and task process events (comparing how the steps of each task are interpreted by each ontology, such as specific misconceptions). The approach of using richer process data on tasks is a novel approach for the field of ITS, and one that is likely to be used by many future systems.

Models of Systems

The level of analysis and scope of a domain model must also be considered when developing an ITS. Chapter 5 explores models for qualitative reasoning (QR), which are designed to model the relationships, causality, and inferences of domain. Key advantages for these models are noted, such as the ability to model domain relationships separately from the model of a specific task. The elements of a QR model are outlined, which include the component ontology (types of things), the process ontology (types of relationships or interactions), and quantities (fluents that represent a given value over time). The ability for these models to represent and tutor causal relationships is discussed. Different types of QR models are also discussed, such as within-state reasoning (e.g., concept maps), multistate qualitative simulations (e.g., discrete-state dynamical systems), and hybrid quantitative/qualitative simulations (e.g., a QR model that is an abstraction of a numerical simulation). This chapter highlights a key capability of QR domain models to help map new learning domains onto students' everyday knowledge, across a range of domains from water cycles to space exploration.

Chapter 6 explores the capabilities of workflow models to model the domain supported by an ITS. Workflow models are designed to represent socio-technological systems, where people and systems hold certain roles and participate in interactions over time. These models can be used to model causality and deci-

sion making in systems, as well as to model faulty or bounded decision making (e.g., acting with insufficient information or costs to gather information). An example application of these approaches to modeling military tactical air operations for a team of jet fighters is presented, which highlights the distinct capabilities of workflows to model complex and dynamically changing systems. As ITS increasingly move into socio-cultural and socio-technological domains, workflows, and other models that capture human capabilities and limitations will be increasingly important.

Final Thoughts

The core insight behind any ITS domain modeling effort must be "essentially, all models are wrong, but some are useful" (Box & Draper, p. 424). As compared to statistics, this holds doubly true for a tutoring system: the ability of the model to be able to complete domain problems and make accurate predictions or any other capability is only useful insofar as it helps the learner. Failing to understand this fundamental issue can very easily lead to unproductive efforts to develop ideal models of the systems or knowledge that the learner needs to know, but that are inefficient (e.g., too hard to author) or ineffective (e.g., not useful for adaptation or interaction with the user) for actually helping a novice or journeyman master the material. It is for this reason that so many domain-modeling approaches exist: they tend to be optimized to the domain and pedagogical strategies of an ITS that is tutoring the domain.

As seen across these chapters, having multiple models of the same system is not necessarily a liability. For example, for ill-defined domains (as noted in Chapter 2), different aspects of the domain must be modeled and tutored differently. Likewise, qualitative reasoning models remove unnecessary precision or calculations, enabling the ITS to focus on general principles and dynamics (Chapter 5). Even in well-defined domains where knowledge components can be readily extracted based on data (Chapter 3), models may go through many iterations to optimize the goal of measuring and optimizing learning. While in the past, this has led to problems in sharing data between systems, research on methods to efficiently align ontologies between different ITS should eventually allow communicating inferences between systems even if they model their domain and knowledge differently (Chapter 4). More generally, the efficiencies from reuse of domain models across learning tasks must be balanced against the alternative of specialized domain models, even if those specialized models need mappings to interoperate meaningfully.

So then, while standards for representing and exchanging domain models and knowledge can still be effective tools (e.g., leveraging an existing ontology, expert system, or simulation), ITS designers must also consider when it is valuable to develop an alternative model that is more efficient for a given tutoring context. This tension is particularly relevant to a framework such as GIFT, which targets generalized and domain-agnostic tutoring. In the long term, being able to build and consume different types of domain models will be important for the growth of the ITS ecosystem. This growth will ultimately require methods to make useful inferences on data drawn from a variety of models (e.g., different models, integrated through the exchange and alignment of recorded learner data and domain model inferences). As such, while domain modeling has traditionally been a difficult field for standardization, there should be a light at the end of the tunnel where models may eventually be both non-standard (e.g., designed pragmatically for system-specific inferences), but automated algorithms analyze and collate data into standardized forms that support useful adaptation to the learner (Q-Matrices, ontologies).

References

Aleven, V., Ashley, K., Lynch, C. & Pinkwart, N. (2008). Proceedings of the ITS 2008 Workshop on Intelligent Tutoring Systems for Ill-Defined Domains: Assessment and Feedback in Ill-Defined Domains. Springer.

Box, G. E. & Draper, N. R. (1987). Empirical Model-Building and Response Surfaces. Wiley. ISBN 0471810339.

Nkambou, R., Mizoguchi, R. & Bourdeau, J. (Eds.). (2010). Advances in intelligent tutoring systems (Vol. 308). Springer Science & Business Media.

Woolf, B. P. (2010). Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann.

CHAPTER 2 – Defining the Ill-Defined: From Abstract Principles to Applied Pedagogy

Benjamin D. Nye[1], Michael W. Boyce[2], and Robert A. Sottilare[2]
[1]USC Institute for Creative Technologies,
[2]US Army Research Laboratory

Introduction

Attempts to define ill-defined domains in intelligent tutoring system (ITS) research has been approached a number of times (Fournier-Viger, Nkambou & Nguifo, 2010; Lynch, Ashley, Pinkwart & Aleven, 2009; Mitrovic & Weerasinghe, 2009; Jacovina, Snow, Dai & McNamara, 2015; Woods, Stensrud, Wray, Haley & Jones, 2015). Related research has tried to determine levels of ill-definedness for a domain (Le, Loll & Pinkwart, 2013). Despite such attempts, the field has not yet converged on common guidelines to distinguish between well- versus ill-defined domains. We argue that such guidelines struggle to converge because a domain is too large to meaningfully categorize: every domain contains a mixture of well- and ill-defined tasks. While the co-existence of well- and ill-defined tasks in a single domain is nearly universally agreed upon by researchers, this key point is often quickly buried by an extensive discussion about what makes certain domain tasks ill defined (e.g., disagreement about ideal solutions, multiple solution paths).

In this chapter, we first take a step back to consider what is meant by a domain in the context of learning. Next, based on this definition, we map out the components that are in a learning domain, since each component may have ill-defined parts. This leads into a discussion about the strategies that have been used to make ill-defined domains tractable for certain types of pedagogy. Examples of ITS research that applies these strategies are noted. Finally, we conclude with practical how-to considerations and open research questions for approaching ill-defined domains.

This chapter should be considered a companion piece to our chapter in the prior volume of this series (Nye, Goldberg & Hu, 2015). This chapter focuses on how to understand and transform ill-defined parts of domains, while the prior chapter discusses commonly used learning tasks and authoring approaches for both well- and ill-defined tasks. As such, this chapter is intended to help the learner understand if and how different parts of the domain are ill defined (and what to do about them). The companion piece in the authoring tools volume discusses different categories of well- and ill-defined tasks from the standpoint of attempting to author and maintain an ITS.

What is a Learning Domain?

It is easy to think of examples of domains (e.g., math, writing, physics, accounting), but harder to find examples of precise definitions of why certain topics are considered learning domains while other topics are typically not (e.g., fire, hearing, the universe). This implies that domains tend to be defined bottom-up, as clusters of knowledge, skills, and tasks that co-occur, either in educational (e.g., curricula) and/or vocational (e.g., expert skillsets) contexts. Treating domains as clusters of knowledge, the boundaries of such clusters would be well defined when their connections and dependencies to knowledge within that domain are much denser than to knowledge considered outside of that domain (i.e., distinct boundaries). So then, a domain can be considered as a set of things that do the following:

(1) Define an expert in some field (e.g., expertise). This view is useful from a training standpoint, where a domain is modeled after an archetypal expert. This connects strongly with assessment literature, in that most assessments attempt to discriminate between novices and experts (Alexander, 1992); or

(2) Are commonly taught together (e.g., pedagogy for an academic discipline). This view is useful for communicating about learning and accommodates knowledge and skills that might underlie a wide range of specializations (e.g., such as reading).

Subject-matter domains appear to be descriptive in nature: learning objectives, tasks, knowledge, and skills are grouped into domains due to either the history of academic disciplines (e.g., how we know to teach it) or the broader societal context (i.e., how we know to use it). While some of these differences appear to be due to relationships between knowledge (e.g., despite being a strong prerequisite for physics, the domain of math is considered separate from physics, since many other domains use math but not physics), others are probably path-dependent historical artifacts (i.e., just how we have always done it).

From a beginning learner's standpoint, any domain may be perceived as poorly defined because its features, rules, and guiding principles are unknown. This is also the first stage of forming a domain from a societal standpoint, since all domains initially begin without experts. In this very beginning phase, the domain is not ill defined (e.g., lack of agreement) but undefined (e.g., not even specified at a level to ground a discussion). From this stage, an understanding of the domain evolves and causes it to become distinct from other domains (useful for categorization), and also enables identifying patterns that are useful to achieve outcomes that are valued within that domain. In this way, society's defining and understanding of a domain must follow phases similar to those of a beginning learner. Initially, some learners "ad-lib" since there are either no formal rules or guiding principles for a domain, or the learner is unaware of them. For example, dancing may appear to be an ill-defined domain to a teenager on a dance floor for the first time, but is well defined for a professional dancer in a Broadway play or a ballerina in a ballet production. The differences in perception of the definition of a domain may rest in the learner's familiarity with successful and unsuccessful models of performance, their capabilities to implement a successful model (e.g., talent), their degree of preparation invested to perform well on a consistent basis (e.g., practice), and also a societal agreement about what success means in that domain (e.g., goals).

Starting from an initial lack of structure, domains gain definition and value from their ability to transfer learning, either from one person to another (e.g., pedagogy, communication) or from one task to another (e.g., leveraging similarities between domains). Starting from an initial state where exploration is arbitrary or random, several theories of how to be successful in that domain will tend to emerge. In other words, learners may have several paths to successful performance but lack consensus on a single best methodology. Over time, the more successful methods should win out and continue (i.e., memetic competition with survival of the fittest). Eventually, a limited set of methods may become popular both because they are effective and because they are sufficiently simple to be shared with the masses. This leads to a convergence and ultimately to well-defined domains (e.g., ones with high levels of agreement). However, complex solutions with higher levels of performance may not have this same level of consensus.

For example, in the 1860s when the game of baseball was evolving, nearly every batter had a different approach to hitting the ball. Some were more successful than others, such as choking up on the bat (shortening it) to increase their bat speed when facing a pitcher who threw very fast. Over time, the task of hitting a baseball went from "undefined" (lacking any rules or principles) to "ill-defined" (e.g., lacking agreement, the domain changed faster than consensus was reached, or the domain changed due to learning about it, such as in some game-theoretic domains). In baseball, part of the problem was that some people defined paths to success and could perform well using these models, but most people could not: players' strategies interacted with their physical capabilities, as well as game-theoretic considerations (e.g., chang-

ing your hitting would change how opponents reacted). This is still true today. While there is consensus on what some elements of what makes a successful baseball hitter, professional hitters still have a variety of strategies and conceptualizations for this process. Hitting instructors model some critical behaviors that are necessary to successfully hit a baseball, but training is limited by the fact that reaction times must be fast (e.g., strategies must be trained until nearly automatic, making them mostly mutually exclusive), that different players' athletic talents may be better-suited for certain approaches, and that the performance domain adapts to them (e.g., pitching opponents react to changes in approach). These factors combine to form a domain where the goals are well defined, but the key knowledge to teach varies based on both individual and contextual factors that vary over time.

The distinction between expertise and pedagogy is important when conceptualizing ill-defined domains. A domain is typically considered to be ill defined because the expertise required to do the tasks is unclear. However, from an instructional standpoint, the pedagogical practices and tools to teach the domain might also be unclear (i.e., ill-defined domain pedagogy). This distinction is the difference between domain knowledge (how to do tasks) and pedagogical domain knowledge (how to teach doing those tasks). The purpose of an ITS (and instruction in general) is to reach learning objectives by applying pedagogical domain knowledge to support a learner in developing domain knowledge.

Well-defined expertise implies the ability to design well-defined pedagogy, but ill-defined aspects of expertise might not necessarily imply ill-defined pedagogy. When domain expertise is well defined, one or more instructional frameworks are typically also well defined. Whenever at least one framework is known to be effective, we would typically consider the domain pedagogy to be reasonably well defined. On the other hand, while cultural competency is poorly defined as expertise, cultural immersion through living in a culture is considered to be highly effective for learning. While pedagogy typically builds on understanding the expertise for a domain, there are other interventions that are general because they either based on human cognition (e.g., affective support) or basic principles of learning (e.g., time on task increases learning). That said, it is possible to imagine or build "trick domains," where near-universal pedagogy fails (e.g., ones where the more time you spend training, the less you know, by rewarding habits that are counterproductive in the long term). However, these are not domains that are actually taught in practice. This raises the final point about domains: while there are many components that can be ill defined, there are fewer that are ill defined for domains that we care about tutoring. As such, it is important to consider that while some ill-defined components of domains may be challenges to overcome through better ITS design, others may be red flags that a domain may not yet be mature or that the domain is framed in terms that become outdated faster than they can be trained.

In summary, we posit the following points that underpin our discussion of ill-defined domains:

- Nearly all domains have a mixture of well- and ill-defined components.
- ITS design for domains can be ill defined due to components of either domain expertise or its pedagogy.
- The range of ill-defined domains is restricted due to the nature of the domains that are desirable to train.
- With those premises stated, we can then attempt to identify components that make up a learning domain and how they can be ill defined.

What are the (Ill-Defined) Components of a Domain?

ITSs can be designed based on either the behavior of domain experts or based on the practices of teachers and other instructional experts. In both cases, a domain is primarily specified in terms of the set of tasks that a person or group performs (Nye, Goldberg & Hu, 2015). Multiple domains may share the same tasks, though those tasks may have different relationships with each other within one domain versus another. Such relationships might be temporal (e.g., prerequisites) or part-whole (e.g., subtasks), or include other types of interactions.

The tasks in a domain can be framed using Markov decision processes (MDPs; Bellman, 1957), in that they can be assumed to have states, actions, and transition probabilities between states based on the actions performed. Both states and actions are constructs, in that states and actions represent the space of features and interventions that are useful for decision making or communicating decisions. In an MDP, these would be the state vector features, state-space, and the action space. Tasks, both individually and in aggregate, also imply goals: certain states or trajectories of states that are preferred to other states (e.g., the utility function for an MDP). Finally, assessments are an important component, because they represent tasks or measures that can monitor goals based on the states and actions (e.g., a function over observable nodes in a partially observable MDP).

Defining Components for Expertise versus Pedagogy

These components are shown in Figure 1, both in terms of how they apply to the domain expertise itself and the domain pedagogy (i.e., the tasks of teaching someone to learn that domain expertise). So then, an expert is someone who can do all the domain tasks that a typical expert can do, as inferred from an existing pool of assumed experts (Hoffman, 1988). Likewise, during instruction, a learner is considered to have mastered the content when they can meaningfully complete all the learning tasks for the domain. Tasks that are useful for learning are often not the same ones seen in real life, but instead tend to emphasize key concepts or are bounded due to contextual constraints such as cost or safety. Theories of assessment bridge expertise with instruction, by aligning learning tasks to domain tasks (Alexander, 1992; Fitzpatrick, Hawboldt, Doyle & Genge, 2015; Gipps, 1994).

Domain Expertise	Domain Pedagogy
Tasks: What tasks do experts do? - **States/Features:** What features do experts monitor? - **Actions:** What do experts do to influence transitions between states?	**Tasks:** What are the training tasks? - **States/Features:** What actions should a teacher/ITS monitor? - **Actions**: What interventions should be used to improve learning?
Relationships: How do tasks interact?	**Relationships:** How should tasks interact?
Goals: What task states or pathways are considered better?	**Goals:** What are the learning objectives? How do these relate to task expertise?
Assessment: What tasks and outcomes measure performance?	**Assessment:** What tasks and outcomes measure changes due to learning?

Figure 1. Components of domain expertise (doing) vs. domain pedagogy (teaching).

For a completely well-defined domain, every component of Figure 1 would be straightforward to specify and there would be no disagreement between different experts. In practice, particularly for domain pedagogy, there is almost always some disagreement or uncertainty about optimal specifications. This is be-

cause instructional design is indeed a design problem, which is characterized as one type of ill-defined task. An ITS requires a well-defined instructional design (Domain Pedagogy, right side Figure 1) that can be implemented computationally. To do this, the instructional design must accommodate the various well- and ill-defined aspects components of the domain expertise.

Assuming these components are for a domain and its pedagogy, what does it mean to be ill defined? In practice, any one of these components is ill defined when there is no convergence to a countable number of agreed-upon elements. For example, domain experts may agree about the key tasks for a domain (well-defined tasks), but may disagree about the set of features that are relevant to those tasks (ill-defined features for task states). Since components can be independently ill defined, the degree that a domain is ill defined depends on not just the level of disagreement but also on which components are ill defined.

As an example, Figure 2 contrasts a well-defined domain task (basic algebra) against a highly ill-defined domain task (country stability). In a fully defined task, it is possible to determine at least the order of value for states and also assign value to the actions that the user took. In a predictable task, the value of actions is directly determined by the value of the states, though in a chaotic or probabilistic domain, action value might be only loosely connected to state utility (i.e., optimal actions still sometimes have bad outcomes, and vice versa). For a highly ill-defined task, it may be difficult to determine the actions and features of the state that are relevant (e.g., a "mess"; Ackoff, 1981). Figure 3 gives examples of domains where either the importance of states and/or actions are better or worse defined. A second major difference is that experts agree about solutions and goals for well-defined tasks (e.g., simplifying an algebra equation), but ill-defined domains can have fundamental disagreements over optimal solutions. For example, the United States and Taliban both have vastly different goals for a stable government in Afghanistan.

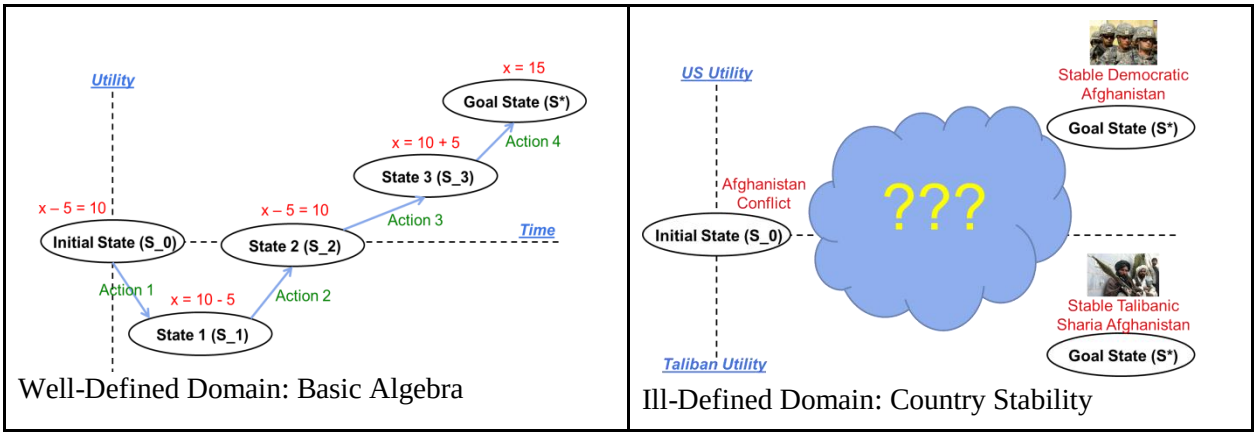


Figure 2. Contrasting the components of a well-defined domain vs. a highly ill-defined domain.

	Action Values Known	Action Value Unknown
State Utilities Known	Can evaluate all actions and suggest good solution paths *Ex. Economic decision tree*	Know the value of states, but can't surely deduce what actions do *Ex. Stock market simulation*
State Transition Gradients Known	Can rank states relatively and can suggest next step or bug fix. *Ex. Solving an Algebra problem*	Know better/worse changes, but can't suggest concrete actions *Ex. Automated essay assessment*
State Values Categorical	Can detect good/bad actions, but can't rank improvement *Ex. Going out of bounds in a race*	Ill-defined task, due to emergent dynamics or subjective values *Ex. Building a better world*

Figure 3. Levels of definition for the value/utility of task states and actions.

Fundamental Ambiguities (What/Why) vs. Underlying Ambiguities (How)

Fundamental ambiguities about the boundaries of a domain are present when experts cannot agree on either the tasks or the goals (i.e., what and why). These ambiguities mean that experts disagree about what important activities occur and what would be considered a good outcome. These indicate uncertainty about what should be taught within a given domain in terms of what a domain is. Selecting tasks determines that one would not teach math in an English curriculum, for example. Ambiguities over good/bad states are when designers agree about how to frame the domain, but do not necessarily agree on the importance of outcomes or actions. For example, history is sometimes criticized for emphasizing events such as wars as compared to other important events such as trade or peaceful regime change (Nash, Crabtree & Dunn, 2000). History is also an example with ambiguities about goals: historians have served many roles within cultures, ranging from neutral archivists who preserve the past, to interpreters who try to communicate it to the present, to revisionists who reinterpret history to try to change our future view of the past (Grele, 1981). The relative importance of different tasks may also be debated. These are fundamental ambiguities about what constitutes expertise and what goals one should be able to accomplish by studying history. Other core disciplines, such as philosophy, ethics, legal argument, and social policy, also hit such issues. ITSs for domains with fundamental ambiguities are relatively rare and tend to focus on the process of studying the domain, rather than arriving at specific outcomes (Pinkwart, Ashley, Lynch & Aleven, 2009; Easterday, Aleven, Scheines & Carver, 2011).

Underlying ambiguities include task states, task actions, and relationships between tasks. These more granular ambiguities are present when experts can agree on tasks enough to make meaningful comparisons between them, but disagree about how to represent what feature space, state-space, or actions are available during a task. Many domains commonly identified as ill-defined are due to these issues, such as programming (Mitrovic, Koedinger & Martin, 2003) and design (Chesler, Arastoopour, D'Angelo, Begley & Shaffer, 2013). Finally, assessment ambiguities can occur when it is unclear which benchmarks should be usefully applied to measure progress toward goals. This type of ill-defined component can occur because the ideal goals are not directly observable (e.g., long-term economic growth or learning), which requires measures that attempt to estimate when goal states and paths are reached.

The above issues that affect domain knowledge and expertise apply similarly to domain pedagogy, where the task is building an instructional design (e.g., a curriculum, either adaptive or linearly sequenced). Much like for the domain itself, the instruction material has tasks (learning tasks), goals (learning objectives), relationships between tasks (e.g., knowledge prerequisites), and assessments to monitor progress toward the learning objectives. Likewise, for each learning activity, a pedagogical feature state and possible interventions exist. While all of these elements can be ill defined, fundamental disagreements at this level occur at the goal level: being unable to agree on the learning objectives. Unlike the domain expertise, disagreements about learning tasks are not necessarily fundamental, since these are the means to an end. Otherwise, the avenues for ill-defined domain pedagogy are similar to those for domain expertise.

So then, ill-defined domain expertise does not necessarily imply ill-defined domain pedagogy, provided that the learning objectives and goals remain well defined. However, ill-defined components of a domain must inherently constrain the available pedagogical options. In short, ill-defined domains can cause certain pedagogical options to become incoherent or impossible. For example, if the desirability of task states cannot be compared, then it is impossible to give error-correction feedback. This has significant implications for the types of instruction and pedagogical strategies that can be applied.

Given that we conceptualize components as ill-defined due to ambiguous components and a lack of agreement between experts, it is important to identify root causes that may underlie such disagreements. Components may be ill defined for a number of reasons (not exhaustive):

- *Subjectivity* occurs when different experts disagree about either the relative importance of tasks or about the value of states within tasks. If the goals are subjective, two experts might see the same events, with one declaring them a ringing success while the other declares them an utter failure.

- *Under-specification* occurs when domain tasks or goals tend to be context-specific, but it is infeasible or impractical to represent the space of contexts that impact the tasks. Under-specification can underlie subjectivity, such as comparing computer code by an author who prefers brief code versus one who prefers extensive code documentation. These preferences are typically due to different design goals, and both experts might agree if one goal or the other was stated or explicitly weighted.

- *Representation/mental-model mismatches* are a related problem, where experts may have different levels of specification for different tasks (e.g., unique mental models; Ososky, 2013) or when experts have similar mental models but communicate them differently (e.g., different names, same concepts; Norman, 1983). This can be observed as disagreement about the specific features or actions involved, despite agreement about goals and the value of final outcomes.

- *Assessment/measurement* confounds occur when the states and values may have relatively high levels of agreement, but are unobservable and assessment methodologies struggle measure those key outcomes.

- *Rapidly evolving domains* occur when domain knowledge becomes outdated faster than it is possible to train the knowledge or skills. For example, sommeliers specialize in the process of tasting wines and typically a specific regional specialty, rather than attempting to be an expert in all wines. This is because the variants of wine released in a year are far more than any one person could taste.

- *Wicked problems* or "messes" occur when multiple tasks are highly interconnected, evolve over time, and also prone to subjective goals (Ackoff, 1981; Rittel & Webber, 1973). Confounds include imperfect information, unintended consequences, probabilistic results, emergent outcomes, and many stakeholders with different value systems (Silverman et al., 2010). These are commonly found in complex social problems, such as public policy. Wicked problems are complex because not only do different experts disagree at a given time; the same expert might have different opinions depending on the time horizon of interest.

- *Uncontrollable problems* occur when the domain is inherently ill defined, in that the tasks are not actually human-controllable because they are inherently chaotic or the information to meaningfully direct them is unobservable (Kalman, 1959). Such tasks are not teachable, since there is only the illusion of control. Related to this are problems where control is only temporary (e.g., cyclical or otherwise non-asymptotic), because entropy rapidly washes out human inputs and their traces from the system (e.g., building sandcastles). Another related category is systems that with high inertia or tendencies toward entropy, where change due to human inputs are only observable after long periods of effort despite no detectable changes (e.g., breaking a boulder by kicking it for years). The latter type might be considered energy-responsive, in that a certain critical mass of inputs (either all at once, or over time) is needed to cross a tipping point where change can be observed.

With the exception of truly uncontrollable problems, pedagogical strategies have been developed to accommodate domains that suffer from these root causes for different aspects of the domain goals, tasks, relationships between tasks, and assessments. Examples of strategies to address ill-defined components are noted in the next section.

Strategies and Solutions: Simplifying, Splitting, Transforming, and Assessing Ill-Defined Components

Four main strategies have been proposed to simply ill-defined aspects of a domain so to make it pedagogically tractable for an ITS: authority-based simplifications that limit their training to a subset of a complex domain (Roberts, 2000), hybrid ITS that split the domain into well- and ill-defined tasks that are tutored differently (Fournier-Viger et al., 2010), transforming ill-defined domain tasks into pedagogically well-defined tasks (Mitrovic et al., 2003), and expanded assessments beyond traditional objective, criterion-based assessments (e.g., explicit handling of subjectivity; Nye, Bharathy, Silverman & Eksin, 2012).

Simplifying the Domain: Authority-Based Simplifications

The first strategy for simplifying ill-defined aspects of a domain is entirely direct: simply choose a point-of-view and stick with it. This approach originates in literature on the study of complex social systems, often-called messes or wicked problems (Ackoff, 1981; Rittel & Webber, 1973). Choosing beneficial courses of action in a socio-cultural system is ill defined because the framing of the problem determines the value of the solution (Rittel & Webber, 1973). Confounds include imperfect information, unintended consequences, probabilistic results, emergent outcomes, and many stakeholders with different value systems (Silverman et al., 2010). This makes the goals for actions and outcomes in social systems subjective, due to differences in value systems or framing. This ambiguity is intractable because it is based in serious moral and philosophical questions, such as "How much money is a life worth?'' and "Is health or security more important?'' Even for an individual, quantitatively answering such questions involving life and death tends to be difficult and inconsistent (Waldmann & Dieterich, 2007). Due to their complexity, issues such as policy problems, economic interventions, and military strategies are often analyzed using forecasting simulations to better understand potential outcomes (Ichikawa et al., 2010; Silverman et al., 2010). However, when such approaches are used, actions often have implications beyond immediate outcomes and contribute to later emergent outcomes.

Roberts (2000) notes that wicked problems can be reduced using authoritative simplifications. This approach transforms the problem by using a subset of expert perspectives to reduce the number of stakeholders. The training objectives selected by funding sponsors and developers intrinsically introduce authoritative simplifications. For example, a training environment for the US military will assess performance in terms of allied interests and US. This can reduce certain globally subjective issues into objective tasks. Domain experts can also provide authoritative structure. However, experts often differ on useful courses of action to improve a social problem so subjectivity cannot be entirely eliminated. The Complex Environment Assessment and Tutoring (CEATS) prototype for simulation-based training of counter-insurgency operations applied this methodology to convert a faction-based predictive model of district stability into a learning environment (Nye et al., 2012). CEATS employed "objective" measures of performance (e.g., increasing alliances with local groups) that were actually just special cases of subjective measures of performance where the subject (US and International Forces) was held constant.

Authority-based simplifications are particularly useful in two cases. The first case is when all other methods to transform or simplify the domain have been exhausted, and it becomes necessary to simply settle on a (possibly even arbitrary) representation. In some cases, this is not necessarily bad. For example,

agreeing on expert knowledge components to adaptively select learning tasks can be quite difficult, while methods to infer useful knowledge components to perform this task are quite mature (Ritter, 2015). As such, if data can be readily collected, it is a reasonable strategy to make best-guess Knowledge Components (KCs) and prerequisites and then apply sufficient sampling to explore the space of alternatives to find a more-optimal set. The second case is when stakeholders that are critical to adoption would prefer to focus on certain tasks or features, as compared to others. In this case, good standards in user-centric design dovetail with the need to simplify ill-defined components. By consulting with the stakeholders involved in adopting, using, and maintaining the system, it may be discovered that certain domain tasks, features, or assessments are more useful and hence more likely to be used to support learning. This is particularly important since the best designed ITS is useless unless people use it.

Splitting Up the Domain: Handling Well-Defined and Ill-Defined Tasks Separately

One strategy applied to address ill-defined domains has been hybrid tutoring which splits the domain into different tasks and features based on their well-defined and ill-defined characteristics (i.e., divide-and-conquer). Hybrid tutoring approaches apply multiple qualitatively different pedagogical interventions to different tasks (Fournier-Viger, et al., 2010). Since many domains have a have a mixture of well-defined and ill-defined problems, the first strategy is to identify and split the domain into the tasks, which are well define versus those that are ill defined. Ideally, a hybrid design would guide tutoring based on the nature of the problem. However, hybrid-tutoring designs are an emerging topic and lack established design principles.

Fournier-Viger et al. (2011) describes one application of a hybrid ITS to assessing and tutor the ill-defined domain of controlling a three-dimensional robot arm with seven dimensions of freedom. This ITS applied model-tracing tutoring to a set of well-defined spatial cognition procedures, partial task models (sequential patterns) that were extracted from annotated actions sequences, and constraint-based modeling to a path-planning simulator to catch critical failures (e.g., damaging the arm). This approach represented the domain from multiple levels, with each level able to support different types of feedback. Model tracing provided feedback, hints, and worked examples for the high-level process and basic controls for the robot arm (e.g., checking cameras for potential obstructions before moving the arm). Partial task models were used to establish similarity of the user's performance to expertise archetypes (e.g., expert, intermediate, novice) and could be used to suggest next-step hints. Finally, the constraint-based model was used to detect and remedy invalid actions.

Guiterrez-Santos, Marvikis, and Magolus (2010) proposed subsumption architectures to break up a complex problem into a series of layers. In this approach, the ITS assumes higher-level tasks with subtasks that determine the state and performance at the higher level as well (i.e., learners do not directly act on the high-level task). A subsumption architecture can provide one mechanism to implement a hybrid ITS, where different layers monitor and attempt to tutor subtasks, and high layers monitor overall task and learning goals to help determine what pedagogical interventions should be applied. Guiterrez-Santos et al. (2010) tutored micro-worlds using this approach, by tutoring a higher-level abstraction of the micro-world. Dividing the system into layers did this: the expresser (the exploratory environment), computation (attempts to solve well-defined tasks), and aggregation (integrates computational solutions into the context of the larger exploration goals). It does this through a rule-based approach where information from different types of data is mapped to satisficing criteria (i.e., conditions for "good enough" solutions as opposed to empirical optima). There is also a check to see if that portion of the solution maps closely to any of the already existing solutions.

Transforming the Domain: Translating Ill-Defined Components into Well-Defined Pedagogy

Transforming the domain imposes additional structure onto ill-defined tasks to support training. Transforming the domain can also partially address the issue of emergent outcomes. Strategies for mid-game chess and Go rely on strategic features of board structure such as territory as a proxy for direct game-state policy evaluation, which is otherwise an intractable problem (Richards, Moriarty & Miikkulainen, 1998). Domain experts in complex socio-cultural problems such as counter-insurgency and economics use similar feature-based approaches to assess the situation (US Army, 2010). Such proxy measures of future value complement the immediate outcomes of actions. However, for complex domains or simulations of such domains, emergent outcomes can occur that conflict with expert expectations. This means that formative and summative assessments may be inconsistent (e.g., good actions, bad outcomes).

Transforming Fundamental Ambiguities

A number of techniques have been applied to try to transform a domain to eliminate fundamental ambiguities, such as what constitutes a good legal case. The most common high-level approach to such domains is to focus on the processes (i.e., actions) that are performed, with less emphasis on the specific content. For example, the PolicyWorld game breaks policy thinking (highly ill-defined) into structured inquiry learning steps (Easterday et al., 2011). As such, it partially sidesteps the issue of which specific policies are good or bad. Instead, it focuses on helping the learner understand the actions and steps that are necessary to make an informed decision about a policy. Systems such as the Legal Argument Graph Observer (LARGO) apply similar transformations where legal arguments (usually expressed as text) are instead represented using graphs to scaffold structured argumentation (Pinkwart, Ashley, Lynch & Aleven, 2007).

A strongly related approach involves having learners perform traditionally unstructured tasks (e.g., argumentation) using highly structured tools that limit them to a better-defined analog. Graphical visualizations (e.g., concept graphs) and form-based entry are common solutions. Lynch and Ashley (2014) proposed a graphical structure that allows for logical relationships, to discover areas where the student arguments contained conflicting information. Similar approaches were used by both PolicyWorld and LARGO (Easterday et al., 2011; Pinkwart et al., 2007). While imposing an artificial task or structure reduces the ecological validity of the learning task as compared to the domain tasks, this sacrifice can give better analysis and traction for other pedagogical goals (e.g., structuring logical arguments) by disentangling them from more complex problems (e.g., writing up a rhetorically persuasive legal argument).

Metacognitive and affective tutors can also be applied to tasks that are not well represented by the ITS, since they attempt to train reusable skills that should improve performance on more than just the current domain. As an example of non-cognitive modeling, Baker and Corbett (2014) identified a series of features that can help to detect whether a student is undergoing robust learning through metacognitive activities, such as the amount of time spent after receiving a feedback message related to lesson content, pauses between answering questions that the student should have knowledge of, attempts to game the system, or participation in off-task behavior. These detectors can serve to support the instructor's understanding of individual student performance, thereby increasing areas of specific emphasis of their pedagogical strategies.

A final category of transformation is to avoid representing the whole domain, but instead use only tutor-specific pathways for complex tasks that have multiple (or even uncountable) solutions. Scenario-based training, path-branching ITS, "what-if" scenarios, and example-tracing tutoring can provide this type of transformation (Lane et al., 2008; Aleven et al., 2009). When the goal of the tutoring is to build basic familiarity or tutor skills that can be readily trained in that specific set of tasks and paths, this approach can

sometimes be more efficient than a higher-fidelity model that accurately represents the probability of certain events. For example, part of the success of both the Sherlock tutor for technician training (Lesgold, Lajoie, Bunzo & Eggan, 1991) and the Digital Tutor (Fletcher, 2011) was attributed to their ability to train rare diagnosis and repair tasks that occur within larger technological systems. Similarly, scenario-based learning for uncommon but important social and cultural competencies might achieve similar impacts.

Transforming Underlying Ambiguities

Different transformations are applied when experts can agree about the high-level tasks and general goals, but cannot agree about how to measure progress or struggle to compare the relative value of specific states or values. These transformations typically convert a task with a poorly defined state-space or valuation of states (e.g., utility) into a simpler set of features or detectors that can be used to drive tutoring.

Constraint-based modeling (CBM) is one of the most well-established approaches to simplify tutoring a task with ill-defined state or action values. Rather than being concerned with what the student has done, the goal in CBM is to give the students as much flexibility as possible so long as the student does not reach a situation that is known to be incorrect. (Mitrovic et al. 2003). In CBM, the ill-defined domain is made more tractable by providing constraints that can create a boundary while still allowing the learner to make choices (Woods et al., 2015). This supports ITSs that act when specific flaws or out-of-bounds conditions are detected, even though the relative utility of different states is difficult or impossible to assess. A CBM ITS only needs features of acceptable solutions in order to work properly (Mitrovic & Ohlsson, 2015), which is advantageous in terms of ill-defined domains because it can solve open-ended problems. Mitrovic has spent the better part of the last two decades analyzing CBM learning through use of a structured query language (SQL)-based tutor. SQL is a good candidate as an ill-defined domain because there can be multiple correct solutions without a clear algorithm to guarantee success. To assist with the ill-defined nature of SQL queries, students are taught search strategies using example problems, which they can then use as reference points to apply those strategies to actual problems (Mitrovic & Weerasinghe, 2009).

Instance-based inference, such as case-based reasoning (CBR), is a second approach to dealing with ambiguity and is particularly useful when the set of useful or agreed-upon features for a state-space are not fully known. These techniques are applicable when cases can be assigned to specific groups (e.g., good/bad examples; expert/intermediate/novice; expert1/expert2/expert3). For example, Goldin, Ashley, and Pinkus (2006) used case-based methods to tutor case analysis as part of legal ethics. Combining a context classifier that detects which concepts have been defined and a CBR manager that compares these concepts to identify similar cases did this. It then reports what concepts have been defined in those cases and how they have been applied. Similar approaches where tagged instances are directly leveraged have also been observed for well-defined domains to identify expert blind spots (e.g., features that experts no longer see; Matsuda, Cohen, Sewall, Lacerda & Koedinger, 2007). Instance-based inference relies on the ability to apply pattern matching to find similar instances or to identify clusters, even though specific rules or expert guidelines have not been developed. If the domain is fundamentally ill defined, instance-based clustering (e.g., good/bad labels) might actually show multiple clusters that represent the convolution of both the expert raters' attitudes and the instance features. As such, this approach can be useful both as a solution and a diagnostic tool for fundamental ambiguities.

Similar approaches have been applied to user trajectory data (i.e., each instance is a pathway taken by a user during a certain task session), rather than a corpus of static cases. These approaches rely on data mining to compare ongoing results to trajectories of data and adapt to the user (Baker & Corbett, 2014). Lazar and Bratko (2014) created an intelligent tutor using this technique to tutor Prolog programming. They do

this by counting the number of times a line and associated line edits appear. By searching the potential paths that are known to get to a successful solution, user edits are ranked and hints are generated. These approaches are analogous to case-based approaches, except that each combination of user-task session represents a case.

Alternative Assessments: Measurements Beyond Performance or Mastery

When the goals of a domain are not well defined (e.g., uncertainty over pedagogical goals or standards), it is difficult to provide learners with guidance to meet those goals. FitzPatrick. Hawboldt, Doyle, and Genge (2015) ran focus groups and document analysis that found that over half of the assessment tasks did not tie in with the respective objectives in terms of Bloom's revised taxonomy (Krathwohl, 2002). Not only were the assessment tasks were not matched to objectives, and over half the assessment questions were categorized at the lowest level ("remember") despite little to no learning objectives targeting that level (FitzPatrick et al., 2015).

In many cases, seemingly ill-defined domains can sometimes be implemented in ITSs by following solid curriculum design principles that help instructors structure the domain and assessments rigorously. This can improve alignment of pedagogical goals to the target domain tasks. In this case, the pedagogical domain knowledge comes from the teacher's personal learning experiences while they were in school, the teacher's education and professional development, and the teacher's experience while teaching others (Friedrichsen et al., 2009). Even if the domain is lacks consensus in some areas, expert teachers may actually still use effective techniques that have been refined over multiple classes, which might only need to be verified using validated assessments that carefully capture the learning objectives.

However, in many cases, objective and criterion-based assessments are not available. Sometimes, this limitation is only due to practical issues (i.e., it only needs to be done). However, in other cases, objective assessment criteria do not exist because experts or instructors cannot agree about the value of different tasks, states, or actions for a domain. From the standpoint of a classical ITS that attempts to support error-correction feedback and next-step hints, this is an insurmountable hurdle: how can one provide feedback if there is no way to detect a right answer? Unfortunately, this is somewhat true: certain tasks simply do not have objective solutions (e.g., subjectivity) and others defy the ability to suggest a next step (or steps) that will guarantee better outcomes (e.g., emergent results).

While such tasks are not amenable to classical ITS interventions, they are not impossible to measure or deliver meaningful adaptive pedagogy. Research on assessment literature has identified a variety of alternative educational assessments. An assessment has at least four key facets: its reference basis, intended usage, objectivity, and relationship of the learner to the assessed concept (Gipps, 1994). The first three are listed in Figure 4. The reference basis represents what the assessment compares the learner against. A criterion-based assessment assumes some particular task or universal benchmark (e.g., running a 5-minute mile). Alternatively, a norm-based assessment compares the learner against some peer group. Ipsative assessments compare the learner against themselves, either at different time points or between different skills/domains. The usage of an assessment can primarily formative (i.e., help adapt to the learner) or summative (i.e., assess the state of the learner, such as for reporting or communication). Finally, the objectivity of the assessment means that it can be either objective (i.e., universally agreed or mandated) or subjective (i.e., depends on viewpoint or rater characteristics).

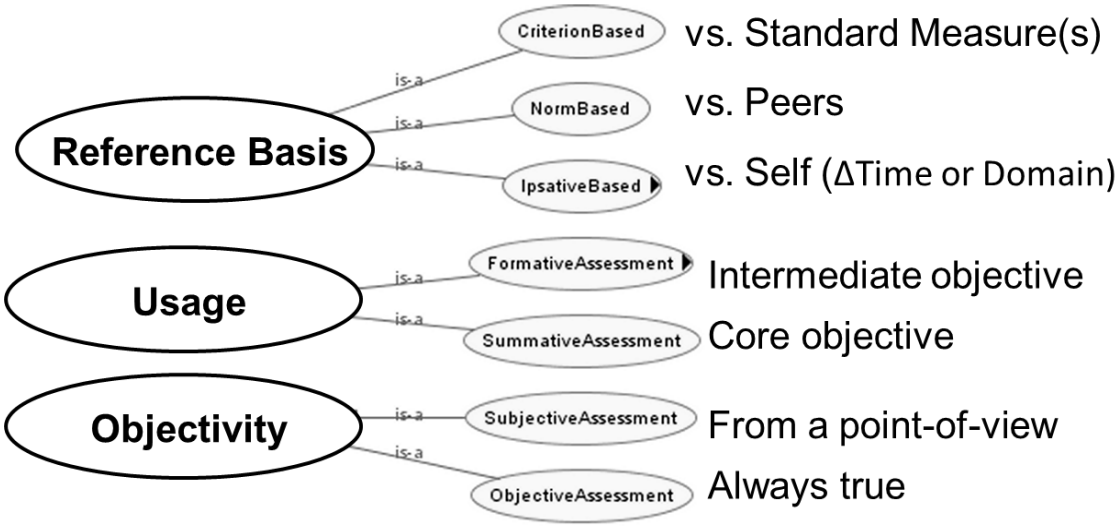


Figure 4. Key assessment characteristics.

Building on these characteristics, it is possible to assess qualitatively different relationships with constructors and knowledge. The vast majority of assessments focus on measuring task performance, mastery, and learning (i.e., change in mastery). However, assessments of academic emotions (e.g., engagement with a topic), attitudes (e.g., motivation to continue studying a topic), frequency of use (e.g., how often a learner tends to use a skill spontaneously), and a variety of other relationships for the learner to knowledge or skills are likely essential for long-term learning and retainment.

Expanding the range of assessments significantly expands the scope of feedback than can be provided. Table 1 notes a list of common pedagogical interventions, as limited by the type of assessment involved. To interpret this table, an assessment can be categorized in terms of Objectivity x Usage x Frame of Reference. If a checkmark exists in each one of these fields, then the assessment should always be able to power the given intervention. Error feedback, providing a right answer, hinting at the next optimal step, and other common interventions (VanLehn, 2006) rely on a well-defined problem structure where objective criteria are established. These can be used for well-defined tasks where traditional right/wrong feedback and hints would be effective. Comparative feedback based on norm-based or impassive assessments fills a complementary role and can be used more broadly, indicating how the user's performance differs from other performances. Alternative approaches used for ill-defined domains are domain non-specific approaches, such as inquiry learning, dynamic collaboration, and Socratic questioning (Fournier-Viger et al., 2010). Inquiry learning appears well suited to consuming impassive assessments, because learners build explanations in a stepwise manner. Consistency between steps is important, so examining a user's next steps in light of their prior steps seems most appropriate. Robust approaches include Socratic questioning (e.g., "Why…") and self-reflection prompts, which can employ knowledge from nearly any type of assessment. Selecting or suggesting collaborative groupings is also robust with respect to the knowledge available, but requires a learning environment that supports collaboration.

Table 1. How common interventions depend on assessments types.

Pedagogical Intervention	Objectivity		Usage		Frame of Reference		
	Objective	Subjective	Summative	Formative	Criterion	Norm	Ipsative
Error Feedback	✓		✓	✓	✓		
Correction Hint	✓			✓	✓		
Next-Step Hint	✓			✓	✓		
Comparative Feedback	✓	✓	✓	✓		✓	✓
Inquiry Learning Prompt	✓	✓		✓			✓
Reflective / Socratic Prompt	✓	✓	✓	✓	✓	✓	✓
Collaborative Grouping	✓	✓	✓	✓	✓	✓	✓

While this is not intended to be a universal guide to mapping alternative assessments to interventions, it demonstrates the possible pedagogy that can be leveraged by mapping alternative assessments to alternative ITS interventions. Future research will hopefully map out a deeper space of guidelines for this type of mapping, which could adaptively select or suggest interventions that match the available assessments.

Recommendations for Handling Ill-Defined Components of Domains

Based on this discussion, a particular set of steps are recommended when building an ITS for a domain that may have ill-defined components. These steps can be integrated into existing approaches to understanding domain knowledge, such as cognitive task analysis (Schraagen, Chipman & Shalin, 2000) and curriculum design methodologies (Wiggins & McTighe, 2005). Considering Figure 1, it is assumed that instructional designers for ITSs would first attempt to understand the domain expertise (Figure 1, left side) and then would attempt to understand and implement domain pedagogy (Figure 1, right side). As noted previously, it is also assumed that a designer would often attempt to identify set of tasks, the features and actions of those tasks, the relationships between tasks, the goals and relative values for task outcomes/actions, and the assessments that measure progress toward goals. During this process, it should be possible to identify and address ill-defined components. The following steps are recommended for handling ill-defined components:

(1) Authority simplifications to the task set,

(2) Segmenting tasks into well-defined and different ill-defined categories,

(3) Transforming ill-defined tasks into well-defined pedagogical tasks (e.g., ones where appropriate feedback and real-time interventions are possible), and

(4) Assessing traditional and alternative (subjective, ipsative, etc.) relationships to domain knowledge.

Authority Simplifications to the Task Set

If the set of tasks that constitute domain expertise is ill-defined, authority-based simplifications are usually applied. Ultimately, education and training is a socially constrained process: skills are trained because some need exists. As such, if experts disagree about what tasks constitute expertise, the stakeholders who are adopting the ITS (e.g., schools, companies, governments) can act as an authority to simplify this set based around their perceived needs. This is important even for well-defined domains, since there is little point in building high-quality activities that stakeholders would not want to use.

Split Tasks into Well vs. Different Ill-Defined Categories

Once the set of tasks is identified, determine the level of definition that is available for each task. As noted, this involves considering the task state features, state-space, action space, and goals/utility of states and actions. Next, bin tasks into different categories based on the ability to specify these components, both for domain expertise tasks and for learning activities. Well-defined tasks have the widest set of possible interventions available. Nye et al. (2015) notes the types of ITS interventions that are possible for different categories of tasks.

Transform Ill-Defined Tasks into Well-Defined Pedagogy

To expand the range of pedagogy that is available for a task, it is possible to transform certain ill-defined components into more tractable forms. This is the most complex step, since it depends on the specific types of ill-defined tasks that must be trained. CBM is often applied to design tasks where the space of invalid designs may be known but the qualities of optimal good designs are not. CBM and comparison against archetypes are applied when tagged examples are available (e.g., good/bad cases), but features or actions are unclear. Example-tracing tutors and branching tutors can also be applied where the state-space or dynamics are large (or possibly not fully known), but tutoring a limited path or scenario through the space is still useful. Authority simplifications can also be applied to task features, actions, or goals as well, if necessary. When applying simplifications or transformations, the most important consideration is that the learning objectives should ideally not be simplified. For example, if a key domain task is to search and choose from a wide array of options, a tutor that only allows one path would be unable to train that skill.

Assess both Traditional and Alternative Relationships to Domain Knowledge

Particularly where task goals and value functions for states and actions are not agreed-upon or hard to measure, alternative assessments may be necessary. Alternative assessments tend to either be decoupled from the specific states of the domain (e.g., measures of learner affect) or parameterized. Subjective assessments, for example, allow comparing learner performance against different subjective benchmarks (e.g., experts applying different theoretical frameworks). Norm-based assessments allow comparing a learner against other learners, even if a "right" answer is unknown (e.g., identifying outliers or clusters of learner types). Parameterized assessments can enable additional types of feedback (e.g., comparative feedback) that would be unavailable otherwise. While alternative assessments have not been traditionally used in ITSs, they open up a wide range of possibilities for addressing ill-defined domains (e.g., explicitly tutoring where experts disagree due to different schools of thought).

By applying these four high-level strategies and implementing lower-level strategies noted in the prior section, it should be possible to provide pedagogically useful instruction to a wide variety of seemingly intractable domains.

Conclusions and Future Directions

Overall, this conceptualization of ill-defined domains makes them appear significantly more tractable for an ITS than they might initially appear. A core goal of this analysis was not to fall back on the "God of the Gaps" definition for ill-defined domains, where a domain or its tasks are ill defined because existing tutoring systems cannot handle that domain. By first establishing a working definition for a domain and its components, it becomes clear that nearly every domain can be supported by adaptive ITS behaviors. While these ITS behaviors might not match the interventions characteristic of traditional ITS behavior (i.e., VanLehn, 2006), they can provide a wide variety of interventions that can be powered by alternative assessments or representations of the domain. When used systematically, hybrid ITS methodologies mean that multiple pedagogies can be combined that use different information and interventions for different tasks in the same domain.

This makes the concept of an "ill-defined domain" somewhat misleading. Even in "well-defined" domains, such as mathematics, problems such as Bertrand's Paradox (i.e., calculating probabilities based on integrals of a circle) continue to be debated and studied (Aerts & Sassoli de Bianchi, 2014). This makes the concept of trying to label domains as ill-defined or defining continuums of ill-definedness for a domain is not very productive. By comparison, identifying and categorizing different types of ill-defined tasks within a domain appears quite valuable because this allows building different types of interventions and user adaptation to the available information about each task. We advocate focusing analysis of ill-defined components and behaviors at the task level for future research. Explicit modeling of each component should help identify correlations or patterns that grow into generalizable principles for both ITS design and other types of instructional design.

Such future research will certainly be needed for ill-defined domains, since there are a large number of unresolved questions about how to handle ill-defined domains. Some of these questions are theoretical, such as proposed guidelines about how to tutor domains that have many interconnected tasks that are all impacted by the same actions. Another major question is how to identify stopping rules for building an ITS for a domain: when is a domain so fundamentally ill-defined that its scope or very existence should be challenged (e.g., tasks and/or goals are arbitrarily and unstably related)? Tasks that are ill defined due to multiple interacting learners (e.g., team learning) also need further study and integrating findings from computer-supported collaborative learning research. Other questions are practical, such as what pedagogical techniques can be efficiently authored and maintained for ill-defined domains (particularly ones where the required skills change rapidly). The long-term goal of exploring ill-defined tasks should be to move beyond the general label of "ill-defined" to establishing the specific information and affordances available for user assessment and pedagogical strategies. This mapping of task characteristics to available pedagogy would be a significant advance for the field, enabling instructional designers to rapidly understand how an ITS can interact with a learner (or learners) based on the tasks that are taught. Such a mapping would be a major asset for authoring tools and for automating ITS interactions, since it would establish boundary conditions for pedagogical strategies and clarify the possible roles of an ITS in different strategies and learning tasks.

References

Aerts, D. & Sassoli de Bianchi, M. (2014). Solving the hard problem of Bertrand's paradox. Journal of Mathematical Physics 55: 083503, doi:10.1063/1.4890291

Aleven, V., Mclaren, B. M., Sewall, J. & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. International Journal of Artificial Intelligence in Education, 19(2), 105–154.

Alexander, P. A. (1992). Domain knowledge: Evolving themes and emerging concerns. Educational Psychologist, 27(1), 33–51.

Ackoff, R. L. (1981). The Art and Science of Mess Management. Interfaces, 11(1), 20-26. doi:10.1287/inte.11.1.20
Baker, R. S. & Corbett, A. T. (2014). Assessment of robust learning with educational data mining. Research & Practice in Assessment, 9(2), 38–50.
Bellman, R. (1957). A Markovian Decision Process. Journal of Mathematics and Mechanics 6, 679–684.
Chesler, N. C., Arastoopour, G., D'Angelo, C. M., Bagley, E. A. & Shaffer, D. W. (2013). Design of professional practice simulator for educating and motivating first-year engineering students. Advances in Engineering Education, 3(3), 1-29.
Easterday, M. W., Aleven, V., Scheines, R. & Carver, S. (2011). Using Tutors to Improve Educational Games. In G. Biswas, S. Bull, J. Kay & A. Mitrovic (Eds.), Artificial Intelligence in Education (Vol. 6738, pp. 63–71): Springer Berlin Heidelberg.
FitzPatrick, B., Hawboldt, J., Doyle, D. & Genge, T. (2015). Alignment of Learning Objectives and Assessments in Therapeutics Courses to Foster Higher-Order Thinking. American journal of pharmaceutical education, 79(1), 1–8.
Fletcher, J. D. (2011). DARPA Education Dominance Program: April 2010 and November 2010 Digital Tutor Assessments. IDA Document NS D-4260. Alexandria, VA: Institute for Defense Analyses.
Fournier-Viger, P., Nkambou, R. & Nguifo, E. (2010). Building Intelligent Tutoring Systems for Ill-Defined Domains. In R. Nkambou, J. Bourdeau & R. Mizoguchi (Eds.), Advances in Intelligent Tutoring Systems (pp. 81–101): Springer Berlin Heidelberg.
Fournier-Viger, P., Nkambou, R., Mayers, A., Nguifo, E. & Faghihi, U. (2011). An Hybrid Expert Model to Support Tutoring Services in Robotic Arm Manipulations. In I. Batyrshin & G. Sidorov (Eds.), Advances in Artificial Intelligence (Vol. 7094, pp. 478–489): Springer Berlin Heidelberg.
Friedrichsen, P. J., Abell, S. K., Pareja, E. M., Brown, P. L., Lankford, D. M. & Volkmann, M. J. (2009). Does teaching experience matter? Examining biology teachers' prior knowledge for teaching in an alternative certification program. Journal of Research in Science Teaching, 46(4), 357–383.
Gipps, C. V. (1994). Beyond testing: Toward a theory of educational assessment: Falmer Press.
Grele, R. J. (1981). Whose Public? Whose History? What is the Goal of a Public Historian? The Public Historian, 3(1), 40–48.
Goldin, I. M., Ashley, K. D. & Pinkus, R. L. (2001). Introducing PETE: computer support for teaching ethics. In Proceedings of the 8th international conference on Artificial intelligence and law, St. Louis, Missouri, USA (pp. 94–98). ACM.
Gutierrez-Santos, S., Mavrikis, M. & Magoulas, G. (2010). Layered Development and Evaluation for Intelligent Support in Exploratory Environments: The Case of Microworlds. In V. Aleven, J. Kay & J. Mostow (Eds.), Intelligent Tutoring Systems (Vol. 6094, pp. 105–114): Springer Berlin Heidelberg.
Hoffman, R. R. (1998). How can expertise be defined? Implications of research from cognitive psychology. Exploring expertise, 81–100.
Ichikawa, M., Koyama, Y. & Deguchi, H. (2010). Virtual City Model for Simulating Social Phenomena. In K. Takadama, C. Cioffi-Revilla & G. Deffuant (Eds.), Simulating Interacting Agents and Social Phenomena (Vol. 7, pp. 253-264): Springer Japan.
Jacovina, M. E., Snow, E. L., Dai, J. & McNamara, D. S. (2015). Authoring Tools for Ill-defined Domains in Intelligent Tutoring Systems: Flexibility and Stealth Assessment. In R. A. Sottilare, A. C. Graesser, X. Hu & K. W. Brawner (Eds.), Design Recommendations for Intelligent Tutoring Systems: Volume 3: Authoring Tools and Expert Modeling Techniques, (pp. 109–121). US Army Research Laboratory.
Kalman, R. (1959). On the general theory of control systems. IRE Transactions on Automatic Control, 4(3), 110–110.
Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. Theory into practice, 41(4), 212-218.
Lane, H. C., Hays, M. J., Core, M., Gomboc, D., Forbell, E., Auerbach, D. & Rosenberg, M. (2008). Coaching intercultural communication in a serious game. In Proceedings of the 16th International Conference on Computers in Education (ICCE 2008), (pp. 35–42).
Lazar, T. & Bratko, I. (2014). Data-driven program synthesis for hint generation in programming tutors. In Proceedings of the 12th International Conference on Intelligent Tutoring Systems July 2014, Honolulu, HI. (pp. 306–311).
Le, N. T., Loll, F. & Pinkwart, N. (2013). Operationalizing the Continuum between Well-Defined and Ill-Defined Problems for Educational Technology. IEEE Transactions on Learning Technologies, 6(3), 258-270.
Lesgold, A., Lajoie, S., Bunzo, M. & Eggan, G. (1991). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. Larkin & R. Chabay (Eds.), Computer-assisted instruction and intelligent

tutoring systems: Shared goals and complementary approaches (pp. 201-238). Hillsdale, NJ: Lawrence Erlbaum Associates.
Lynch, C. & Ashley, K. (2014). Empirically valid rules for ill-defined domains. In Educational Data Mining (EDM) 2014, (pp. 237-240). London, U.K.
Lynch, C., Ashley, K.D., Pinkwart, N. & Aleven, V. (2009). Concepts, structures, and goals: Redefining ill-definedness. International Journal of Artificial Intelligence in Education, 19(3), 253-266.
Matsuda, N., Cohen, W. W., Sewall, J., Lacerda, G. & Koedinger, K. R. (2007). Evaluating a Simulated Student Using Real Students Data for Training and Testing User Modeling 2007 (pp. 107–116), Springer.
Mitrovic, A., Koedinger, K. & Martin, B. (2003). A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling. In P. Brusilovsky, A. Corbett & F. de Rosis (Eds.), User Modeling 2003 (pp. 313–322). Springer Berlin Heidelberg.
Mitrovic, A. & Ohlsson, S. (2015). Implementing CBM: SQL-Tutor After Fifteen Years. International Journal of Artificial Intelligence in Education, 1–10. doi:10.1007/s40593-015-0049-9.
Mitrovic, A. & Weerasinghe, A. (2009). Revisiting ill-definedness and the consequences for ITSs. Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009). V. Dimitrova and R. Mizoguchi (Eds), 375–382. IOS Press.
Nash, G. B., Crabtree, C. A. & Dunn, R. E. (2000). History on trial: Culture wars and the teaching of the past (2nd ed.). New York, NY: Vintage.
Norman, D. A. (1983). Some observations on mental models. Mental models, 7(112), 7–14.
Nye, B., Bharathy, G., Silverman, B. & Eksin, C. (2012). Simulation-Based Training of Ill-Defined Social Domains: The Complex Environment Assessment and Tutoring System (CEATS). In S. Cerri, W. Clancey, G. Papadourakis & K. Panourgia (Eds.), Intelligent Tutoring Systems (Vol. 7315, pp. 642–644): Springer Berlin Heidelberg.
Nye, B. D., Goldberg, B. & Hu, X. (2015). Generalizing the Genres for ITS: Authoring Considerations for Representative Learning Tasks. Design Recommendations for Intelligent Tutoring Systems: Volume 3: Authoring Tools and Expert Modeling Techniques (pp. 47–64). US Army Research Laboratory.
Ososky, S. J. (2013). Influence of Task-Role Mental Models on Human Interpretation of Robot Motion Behavior (Doctoral dissertation, University of Central Florida, Orlando, Florida).
Pinkwart, N., Aleven, V., Ashley, K. & Lynch, C. (2007). Evaluating legal argument instruction with graphical representations using LARGO. In Frontiers in Artificial Intelligence and Applications, 158, (pp. 101–108). Springer: Berlin.
Pinkwart, N., Ashley, K., Lynch, C. & Aleven, V. (2009). Evaluating an intelligent tutoring system for making legal arguments with hypotheticals. International Journal of Artificial Intelligence in Education, 19(4), 401–424.
Richards, N., Moriarty, D. & Miikkulainen, R. (1998). Evolving Neural Networks to Play Go. Applied Intelligence, 8, 85–96,
Rittel, H. & Webber, M. (1973). Dilemmas in a general theory of planning. Policy Sciences 4(2), 155–169 (1973)
Ritter, S. (2015). Authoring for the product lifecycle. Design Recommendations for Intelligent Tutoring Systems: Volume 3: Authoring Tools and Expert Modeling Techniques (pp. 137–144). US Army Research Laboratory.
Roberts, N. (2000). Wicked problems and network approaches to resolution. International Public Management Review 1(1), 1–19.
Scheuer, O., McLaren, B. M., Loll, F. & Pinkwart, N. (2012). Automated analysis and feedback techniques to support and teach argumentation: A survey. Educational technologies for teaching argumentation skills, 71–124.
Schraagen, J. M., Chipman, S. F. & Shalin, V. L. (2000). Cognitive task analysis. Mahwah, NJ: Lawrence Erlbaum Associates.
Silverman, B.G., Bharathy, G.K., Nye, B.D., Kim, G.J., Roddy, M. & Poe, M. (2010) M&S methodologies: A systems approach to the social sciences. In Sokolowski, J.A. & Banks, C.M. (eds.) Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains, (pp. 227-270). Wiley & Sons, Hoboken, NJ.
US Army (2006). Counterinsurgency Field Manual 3-24. Department of the Army.
Vanlehn, K. (2006) The behavior of tutoring systems. International Journal of Artificial Intelligence in Education 16(3), 227-265 (2006)
Waldmann, M. & Dieterich, J. (2007) Throwing a bomb on a person versus throwing a person on a bomb. Psychological Science 18(3), 247.
Wiggins, G. P. & McTighe, J. (2005). Understanding by design (2nd Ed.): Pearson.

Woods, A., Stensrud, B., Wray, R. E., Haley, J. & Jones, R. M. (2015). A Constraint-Based Expert Modeling Approach for Ill-Defined Tutoring Domains. In Florida Artificial Intelligence Research Society (FLAIRS) 2015 (pp. 469–474). AAAI Press: Hollywood, FL.

CHAPTER 3 – Methods to Refine the Mapping of Items to Skills

Michel C. Desmarais and Peng Xu
Polytechnique Montréal

Introduction

A critical part of domain modeling in intelligent tutoring systems (ITSs) is to determine how the domain content relates to the skills and knowledge that we aim for the student to learn about. This task can be achieved through data-driven techniques. If domain content is associated with a set of questions or exercises, student performance data over these items can be used to find the latent skills behind the content.

Data-driven techniques that map items to skills fall within two main categories: (1) entirely driven from student data performance or (2) starting with expert-given mapping and refined based on these data. Entirely data-driven techniques are appealing because they dispense the tedious efforts required to do the mapping by content experts. However, the mapping obtained from such methods may be hard to interpret and will almost surely contain latent skill factors that do not match the pedagogical structure of the learning content. For this reason, the refinement of expert given mappings has greater utility in most contexts and this chapter focuses on this specific problem.

Another fundamental distinction to be made is whether the data include a time component. In learning environments, students learn as they interact with the system. A single student's skills mastery profile changes in time and within the same data sample. A number of studies have focused on refining the item to skills mapping with this type of dynamic data, namely, Stamper and Koedinger (2011), Koedinger et al. (2013), and Aleven and Koedinger (2013). Simplifying measures such as taking into account only the first attempt and ignoring hints and scaffolding can alleviate the complexity of analyzing this type of data (see, for example, González-Brenes, 2015).

The modeling of task to knowledge components (KCs)/skills for models that include a time dimension, and from data that has a time dimension, is the focus of a previous volume's chapter of the current series (Aleven and Koedinger, 2013). In this chapter, we focus on static data, where we assume the data are a snapshot in time of the student's skill profile and return in the discussion on the question of how this assumption can be dealt with.

Background Concepts

Before discussing the item to skills refinement models, let us introduce the background concepts and models.

Q-Matrices

The mapping of items to latent skills is often referred to as a Q-matrix. Rows of the matrix are the items and columns are the skills. Q-matrices are generally represented as Boolean matrices. The items can represent question, exercises, or any task that has a clear outcome. Skills are also termed KCs (Aleven & Koedinger, 2013).

There are three distinct ways how skills can be considered related to tasks:

- *Conjunctive*: all skills are required to successfully complete the task.
- *Disjunctive*: any skill is sufficient.
- *Compensatory or additive*: all skills contribute to increase the chances of success.

All three versions of Q-matrices obviously entail different student skills models that might be used in a running system. Furthermore, skills can be continuous or discrete, which also entails different models, as does the time factor. If learning occurs in the data, the models have to account for it.

However, we should emphasize that although the skills model depends on the type of Q-matrix and on whether the data have a time dimension, the Q-matrix does not. Whether a task involves skills or not is independent of the model and of the time factor.

Factorization Framework

A valuable framework for conceptualizing the relationship between a Q-matrix $\mathbf{Q}$, the student skill profiles matrix $\mathbf{P}$, and the student test outcome results matrix $\mathbf{R}$, is to define the relationship as a product:

$$\mathbf{R} = \mathbf{P}\mathbf{Q}^{\mathrm{T}} \tag{1}$$

For example, the following matrix product would correspond to a compensatory version of a Q-matrix in which the rows are normalized to sum to 1:

$$\mathbf{R} = \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \end{array} \begin{array}{c} \begin{array}{ccc} i_1 & i_2 & i_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1/2 \\ 1 & 1 & 1/2 \end{bmatrix} \end{array} = \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \times \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1/2 & 1/2 \end{bmatrix} \end{array}^{\mathrm{T}}$$

Using the negation operator $\neg$ defined as

$$\neg x = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

we can define a conjunctive Q-matrix as the product (assuming a normalized Q-matrix)

$$\neg\mathbf{R} = \neg\mathbf{P} \times \mathbf{Q}^{\mathrm{T}} \tag{3}$$

and a disjunctive Q-matrix as

$$\neg\mathbf{R} = \mathbf{P} \times \mathbf{Q}^{\mathrm{T}} \tag{4}$$

by redefining the first condition of the negation operator ($1 \quad \text{if} \quad x > 0$).

Cognitive Diagnosis Models and Q-Matrix Refinement

Every Q-matrix refinement model has an underlying model, often called a cognitive diagnosis model (CDM). Common models can be described using the matrix factorization framework. For example, the DINA and DINO models described below.

DINA and DINO are two well-known models for cognitive modeling. The DINA model relies on a conjunctive Q-matrix that corresponds to Eq. 3, whereas the DINO relies on a disjunctive Q-matrix that corresponds to Eq. 4. However, these models also include the guess and slip factors that, respectively, define the probability of a good response given a predicted 0 in the $\mathbf{R}$ matrix and a probability of an incorrect response given a predicted 1. An adaptation of the factorization framework to accommodate the uncertainty introduced in DINA and DINO is to consider the $\mathbf{R}$ matrix as a probability matrix and to substitute the 0's and the 1's, respectively, by the $guess$ and $(1-slip)$ factors.

Let us introduce a distinction in the notation and refer from here on to $\hat{\mathbf{R}}$ as the predicted student response outcome (for example, the product of an estimated student profiles matrix, $\hat{\mathbf{P}}$, and a Q-matrix, $\mathbf{Q}$) and to $\mathbf{R}$ as an observed student response outcome matrix.

Many models consider the estimated student response outcome matrix, $\hat{\mathbf{R}}$, to be a probability matrix. For example, taking the logit of $\hat{\mathbf{R}}$ ($\log(r_{ij} / (1-r_{ij}))$ and the log of $\hat{\mathbf{P}}\mathbf{Q}^{\mathrm{T}}$ leads to flexible log-odds CDM (Hensen, Templin & Willse, 2009).

But regardless of the specific model used for Q-matrix refinement, the principle of refinement follows a general framework. Given a cognitive diagnostic model and an estimated student profiles matrix $\hat{\mathbf{P}}$, the refinement process can be considered as searching the space of Q-matrices for a matrix that will minimize the difference between $\mathbf{R}$ and $\hat{\mathbf{R}}$. The difference can be the RSS ($\| \mathbf{R} - \hat{\mathbf{R}} \|^2$), or any other reasonable loss function.

In fact, minimizing the loss function is the same objective whether we are looking at estimating the student profiles or any other parameter of the model. What is unique to the process of Q-matrix refinement is the initial starting point: the expert's matrix. If we were to look for a Q-matrix that solely optimizes prediction, we could instead start from a random Q-matrix or use heuristics to guess a strategic initial Q-matrix. In other words, the process looks for a local minima starting from a point in the Q-matrix space that corresponds to the expert Q-matrix.

Refinement Methods

The literature on Q-matrix refinement methods has exploded in recent years, both for static data (de la Torre & Chiu, 2015; Barnes, 2010; Desmarais & Naceur, 2013; Xiang, 2013; Chung, 2014; H. Li & Suen, 2013; Qin et al., 2015; Romero, Ordoñez, Ponsoda & Revuelta, 2014; Köhn, Chiu & Brusco, 2015; Nižnan, Pelánek & Řihák, 2014; Xu & Zhang, 2015) and dynamic data in which student learning occurs (Stamper & Koedinger, 2011; Koedinger et al. 2013; Aleven & Koedinger, 2013; González-Brenes, 2015; N. Li, Cohen, Koedinger & Matsuda, 2011). In addition to algorithms that take as input student response outcome data, we find the emergence of methods that integrate text analysis to label and optimize the

search for better Q-matrices (Goutte, Léger & Durand, 2015; N. Li, Cohen & Koedinger, 2013; Matsuda, Furukawa, Bier & Faloutsos, 2014).

This is a clear sign of the importance of the problem as well as of the vitality of the research on the topic. Some studies have shown that using data-driven techniques of refinement generally result in Q-matrices that have better predictive power and are a better fit to the data in general (Aleven & Koedinger, 2013; Durand, Belacel & Gutte, 2015; Matsuda et al., 2014).

We first review three examples of Q-matrix refinement algorithms and show how such algorithms can be combined to obtain substantial gains in the next section. The first two are based on the DINA model, whereas the last one is based directly on Eq. 3.

minRSS

For a given Q-matrix, there is an ideal response pattern (ideal response vector) for each skill pattern (profile vector). If there are no slip and guess factors, then the response pattern for every category of student profile is fixed. A reasonable assumption is to assume the real response pattern should not differ much from this ideal response pattern. Then the problem is how to measure the difference between the real pattern and ideal pattern. The most common metric for binary data is Hamming distance, that is,

$$d_h(r,\eta) = \sum_{j=1}^{J} | r_j - \eta_j |$$

where r is the real response vector while η is the ideal response vector. J is the number of latent skills. Chiu and Douglas (2013) refined this metric based on the idea that if an item has a smaller variance (or entropy), then it should be given higher weight. The formula is

$$d_{\omega h}(r,\eta) = \sum_{j=1}^{J} \frac{1}{\overline{p}_j(1-\overline{p}_j)} | r_j - \eta_j |$$

where $\overline{p}_j$ is the proportion of correct answers of item j. Equipped with this metric, we can find the ideal response matrix that best fits the data, and then find the correspondent profile matrix $\mathbf{P}$. With these results, a powerful method was proposed to update the Q-matrix (Chiu, 2013). First, a squared sum of errors for each item k can be computed by

$$RSS_k = \sum_{i=1}^{N} (r_{ik} - \eta_{ik})^2$$

where N is the number of respondents. Then, the item with the highest RSS is chosen to update its correspondent q-vector. All the other possible q-vectors are tested to calculate their RSS and the q-vector giving the lowest RSS is chosen to replace the original one. We name this method minRSS based on this minimization objective. The Q-matrix is thus updated accordingly, and the whole process is repeated. The previous changed q-vector is taken out of searching pool for the next iteration. The whole procedure terminates when the RSS for each item no longer changes. This method has a consistency property, which

was shown by Wang & Douglas (2015). That is, it has good performance under different underlying conjunctive models.

maxDiff

Under the setting of DINA model, for every item j, there are two model parameters: slip s_j and guess g_j. De la Torre (2008) proposed that a correctly specified q-vector for item j should maximize the difference of probabilities of a correct response between examinees who have all the required attributes and those who do not. For a model involved with K possible skills, there are 2^K possible q-vectors (i.e., skill combination). Denote these possible q-vectors by α_l, $l = 0,1,...,2^K - 1$, then q_j is the correct q-vector if

$$q_j = arg\,max_{\alpha_l}[P(X_j = 1 | \xi_{ll'} = 1) - P(X_j = 1 | \xi_{ll'} = 0)] = arg\,max_{\alpha_l}[\delta_{jl}]$$

for $l, l' = 1, 2, ..., 2^K - 1$ and $\xi_{ll'} = \prod_{k=1}^{K} \alpha_{lk}^{\alpha_{l'k}}$. That is also why we call it maxDiff. An interesting observation is since $P(X_j = 1 | \xi_{ll'} = 1) = 1 - s_j$ and $P(X_j = 1 | \xi_{ll'} = 0) = g_j$, then

$$q_j = arg\,max_{\alpha_l}[1 - (s_j + g_j)]$$

that is, maximizing the difference is equivalent to minimize the sum of the slip and guess parameters. A natural idea is to test all q-vectors to find the maximum δ_{jl} but that is computationally expensive. De la Torre (2008) proposed a greedy algorithm that adds skills into a q-vector sequentially. First, δ_{jl} is calculated for all q-vectors that contain only one skill and the one with biggest δ_{jl} is chosen. Then, δ_{jl} is calculated for all q-vectors that contain two skills including the previously chosen one. Again the q-vector with the largest δ_{jl} is chosen. This whole process is repeated until no skills increases δ_{jl}. However, this algorithm requires knowing s_j and g_j in advance. For real data, they are calculated by an expectation maximization (EM) algorithm (de la Torre, 2009).

ALSC

Conjunctive alternating least squares factorization (ALSC) is a common matrix factorization (MF) technique. Desmarais and Naceur (2013) proposed to factorize student test results into a Q-matrix and a skills-student matrix with a least squares estimate.

Contrary to the other two methods, it does not rely on the DINA model as it has no slip and guess parameters. ALSC decomposes the results matrix $\mathbf{R}$ based on the least squares estimate.

The factorization consists of alternating between estimates of $\mathbf{P}$ and $\mathbf{Q}$ until convergence. Take conjunctive model, for example, starting with the initial expert-defined Q-matrix $\mathbf{Q}_0$, an initial least-squares estimate of $\mathbf{P}$ is obtained:

$$\neg\hat{\mathbf{P}}_0 = \neg\mathbf{R}\mathbf{Q}_0(\mathbf{Q}_0^{\mathrm{T}}\mathbf{Q}_0)^{-1} \tag{5}$$

which is the least squares solution of Eq. 3. Then, a new estimate of the Q-matrix, $\hat{\mathbf{Q}}_1$, is again obtained by the least-squares estimate:

$$\hat{\mathbf{Q}}_1^{\mathrm{T}} = (\neg\hat{\mathbf{P}}_0^{\mathrm{T}}\neg\hat{\mathbf{P}}_0)^{-1}\neg\hat{\mathbf{P}}_0^{\mathrm{T}}\neg\mathbf{R} \tag{6}$$

and so on until convergence. Alternating between Eqs. 5 and 6 yields progressive refinements of the matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{P}}_i$ that more closely approximate $\mathbf{R}$, the observed student response outcome matrix. The final $\hat{\mathbf{Q}}_\mathbf{i}$ is rounded to yield a binary matrix.

Combining Techniques with Ensemble Algorithms

The effectiveness of a Q-matrix refinement technique may depend on the specific characteristics of the data or on the characteristics of the matrix itself. An algorithm that can learn the conditions under which an approach is more likely to give a reliable answer can, in principle, provide better refinements than any algorithm alone.

A critical factor for the success of the combination approach is defining the effective factors that best show how to combine the output of the Q-matrix refinement algorithms. Let us focus on one of these factors, stickiness, as an informative example.

Stickiness represents the rate of a given algorithm's false positives for a given cell of a Q-matrix. A false positive is considered a recommended change in the Q-matrix when the ground truth tells us it is wrong, that is, no changes should be recommended. The rate is measured by "perturbating" in turn each and every cell of the Q-matrix and counting the number of times the cell is a false positive. The decision tree can use the stickiness factor as an indicator of the reliability of a given Q-matrix refinement algorithm's suggested value for a cell. Obviously, if a cell's stickiness value is high, the reliability of the algorithm's suggestion will be lower.

The question is, how does one train a decision tree with enough data to use stickiness and other factors? An original idea introduced in our approach is to use synthetic data for which we know the Q-matrix ground truth. Random matrices with a similar ratio of 0/1 are generated and the perturbation process described above is applied to generate tens of thousands of tuples with the following elements:

(1) Target value

(2) Predicted values from the three algorithms studied

(3) Stickiness

(4) A few other characteristics of the skill and the item involved

These elements represent the input to a decision tree that essentially learns which of the predicted values (2) are most likely to be correct given the contextual factors (3 and 4)

Based on the decision-tree combination approach described above, Desmarais et al. (2015) obtained a substantial gain in accuracy over the best of three refinement algorithms. Considering on an equal basis an error as not recovering the perturbated cell and recommending changes to non-perturbated cells, they obtained an error reduction in the range of 50% using real data over Q-matrices defined by experts and around 85% using DINA-based synthetic data for the same matrices.

The combination approach can be considered an ensemble technique in the machine learning field. Another well-known ensemble technique is to combine a decision tree with boosting. Boosting consists in assigning a weight to each individual observation in the loss function. The weight is increased when the predicted value differs from the observed one, and the classifier, namely, the decision tree in our case, is trained with the new weighted loss function. Using the Adaboost boosting algorithm, an additional improvement in error reduction of 18% for real data and 46% for synthetic data was obtained (Xu & Desmarais, 2016).

In terms of correct and incorrect refinements, the ensemble technique, which combines the three algorithms with a boosted decision tree, is able to recover almost all of the perturbated cells to their original value. It improves the rate of recall of perturbated cells from around half to close to all. However, despite these improvements, it still introduces a small number of incorrect refinements (proposed changes to cells that were not perturbated). These incorrect refinements can prove disrupting to an expert who uses a Q-matrix refinement tool, as it entails an effort to analyze and assess the proposed refinements, and future efforts should focus on reducing this number.

Discussion

An important finding from the work with ensemble techniques is the demonstration of the complementarity of Q-matrix refinement algorithms, at least with the three algorithms used in the ensemble studies reported in the previous section. The gains to recover perturbated cells to their original value are remarkable and the performance is close to perfect. While the number of false refinements remains disrupting, it is not to the extent that they undermine the value of the recommended changes.

Can ensemble techniques extend to algorithms to map items to skills with dynamic data? Can we assume the algorithms to refine the mapping of item to skills/KCs for dynamic data are also complementary? It is reasonable to believe that similar gains could be obtained if we can assume the complementarity is present. Simplification such as using the first trial (for example, González-Brenes, 2015) may permit greater variability of approaches, and therefore, complementarity in the sources of information that can be combined in an ensemble technique.

Recommendations and Future Research

The Cognitive Tutor Authoring Tools (CTAT) described in Aleven and Koedinger (2013) for helping an expert to map exercises, items, and tasks to underlying skills are excellent examples of valuable outcomes of models and algorithms that can use data to help an expert refine a Q-matrix (or a KC model in their terminology). These tools allow the comparison of Q-matrix versions, assessment of their fit in terms of predictive power and other measures of fit, and provide different means to refine them.

The approaches and algorithms reviewed in this chapter should lead to such tools, or complement them. They help identify weaknesses in a Q-matrix at the item level, pinpointing missing or irrelevant skills associated with tasks and also at the skill level: a whole column may show anomalies that suggest a skill may be ill defined and unfit to the data.

As emphasized before, the Q-matrix refinement algorithms reviewed in this chapter make the assumption that learning does not occur in the data. This is a serious limitation for data collected from learning environments where learning does occur. Important questions to address are what are the consequences of the violation of this assumption and how can we mitigate the adverse effects and work around them?

An avenue to explore is to transform the dynamic data into a static view. The dynamic cognitive modeling models can generally be conceptualized as a factorization model, akin to the factorization of Eq. 1, but the $\mathbf{P}$ and the $\mathbf{R}$ matrices have a third dimension, which is time. They are then considered tensors models (see Thai-Nghe, Horváth & Schmidt-Thieme, 2011). In such models, it is possible to predict the student response outcome data at a given time slice, thereby transforming the dynamic data into a static view that the models reviewed in this chapter can handle.

References

Aleven, V. & Koedinger, K. R. (2013). Knowledge component (KC) approaches to learner modeling. Design Recommendations for Intelligent Tutoring Systems, 189–203.

Barnes, T. (2010). Novel derivation and application of skill matrices: The Q-matrix method. Handbook on educational data mining, 159–172.

Chen, Y., Liu, J., Xu, G. & Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. Journal of the American Statistical Association, 110(510), 850–866.

Chiu, C.-Y. (2013). Statistical refinement of the Q-matrix in cognitive diagnosis. Applied Psychological Measurement, 37(8), 598–618.

Chiu, C.-Y. & Douglas, J. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. Journal of Classification, 30(2), 225–250.

Chung, M.-t. (2014). Estimating the Q-matrix for cognitive diagnosis models in a Bayesian framework (Unpublished doctoral dissertation). Columbia University.

de la Torre, J. (2008). An empirically based method of Q-matrix validation for the DINA model: Development and applications. Journal of educational measurement, 45(4), 343–362.

de la Torre, J. (2009). Dina model and parameter estimation: A didactic. Journal of Educational and Behavioral Statistics, 34(1), 115–130.

de la Torre, J. & Chiu, C.-Y. (2015). A general method of empirical Q-matrix validation. Psychometrika, 1–21.

Desmarais, M. C. & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In Artificial intelligence in education (pp. 441–450).

Desmarais, M. C., Xu, P. & Beheshti, B. (2015). Combining techniques to refine item to skills q-matrices with a partition tree. In Educational data mining 2015.

Durand, G., Belacel, N. & Goutte, C. (2015). Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In Design for teaching and learning in a networked world (pp. 56–69). Springer.

González-Brenes, J. P. (2015). Modeling skill acquisition over time with sequence and topic modeling. In Aistats.

Goutte, C., Durand, G. & Léger, S. (2015). Towards automatic description of knowledge components. In Proceedings of the tenth workshop on innovative use of nlp for building educational applications (pp 75–80).

Goutte, C., Léger, S. & Durand, G. (2015). A probabilistic model for knowledge component naming. In Proceedings of the 8th international conference on data mining (pp. 608–609).

Henson, R. A., Templin, J. L., &Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. Psychometrika, 74(2), 191–210.

Koedinger, K. R., Stamper, J. C., McLaughlin, E. A. & Nixon, T. (2013). Using data-driven discovery of better student models to improve student learning. In Artificial intelligence in education (pp. 421–430).

Köhn, H.-F., Chiu, C.-Y. & Brusco, M. J. (2015). Heuristic cognitive diagnosis when the Q-matrix is unknown. British Journal of Mathematical and Statistical Psychology, 68(2), 268–291.

Li, H. & Suen, H. K. (2013). Constructing and validating a Q-matrix for cognitive diagnostic analyses of a reading test. Educational Assessment, 18(1), 1–25.

Li, N., Cohen, W. W. & Koedinger, K. R. (2013). Discovering student models with a clustering algorithm using problem content.

Li, N., Cohen, W.W., Koedinger, K. R. & Matsuda, N. (2011). A machine learning approach for automatic student model discovery. In Edm (pp. 31–40).

Matsuda, N., Furukawa, T., Bier, N. & Faloutsos, C. (2014). Machine beats experts: Automatic discovery of skill models for data-driven online course refinement. In Educational data mining 2014 (pp. 101–108).
Nižnan, J., Pelánek, R. & Řihák, J. (2014). Mapping problems to skills combining expert opinion and student data. In Mathematical and engineering methods in computer science (pp. 113–124). Springer.
Qin, C., Zhang, L., Qiu, D., Huang, L., Geng, T., Jiang, H., . . . Zhou, J. (2015). Model identification and Q-matrix incremental inference in cognitive diagnosis. Knowledge-Based Systems, 86, 66–76.
Romero, S. J., Ordoñez, X. G., Ponsoda, V. & Revuelta, J. (2014). Detection of Q-matrix misspecification using two criteria for validation of cognitive structures under the least squares distance model. Psicologica: International Journal of Methodology and Experimental Psychology, 35(1), 149–169.
Roussos, L. A., DiBello, L. V., Stout, W., Hartz, S. M., Henson, R. A. & Templin, J. L. (2007). The fusion model skills diagnosis system. Cognitive diagnostic assessment for education: Theory and applications, 275–318.
Sinharay, S. & Almond, R. G. (2007). Assessing fit of cognitive diagnostic models a case study. Educational and Psychological Measurement, 67(2), 239–257.
Stamper, J. C. & Koedinger, K. R. (2011). Human-machine student model discovery and improvement using datashop. In Artificial intelligence in education (pp. 353–360).
Templin, J. (2015). Diagnostic assessment. Technology and Testing: Improving Educational and Psychological Measurement, 285.
Thai-Nghe, N., Horváth, T. & Schmidt-Thieme, L. (2011). Factorization models for forecasting student performance. In Edm (pp. 11–20).
Wang, S. & Douglas, J. (2015). Consistency of nonparametric classification in cognitive diagnosis. Psychometrika, 80(1), 85–100.
Xiang, R. (2013). Nonlinear penalized estimation of true Q-matrix in cognitive diagnostic models (Unpublished doctoral dissertation). Columbia University.
Xu, P. & Desmarais (2016). Boosted Decision Tree for Q-matrix Refinement. 9th International Conference on Educational Data Mining (EDM2016), Raleigh, NC (submitted).
Xu, G. & Zhang, S. (2015). Identifiability of diagnostic classification models. Psychometrika, 1–25.

CHAPTER 4 – Ontology Alignment of Learner Models and Domain Models

Douglas B. Lenat
Cycorp

Introduction

The people designing and building an intelligent tutoring system (ITSs) certainly have some domain model and user model (or distribution of likely user models) in mind. It is possible to completely "compile those out" so that the ITS has no explicit representation of either model – e.g., a fixed decision tree determines which problem to present to the learner depending on responses to the earlier problems – but there is a tremendous wasted opportunity, if that course is followed.

To put this positively, there is great potential value to explicitly representing the domain and learner models in the ITS. For example, if the learner model tracks the ongoing progress of the learner, the teacher may inspect that model during or after the use of the ITS to assess whether continued use of the ITS is cost-effective and determine what ITS might best serve the learner next.

This chapter focuses on an even more exciting benefit to having those models explicit in the ITS: the synergy of shared models across multiple ITSs that were developed completely independently of each other. The following is a preview a couple cases of this:

A learner has worked for a while with ITS#1 and starts to use ITS#2. Instead of it having to build up a model of that learner from scratch, it imports the ITS#1 learner model and maps it (using an ontology#1 ←→ ontology#2 mapping) into a learner model it can use, treating the learner as correctly and individually as though that person had worked with ITS#2 for all that time.

ITS#2 teaches generalization and specialization relationships using a certain game, let's say one involving helicoptering around in 3D node-and-link space. The ITS#2 developer wants to extend this into a new domain – let's say from biology to ecology – and would normally have to build up a domain model in that new area. But instead, the developer imports the ITS#1 domain model of ecology, perhaps combining a few types of generalization relationships that ITS#1 distinguishes into one relationship and ITS#2 can then immediately

An *ontology* is a vocabulary of terms – representing individuals, types, and relationships or predicates – along with a set of rules/assertions/axioms involving those terms.

Think of the terms like English words, and the assertions as sentences. Some of the relations are taxonomic (element-of, superset, etc.), some specify type constraints on arguments of relations, and so on. Together, the grammatical sentences are said to be *well-formed.* Some of these representation
languages stop there, only allowing (r a b) triples; some allow variables and quantifiers like for-all and there-exists; some even allow some of the assertions to be about other assertions, to be about modal attitudes such as "Israel fears that Syria wants the UK to prohibit…."

Using this machinery, the assertions can be assembled into a coherent knowledge base, to which some inference procedures can be mechanically carried out, and in the process, some application functionality can result, much as though a program had been written to do some application of interest. In this case, though, the only "program" is the general inference engine.

begin teaching learners about ecology by having them helicopter around the imported domain model.

Individual ITS-ITS pairwise mappings hold great potential, and the sheer number of possibilities here is astronomical.[1]

All this hinges on being able to work out the mapping or alignment of the ontology elements, which is generally going to be a many-to-many mapping, perhaps involving some conditional rewriting rules that make it more accurate than just leaving these links as "system#1's x is-related-to system#2's y"). That can be a substantial manual task, to do those mappings and write those rules, as there may be thousands of domain model elements, and/or learner model elements, in each of system#1's and system#2's ontologies. But there are some exciting ways in which that can be partially or even completely automated, at least in certain cases, and this chapter concludes by presenting some of those promising research directions.

Discussion

Many excellent articles and surveys and entire books have been written on ontology alignment (Otero-Cerdeira et al., 2015; Huang, 2008; Shvaiko & Euzenat, 2013). Therefore, rather than trying to be comprehensive here, we present a few "steps along the way" from manual to increasingly automated ontology alignment, culminating in some novel methods that are particularly suited to ITSs.

We recently spent about 50 hours manually aligning our Cyc-based MathCraft™ (Lenat & Durlach, 2014) middle-school math domain and learner ontologies with the Carnegie Learning Group's ontologies (Fancsali & Ritter, 2014), each of which contained on the order of 1000 elements representing concepts (e.g., commutativity), skills (e.g., borrowing when needed during subtracting), and bugs (e.g., borrowing even when it is not necessary). In both cases, the learner model is largely isomorphic to the domain model, but with some meta-level indication of whether and how this learner has mastered or is confused by this domain element. The time-consuming part of the mapping comes because each ontology was developed independently, starting from a common core – in this case, literally The Common Core standard – which unfortunately only provides on the order of tens of distinct concepts and skills, two full orders of magnitude less than both Cycorp and CLG empirically found necessary to have in their respective ITSs.

A typical one-to-one mapping was between the Cyc ontology term "(FindingTheVolumeOfShapeTypeFn Wedge)" and the CLG term "Calculating Volume of Right Prisms: Find triangular prism volume".

A typical one-to-many mapping was between the Cyc term "AddingDenominatorsOfFractions" and the dozen CLG terms for actually doing such additions in different cases, such as when both numbers did vs. did not have any integral part and the denominators were the same vs. different.

A typical many-to-many mapping that had to be identified and articulated was "Adding involving small whole numbers and no carrying" (and similarly but with carrying) on the Cycorp side and "Calculate sum digit with 1 digit" (and similarly but with 2 digits) on the CLG side. In this case, articulating the mapping means writing the Cyc rule, which explains that if the addition happens to involve 1 vs. 2 digits then map to the …1… vs. …2… CLG ontology term; and writing the CLG rule, which explains that if the addition happens to involve carrying then map to the …with… vs. …no carrying… Cyc ontology term.

[1] At an April 29, 2015, workshop on ITS ontologies hosted by and at Advanced Distributed Learning (ADL) in Orlando, FL, Prasad Ram of Ednovo reported having compiled a database of over 6 million available online ITS resources for K–12 students; that sample alone admits 36 trillion potentially valuable pairwise alignments.

When the mapping is not one-to-one, stating the articulation rules is completely optional, but doing so provides much more traction than just leaving things at "this set S1 of Cyc ontology terms is closely related to this set S2 of CLG ontology terms".

This manual process is tedious and labor-intensive enough when the number n of ontologies is 2, but threatens to increase quadratically with n; squaring the previously mentioned 6 million available resources and multiplying by 50 hours yields an appropriately daunting number of thousands of trillions of person-hours of work looming ahead, if we were to do this manually.

One solution would be to develop and agree on a sort of "gold standard" domain ontology (for each domain) and learner model ontology (for each type of situation related to the type of task, the type of domain, and the type of learner). Then the alignment task drops back from quadratic in the number n of ontologies to linear in n. Often the situation is that the gold standard ontology has not yet been worked out or agreed to, or (as in the case of The Common Core) is far too coarse-grained to serve the necessary function[2] or there are too many separate "standards", often for slightly different purposes, such as healthcare's morass of HL7, DICOM, IHE, ICD, LOINC, MeSH, UMLS, SNOMED, GALEN, etc.

A step toward automatically aligning two ontologies is to algorithmically notice similarities between them and use that to "grow" the mapping.

Sometimes this can be as obvious as string similarity between the name of a term in one ontology and the other. One can point out places this heuristic fails (e.g., when a term such as myocardial infarction is redefined but the old term is kept around with the new meaning), but it is not a bad heuristic to use when it happens to apply.

Often the topology of the knowledge graphs (combined with the growing partial mapping) will suggest new candidates for aligning. A set with exactly 50 elements, one of which appears to be the state of Texas, is likely to be the set of US states. More subtly, in the diagram in Figure 1, where corresponding terms for Mammal and Dog (and the subset relationship) are already known, one of the two question-mark terms is likely to correspond to the concept of Canine:

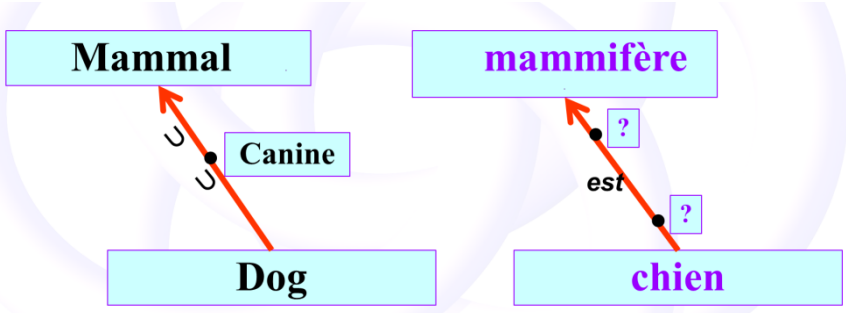


Figure 1. Given the Mammal-mammifère and Dog-chien and subset-est alignments, one of the two "?" concepts is likely to correspond to "Canine".

[2] In some domains, an *enumerative* ontology exists that has the opposite problem: it is actually too verbose! For example, the medical International Classification of Diseases (ICD) ontology takes all the independent attributes and combinatorically explodes the size of that ontology to the point where there is a separate numbered term for, say, congenital enlargement of the left pulmonary artery, rather than factoring out separately independent attributes like congenital vs. acquired, enlargement vs. diminution, left vs. right, and so on.

If one has massive amounts of data available, statistical correlation can sometimes help suggest extensions to an existing partial alignment of two ontologies. Let's suppose that millions of students have taken standardized tests 1 and 2, and we graph the frequency of correct responses requiring particular skills and conceptual knowledge (Figure 2). If we can rearrange the columns of the two test result graphs so that the peaks' heights match – so that the spectra look the same – then there is a chance that the particular meaning of each peak in test#1's empirical results corresponds to the concept or skill that is the particular meaning of the corresponding peak in test#2's empirical results. This sounds of very dubious reliability, but with sufficient data it can work, especially in cases where the peaks are of mutually distinctive heights. In effect, this is tapping into the same source of power that Amazon and Netflix use when they recommend a book or movie you might enjoy, based on similarities in buying/satisfaction data.

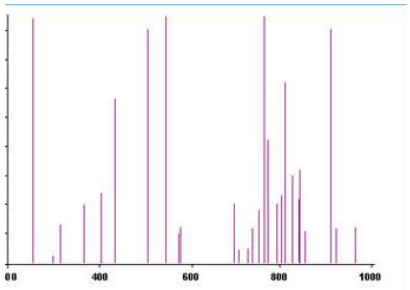


Figure 2. The distribution of scores (y-axis) of millions of students on various skills (x-axis) in two separate but similar tests with different score *labels* for the same N skills. Rearranging the columns to make the two spectra appear maximally similar, one can weakly infer that the corresponding *labels,* in order, now pertain to what test#1 and test#2 happen to call the same underlying skill.

To see how one could do something better than these statistical sorts of graph-matching heuristics, consider the 35+17 math problem that a student is solving incorrectly in Figure 3a. Even though there is only one data point, one can quickly guess that what the student is doing wrong is failing to carry a 1, getting 42 instead of the correct answer 52. In the next example, Figure 3b, the student is "obviously" just adding the numerators and adding the denominators to get their answer: It's obvious because you have models of how students do addition, models of the common bugs they have in those models, and so on.

(a)

$$\begin{array}{r} 35 \\ +17 \\ \hline 42 \end{array}$$

(b)

$$\frac{4}{5} + \frac{1}{3} = \frac{5}{8}$$

Figure 3. Even though you've never seen these problems before, you can easily – and correctly – think "What mistake could have made *me* do that?" This is the power of small data + understanding, as opposed to big data + statistics. (a) The student has failed to carry a 1. (b) Probably due to cognitive dissonance with multiplication, the student has added the two numerators and added the two denominators.

In the case of solving the first order equation 11x – 34 = 10, many students simplify by adding 34 to both sides of the equation, which is good, getting 11x = 44, but then a common error these days is for the students to confuse 11x as appending rather than multiplication, so they metaphorically subtract 11 from 11x to get x, and they literally subtract 11 from 44 to get 33, i.e., they rewrite 11x = 44 as x = 33.[3] The first

[3] Students who always borrow (even when unnecessary) get the bizarre answer $x = 213$, which would be unlikely to have been acceptable to students living before the advent of calculators.

point about these examples is that even with only one single exemplar, you have no trouble diagnosing what the student does and doesn't understand, in each case, what their difficulty is. In other words, you can infer several concepts and skills they do understand and one that they do not understand. The second point to make about these examples is that model-based ITS can have models of the various correct and incorrect skills and concepts of the domain, not just the correct ones, and can use those models exactly the same way that you used your models of (students struggling to learn) arithmetic to diagnose the above examples.

The third point to make about this example, and the most important one from the point of view of automating the ontology-aligning process, is that you or I or MathCraft can immediately and correctly infer several things just from finding out that a learner got the answer 33 to this problem, or the answer 213, or the answer 4. ITS#1 can work with the learner on various problems, and regardless of what domain and learner model it has – even if it has none! – ITS#2 can make use of problem/answer data that ITS#1 has collected. In case ITS#1 does have a domain model and a user model, it can say to ITS#2, in effect, "The students who get 33 as their answer to this problem suffer from what I happen to call bug-80913. The students who get 213 as their answer suffer from bug-1411. The ones who get 4 as their answer generally understand what I happen to call concepts 2923 and 715." ITS#2 independently generates the graph in Figure 4, which enables it to identify what terms in its ontology correspond to ITS#1's bug-80913 and so on.

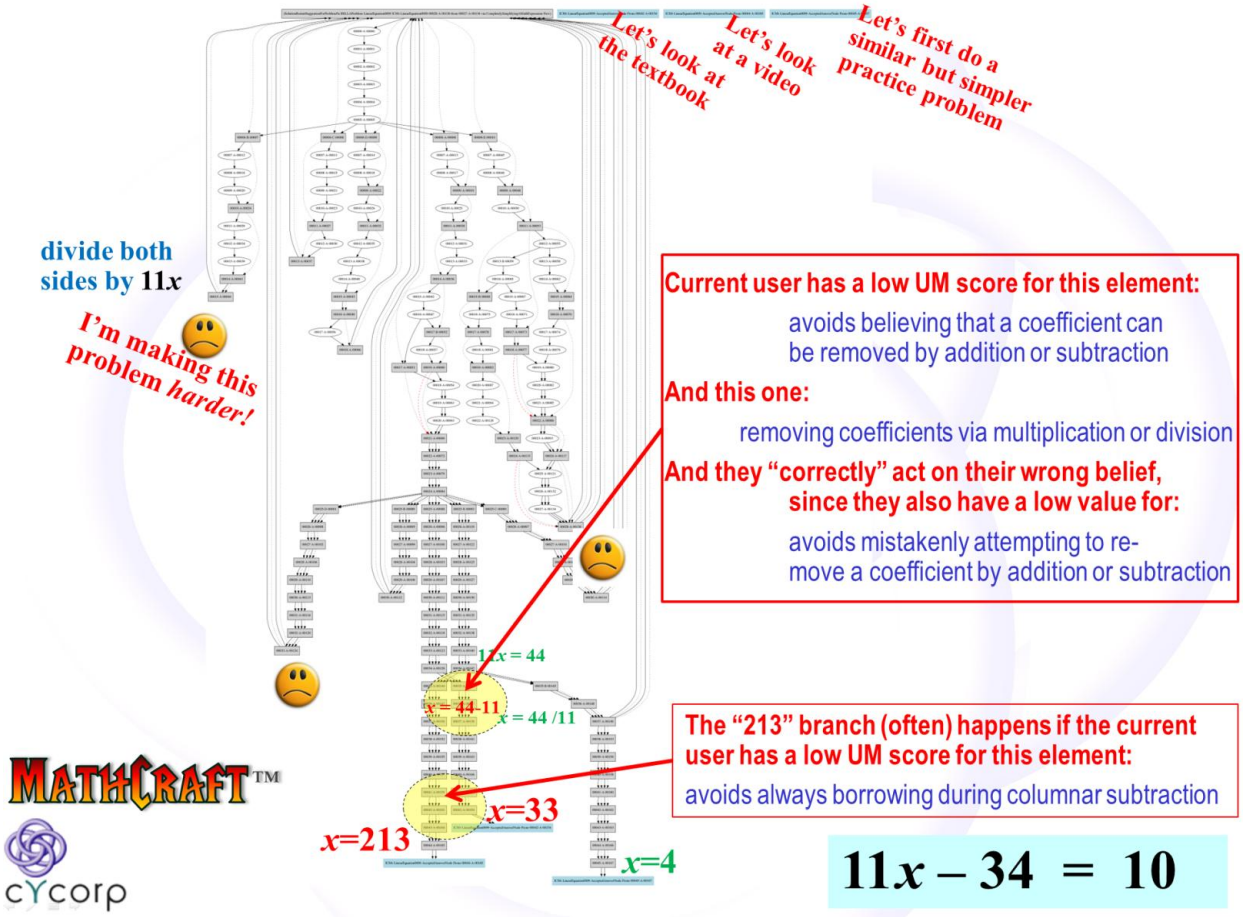


Figure 4. MathCraft model of the correct and (most common) incorrect actions for the 6th grade math problem 11*x* – 34 = 10, showing a graph of the right and wrong paths, which have a few dead ends ("*I'm making this problem harder!*"), a few wrong answers (33), and a right answer (4). If some other ITS tells MathCraft

that a learner got *x*=33 to this problem, MathCraft can immediately set several learner model elements correctly, without knowing or needing to know what that other ITS's domain or learner model looked like.

A relatively small number of examples, therefore, enable the two ITSs' ontologies to be aligned automatically, with no manual effort whatsoever. Notice that this same technique results in aligning the domain ontologies and aligning the learner models. In hindsight, this should not be too surprising, since a large part of the learner models capture their correct and incorrect understanding of the domain.

To be fair, this technique depends on there being – at least in part – an objective way to generate common learner answers to problems and questions; in some domains, such as creative writing where objective testing stumbles, this technique will also stumble. On the other hand, even a mediocre ontology-alignment is vastly better than none at all and is almost as useful as a perfect one. If I am a human teacher, or an ITS, and I need to teach a student 6th grade math, then even a few sentences about that student's level of mastery and depth of understanding of topics and skills from earlier grades goes a long way to giving the information sufficient to treat the student correctly, to not waste too much time on things that the student already knows or that require prerequisites the student does not yet know.

This type of reasoning is not a logically sound deduction or proof; it is what logicians call abduction. If someone walks into a movie theater dripping wet and carrying a wet umbrella, patrons abduce that it is raining outside; there could be several explanations but that is the most likely one. Occam's Razor advises us to eschew unnecessary complication, when you see hoofprints, think horses, not zebras. The situation is somewhat confused terminologically by Sir Arthur Conan-Doyle having his character Sherlock Holmes perform feats of abductive reasoning but incorrectly say that he's doing deductive reasoning. Abduction can also be thought of as a type of inductive reasoning, which is similarly not guaranteed to be correct in general. If a student correctly solves 1041.02 x 89¾, we induce their understanding of a large number of concepts and skills. We do not hesitate to rely on these unsound abductive and inductive conclusions, because they usually end up being correct, and the world is complicated and rich, and we have limited resources to expend before we need to act on our best informed guesses. This is akin to the bounded rationality Herb Simon proposed in (Simon, 1947).

Recommendations and Future Research

We have surveyed a continuum of ontology-aligning methods, ranging from purely manual to semi-automated graph-matching to purely automated abduction. We encourage future study and research by the Generalized Intelligent Framework for Tutoring (GIFT) and ITSs in general on when and how to best use these various sources of power, which have different characteristics and guarantees. The automated techniques we presented are strictly speaking unsound yet empirically often correct; that sounds off-putting, but much the same can be said for statistical inference techniques.

The benefits of aligning multiple ITSs' domain ontologies include, among other things, the ability for an ITS built for one domain to be "run" to teach another domain. The benefits of aligning multiple ITSs' learner models include, among other things, the ability for a student to move to a fresh ITS and be treated as individually and correctly as though having already used it for a long time. The benefits of having a whole community of ITS resources explicitly model both the domain model and the learner model include the ability to automatically infer that/when/how students start using some ITS that they hadn't been using before, and which would be of particular suitability and value to them at present. As the number of ITS resources (already in the millions) increases, the need for and the value of this meta-level reasoning and match-making ability will soar, and the wastefulness of each ITS treating each new user as a tabula rasa will dwarf the cost of forestalling that through ontology-alignment.

Our example of how to automate the domain model-aligning and learner model-aligning is just that: an example. We encourage GIFT and ITS researchers to consider this an important and valuable component in their future systems and their future research, and share empirical data and new insights about how to break the ontology alignment bottleneck.

References

Fancsali, S. E. & Ritter, S. (2014) Context personalization, preferences, and performance in an intelligent tutoring system for middle school mathematics. In Proceedings of the Fourth International Conference on Learning Analytics and Knowledge (LAK '14). New York: ACM. 73–77.

Huang, J. (2008) Toward Mutual Understanding Among Ontologies: Rule-Based and Learning-Based Matching Algorithms for Ontologies. Saarbrücken, Germany: VDM Publishing.

Lenat, D. & Durlach, P. (2014) Reinforcing Math Knowledge by Immersing Students in a Simulated Learning-By-Teaching Experience , Intl. Journal of Artificial Intelligence in Education, 24, 216–250.

Otero-Cerdeira, L., Rodriguez-Martinez, F. J. & Gomez-Rodriguez, A. (2015) Ontology matching: a literature review. Expert Systems with Applications, 42, 942–949.

Shvaiko, P. & Euzenat, J. (2013) Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, 25(1), 158–176.

Simon, H. A. (1947) Administrative behavior: A Study of Decision-making Processes in Administrative Organization, New York: The Macmillan Company.

CHAPTER 5 – Qualitative Representations for Education

Bert Bredeweg[1] and Kenneth D. Forbus[2]
[1]University of Amsterdam, The Netherlands
[2]Northwestern University

Introduction

Qualitative models capture conceptual knowledge about continuous phenomena and systems. This knowledge ranges from that of the person on the street, who has never taken formal mathematics or physics courses, to that of experts, such as scientists and engineers. Reasoning and learning techniques over qualitative representations have been used to create computational models of human commonsense reasoning, models of conceptual change, and models of how scientists and engineers reason in their professional work. Qualitative representations have been particularly useful in education, since they provide a level of knowledge that captures causality and everyday reasoning directly, while providing a substrate for professional knowledge.

The interactive power of instructional software is significantly influenced by the richness and accessibility of the available domain knowledge. Qualitative reasoning (QR) formalisms are well suited from this perspective and are open to support learning for all kinds of tasks related to science, technology, engineering, and math (STEM). The representations are to a large degree domain-independent and can be used for a variety of systems, e.g., natural biological systems (cf. Kuipers and Kassirer, 1984; King, et al., 2005; de Jong et al., 2005; Noble et al., 2009) as well as human-created technical artifacts (cf. Shimomura et al., 1995; Price, 2000; Ironi and Tentoni, 2005).

Qualitative Reasoning – An Example

To understand qualitative representations and reasoning, let us consider an example of a qualitative model. Figure 1 shows a model of an iron block, suspended from a spring fixed to a tree. The tree is positioned on the earth. The model represents the physical structure of the system, including the objects involved (e.g., "Block" and "Spring") and their arrangement (e.g., "Suspended from"). The objects have quantities associated to them (e.g., Block's Position). Quantities have value ranges, referred to as quantity spaces (e.g., Elasticity can be one of {Zero, Plus, Max}), which allow specifying their current magnitude (Elasticity is initially {Plus} as indicated by the blue pointer), and a derivative detailing whether the quantity is decreasing (downward arrow), steady (zero sign), or increasing (upward arrow). The derivative for Elasticity is not assigned and thus unknown according to the details shown in Figure 1, while for Gravity it is steady. Particularly important are the dependencies defined between the quantities. The I+ implements a positive direct influence (Gravity influences Momentum positively), while the P+ implements a positive indirect influence, also known as proportionality (e.g., Elasticity is negatively proportional to Position, hence P-). The model has one inequality statement, indicating that at start, the 'Elasticity is smaller than Gravity'.

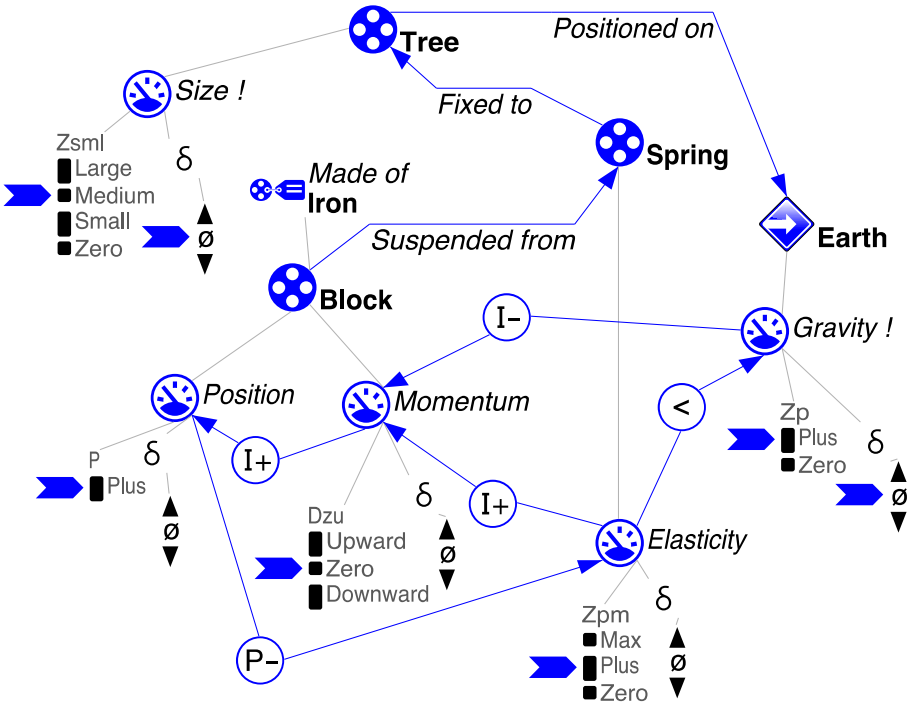


Figure 1. Qualitative model of a block suspending from a spring fixed to a tree. Friction is ignored. Created with DynaLearn, using Learning Space 4 (Bredeweg et al., 2013).

Figure 2 shows the results of a qualitative simulation of the spring model. It consists of two parts. The state graph on the left consists of qualitative states of behavior linked by arrows indicating what transitions can occur between states. Notice that, unlike traditional simulations, qualitative simulations can be ambiguous, as indicated by multiple possible transitions (e.g., the transition from state 10 to either state 11 or state 1). This ambiguity is a consequence of the lower level of detail in qualitative representations. While ambiguity in traditional simulation is considered a problem, in qualitative simulation it is useful because it shows alternate possible behaviors. In other words, qualitative simulations frame the possible behaviors for a system. Here, there are four unique paths, the shortest being [1 → 2 → 3 → 4 → 6 → 7 → 9 → 10 → 1]. Since they are all loops, we can infer that the system will oscillate and, under the current assumptions, the oscillation will never end.

The right part of Figure 2 shows a set of value histories for a specific trajectory through the qualitative simulation, with qualitative state identifiers being shown on the x-axis. The value history can be interpreted using the qualitative model to construct a causal explanation for this behavior. From Figure 1, a qualitative reasoner can identify that the beginning of the causal chain are Gravity and Elasticity, since they participate as sources in direct influence relationships and are not themselves directly influenced. Both quantities have a magnitude greater than zero (namely, plus), and both directly influence Momentum, although in opposite directions. Since Elasticity < Gravity initially, the negative influence will dominate, causing Momentum to start decreasing. This leads to state 1 in Figure 2. Since Momentum was equal to zero, this negative influence will cause an immediate state transition to state 2, when Momentum is at the Downward value of its quantity space. Because Momentum positively influences Position (I+), this causes Position to decrease. Moreover, because Elasticity is negative proportional to Position (P-), Elasticity will change in the opposite direction and start increasing. Both these results are generated during simulation and part of the reasoning used to construct state 2. The dynamics continue, ultimately leading to the cycle of states shown in Figure 2.

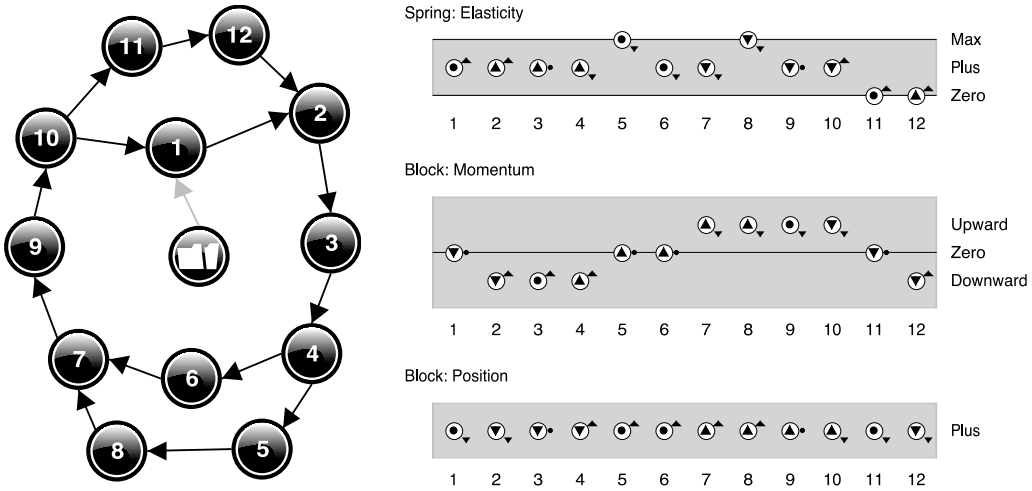


Figure 2. Simulation results for the qualitative model in Figure 1. The state-graph with four behavior-paths is shown on the left-hand side. The folder symbol (middle of the state-graph) denotes the initial situation. The value history for the changing quantities is on the right-hand side. Black vertical lines indicate that the value is a point (points and intervals alternate). Each circle indicates the current value for the quantity in that particular state. The symbol inside the circle represents the derivative, and indicates whether the quantity is decreasing (pointing down), steady (circle), or increasing (pointing up). Quantity Size and Gravity are not shown, because they remain unchanged. Note that state numbers are identifiers and do not define state order.

To summarize, this model captures the idea that there is a balance between elasticity and gravity that drive momentum, which, in turn, drives the block's position, which provides negative feedback on elasticity (Figure 1). When started in an unbalanced initial situation, QR over this model generated the prediction that the block will oscillate up and down. Detecting oscillation and other higher-order patterns of behavior is possible with numerical simulations, but only by providing many more concrete numerical parameters and constructing accurate equations. For this example, the burden of constructing such a numerical simulation would not be too onerous. But for many STEM topics accurate simulations become time consuming to construct and analyze, and end up being black boxes rather than artifacts that students can construct and modify to represent their ideas. Moreover, for early education (up through US middle school), students do not have the numerical sophistication to construct such models. Most elementary and middle school science curricula rely on qualitative, causal explanations. Qualitative models provide a formal representation for directly handling such information and reasoning with it. Existing qualitative reasoning software was used to construct the model of Figure 1 and automatically generate the results shown in Figure 2, and other research has been done to automatically generate natural language descriptions of causal predictions created from QR models. In other words, qualitative representations provide a representation language for domain modeling involving causal, conceptual models about the continuous world. This includes physical situations, but also economics and aspects of social science ideas.

Related Work

This section outlines the key ideas of qualitative representations. We start with a brief historical perspective. We summarize representations for ontologies, quantities, and causal reasoning. Then we summarize some representative prior projects that use qualitative representations for education.

Motivations for Qualitative Representations

Improving education was one of the original motivations for the development of qualitative representations. Several early projects that used numerical models and simulations to do domain reasoning in tutoring systems, such as SOPHIE (Brown et al., 1982) and STEAMER (e.g., Hollan et al., 1987), found that numerical models were a poor fit for the kinds of knowledge that students really needed to learn. In the case of SOPHIE, which was aimed at improving electronics troubleshooting skills, the numerical simulation did not capture the causal knowledge that expert technicians had. In the case of STEAMER, experimenting with a numerical simulator of a frigate's steam propulsion plant turned out to only go so far in training operators of such plants. Like electronics troubleshooting, the operators of complex physical plants need to have causal, conceptual models of how the plant works, so that they can interpret the detailed numerical observations that they are getting from their instruments. The idea of qualitative modeling was born as a way to formalize the tacit knowledge that experts have about such systems.

Approaching qualitative representations as a modeling enterprise led naturally to a separation of domain modeling from task modeling, a distinction that is important for scalability. Traditional expert systems, for example, typically failed to make this distinction, thus interleaving general domain knowledge with knowledge about a specific artifact and a specific task involving that artifact (cf. Steels, 1990). While qualitative representations are often motivated by traditional mathematical models, they have different concerns. Consider for example the ideal gas law,

$$PV=nRT.$$

Traditional mathematical formalisms do not formalize the conditions under which this equation holds. That requires constructing ontologies that distinguish gasses from liquids and solids, and ideal gases from other gasses for which this equation is too inaccurate to be used. Qualitative representations include formal representations of applicability and modeling assumptions. Traditional mathematical formalisms do not incorporate a notion of causality. When solving this equation numerically, it is just as reasonable to solve for n (the number of molecules) given T (temperature) as it is to solve for T given n. However, causally, a change in temperature cannot, by itself, cause the number of molecules in a sealed container to rise or fall due to the properties of an ideal gas! Qualitative representations include forms of qualitative mathematics, which can be used to provide human-like causal arguments.

Ontologies

Ontologies are theories of the kinds of things that exist. As such, ontologies form a means of providing information about the applicability of modeled concepts. A scientific or engineering domain is formalized by selecting an ontology and constructing a domain theory, a set of constructs using that ontology that define the kinds of things that models can be constructed out of. Specific models are created via a model formulation process, which constructs models for specific systems or situations out of model fragments from the domain theory (Falkenhainer & Forbus, 1991; Bredeweg, 1992). The particulars of model fragments and the formulation process vary depending on the kind of ontology. Ideally, the domain theory can correctly model the behavior of any system expressible via its model fragments, and provide models that are useful for different tasks. In practice, domain theories tend to be built out initially to cover some initial subset of systems of interest, but unlike more ad-hoc strategies, they can more easily be extended as the need to expand coverage grows.

Five different classes of ontologies have been used in qualitative representations: components, processes, constraints, bond graphs, and spatial aggregates. We ignore constraints, bond graphs and spatial aggregate

here, because we do not know of cases where those ontologies have been used in education. We discuss each of the others in turn.

Component Ontology

The idea of the component ontology (de Kleer & Brown, 1984; de Kleer 1984; Williams, 1984; Davis 1984; Genesereth, 1984; Barrow, 1984) is to model systems as components that manipulate materials and conduits that transport materials. The canonical example of a domain well modeled via components is analog electronics. The components are resistors, capacitors, transistors, etc., and the conduits are wires, which transmit charge. Component models are a natural choice for engineered systems that can be characterized as lumped-parameter systems. Automobiles, photocopiers, and power plants all are well characterized by component models at various levels of abstraction.

By associating qualitative mathematical laws with types of components, a model for a specific system can be assembled by "hooking up" instances of component types with the appropriate instances of conduits, just as one would build an electronic circuit by connecting resistors, capacitors, and other components by wires. The library of component types and conduit types constitutes the domain theory for this ontology. An important principle in constructing elements of a domain theory (for all ontologies, but it was first articulated for component ontologies) is the no function in structure principle (de Kleer & Brown, 1984). That is, a model of a component must not include tacit assumptions about other parts of the system that it is connected to. Consider for example a simple electrical circuit consisting of a battery, a switch, and a light-emitting diode (LED). A poor way to model the switch would be "IF the switch is closed, THEN the LED is lit", since that assumes the existence of the other components and that the battery is not fully discharged. In other words, component models should not assume anything about the system that they are connected to. To support compositionality, properties of system models built with components must have the system properties arise from their combination.

No ontology is universally useful. The component ontology is not a good fit for distributed systems (e.g., modeling the weather), where spatial aggregation models are a more appropriate choice. The component ontology is also less appropriate when modeling systems whose connectivity changes radically during normal behavior. For example, modeling a bouncing ball as a component involves some fairly unnatural choices (e.g., momentum flow, and constantly changing the "circuit" used to model it, due to it striking different objects). Finally, when the material substance itself has substantial properties, and whether or not it exists in particular components can change, a process ontology becomes a better choice.

Process Ontology

The process ontology reifies continuous processes as the agents of causal change. For example, in Qualitative Process (QP) theory (Forbus, 1984), the world is modeled as physical objects whose continuous properties are described by quantities. Complex fluid/thermal systems, chemical engineering, commonsense physics, and container models in physiology have been modeled via QP theory.

The domain theory in a process ontology consists of types of objects (e.g., thermal objects, contained liquids) and types of processes (e.g., fluid flow, boiling). Given a description of a specific situation, model fragments are instantiated to provide the causal laws governing entities in the system (e.g., treating the coffee in a cup as an instance of the contained liquid) and introduce new entities (e.g., the flow of heat from the coffee in the cup to the surrounding environment) as needed to construct a causal model of what can be happening in that situation. The relevance of model fragments is gated to large degree by ordinal relationships between parameters, hence a qualitative simulator can infer that once coffee reaches room temperature, it will remain at constant temperature.

While the process ontology is useful in many domains, it is not appropriate for analog electronics. Modeling electronics in terms of processes would require thinking about charge flow, and reifying processes along each conduit in the system. For most components (including nodes where multiple connections come together), there is no charge accumulation. Thus a process model would involve many more conceptual entities while providing no additional insights. Any modeling idealization where the materials do not accumulate within the system (except for certain components) will similarly be a poor choice for process modeling.

Representing Quantities

The continuous properties of an entity or system, such as volume, temperature, and height, are treated as quantities in qualitative reasoning. By quantity, we mean what in logic is called a fluent (i.e., something that can take on specific values at particular times). The coffee in a cup has a temperature, that is, a quantity, and that temperature takes on different values as the coffee is heated or cooled. In traditional mathematics, the values of quantities are specified as elements of the real numbers, and in numerical simulation, as floating point numbers. Qualitative representations involve weaker, less detailed information than either of these. The goal of qualitative representations of quantities is to find minimal amounts of information that suffice to draw powerful conclusions about continuous systems.

Two important kinds of qualitative representations that were used in the spring-block example of Figures 1 and 2 are ordinal relations and signs of derivatives. Ordinal relations are statements about whether or not two quantities have the same value, or if one is less than or greater than the other. Ordinal relations are an important qualitative description because applicability conditions for model fragments often depend upon them. For example, heat flow stops when the temperatures of two connected entities become equal. Signs of derivatives – which are ordinal relationships between the derivative of a quantity and zero – are important because they capture directions of change. For example, the temperature of hot coffee is initially decreasing, after first being poured, and after some time, when the temperatures equalize, it becomes steady. This is why ordinal relationships are such an important constituent of qualitative state.

It is often useful to completely specify quantity spaces, so that ranges of values are mapped to a finite set of symbolic values. In Figure 1, for example, Momentum is mapped to three values {Upward, Zero, Downward}. Temperatures when phase changes are being considered might be mapped into {AbsoluteZero, Solid, Melting, Liquid, Boiling, Gas}, where each value is less than the subsequent value. Notice that these come in alternating interval/instant pairs, i.e., AbsoluteZero, Freezing, and Boiling are all points, whereas Solid, Liquid, and Gas are intervals. Point values are often called landmark values because they are typically introduced to capture when phenomena changes. However, some simulation strategies allow the introduction of additional landmarks, so that phenomena such as decaying oscillations or pumped circuits can be described in value histories.

Model fragments are one way to introduce landmark values. However, other criteria can be useful as well. For example, in reasoning about economics it is common to talk about when mortgage rates are high versus low, and large versus small population sizes in ecology. Some research has been done on ways to do this automatically (e.g., Guerrin, 1995), but making such choices still requires skill on the part of the domain modeler.

Key to QR about quantities is the notion of continuity. Even qualitative values preserve a form of continuity: If in one state A>B, then in the next state it cannot be the case that A<B. Instead, there must be an intermediate state in which A=B, barring some kind of discontinuous change. (Models of impulses have been created, but these are not widely used.) There is also a constraint between the duration of a qualitative state and the change in ordinal relationships that gave rise to it. A change from equality takes only an

instant (as with State 1 of Figure 1), whereas a change to equality requires an interval of time. This means that the duration of qualitative states can be divided into two categories: instant or interval.

Causality

Quantities are constrained by qualitative mathematical relationships. There are two broad families of such relationships. The first are confluences, a form of equation typically involving sign values. Confluences are almost always used with the component ontology. The second are influences, which provide directional constraints. The choice of relationship also implies a choice of causal argument construction for between-quantity relationships. With confluences, it is assumed that some perturbation is applied to the system, and the flow of information during propagation of that perturbation through the system is treated as the causal explanation for the changes. In a typical transistor amplifier, for example, increased input voltage causes an increased voltage to the base of the transistor, which causes an increased flow of current between the transistor's emitter and collector, which causes a drop in the voltage at the output of the transistor. Notice that this explanation moves freely between changes in voltage causing changes in current, and vice versa.

With directed relationships this does not happen. Instead, the direction of causality is encoded in the direction of the relationships. In Figure 1, for example, changes in Elasticity cause changes in Momentum, thanks to the I+ relationship between them (called a direct influence). To be sure, tracing through the influences in Figure 1, the alert reader will notice that there is causal chain of I+, and P- that goes back from Momentum to Elasticity. This is an example of feedback. The P+ and P- relationships, also called indirect influences, are functional, that is, they indicate that one property (here Elasticity) is a function of another (here Position), but in keeping with the nature of qualitative mathematics, we do not know the precise form of the relationship. We only know that for P+, if all else is equal, increasing one causes an increase in the other, and for P-, again if all else is equal, increasing one causes a decrease in the other. The direct influences, on the other hand, indicate integrals. That is, I+ indicates that the derivative of Momentum is a sum that includes Elasticity as part of it, and the I- indicates that Gravity provides a negative contribution. Direct influences break causal loops, leading to a clean ordering of causal events. That is, causality proceeds from direct influences (so named because they are considered to be the imposed by some (perhaps unnamed) process) through indirect influences, based on the directions encoded in the primitives themselves.

Examples of Educational Software using Qualitative Representations

Given that education was one of the motivations for exploring qualitative representations, it is perhaps not surprising that multiple types of educational software have been built using qualitative models. We summarize three kinds of such software here: within-state QR, multi-state qualitative simulations, and hybrid qualitative/quantitative systems.

Within-state QR

Much of the curriculum content of elementary and middle-school science concerns qualitative causal models of what is happening in a single qualitative state. Helping students understand how to carve up the unruly world into idealizations that can be more concisely modeled has been the focus of several systems that use concept maps as a visual means of enabling students to express their qualitative models. The vocabulary of relationships is constrained to be constructs from qualitative modeling, albeit with student-friendly names. For example, Betty's Brain (Biswas et al. 2015) asks students to teach Betty, a simple artificial intelligence (AI) system, about domains such as stream ecology. They do this by constructing qualitative models, using concept maps, connecting quantities via influences. Their Betty is then quizzed,

and the students vie to see whose Betty does the best. This teachable agent approach is extremely motivating. Similarly, VModel (Forbus et al. 2004) has students building concept maps involving entities, quantities, and processes as well as influences. This helps them learn about the distinction between extensional and intensional quantities, as well as specific domains such as ecosystems. A key feature is that VModel automatically produces English explanations of causality based on the representations they use in their model. When an explanation does not "sound right" to the student, they are motivated to fix their model in ways that, due to the design of the system, make it more accurate.

Multi-state Qualitative Simulation

DynaLearn is an interactive learning environment (Bredeweg et al., 2013) that allows learners to acquire conceptual knowledge by constructing and simulating qualitative models. DynaLearn is based on Garp3 (Bredeweg et al., 2009) and uses diagrammatic representations for learners to develop their ideas. A key feature of DynaLearn is the notion of learning spaces (LSs), a set of knowledge construction workspaces that progress from simple to complex, and as such scaffold the development of systems thinking on behalf of a learner.

DynaLearn has 6 spaces. LS1 focuses on concept maps. LS2 introduces learners to the distinction between structure and behavior, particularly, the notion of quantities and changes (increase/decrease) propagating through the system; LS2 relates to the approach taken in Betty's brain. LS3 introduces simple multi-state simulation. Landmarks can be defined that identify threshold values of quantities for which the system changes behavior. LS4 supports learners in distinguishing different types of causality (notably direct and indirect influences) and working with feedback loops. A LS4 model is shown in Figure 1. LS4 relates to VModel, but also allows for multi-state simulation. LS5 refines LS4 by making it possible to specify conditions under which specific parts of the model are true. The goal of LS6 is to have learners acquire generic knowledge of processes and system behavior, and how that generic knowledge instantiates to particular situations. Learners therefore create scenarios and domain theories. The latter is captured as a set of model fragments. LS6 models can become advanced, reflecting expert-level understanding.

To further aid learners in systems thinking DynaLearn is equipped with recommendation components capable of generating knowledge-based feedback, and virtual characters implementing an engaging interaction with learners.

Hybrid Qualitative/Quantitative Systems

The explanatory capabilities of qualitative representations can provide a useful complement to numerical analyses and simulation. For example, CyclePad (Forbus et al., 1999) is an articulate virtual laboratory that enables students to explore ideas in engineering thermodynamics by "building" designs and analyzing their performance (Figure 3). CyclePad uses qualitative representations in several ways. First, it uses ordinal constraints as reality checks on student designs: a design that tacitly assumes that a pump creates, rather than consumes, work is contradictory, and such contradictions and the assumptions underlying the contradiction are displayed for students. Second, it uses an automatically constructed teleological model of the student's design (Everett, 1999) and quantitative benchmarks to help students understand the real-world impact of their design choices. A power plant that produces less energy than a candle, for example, is a poor design. CyclePad remains in use by students and classes in multiple countries, over 15 years after the end of its funding.

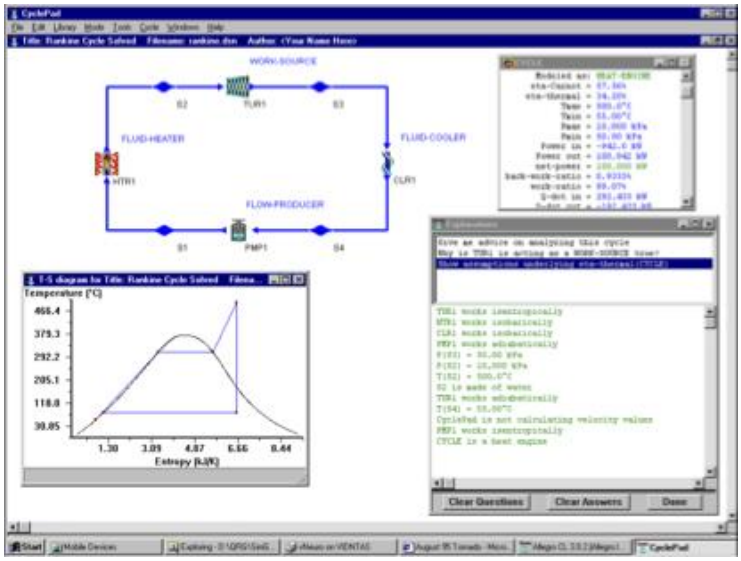


Figure 3. CyclePad – Using qualitative representations with numerical analysis to help detect inconsistent designs.

A problem that bedevils STEAMER-like systems in this category, where qualitative explanations are layered afterwards on top of an existing numerical simulator, is that there can be mismatches between the qualitative and quantitative aspects of the system. One way to avoid this is to automatically construct the numerical simulator from the qualitative model, using a domain theory that contains both qualitative and quantitative model fragments. This idea, called self-explanatory simulators (Forbus & Falkenhainer, 1995), encodes a form of the qualitative, causal reasoning that a domain expert would do in creating a simulator into the software itself. The simulator produces numerical values, the way a traditional simulator does, but also produces a concise history of qualitative episodes that, with an index of causal relationships produced at compile time, enables systems to accurately generate causal descriptions of behavior to complement the numerical behaviors. Self-explanatory simulators were used in a NASA-sponsored Principles of Operations resource about Deep Space One aimed at middle school students4 and in a curriculum on ecosystems (Mars Survival Station) fielded in the Chicago Public Schools (Forbus, 1996).

Discussion

This chapter reviewed ideas underlying qualitative representations and their use in qualitative causal models. It discussed alternative representation and reasoning approaches, the construction of qualitative models, and ways in which such models are used for education.

Although there have been several successful applications of qualitative representations in education, these only scratch the surface of what is possible. There are opportunities to apply qualitative representations across STEM education. For elementary and middle school, simpler within-state qualitative models are likely to suffice for most purposes. For high school and university students, multi-state qualitative simulations should be added to the mix. To fully realize these opportunities, however, several hard problems need to be tackled:

[4]http://www.qrg.northwestern.edu/projects/vss/docs/index.html

(1) Helping students map from the everyday world to abstractions. For tutoring systems to do this, they need to have a much broader understanding of the domains in which they are operating. In the limit, natural language interaction about the situation or system being modeled would be useful. This requires broader knowledge bases, containing knowledge about the everyday world as well as cultural knowledge, the latter which might be harvested from the Semantic Web.

(2) Providing good summarization and debugging facilities for multi-state simulations. There has already been useful research on this (de Koning et al., 2000, Bouwer and Bredeweg, 2010), but more needs to be done.

(3) Building an open, common repository of interoperable domain theories would greatly facilitate the construction of future intelligent tutoring systems.

It is worth mentioning that the use of qualitative representations within tutoring systems can support different pedagogical strategies. In the "learning by modeling" approach as deployed in DynaLearn, learners actively construct knowledge graphs and simulate these. The teachable agent paradigm also requires learner to create diagrams, but opens up a new dimension by suggesting that the learner teaches the virtual character and thereby transforming the situation into "learning by teaching". In approaches such as CyclePad the qualitative representations are not shown explicitly to learners. Rather, they act as an instrument to detect inconsistencies in the designs created by a learner and to automatically generate explanations. It would be interesting to further explore alternative pedagogical strategies. For instance, using qualitative representations as the basis for interactive virtual environments (Cavazza and Simo, 2003) and applied games.

Recommendations and Future Research

Regarding the Generalized Intelligent Framework for Tutoring (GIFT) specifically, we recommend the following:

- Incorporating both within-state and multi-state qualitative reasoners as modules that can be used by GIFT authors. We suggest using Garp3/DynaLearn for multi-state QR, since it is publically available and has the most developed suite of tools for such purposes. Within-state qualitative reasoners have also been built in the context of DynaLearn.

Regarding future research, we recommend the following:

- Research on the tradeoffs in using different levels of qualitative modeling across different STEM education topics and age levels. Our hypotheses about when within-state versus multi-state reasoning are appropriate are based on our experience and informal surveys of relevant curricula, but there is not enough experience yet to have solid evidence about this issue, as well as which hybrids of qualitative and quantitative techniques work best for different educational purposes.

- Research on scaling up domain theories. This includes both how such domain theories can be constructed, via learning by reading, interactive instruction of AIs via domain experts, and models of misconceptions by data mining over fielded tutoring systems. It also should include research on knowledge integration, e.g., how to combine domain theories that involve different levels of assumptions concerning granularity, perspective, and operating assumptions, and how to do model formulation with domain theories that are closer in size and scope to the best human experts.

- Further developing and evaluating the use of qualitative representations in the context of alternative pedagogical strategies, including virtual environments and applied games.

References

Barrow, H.G. (1984). A program for proving correctness of digital hardware designs. Artificial Intelligence, 24(1–3), 437–491.

Bouwer, A. and Bredeweg, B. (2010) Graphical means for inspecting qualitative models of system behaviour. Instructional science, 38, 173–208.

Biswas, G., Segedy, J.R. & Leelawong, K. (2015). From Design to Implementation to Practice - A Learning by Teaching System: Betty's Brain. International Journal of Artificial Intelligence in Education.

Bredeweg, B., Liem, J., Beek, W., Linnebank, F., Gracia, J., Lozano, E., Wißner, M., Bühling, R., Salles, P., Noble, R., Zitek, A., Borisova, P. and Mioduser, D. (2013). DynaLearn – An Intelligent Learning Environment for Learning Conceptual Knowledge, AI Magazine, 34(4), 46–65.

Bredeweg, B., Linnebank, F., Bouwer, A. and Liem, J. (2009). Garp3 - Workbench for Qualitative Modelling and Simulation. Ecological Informatics 4(5–6), 263–28

Bredeweg, B. (1992). Expertise in Qualitative Prediction of Behaviour. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands.

Brown, J. S., Burton, R. R. and Kleer, J. de (1982). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III. In D. Sleeman & J. S. Brown (Eds.), Intelligent Tutoring Systems. Academic Press, New York, pp. 227–282.

Cavazza, M. and Simo, A. (2003). Qualitative physiology: From qualitative processes to virtual patients. Proceedings of QR03.

Davis, R. (1984). Diagnostic reasoning based on structure and behavior. Artificial Intelligence, 24(1–3), 347–410.

Everett, J. O. (1999). Topological inference of teleology: Deriving function from structure via evidential reasoning. Artificial Intelligence, 113 (1–2).

Falkenhainer, B. and Forbus, K. (1991). Compositional modeling: finding the right model for the job. Artificial Intelligence 51(1–3), 95–143.

Forbus, K.D. (1984). Qualitative process theory. Artificial Intelligence, 24(1–3), 85–168.

Forbus, K. (1996). Self-explanatory simulators for middle-school science education: A progress report. Proceedings of QR96.

Forbus, K. and Falkenhainer, B. (1995). Scaling up self-explanatory simulators: Polynomial-time compilation. Proceedings of IJCAI-95, Montreal, Canada.

Forbus, K.D., Whalley, P., Everett, J., Ureel, L., Brokowski, M., Baher, J. and Kuehne, S. (1999). CyclePad: An articulate virtual laboratory for engineering thermodynamics. Artificial Intelligence, 114, 297–347.

Forbus, K., Carney, K., Sherin, B. and Ureel, L. (2004). VModel: A visual qualitative modeling environment for middle-school students. Proceedings of the 16th Innovative Applications of Artificial Intelligence Conference, San Jose, July 2004

Genesereth, M.R. (1984). The use of design descriptions in automated diagnosis. Artificial Intelligence, 24(1–3), 411–436

Guerrin, F. (1995). Dualistic algebra for qualitative analysis. Proceedings of QR95. Amsterdam, The Netherlands.

Hollan, J.D., Hutchins, E.L. and Weitzman, L. (1987). STEAMER: An interactive inspectable, simulation-based training systems. In G. Kearsley (ed.). Artificial intelligence and instruction: applications and methods. Addison-Wesley, Reading (Mass), pp. 113–134.

Ironi, L. and Tentoni, S. (2005). In Electrocardiographic Imaging: Towards Automated Interpretation of Activation Maps, Lecture Notes in Artificial Intelligence, vol. 3581, pages 323–332. Springer.

de Jong, H., Geiselmann, J., Batt, G., Hernandez, C. and Page, M. (2004). Qualitative simulation of the initiation of sporulation in Bacillus subtillis. Bulletin of Mathematical Biology, 66(2), 216–300.

King, R., Garrett, S. and Coghill, G. (2005). On the use of qualitative reasoning to simulate and identify metabolic pathways. Bioinformatics, 21, 2017–2026.

De Kleer, J. and J.S. Brown, J.S. (1984). A qualitative physics based on confluences. Artificial Intelligence, 24(1–3), 7–83.

De Kleer. J. (1984). How circuits work. Artificial Intelligence, 24(1–3), 205–280.

De Koning, K., Bredeweg, B., Breuker, J. and Wielinga, B. (2000). Model-based reasoning about learner behaviour, Artificial Intelligence, 117(2), 173–229.
Kuipers, B. and Kassirer, J. (1984). Causal reasoning in medicine: Analysis of a protocol. Cognitive Science, 8, 363–385.
Noble, R.A.A., Bredeweg, B., Linnebank, F.E., Salles, P. and Cowx, I.G. (2009). A qualitative model of limiting factors for a salmon life cycle in the context of river rehabilitation. Ecological Informatics, 4(5–6), 299–319.
Price, C.J. (2000). AutoSteve: Automated electrical design analysis. Proceedings ECAI-2000, pages 721–725, August.
Shimomura, Y., Tanigawa, S., Umeda, Y. and Tomiyama, T. (1995). Development of self-maintenance photocopiers. Proceedings of IAAI-95, pages 171–180.
Steels, L. (1990). Components of expertise. AI Magazine, 11(2), 28–49.
Williams, B.C. (1984). Qualitative analysis of MOS circuits. Artificial Intelligence, 24(1–3), 281–346.

CHAPTER 6 – A Work Practice Simulation Approach to Modeling Socio-Technical Domains

Benjamin Bell,[1] William J. Clancey,[2] and Winston Bennett, Jr.[3]
[1]Aqru Research and Technology, [2] Florida Institute for Human and Machine Cognition,
[3] US Air Force Research Laboratory

Overview

A domain model can incorporate a rich and realistic context in a tutoring environment. Domains can be modeled to provide an environment with "real-world" look and feel with a suite of behaviors that create a dynamic context for learning. The construct of "context" is often seen as something objective or completely physical in the world. A domain model for a construction crane simulator may define properties and behaviors of the crane to include its lift capacity and boom length, and will define objects in the world that act upon the crane and vice versa. However, the context of human behavior is also conceptual and subjective. Conceptual context is relational and affected by perceptions, and thus is often dynamic. Domain models should capture how context is conceptually constructed through and during interactions of monitoring, moving in, and manipulating the world.

We adopt a socio-technical approach to domain modeling to adequately capture and model how understanding of context develops in activity, and people's activity within that understanding. Our approach to domain modeling includes a spatial/geographical model, cultural features and artifacts, and information systems, to reflect a broad meaning of "context" that includes what the learner is doing physically and mentally, how the learner is conceiving what the learner is doing (which in more general terms is less defined than tasks), and what the learner is perceiving in the environment. In other words, the context for behavior is always conceptual, and is about a socio-technical, physical world in which the person is an actor.

In this chapter, we present a methodology for modeling domains that employs a socio-technical framework for capturing the activities of individuals and the socio-technical context. We discuss the need for both task analyses and psychosocial activity analyses and how those methods vary for different domains. We describe a computational tool used for building and running these models and summarize several applications of this approach. Finally, we present new work that employs these models for predictive analysis in simulations of uncertain, complex threats, and conclude with implications for domain modeling in intelligent tutoring systems (ITSs).

Introduction

Remarkable advances in artificial intelligence (AI), human-systems integration, and digital communications are providing automation systems of unprecedented sophistication. In this chapter, we use the term AI to describe an agent that can serve one or more humans as an intelligent assistant in either actual systems or in training simulations. An AI onboard an aircraft could, for instance, monitor the aircraft, interpret and carry out pilot commands, and advise the pilot as to aircraft and system status, mission progress, threats and alerts. People and agents are part of a socio-technical system, so the functions of the AI and the pilot in our example go well beyond the mechanics of maneuvering the aircraft. Rather, people and agents are inherently part of a network of operations that may involve ambiguous communications and dynamic roles and responsibilities. Combat sorties, for example, require considerable interaction within and across aircraft and remotely with people (and agents) on the ground.

ITSs employ domain models that simulate the behaviors and performance parameters of a diverse range of entities, from space shuttles to tactical radios, including the simulation of human activity – from air battle managers (Freeman, et al., 2003) to village tribal elders (Johnson & Wu, 2008). However, there is seldom evidence of socio-technical factors in conventional approaches to domain modeling. Most modeling frameworks are not adequate to represent the effects of socio-technical phenomena because they focus on functions and tasks rather than group interactions and chronological activities, do not relate perception to activities and physical interaction, and/or focus on prescribed processes rather than practices.

The need for domain models to capture and reflect socio-technical processes arises in most contemporary tutoring applications, as today's workforce must be trained to be proficient in an information-rich, networked, digital, automated world. In some instances, the socio-technical context is not just backdrop enrichment but an explicit focus of the learning; ITSs are being developed that include training for detecting and overcoming problems with socio-technical lapses.

For ITS developers serving such communities, the need to train personnel to overcome such disruptions calls for new domain modeling constructs. Simulation approaches grounded in discrete models sensing and acting independently serve only to exercise how each model behaves and reacts within the confines of the assumptions of the modeler, which are often limited to technical factors. While a communications disruption, for example, would adversely affect discrete entities comprising an overall domain model, it may also have devastating systemic effects on the way people and intelligent systems work together. The unanticipated, subtle socio-psychological effects of factors like denied communications can disrupt coordination and degrade trust. Many unknowns surround how even slight disturbances in team interactions can portend serious downstream consequences.

Domain models are needed that can properly capture work practices of the socio-technical system (Clancey 1993). With teams in complex, highly engineered environments relying increasingly on advanced collaborative and technological work systems, an urgent need has surfaced for a more robust capability to train these teams to succeed when faced by both nominal and adverse conditions. In the rest of this chapter, we present an innovative application of Brahms, a sophisticated and proven work practice analysis simulation (Clancey, et al., 1998; Clancey, et al., 2013). We introduce models and a testbed built to support training, research and concept development for human-automation teams to achieve success in overcoming socio-technical disruptions. Before we present our socio-technical domain modeling approach, we provide in the next section a brief discussion of Brahms to contextualize the remainder of this chapter.

The Brahms Work Systems Framework as Domain Model

Brahms is a work practice modeling and simulation framework used in designing and implementing work systems. It is particularly useful for activity systems with distributed teams with different roles and responsibilities using automated information processing and model-based tools. In that sense, it is well suited for modeling a diversity of domains with rich levels of interpersonal and physical interactivity, as is required for many ITS applications.

Socio-Technical Work Practice Simulation

A work practice simulation represents chronological, located behaviors of people and automated systems. In contrast with task models, which represent abstractly what behaviors accomplish (i.e., technical functions; Schön, 1987), a behavioral model represents what people and systems do, called activities (Leont'ev, 1979; Clancey, 2002). Activities include monitoring (looking, attending), moving, communicating, reading and writing, all of which require time and occur in particular places with other people,

tools, materials, documents, and so on (Suchman, 1987; Lave, 1988; Ehn, 1989; Wynn, 1991). In terms of work, a function/task model abstracts what a person or system does (e.g., "determine location"), while a cognitive-behavioral model of practice represents how the work is carried out in the world (e.g., simulate a person moving, changing the state of a control, perceiving a display's representation, and recognizing a problem). An activity framework enables modeling how knowledge and expertise of distributed teams are applied in practice, which includes noticing and characterizing a situation as being problematic and defining goals and methods for handling the situation (Jordan, 1992; Clancey, et al., 2005).

In simple terms, most cognitive models describe knowledge and operations for transforming information and world states. A Brahms activity model includes this cognitive aspect but casts it within the larger simulation of processes and physical interactions in which operations occur in the world. From a psychological perspective, activities are how people conceive their day-to-day affairs as social actors manipulating tools, representations, and other objects while they move and communicate in some physical setting (e.g., going to the grocery store to buy dinner). Activities are ongoing conceptions of "what I'm doing now" (Clancey, 1997); they encapsulate roles ("whom I'm being now"; Wenger, 1998), norms ("what I should be doing now"; Lave and Wenger, 1991), and progress appraisals ("how well I'm doing"; Feltovich, et al., 2008). In this respect "the context" and "the situation" are conceptual; the Brahms engine is designed to simulate how activity conceptualizations are activated, prioritized, interrupted, and resumed psychologically, as perception, inference, and communications modify the agent's concept of "what I should be doing now."

The Brahms multiagent framework can thus model: (1) people's beliefs and behaviors (their activities, including communicating and moving), (2) objects with behaviors (e.g., instruments, devices, vehicles, computer agents), and (3) representations (e.g., documents, displays, diagrams) (4) in a physical setting (e.g., cars, offices, roads, regions). These are all constructs in the Brahms modeling language, which is a formal programming language that is compiled and interpreted by a Brahms virtual machine (VM). The VM loads compiled models, runs a simulation by interpreting the compiled code, and generates log files and history files that can be used to analyze the results of the simulation. The VM enables Brahms to be integrated with simulations running on other platforms, which may require integration modules using network application programming interfaces (APIs) to establish communications and synchronization.

Although our focus here is domain modeling for ITSs, we note briefly that Brahms simulations can serve many purposes. Brahms was developed as a tool for facilitating systems-level thinking about both formal procedures and implicit cultural practices. As a behavioral simulation, it enables what-if analysis and experimentation with more scenarios having complicated interactions than anyone could either generate manually or anticipate (e.g., see Clancey, et al., 2013). Brahms can serve as a "total system" design tool to promote multidisciplinary collaboration at design time, enabling relating and objectifying perspectives so people are learning about each other's work, methods, and concerns and thus are better able to discover and reconcile design tradeoffs. A Brahms simulation is also useful for communicating designs to other stakeholders, such as managers and adopters. Brahms simulations can generate metrics for predicting probabilities of different kinds of outcomes, costs, timings, etc., to justify further concept elaboration and verify designs; these metrics are more precise than task and workflow models by virtue of representing spatial-temporal interactions in more detail.

Brahms Design Process Overview

Brahms agents can run on different computer platforms and communicate using a variety of network protocols using a service-oriented architecture with VMs. Agents send messages in structured natural language—facilitating interaction with people. These agents can be integrated with simulation-based ITS environments (e.g., to simulate other entities in a tactical simulation), as well as software and hardware

elements. Bringing together the systems engineering phases that Brahms facilitates—analysis, design, evaluation, and experimentation—with implementation itself, Brahms enables a simulation-to-implementation research and development (R&D) methodology (Clancey et al. 2008).

The work systems design approach seeks to properly relate people, technology, facilities, and work processes into a coherent system of interactions. In considering the larger sociotechnical system in which human-automation interactions occur, work systems design aims to develop human-centered systems by which technology is designed as a tool that fits organizational roles, protocols/procedures, communication practices, facilities (e.g., physical layout), and especially the need for strategy, improvisation, and workarounds. Accordingly, we have developed a methodology of empirical requirements analysis that iteratively improves system designs by experiments with prototypes that combine ethnographic observation, work practice modeling, and what-if redesign of the system (Clancey et al. 2005).

To evaluate a work system design and its technology, we need to know how the total system will perform. For example, the effectiveness of an Apache pilot is evaluated in the context of the overall mission objectives. The information it provides and its actions are potentially relevant and important in the coordination and outcome of operations potentially involving many distributed agencies and teams. Work practice simulations enable us to understand and predict how distributed people and automated systems will interact, producing sequences of events that may be unanticipated and together constitute the output or accomplishment of the total system.

Crucially, Brahms models are crafted to be general and adaptable, so they can be configured to simulate a large space of scenarios. This enables verifying that the total system design satisfies success criteria (or fails appropriately) in a wide range of conditions and finding ways of optimizing it under different workload and other contextual assumptions. In this respect, the simulation becomes a research and experimentation tool for defining, exploring, and understanding the range of scenarios that may occur, refining roles, protocols, tools, and automated systems to cope with extreme, non-routine variations that may occur (e.g., see Clancey et al., 2013).

To accomplish this flexibility, people, technology (manned aircraft, unmanned aerial vehicles, computer systems), and the environment (e.g., remote facilities and the topographic model of the operational environment) are modeled as separate entities with their own internal behaviors (i.e., interactive, dynamic processes). Different entities interact by communicating (e.g., data transmission, email, loudspeaker), observing through sensory systems (e.g., human perception), and direct manipulation (e.g., external controls).

In summary, the main idea of work systems design is to develop technology by a systematic, scientific methodology that contextually relates technology to human practices that involve people playing different roles using automated systems in a simulated environment (Sachs, 1995). The work system design methodology incorporates participatory design (making "users" of technology part of the R&D team; Greenbaum & Kyng, 1991), participatory observation (i.e., providing researchers access to the operations team & work setting; Spradley, 1980), ethnographic studies (placing work systems in a socio-cultural context; e.g., Suchman, 1987; Jordan, 1992), and work practice simulation (modeling the behavior and interactions of people and technology on multiple design levels in different simulated conditions). Brahms simulations incorporate a great deal of detail, but provide a way of organizing and understanding the details of different system components and how they interact in time and space.

Domain Modeling: Socio-Technical Factors and Requirements

Domain Modeling Characteristics and Requirements

Developing tutoring systems to train personnel to succeed with each other and in concert with information systems requires domain models that capture the context and contribution of human and autonomous system behaviors, particular for learning objectives related to conditions of degraded command, control, and communications (C3) and automation support. An overall simulation or ITS requires fully interoperable models and valid environments grounded in theories of human cognitive and social systems. The domain modeling must therefore support total system simulations—incorporating models that define and relate assumptions about people and their work practices, roles, and protocols; representational tools and interfaces; facilities and layout; autonomous vehicles and other forms of autonomy; external threats; and the physical environment. This approach to domain modeling can best support training for the complexity and range of possible scenarios in complex, fluid socio-technical environments.

Domain Modeling Gap

Tutoring systems and simulations that situate instruction in socio-technical contexts generally adopt an object-oriented domain modeling paradigm where affordances in a domain (e.g., a computer display) encapsulate a set of behaviors. Behaviors can be scheduled or triggered through an action or event, or by proximity of the user or another entity. While tractable and reusable, this approach is not intended to capture a continuum of disruptions and thus generally embodies fixed assumptions, including conditions (nominal), user roles (immutable), order and duration of tasks (consistent), and access to and reliability of information (uninterrupted and correct).

Another approach that could support more robust socio-technical domain models is workflow modeling, supported by recent industry standards like the Business Process Execution Language (BPEL), an extensible markup language (XML)-based language with structured programming concepts. Workflow models, though, are based on a functional flow-based abstraction that models defined tasks and operations such as those formalized in business procedures and are thus ill suited to capturing the dynamics and uncertainties of complex operations in contested environments. Some systems such as I-X (Wickler, 2007) have an "activity" construct, but the interpretation is in terms of functions (tasks) and plans. Such models do not include or model how "off task" activities are performed, such as managing disruptions and providing assistance to teammates. For this reason they typically do not model "a day in the life" of an individual working with other humans and with automation, particularly under off-nominal conditions.

A different approach to modeling agency in a domain is to apply a Belief, Desire, Intention (BDI) framework (Bordini, et al., 2005). While this approach offers some utility in agent-based modeling, BDI systems are not sufficient because they do not, generally speaking, model objects, people, and systems as independent processes interacting in a simulated environment. For ITS domain modeling, in particular, BDI frameworks would have deficiencies in modeling the "total picture" including the environment (places, buildings, roads), where the agent is located in the world, how perception is affected by the agent's location, how perception is affected by the agent's conception of the present activity, and behavioral interactions among objects and systems. For domain modeling of the type we are discussing, namely, training personnel for human/autonomous operations in denied or contested environments, BDI frameworks are not equipped to model how, when, and where people interact with computer interfaces; protocols of how people carry on a conversation; or chronological activities in "a day in the life" of each agent.

Another, related approach is to employ cognitive modeling principles for capturing the behavior of intelligent agents and systems in a domain model. Commonly used architectures such as Soar and Adaptive Control of Thought—Rational (ACT-R) share with Brahms the basic cognitive distinction between declarative memory (beliefs) and procedural memory (activities/workframes). Cognitive modeling approaches, though, emphasize how behavior is the product of internal neuropsychological encoding, retrieval, latencies, and other cognitive phenomena. While they may model relevant reactive and inferential capabilities, cognitive agent architectures based on theories of human thought and reasoning are not designed to model the sort of embodied theories of attention, deliberation, and action that locate all actions, including mental operations, within a physical setting. This limits the reach of strictly cognitive approaches in modeling how distributed, multi-actor activities, functions, and tasks are accomplished in practice (Lave, 1988).

Bridging the Domain Modeling Gap with Brahms

Domain models for ITSs must support, as we have discussed, a broader socio-technical context, particular for training learners to successfully continue to perform with each other and with their automation systems under degraded circumstances. The gaps presented above can be bridged through a modeling approach using Brahms, because as described above, it is based on socio-cognitive theories of perception, inference, communication, and collaboration (for example, Ehn 1989; Greenbaum & Kyng 1991; Sachs 1995).

Brahms includes the reactive and inference modeling capabilities of other computational architectures based on theories of human thought and reasoning, but is based on an embodied theory of attention, deliberation, and action. Brahms combines an agent's internal state (possible/incomplete activities, plus beliefs, which can represent plans and goals) with a complex modeled environment to determine next behaviors.

In contrast to workflow models, Brahms' activity-based approach represents how functions are carried out in practice (Greenbaum & Kyng 1991; Sachs 1995). This emphasis on "what actually happens in practice" distinguishes Brahms from frameworks that abstract work into functions and tasks. Brahms is a descriptive (behavioral-cognitive) model with prescriptive design implications. In contrast, many representations used for instruction are purely prescriptive/normative; they specify what is supposed to happen, omitting particulars of when, where, with whom, using what tools, etc., that an activity model represents. For example, a workflow model might indicate that a job is conveyed from one office to another; an activity model would indicate how that happens, such as using an online dropbox, using the server for the dropbox, defining who uses the dropbox software and who uses what computer, noting its location, determining how the software is used (e.g., referring to a sheet of paper?), and detailing what the person does when required data are missing. This inclusion of "practical" and "logistic" aspects of work, beyond often well-articulated definitions, rules, and procedures, gives Brahms-style activity simulations advantages for tutoring how to carry out a technical task in a complicated environment.

Brahms is distinguished from BDI systems (e.g., Jason and AgentsSpeak) by modeling an agent's conceptualization of activities as parallel-hierarchical processes in the subsumption architecture. This allows modeling how activities are like identities that blend and contextually change what is perceived, how communications are interpreted, and how tasks are prioritized.

Brahms also contrasts with cognitive modeling frameworks that focus on simulating "the mind of the agent". Of particular importance to modeling human/autonomous operations in austere or contested environments is locating all mental activity and physical actions in space, which Brahms inherently supports. The Brahms engine simulates where every agent and object is located at every clock tick, as well as what

can be heard or seen at each location, such as whether an agent can hear someone speaking nearby. Although not intended to be a general spatial representation language, the Brahms language provides constructs and built-in operations that support modeling the environment and processes within it. If, for instance, an agent/object contains or holds another agent/object, as in an aircraft with a pilot, then the location of the contained object (the pilot) is updated automatically when the object that holds it (the aircraft) moves.

Modeling domains characterized by interactions and, potentially, disruptions to those interactions is support by Brahms' capabilities to simulate chronological, located behaviors, meaning, how functions and tasks are accomplished in practice. Actions involve perception and motion in space and may involve interruptions, metacognition, informal assistance from others, or workarounds. Brahms explicitly models context-sensitive perception, and its subsumption architecture allows composed activities at different levels of abstraction to "run" in parallel (e.g., handling an incoming request is contextual, reflecting the agents' current roles and activity norms). This subsumption architecture distinguishes Brahms from conventional object-oriented programming as workframes may become active and interrupted or resumed in an activity hierarchy.

Brahms is thus designed to simulate the interactive behavior among people, systems, and the environment (e.g., how a pilot interacts with an onboard associate and coordinates with unmanned aerial system [UAS] crew operating supporting aircraft). Such behaviors can be observed, described, formalized, quantified, and predicted in Brahms. Emphasis is on modeling the "total system" to understand and simulate emergent outcomes, which are the product of what is happening in the world.

Brahms-CAST: Socio-Technical Domain Modeling of Denied Environments

In this section we present an example domain modeling effort intended to enrich simulation-based training in a specific tactical context. Although the training system in this example is probably better characterized as a simulation than as an ITS, we believe that the domain modeling aspect of this example illustrates how a socio-technical modeling framework can support technology-mediated training in general, and that the lessons from employing this approach in tactical simulations apply equally well to ITSs.

The abbreviated description of our preliminary design in this section illustrates a socio-technical approach to address the gaps detailed earlier in this chapter. Specifically, we are creating a simulation testbed that models a complete scenario, incorporating the agents, aircraft, and systems for C3, navigation and targeting as well as facilities, geography and threats. This domain model, called the Brahms Contested Airspace Simulation Testbed (Brahms-CAST), can be integrated with simulation-based training environments and enrich learning by incorporating socio-technical factors and effects into a mission scenario (such as in Farkin, et al., 2004). Brahms-CAST can also support the development of ITS, by providing domain models with the ability to exhibit and respond to socio-technical actions and events.

The example focuses on training personnel who work closely with automation systems to detect, diagnose, and work around disruptions. "Disruptions" can interfere with human-human teamwork as well as with interaction between an individual and an automation system. Consider, for example, disruptions to communication. Benign events like atmospheric or solar storms can degrade communications; hostile acts such as jamming, spoofing, and cyber-attacks can also adversely influence human-automation teams. Individuals and teams must be prepared to detect, diagnose, repair, and restore services while maintaining continuity and minimizing risk. And they must do so in close collaboration with their automated systems (which may or may not be reliable or even operational).

Willful disruptions to communication and networks is one of a cluster of hostile actions categorized as anti-access/area denial (A2/AD), a growing concern that is occasioning discussions among military leaders about training. A2/AD is a broad label intended to include threats to US forces' regional access and freedom of action. Because A2/AD strategies may include cyber-attack, electronic jamming, or kinetic damage to network facilities, military personnel must be trained to overcome disruptions to C3 and lapses in automated decision support capabilities while continuing the mission. There is thus a need to train combat forces to overcome threats that alter the way people work with each other and with their information systems.

Scenario

A key requirement in developing the domain model is identifying the critical phase(s) of an overall activity during which emergent interactions may occur and have a bearing on the intended purpose of the simulation, such as planning, training, or developing procedures. Our methodology thus begins with developing a scenario to identify the players and properly contextualize the activities to be modeled. In the case of Brahms-CAST, the scenario is a hypothetical sequence of events depicting an A2/AD action against US air combat forces by a fictitious near-peer adversary. In the nominal, or baseline scenario, four F-16C tactical fighter jets launch to conduct a mission with a standard sequence of preflight, taxi, takeoff, tanking, push point, initial point, target, and return to base. The formation departs Hill Air Force Base (AFB), performs air refueling with a KC-135 Stratotanker, conducts tactical mission checks with the E-3 airborne warning and control system (AWACS) aircraft, and flies to the target in the Tonopah Test Ranges. The scenario defines the players, locations, timelines, and in general terms, the sequence of events.

The top-level entities we created are a formation of four F-16s, the AWACS, and the Stratotanker. The latter two were modeled to fly appropriate flight paths and to communicate with the F-16s for air battle management and air refueling, respectively. We placed greater emphasis on the fidelity of the F-16 models. The adversary agents in the scenario represent the source of electronic warfare (EW) effects (i.e., jamming and spoofing).

We also identified the equipment and systems needed to complete the mission. This enables the Brahms-CAST system to simulate, for instance, the processes activated when one pilot communicates with another. Figure 1 demonstrates the intermediate steps of such a transmission.

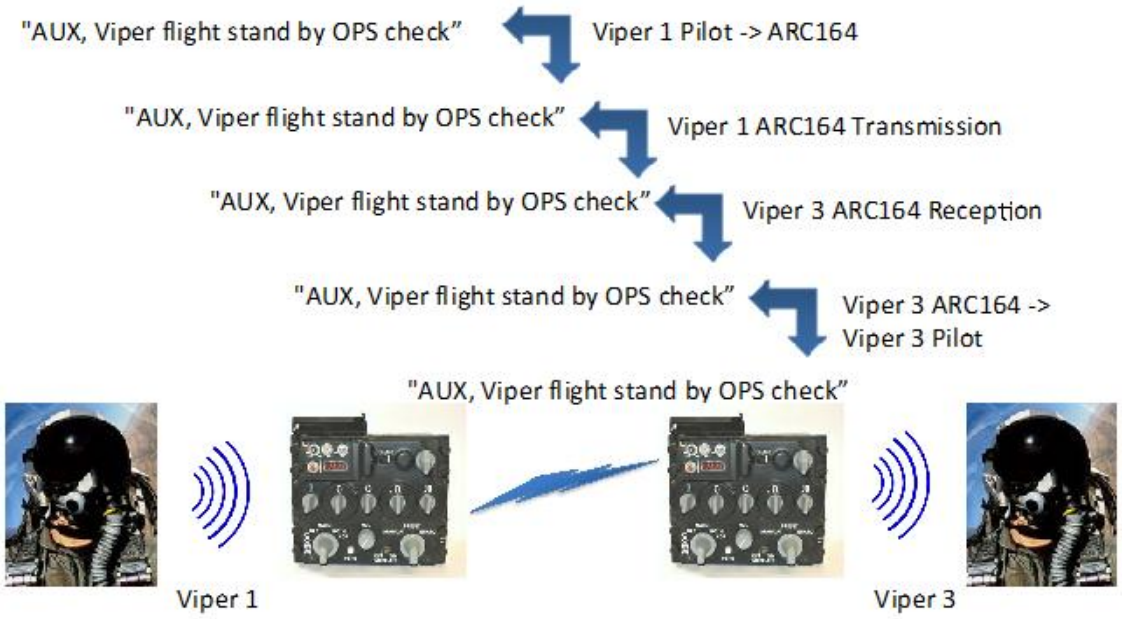


Figure 1. Schematic depiction of socio-technical modeling of air-to-air communication.

Domain Modeling

Once the scenario has been specified, our domain modeling process proceeds in two principal steps: specify (the outcome of which we refer to as a model sketch) and design (the outcome of which is the domain model created in Brahms). While scenarios guide the model specification process, the focus must be on modeling the domain in a general way, including the people, instruments, automated systems, aircraft, and other objects. Conventional approaches to modeling failures and other off-nominal states tend to focus on the details of the particular failure mode but seldom go farther to understand the normal operations in the manner that a general model requires (i.e., a model that can be run on a wide variety of scenarios). These approaches are unlikely to yield models that represent how an activity is carried out normally in practice or how an automated system such as an instrument or display normally operates. A baseline domain model should therefore capture in considerable detail the normal practice for activities that will be affected by scenarios in which special events of interest occur. The specification process is thus performed in great detail because of the importance of getting the models correct – all subsequent scenario runs, variations, and analyses are based upon the correctness and completeness of the model sketch.

From the scenario, we developed the domain model using our work practice methodology and implemented in Brahms. It is important to note that the model, although derived from the scenario, is not a "hard wired" model but a domain blueprint that is variable depending on initial conditions and assumptions. Part of the novel utility of this approach is the ability to run the model over numerous scenarios and analyze outcomes and how different assumptions and conditions can lead to different patterns of results.

The model sketch elaborates the scenario by describing in detail the specific actions, behaviors, and alternative courses of action the agents in the scenario are capable of exhibiting. As a brief example, we discuss the portion of the model sketch devoted to the Push phase of the scenario. As the formation (designated Viper 01 Flight) approaches the push point, they check in with the Airborne Warning and Control System (AWACS), call sign "Darkstar". The model sketches both communications and activity, as illustrated in the short excerpt below:

Tactical Check-in:

- Flight Lead checks in with AWACS and passes the following items at a minimum:
    - Flight/Package status with any alibis, direction, requests (as fragged, rolex, msn change, etc.). AWACS will initiate a Bullseye Check and Flight Leads will acknowledge, #2/3/4 by exception. Request Parrot/India check to ensure all Modes/Codes are properly identified.

Aircraft Check-In:

- Bullseye Check: Accomplished during aircraft check-in with TAC C2 (AWACS, call sign "Darkstar").
    - Viper 01: "AUX, Viper flight push PRI FMT 001, AUX SECURE."
    - Viper 01: "Viper 01 check FMT 001."
    - Viper 02-04: "2,3,4."
    - Viper 01: Viper 01 check AUX SECURE."
    - Viper 02-04: "2,3,4."
    - Viper 01: "Darkstar, Viper 01, Checking in as FRAGGED (all A/C are here), request bullseye check."
    - Darkstar: "Viper 01, Darkstar, bullseye check 360/27, Angels 21."
    - Viper 01: "Viper 01 Same."
    - Viper 01: "Darkstar, Viper 01, has AO Update Delta, standing by for additional words."
    - Darkstar: "Viper 01, Darkstar, AO Update Delta is current."

Roll call is initiated by AWACS after general communications and tactical parameters have been passed (e.g., words changes, package status, rolex, weather plan, key mission enablers, and LOWDOWN). The roll call order should be briefed during mission planning. Command and control (C2) will poll any flights that do not answer the roll call.

Viper 01 establishes communications (comms) with AWACS and receives all the required updates prior to the push. Viper 01 then enters a holding pattern east of the push point (approximately 135 nautical miles [nm] from the target area). At this time, Viper 01 continues to remain in the holding pattern until a push time of 0400L.

Off-Nominal Variations

The model sketch also defines how failures or disruptions are represented and how the agents respond. For instance, under A2AD-like conditions, upon nearing the target area the pilots begin noticing some minor comms jamming. Specifically some of the transmissions (approximately 25%) are clipped on the

Have Quick radio when jamming is active (we use "x" to replace characters in a transmission to denote garbled communications):

- Viper 01: " AUX, Viper 01, flow 18x, Vixxer, x1, set 40x."

In this situation, each pilot in the formation would recognize that an ARC-164 secure radio is being jammed and notify Viper 01. This has a minor effect initially, but as they continue to get closer to the target area the jamming would become progressively worse:

- Viper 03: "AUX, Xxper 01, Vixxx 03, Chaxxermark AUX FMT 001."
- Viper 01: AXX, Vipxx X1 fligxx, Chattermark AUX XMT 001."
- Viper 01: "AUX, Viper 01 flight check HQ FMT 001."
- Viper 02-04 "Viper 02 loud and clear, Viper 03 loud and clear, Viper 04 loud and clear."

At first, the Have Quick radio hopping frequency radios would help but as the 4-ship continues to get closer the four pilots would begin to see degradation in these radios as well. The flight lead might attempt to try other filtered multitone (FMT) channels but with no luck. Furthermore, Viper 01 attempts to contact Darkstar on satellite comms (SATCOM), but SATCOM is jammed as well:

- Viper 01: XXX, Vixxx X1 Flixxx, Cxxxxxxxxxk, XXX FXX 0XX."
- Viper 03: XXX, Vxxer X1, Xxxxr 03, say, axxxn Lxxxx transmxxxxxx?"

Approximately 50 nm from the target area all radios are unusable due to heavy jamming. The F-16s anticipate the comms jamming and continue to press toward the target relying on other equipment other than comms to keep situational awareness. The flight continues to maintain its current formation position using the comms out plan that was discussed in the mission briefing.

The model sketch further defines how conditions influence mission outcomes. Under denial measures such as radio jamming or global positioning system (GPS) spoofing (or both), effectiveness degrades, workload increases, and the reliability of targeting systems erodes. For a strongly degraded scenario, the mission is unlikely to succeed as exemplified in the model sketch excerpt below:

- With GPS spoofing present, the embedded GPS inertial navigation system (EGI) over time begins to think it is in a different geographical position than it really is. To further complicate the problem, approximately 30 nm from the target, Link 16 surveillance and precise participant location and identification (PPLI) tracks begin to fall off line. This is recognized by the wingman and they maneuver their aircraft closer to their flight leads. This action though makes it much more difficult for the wingman pilots to work their radar and targeting pod (TGP) sensors.

- Approximately 25 nm from the target, the F-16s begin searching the target area. Unfortunately, the TGP picture does not display the location that the pilots were briefed to search. The target area is in and around the location of a town/village. The pilots recognize the town, but, looking out into the dark desert, cannot figure out where the TGP is looking. To further complicate the problem, the wingmen have very limited cross-check time to figure out where their TGP is looking because they are spending so much time trying to stay visual. Furthermore, they are unable to communicate this because of the comms jamming that is present.

Scenario Implementation

The goal of the work being reported here was to implement enough of the model sketch in Brahms to validate the application of this approach to modeling denied environments and assess the viability of Brahms working in concert with other software technologies. We created detailed F-16 models (pilot, aircraft, flight controls, navigation, comms), as well as lower fidelity models of AWACS and Stratotanker aircraft. We also constructed smaller models for air control agencies and adversary electronic warfare (EW) agents.

Models were created within either agent or object hierarchies using the Brahms Composer (Figure). The Brahms Composer is a model development environment supporting classes and subclasses with full inheritance. The Brahms Composer includes a graphical editor for creating agent, object, conceptual object, and geography models, with access to source code editors. The Composer provides a common interface to build, execute, and analyze models. Models created using text editors can be imported. The object class selected in the right pane of Figure 2, for instance, shows the ARC-164 as a subclass of AircraftRadio, inheriting its parent properties (as well as the properties of the classes to which AircraftRadio belongs).

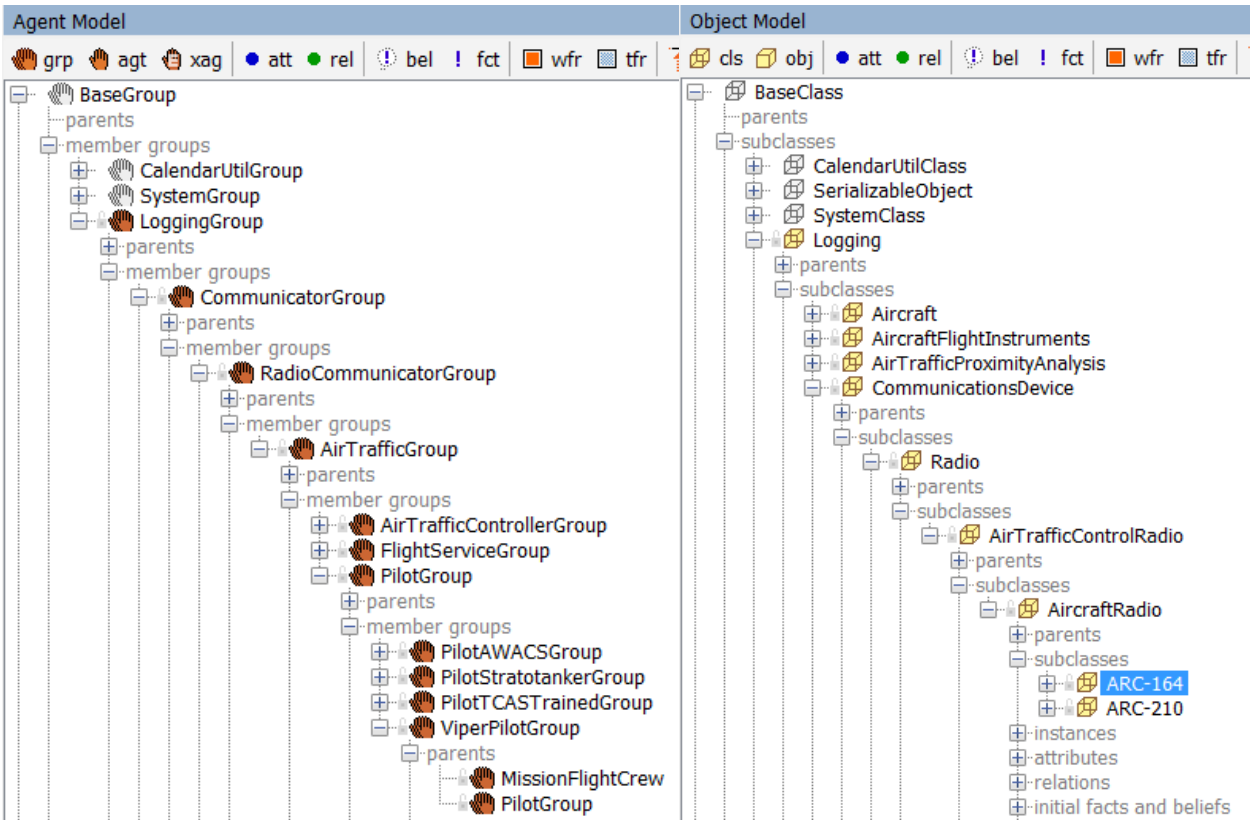


Figure 2. Models in Brahms-CAST showing agent (left) and object (right) hierarchies.

Brahms agents reason about beliefs, desires, and intentions using Thoughtframes. Behaviors in Brahms agents are encapsulated in representations called activities (describing what agents do) and workframes (describing when activities are performed). Figure 3 shows an excerpt from the F-16 pilot model that graphically depicts a portion of the activities hierarchy. Orange boxes are workframes, a torch icon sig-

nals activities like communications and moves, and crossed torches indicate composite activities that have several workframes and/or activities within them. For instance, the selected activity in the left pane, "Next_Flight_Route", is part of the composite activity "CheckPlaneLocation".

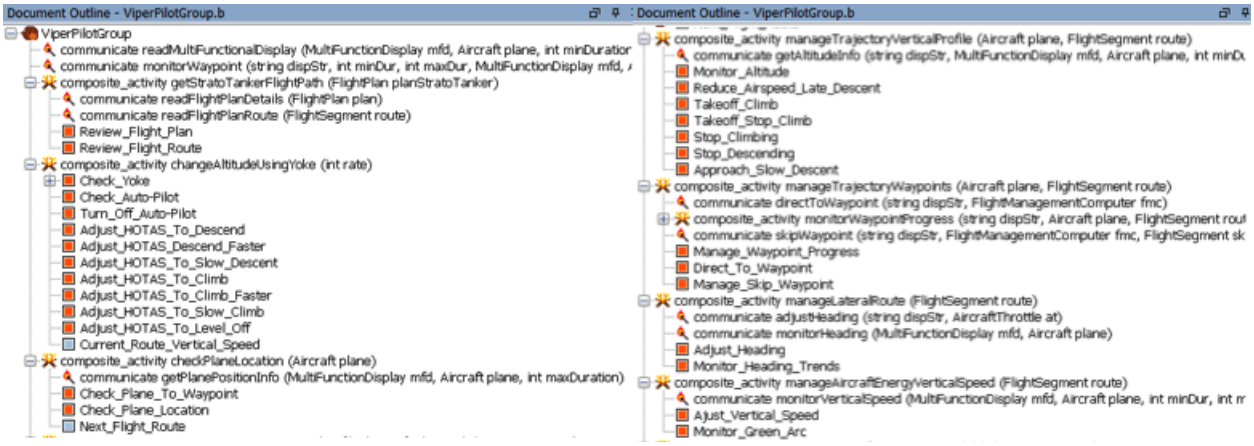


Figure 3. Model excerpt showing F-16 pilot activity hierarchy.

Once models are created, they are run and all event data are published both to an structure query language (SQL) database and to a log file. Multiple runs under differing sets of assumptions permit detailed analyses of how initial conditions and assumptions affect agent performance, mission outcomes, or other measures of interest.

Analyses of scenario runs are supported in Brahms by the AgentViewer, which parses history files generated by the Brahms VM. The AgentViewer provides a detailed trace of all events during a simulation, enabling a modeler to analyze and review the behavior of the agents, objects, and their interactions. The AgentViewer displays a timeline with agent activities, including inferences, communications, and movements. Drilling down displays details of an agent's belief state and world facts at various points in time, linking to workframes and thoughtframes that changed facts and beliefs. It thus permits the analyst to view in numerous ways, time scales, and levels of granularity the activities executed during a scenario run.

We ran scenarios for both and off-nominal conditions. Outputs were exported via a keyhole markup language (KML) export module developed for this project and to the SQL database. A glimpse of the utility of Brahms in simulating important work practice details is offered in Figure 4, which depicts air-to-ground transmissions between the flight lead and departure control early in the mission (vertical blue lines indicate communication). In the context of an ITS, for example, a learner could employ the AgentViewer to drill down to understand how communications are managed and other resources enlisted when workload exceeds timely response to all matters requiring attention, and what the consequences are of various task management alternatives.

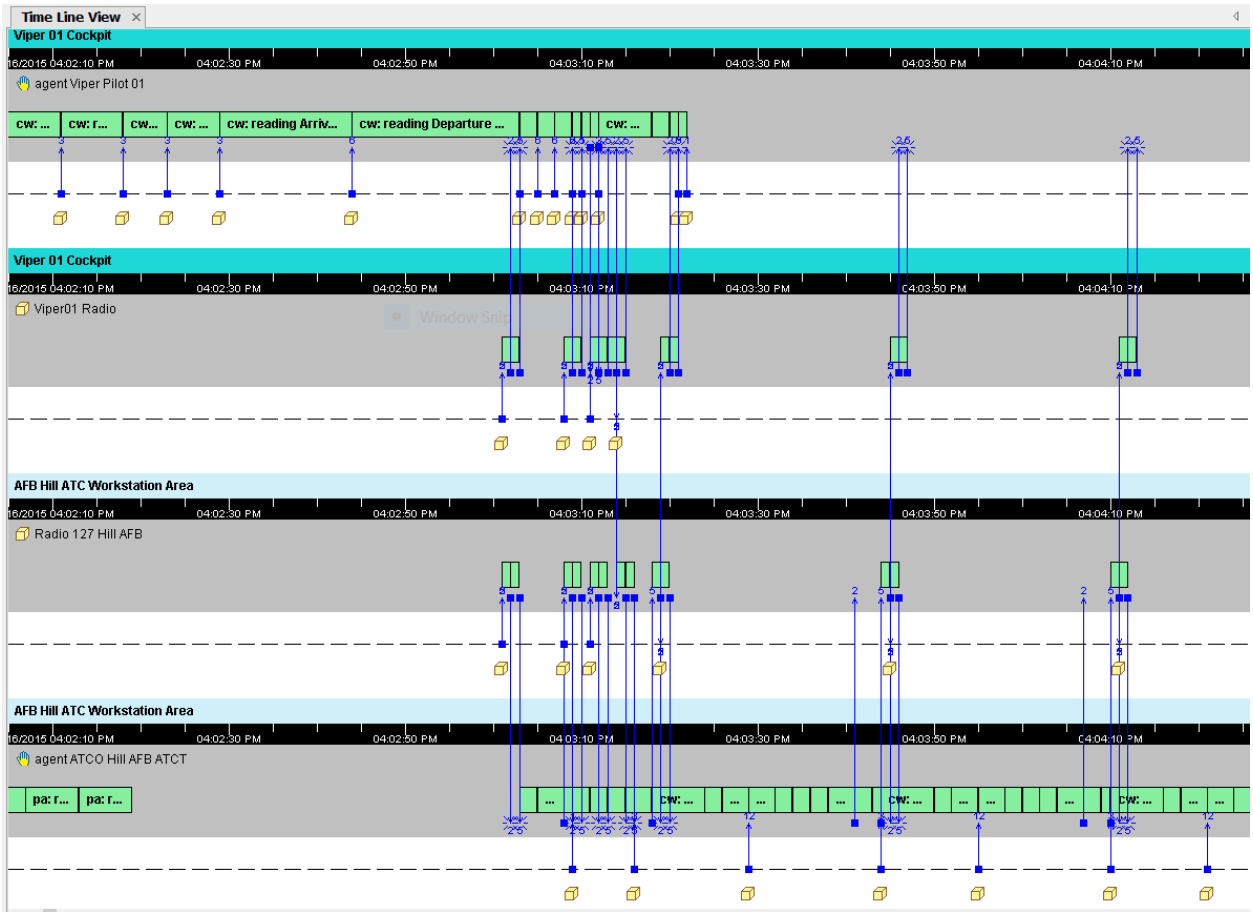


Figure 4. Time-expanded AgentViewer display showing flight lead communications with departure.

Contested Environment Simulations

As a proof-of-concept illustrating Brahms-CAST applied to mission plan evaluation and related assessments, we designed denied scenarios with A2AD effects, and then ran the model against these scenarios. The outcomes data generated from these scenarios demonstrates how Brahms-CAST can be employed for mission plan evaluation.

To model communications jamming and GPS spoofing, we incorporated into the radio models a jammed state during which the radio transmits and receives garbled messages. The level of signal attenuation is modeled by garbling a percent of the message based on distance of the aircraft from the jammer. Garbled messages are modeled as transformed text strings, where GARBLE tokens are substituted randomly for tokens in the message (e.g., "AUX, Viper GARBLE GARBLE, 112 is 256, 346, 1200"). The jamming percent determines the number of GARBLE tokens relative to all tokens in the message. We further model jamming percent as a normal probability function, so, for example, 75% = 6.75 tokens on average (i.e., actual garble for a message with 9 tokens might be 5 tokens or 8 tokens, it is not simply rounded to 7 tokens). Note that transmissions are not actually text strings but structured objects, which enables richer simulation of the effects of denials and countermeasures on mission outcomes.

We created a test variation matrix to capture which conditions were being varied and how for each scenario. Two variables manipulated in the matrix are GPS spoofing and radio jamming. Radio jamming can

have a value of “off”, “always”, or “probabilistic” (governed by a probability function). GPS spoofing can have a value of “off”, “denied always”, “denied probabilistic”, “spoofed always”, or “spoofed probabilistic”. These variables, arranged in the test variation matrix, yield 15 variants that are incorporated into the test plan.

We developed probability functions that determine the conditions for any given scenario. The jamming/spoofing (J/S) probability function is defined as (D – d)/D, where D is the distance between the push point and target, and d is the aircraft’s distance from the target. In other words, the probability of jamming/spoofing is inversely proportional to the aircraft’s distance from target. This suggests how Brahms-CAST can be used as an evaluation testbed to simulate mission plans and gather metrics on mission outcomes. Note that probability functions can be arbitrarily complex in order to more realistically model the effectiveness of A2AD tactics and countermeasures.

We capture the results in the SQL database and event logs for analysis and use the KML export module to view the results in Google Earth Pro. The behaviors presented below are for demonstration purposes and do not realistically model jamming technologies or actual operations. For instance, in the jamming scenario the wingman is unable to locate the target, though in practice could continue to the target; theater rules of engagement would define lost comms procedures and 4th-generation fighters would likely remain in visual contact in any case.

For our experimentation under jamming conditions, results show that the flight lead is able to locate the target (Figure 5). However when we investigate the outcomes, we notice that comms were disrupted and that Viper 02 was unable to locate the target. Figure 6 shows three panels from the AgentViewer depicting jamming (left panel), garbled voice communications (center panel), and the resulting failure of Viper 02 to locate the target (right panel).

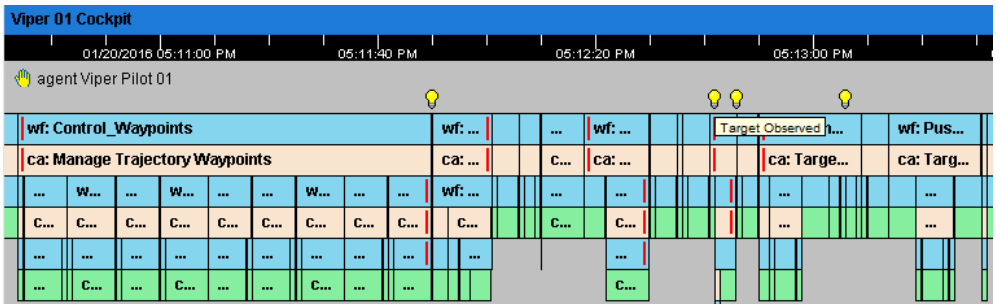


Figure 5. Viper 01 locating target in jamming conditions.

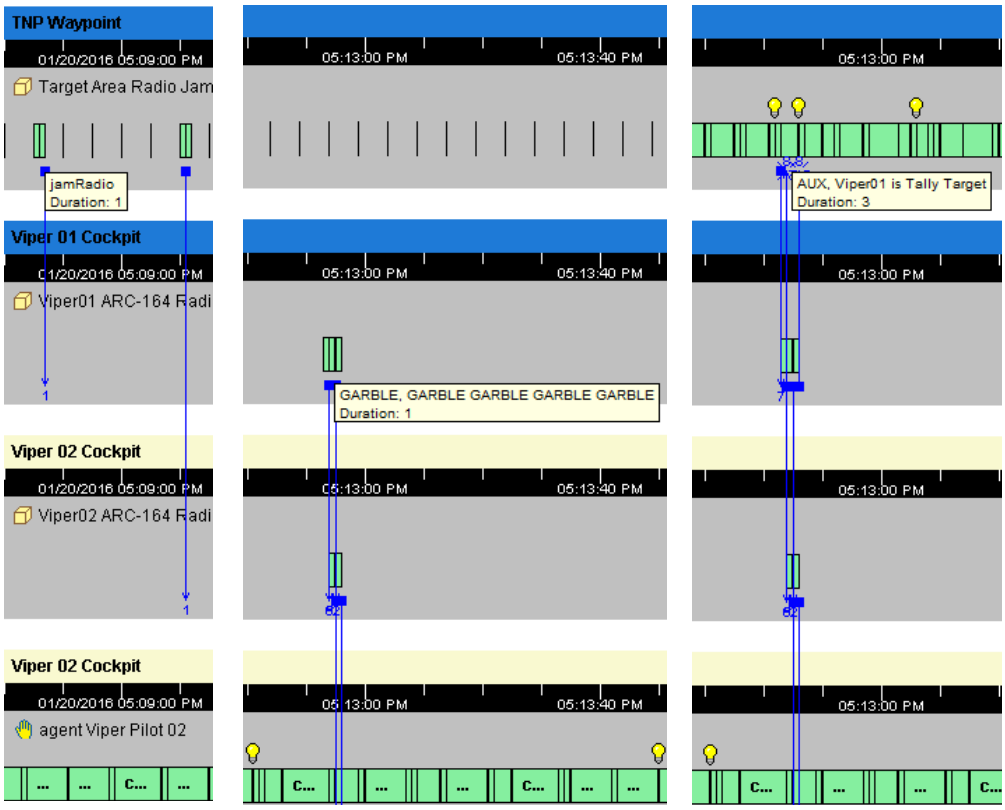


Figure 6. Jamming effects (left) result in garbled comms (center) and Viper 02 failing to locate target (right).

To model GPS jamming effects, we introduced a delay in the GPS receiver that causes a lag in updates to navigation display. This time delay is not intended to be a valid model of how GPS jamming actually manifests but will affect simulations of where the aircraft is headed during piloted flight to a target location. The GPS jamming device is modeled as an object located at the target location. This object monitors for aircraft within 100 nm of its location. The strength of the jamming effect is modeled to be configurable and depends on the distance of the aircraft from jamming device; the closer an aircraft is to device, the more it is subject to jamming.

For GPS spoofing scenario runs, we located a GPS spoofing object (an abstraction of an EW emitter) at the target shelter. In a typical spoofing scenario run, as an aircraft approaches, the spoofing object generates erroneous GPS coordinates and sends these coordinates to the F-16's GPS receiver. The left panel in Figure 7 shows one of the F-16s' radio ("Receiver Viper 01") receiving signals transmitted by this spoofing object model during the mission timeline. The center panel in Figure 7 shows a portion of one of the F-16 pilot models as it approaches the target and erroneously diverts away from the target. The right panel in Figure 7 shows one of the F-16s' TGP object model searching a target location offset from the desired location as a result of the spoofing. To visualize a GPS spoofing scenario, Figure 8 shows the flight lead successfully completing a target pass but the rest of the formation beginning to exhibit the effects of spoofing, deviating from course, and missing the target.

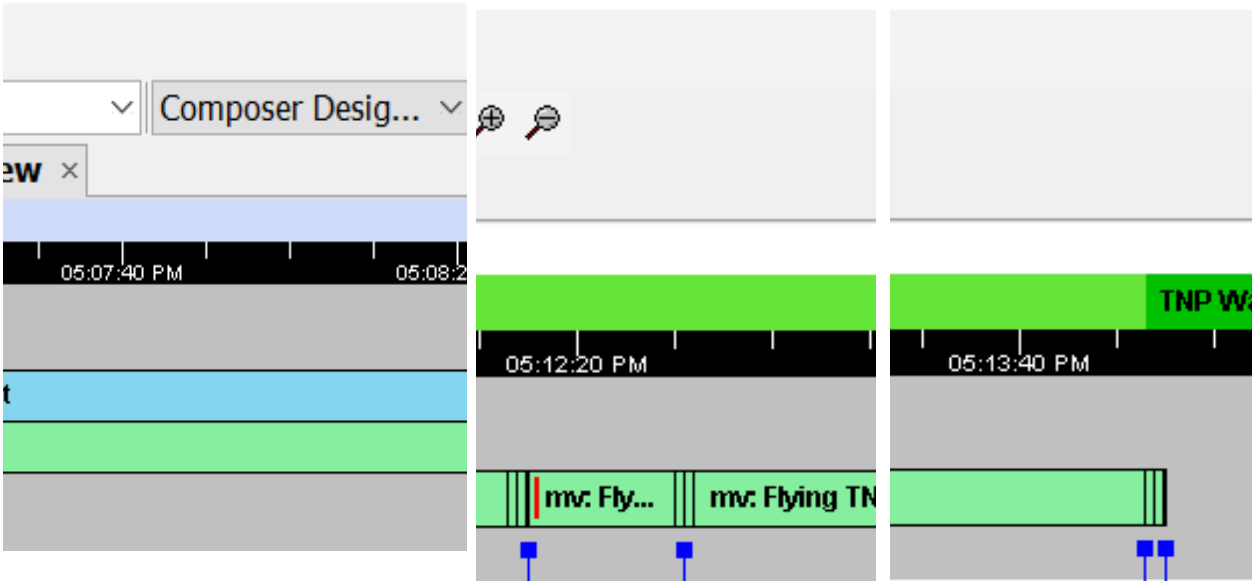


Figure 7. GPS spoofing effects (left) trigger erroneous heading change (center); targeting pod error (right).

Figure 8. Map view of course deviations caused by GPS spoofing, as target is approached and missed.

The utility of Brahms-CAST as a domain modeling tool for analysis and experimentation arises from the ability to run the model numerous times for any given set of initial assumptions. Outcomes can be tabulated as depicted in the example in Figure , where parameters like whether the target was located and how close each aircraft was to the target location can be collected and analyzed. Our tactical results are not intended to be realistic, since assumptions in the model are based on unclassified information and simplifying assumptions. These demonstrations, however, illustrate the potential of Brahms for supporting a diverse array of mission planning evaluations, after-action reviews, contingency analyses, and other assessments.

Run	GPS Spoofing Starts	Target Observed?				Target Observed Time				Target Observed at Distance (nm)				Closest Distance to Target (nm)			
		Viper01	Viper02	Viper03	Viper04	Viper01	Viper02	Viper03	Viper04	Viper01	Viper02	Viper03	Viper04	Viper01	Viper02	Viper03	Viper04
1	1:15:56	Yes	Yes	Yes	No	1:14:30	1:15:17	1:14:17		9.98	7.68	11.91		1.05	0.45	0.44	3.05
2	1:14:13	Yes	Yes	Yes	Yes	1:14:53	1:14:34	1:16:00	1:15:05	10.11	11.29	5.68	9.77	1.1	0.73	0.73	0.46
3	1:13:13	Yes	Yes	Yes	Yes	1:13:13	1:15:23	1:14:19	1:16:24	11.24	4.21	12.34	10.67	0.43	0.73	0.74	1.67
4	1:13:21	Yes	No	Yes	Yes	1:13:31		1:15:21	1:16:01	7.9		10.25	9.8	0.43	2.97	6.25	1.45
5	1:14:24	Yes	Yes	Yes	Yes	1:13:58	1:14:36	1:14:55	1:14:55	7.45	9.88	10.41	12.14	0.42	0.44	0.84	0.71
6	1:12:32	Yes	Yes	Yes	Yes	1:12:13	1:15:00	1:15:50	1:16:07	8.31	10.98	12.44	10.76	6.09	6.26	1.47	1.06
7	1:13:36	Yes	Yes	Yes	Yes	1:14:02	1:13:52	1:14:06	1:14:44	9.25	12.98	12.22	10.64	0.84	0.96	1.61	5.25
8	1:16:40	Yes	Yes	Yes	Yes	1:14:07	1:13:42	1:14:16	1:16:13	11.04	11.42	10.38	7.09	0.7	0.5	0.46	1.48
9	1:11:50	Yes	No	No	Yes	1:13:39			1:15:55	10.2			9.22	0.8	8.99	8.97	1.48
10	1:12:52	Yes	Yes	Yes	Yes	1:13:14	1:14:36	1:15:14	1:15:39	10.08	10.84	9.84	10.96	0.63	6.2	1.46	0.99

Figure 9. Tabulating GPS spoofing results for given initial conditions and assumptions.

Of greatest relevance to ITSs is that, as a generative simulation, Brahms offers a mechanism for a tutoring system to vary assumptions and conditions in the domain model, which can enable an adaptive tutor to modify the level of difficulty or to emphasize key principles.

Discussion: Socio-Technical Factors

This series of events is described above in a manner similar to how conventional modeling paradigms could simulate this scenario. However an ITS or simulation can be far more effective if the underlying domain modeling supports simulation of the socio-technical effects, such as the interaction of ongoing coordinated activity, movement, perception, and manipulation of instruments.

There are numerous interacting factors, not generally accommodated by most modeling approaches, that a socio-technical paradigm could capture and thus enrich a simulation or ITS. A circumstantial combination of events can lead to unanticipated interactions such that people, objects, and systems become causally interdependent with undesirable feedback. Simulating these as independent processes in a simulated world, as in Brahms, has the potential to reveal emergent effects. For example, jamming mentioned above would increase the overall stress level of the pilots (they hear tones in their cockpit from the jamming). Turning down the radio to reduce the volume of the ringing makes it difficult to hear radio transmissions, so communication among the team is degraded. Under conditions of SATCOM jamming, pilots can no longer receive broadcast control from AWACS, which would as a result spend more time checking their air-to-air radar to ensure that no hostile aircraft are airborne. They also devote time during their cross-checks to ensure no surface-to-air missile (SAM) systems are active and emitting. Less time is spent looking at their TGP and at the target area. These user actions and configurations shape the information landscape a user operates in and influence workload, stress, coordination, and team effectiveness. Loss of routinely automated functions requires shifting activities to a different mode of operation that makes accomplishing the mission more difficult.

In particular, when the four ships in the scenario change to a 2+2 line abreast, lost comms means the pilots devote more time to looking outside to confirm the formation is correctly. Under conditions of Link 16 dropout, tightening the formation makes it much more difficult for the wingmen to work their radar and targeting pods. The spoofing and jamming also require each pilot to align the TGP to look in the direction of the target by correcting where the pod is currently looking. This can be a difficult task for the pilot while maintaining a good cross-check with navigation instruments as well as looking outside to maintain formation. This all suggests the need for socio-technical domain modeling that captures the interactions among pilots, aircraft, Link 16, radar and targeting systems in terms of the pilot's overall coordinated activity of monitoring/perceiving, inferring, communicating, and manipulating systems.

Implications of Socio-Technical Representations for ITS Domain Modeling

Domain models generally support a simulation or ITS by furnishing an environment that responds to user actions and generates and correctly propagates events. As our example illustrates, Brahms offers a capability to create a large collection of possible outcomes from initial conditions, each of which documents and reflects the socio-technical factors shaping an event sequence. Brahms-CAST is therefore not merely a hand-crafted, one-off replication of an A2/AD scenario. Rather, it consists of a generalization of roles and automated systems (e.g., pilots, air battle managers, radar, navigation and communication systems, etc.) that play a role in contested environments. Rather than representing only the states and behaviors of subsystems that obtain at a given point in time, Brahms-CAST represents nominal states and behaviors and allows for them to be configured for a diversity of scenarios to characterize alternative behaviors, including absent, alternative, and dysfunctional or off-nominal forms (e.g., dropped communications; inoperative navigation). Initial conditions can be configured to simulate infinitely different workloads (e.g., equipment readiness, threats, weather).

In this way, domain models can be employed to generate scenarios that ITS authors can sequence as part of a syllabus. Each of the many possible configurations of Brahms-CAST parameters (initial facts, beliefs, and properties/states) defines a scenario. The combinations of all possible parameter settings define a space of scenarios that Brahms-CAST is able to simulate with operational validity.

Brahms also enables ITS domain models that have probabilistic outcomes, deriving from (1) variable durations of primitive activities (those not modeled as composite conditional actions) and (2) how perception (e.g., noticing an indicator on a display) is modeled contextually as probabilistic "detectables." Thus, each simulation run of a given scenario (pre-defined initial conditions, e.g., location and effectiveness of the jamming and spoofing sources) produces time-space-state interactions with potentially different outcomes, some with emergent, potentially unanticipated sequences of events that reveal the effects of proximity and timing (e.g., when a pilot scans a display and notices a problem). A behavioral simulation with a short time increment (e.g., 1 s) is particularly advantageous for simulating such effects. Continuing our example scenario, it possible that in some simulation runs of Brahms-CAST a wingman notices the GPS spoofing and advises the lead, far enough in advance of reaching the target area to allow for compensatory measures.

Training developers can use this capability to examine complexities, workloads, and outcomes that result from different sets of initial conditions. Brahms-CAST components can also be adapted to model many related work systems (e.g., different aircraft, mission sets, threats). This flexibility to define additional combinations of people and aircraft of different types enables ITS authors to develop training variations of existing and future systems. Of particular interest in this example are those training systems that address new forms of autonomy and change how people interact with automation under different environmental conditions (e.g., A2/AD).

Brahms-CAST can also help ITS authors develop assessments, by revealing how the timing of events at the level of a few seconds can make a substantial difference in outcomes. In particular, Brahms-CAST simulates how subtle issues of timing in human-automation interactions arise when degraded or missing subsystems result in lack of information and inability to communicate, transforming a given configuration of routine tasks in a normal work system to a situation too complex for the overall work system to successfully manage.

In our example, the events in a tactical sortie reveal how people develop work practices in which they rely on automation, and how the absence of automation may cause the workload to increase and the evolving situations to become too causally codependent to appropriately prioritize tasks or delegate responsibility. That is, the workload has become cognitively complex relative to the person's knowledge, beliefs, roles,

habitual procedures, and tools (see Clancey et al. 2013 for a detailed example of emergent human-automation situations in air traffic control). This framework is thus ideally suited to train personnel in mitigating A2/AD effects.

Brahms-CAST demonstrates the strength of the Brahms framework for simulating behaviors of asynchronous (or loosely coupled), distributed processes in which the sequence of interactions among people and automated systems becomes mutually constrained and unpredictable. Creating and experimenting with Brahms work practice models reveal system interactions that may be omitted, glossed over, or difficult to comprehensively describe in after-action reviews. The simulation can be crafted to generate metrics that can be compared to observational data and/or make predictions for redesign experiments.

Conclusion

In this chapter, we apply a socio-technical modeling framework, called Brahms, that ITS and simulation developers can adapt as a domain modeling approach. Brahms is designed to simulate activities as chronological, located behaviors, that is, how functions and tasks are accomplished in practice. Activities involve perception and motion in space that are modeled explicitly through direct support of the Brahms language and engine (e.g., perception is context-sensitive). In general, activities might involve interruptions, metacognition, informal assistance from others, or workarounds, which can be simulated within the Brahms architecture.

The activity modeling and simulation framework at the heart of Brahms-CAST offers ITS authors a powerful way to think about domain knowledge and expertise because it integrates cognitive processes (perception, inference, belief maintenance) with procedural, interactive behaviors (sequential conditional actions) in a modeled "geography" of areas and paths, an environment with objects/systems having their own behaviors (e.g., navigation displays, unmanned aircraft). Agent (and object) properties and behaviors can be inherited from groups (classes), facilitating model creation and reuse. Through multiple inheritance and blending, behaviors can be represented at different levels to model the effects of training and group practices, as well as individual preference. This framing of work practice in terms of activities, in contrast with task-oriented workflow models, facilitates simulating interacting, choreographed joint behaviors of a team in which individual agents may have complementary, incompatible, and/or the same capabilities.

Modeling domains and expertise in such an activity framework helps training developers capture nuances of the domain that are fundamental to what must be understood and controlled. In this chapter, we discussed an example of how coordination of assets in contested environments becomes highly dynamic and complex when communications are unreliable, intermittent, or not secure. In particular, the constructs provided by Brahms helps ITS creators model how data uncertainty and disengaged command will require humans and autonomous systems to adapt quickly, reconfiguring operational strategy and decision-making authority.

References

Bordini, R. H., Dastani, M., Dix, J. & Seghrouchni, A. E. F. (Eds.) (2005) Multi-Agent Programming: Languages, Platforms and Applications. New York, NY: Springer Science+Business Media, Inc.

Clancey, W. J. (1993) The knowledge level reinterpreted: Modeling socio-technical systems. International Journal of Intelligent Systems, 8(1), 33–49.

Clancey, W. J. (1997) Situated Cognition: On Human Knowledge and Computer Representations. New York: Cambridge University Press.

Clancey, W. J., Sachs, P. Sierhuis, M. & van Hoof, R. (1998) Brahms: Simulating practice for work systems design. International Journal of Human-Computer Studies, 49, 831–865.

Clancey, W.J. (2002) Simulating activities: Relating motives, deliberation, and attentive coordination, Cognitive Systems Research 3(3) 471–499, September, special issue on situated and embodied cognition.

Clancey, W.J, Sierhuis, M., Damer, B., Brodsky, B. (2005) Cognitive modeling of social behaviors. In R. Sun (Ed.), Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, pp. 151–184. New York: Cambridge University Press.

Clancey, W.J., Sierhuis, M., Alena, R., Berrios, D., Dowding, J., Graham, J.S., Tyree, K.S., Hirsh, R.L., Garry, W.B., Semple, A., Buckingham Shum, S.J., Shadbolt, N. and Rupert, S. (2007) Automating CapCom using Mobile Agents and robotic assistants. NASA Technical Publication 2007–214554.

Clancey, W.J., Sierhuis, M., Seah, C., Buckley, C., Reynolds, F., Hall, T., Scott, M. (2008) Multi-agent simulation to implementation: A practical engineering methodology for designing space flight operations. In A. Artikis, G. O'Hare, K. Stathis & G. Vouros (Eds.), Engineering Societies in the Agents' World VIII. Athens, Greece, October 2007. Lecture Notes in Computer Science Series, Volume 4870. Heidelberg Germany: Springer, pp. 108–123.

Clancey, W.J., Linde, C., Seah, C., Shafto, M. (2013) Work Practice Simulation of Complex Human-Automation Systems in Safety Critical Situations: The Brahms Generalized Überlingen Model. NASA Technical Publication 2013–216508, Washington, D.C.

Ehn, P. (1989) Work-oriented design of computer artifacts. L. Erlbaum Associates Inc., Hillsdale, NJ.

Farkin, B., Damer, B., Gold, S., Rasmussen, D., Neilson, M., Newman, P., Norkus, R., Bertelshems, B., Clancey, W.J., Sierhuis, M., Van Hoof, R. (2004) BrahmsVE: From human-machine systems modeling to 3D virtual environments. Proceedings of the 8th International Workshop on Simulation for European Space Programmes (SESP 2004), Noordwijk Holland, October 19–21.

Feltovich, P. J., Bradshaw, J. M., Clancey, W. J., Johnson, M., Bunch, L. (2008) Progress appraisal as a challenging element of coordination in human and machine joint activity. In A. Artikis, G. O'Hare, K. Stathis & G. Vouros (Eds.), 2008, Engineering Societies in the Agents' World VIII. Lecture Notes in Computer Science Series (pp. 124–141). Heidelberg Germany: Springer.

Freeman, J., Diedrich, F. J., Haimson, C., Diller, D. E., and Roberts, B. (2003). Behavioral representations for training tactical communication skills. In Proceedings of the 12th Conference on Behavior Representation in Modeling and Simulation, Scottsdale, AZ.

Greenbaum, J. & Kyng, M. (Eds.) (1991) Design at Work: Cooperative design of computer systems. Hillsdale, NJ: Lawrence Erlbaum.

Johnson, W.L. and Wu, S.M. (2008). Assessing aptitude for learning with a serious game for foreign language and culture. In B. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (Eds.), International Conf. on Intelligent Tutoring Systems (pp. 520–529). Berlin: Springer-Verlag.

Jordan, B. (1992) Technology and social interaction: Notes on the achievement of authoritative knowledge in complex settings. IRL Technical Report No. IRL92-0027. Palo Alto, CA: Institute for Research on Learning.

Lave, J. (1988) Cognition in practice. Cambridge: Cambridge University Press.

Lave, J. and Wenger, E. (1991) Situated learning: Legitimate peripheral participation. New York: Cambridge University Press.

Leont'ev A. N. (1979) The problem of activity in psychology. In Wertsch, J. V. (editor), The concept of activity in soviet psychology (pp. 37–71). Armonk, NY: M. E. Sharpe.

Sachs, P. (1995) Transforming Work: Collaboration, Learning, and Design. Communications of ACM 38(9), 36–44.

Schön, D. (1987) Educating the reflective practitioner. San Francisco: Jossey-Bass Publishers.

Spradley, J. P. (1980) Participant observation. Fort Worth: Harcourt Brace College Publishers.

Suchman, L. A. (1987) Plans and situated actions: The problem of human-machine communication. Cambridge: Cambridge Press.

Wenger, E. (1998) Communities of practice: Learning, meaning, and identity. New York: Cambridge University Press.

Wickler, G., Tate, A. & Hansberger, J. (2007) Supporting Collaborative Operations within a Coalition Personnel Recovery Center. Paper presented at the International Conference on Integration of Knowledge Intensive Multi-Agent Systems, Waltham, MA.

Wynn, E. (1991) Taking practice seriously. In J. Greenbaum and M. Kyng (Eds.), Design at work: Cooperative design of computer systems (pp. 45–64). Hillsdale, NJ: Lawrence Erlbaum Associates.

SECTION II

METHODS OF DOMAIN MODELING

Andrew M. Olney and Arthur Graesser, Ed.

CHAPTER 7 – Design and Construction of Domain Models

Andrew M. Olney and Arthur Graesser
University of Memphis

Introduction

The problem of domain modeling is to represent domain content so that it can be efficiently authored, optimally delivered to students, and precisely tracked with respect to student mastery. While the previous section focused primarily on the representation aspect of domain modeling, the present section focuses on various methods and concerns related to authoring, delivery, and mastery.

An overarching method that unites most of the chapters in the present section is a data-driven approach to domain modeling that uses machine learning. This is consistent with the current zeitgeist of Big Data, enabled by Internet-scale data sets and cloud-computing resources. With these data, researchers are able to parameterize increasingly complex models from data, perform model selection on alternatives of such models, and even author content using crowdsourcing or semi-supervised machine learning.

Data-driven approaches are just one part of the story, and if we look behind the curtain, we see that the following chapters are deeply nuanced in their treatment of domain models. Each chapter raises issues and concerns not only for what domain models currently are but also what they could be. This section provides an excellent overview of domain modeling and a guide to future research. For the purposes of this introduction, we focus on three major themes arising from this work: bringing external information into domain models, mapping items to skills, and creating domain models.

Theme 1: Bringing External Information into Domain Models

The chapter by Brawner, Goodwin, and Regan discusses the problem of hybrid systems where training is not bounded by a computer based environment but instead is also distributed across human based training. Ideally, the Generalized Intelligent Framework for Tutoring (GIFT) should be able to know about training delivered outside of itself in order to best adapt the training it delivers. This external training information could be longitudinal (in the sense of slowly acquired mastery) or immediate (in the sense of a lecture delivered by an instructor just prior to interaction with GIFT). Sharing information can be accomplished in three ways. First, sharing can use a high-level competency framework that is loosely coupled with actual skills. This essentially is a shared domain model at a high level, analogous to an upper level ontology. Second, sharing can use a low level approach where the set of problems and student responses is shared, and each system is responsible for inferring the competencies in which it is interested from this interaction data. Thirdly, machine learning could be used to unify these two levels to model what actions best predict competencies at the organization level. That is, the learning records of various systems are forwarded to the organization level system (e.g., Army-wide) for inference and matching to high level competencies.

The chapter by Sinatra shares this concern in the sense of wanting to use system-external information to adapt to the student, but the emphasis is on interest- and motivational-based enhancement of memory rather than adaptive problem selection or mastery. In other words, the focus is less about what to teach and more about how to teach. One technique is the Self-Reference Effect (SRE), a way of enhancing recall, retention, and motivation across many domains and age groups. The SRE can be created simply by changing the pronouns used in the delivery of instruction to reference the student ("your respiratory system") rather than the typical description in which the student is not a participant ("the respiratory system"). Sim-

ilar to SRE, context personalization, in which the interests of the students are incorporated into the material, can increase problem-solving performance, knowledge transfer to new problems, and positive attitudes about the material. These effects could be realized in GIFT by (1) authoring guidelines to enhance SRE, (2) adding user interest surveys to populate templated GIFT materials, such that interests could be substituted for topics as needed, or (3) supporting a diversity of interest-aligned content such that once interests are established, GIFT could preferentially select content matching those interests.

Theme 2: Mapping Items to Skills

The chapter by Goldin, Pavlik, and Ritter is concerned with mastery of particular knowledge components (KCs) and also defines domain models primarily in Q-matrix terms. Like the Desmarais and Xu chapter in the previous section, the focus is on refining the Q-matrix, but unlike the previous chapter, the objective is not to re-parameterize an initial Q-matrix (e.g., by changing the value in a particular cell) but to instead refine the structure of the Q-matrix itself by collapsing or splitting existing KCs and their corresponding matrix columns. The approach is based on learning curves, which are models of the error rate for a population of students as they attempt to learn a KC. Given a set of alternative domain models and associated learning curves, one can rank and select the domain model that best represents the learning curve data. The major focus of this chapter is on the qualitative depiction and analysis of learning curves for the refinement of Q-matrices.

Theme 3: Creating Domain Models

The chapter by Williams, Kim, Glassman, Rafferty, and Lasecki represents an emerging area of domain model authoring that relies on crowdsourcing. The goal of crowdsourcing is to shift the burden of authoring domain models from a handful of experts to capable students or novices. One of the principal problems in this approach is maintaining quality. This chapter focuses not on the creation of intelligent tutoring systems (ITSs) wholesale but instead on the addition of new content to an existing system. After solving math problems, students contribute explanations of their solutions that are rated by other students, and then machine learning is used to select the best explanations. Results show that students both learn more and that explanations (as ranked by the machine learning algorithm) improve over time. By focusing on self-explanations, this technique appears to be fairly generalizable and therefore useful for systems implemented in GIFT.

The chapter by Barnes, Mostafavi, and Eagle presents interaction networks, which are data-driven, problem-specific domain models based on student actions during problem solving. Interaction networks can be used to mark the correctness of student actions, model student ability, and adapt delivery of instruction. Interaction networks are particularly suited to state-space domains (like algebra and logic) where student actions can be viewed as edges between nodes representing states. Given even a small amount of data to characterize the solution space, interaction networks can be used to generate next-step hints. By clustering student action sequences in the interaction network, the problem-solving behaviors of groups of students can be grouped not only into successful versus unsuccessful approaches but also different successful approaches to the same problem. These methods can be extended to implement data-driven knowledge tracing and associated problem selection. This approach could be used in GIFT for state-space domains where a complete system exists (necessary to collect the student interaction data) but has no associated pedagogy.

The chapter by Dargue, Pokorny, and Biddle presents a specific variant of cognitive task analysis, called Precursor, Action, Results, Interpretation (PARI), for the purpose of building domain models based on expert mental models. PARI proceeds with a problem-solving dyad of subject-matter experts (SMEs), one

of whom simulates outcomes (e.g., equipment responses) to the SME engaged in problem solving. Using a set of pre-selected tasks to structure these interviews, PARI records the problem-solving process, step by step, in terms of four pieces of data: action precursor, action, result, and result interpretation. While the first two pieces of data may be considered as part of a production rule, the last two pieces correspond to the revision of a mental model given a result. In several respects, this can be viewed as the manual approach to the problem that the previous chapter tries to automate. However, this manual process could be used to author a domain model directly from experts rather than needing a system to collect data for data-driven methods to be applied. Because cognitive task analysis approaches are the most comprehensive, they continue to represent the gold standard for domain model authoring.

Implications for GIFT

These chapters provide guidance for the continued development of GIFT by illustrating what domain modeling methods are valued in the community and what impacts these methods have in past and ongoing research. Briefly stated these are as follow:

- What methods can be used for sharing domain modeling data across systems such that mastery can be tracked across systems and instructional adaptation can occur across systems?

- How can GIFT represent item-to-knowledge mappings (e.g., Q-matrices) in a flexible way that allows the Q-matrices to be restructured/re-parameterized as needed?

- How can GIFT support domain model authoring across the lifecycle, stretching from initial expert interviews, to data-driven refinement from use, to crowdsourced authoring of additional content?

CHAPTER 8 – Scaling Across Domains and the Implications for GIFT

Keith W. Brawner[1], Gregory Goodwin[1], and Damon Regan[2]
[1]US Army Research Laboratory, [2]Advanced Distributed Learning Initiative

Introduction

Currently, the Generalized Intelligent Framework for Tutoring (GIFT) adapts training in a somewhat closed system. That is, current GIFT applications are only concerned with modeling the learner and domain within the confines of a single block of training delivered exclusively through the GIFT framework. While this approach has been advantageous from the point of view of providing experimental control while exploring GIFT's utility and effectiveness, it only partially addresses what we know to be the "real world" requirements of an adaptive training system. Adaptive training systems like GIFT will be expected to function within a rich ecosystem of training events and include classroom lectures, group discussions, and hands-on training, as well as simulation, gaming, other computer-based training, and training outcomes. A single course might easily include all of these forms of training, while GIFT may be responsible for only a subset. For GIFT to be most effective in such an environment, it will have to be able to model the learner and domain at a higher level and not just for the specific blocks of training that it is responsible for delivering.

The reader may consider the following specific illustrative example. Suppose an instructor delivers a lecture on a topic and then the students are expected to build their understanding of that topic by completing some scenario-based training delivered by GIFT. For GIFT to adapt the scenario-based training to the students, it needs information on how well each learner understands the concepts taught in the lecture, perhaps by quizzing the students at the end of the lecture. As a result of the student's quiz performance, it may be instructionally relevant for GIFT to provide remediation for the concepts taught in the lecture. In other words, GIFT's domain model would need to include models of the concepts taught in the lecture, even though GIFT did not deliver that training. As another example, if GIFT were to function just like a human tutor, helping learners understand concepts with which they were struggling, GIFT would need to have a domain and learner model for the entire course, not just for blocks of instruction delivered by GIFT.

Language training and marksmanship training are real-world examples of the type of training described above. Neither language proficiency nor marksmanship expertise develops through training in single applications or over short periods of time. Rather, both require hundreds of hours of deliberate practice after mastering the basics. Training for both includes lecture, computer-based training, and live practice. While technology has been shown to help language learning (Zhao, 2003) and marksmanship training (Chung et al., 2011), studies are typically isolated to a single application over a short period of time (although there are some exceptions, notably Adair-Hauck, Willingham-McLain & Youngs, 2000; Green & Youngs, 2001; Spain et al., 2013). This chapter explores the challenges of scaling the domain model to encompass an entire course, specifically through the examination of these two training domains.

A Long-Term Vision for Technology Integration

The Army's long-term vision for the integration of technology into its courses blends classroom instruction, digital resources, and deliberate practice in virtual and live simulations supported by computer and human tutors. The Army Learning Concept 2015 (Command, 2011) proposes that this is the future of training. Blending these activities together over time requires coordination around shared learning data:

data about performances, learner models, competencies, and available instructional resources. Learners, instructors, human tutors, digital resources, and intelligent computer tutors could all use this data to adapt learning to the specific needs of learners across a set of interrelated lessons and courses.

Consider a possible near-future US Army Basic Rifle Marksmanship (BRM) scenario. A classroom instructor uses a dashboard to see and contribute to a persistent, interoperable learner model for each soldier (Goodwin, Murphy, Hruska & Consulting, 2015) to differentiate instruction (possibly with different available instructional resources or just a more flexible time and pace), engage in dialogue, and create enduring relationships. When a soldier moves on to practice what they've learned using a simulator, the simulator can capture performance data to provide intelligent tutoring when performance and behavior deviate from experts (Goldberg & Amburn, 2015). Later, soldier marksmanship performance data from an instrumented range is captured with acoustic sensors and can be used for immediate feedback and relevant digital resources provisioning. Instructors monitoring the data can quickly see performance from all learners at a glance. Qualification data could be immediately shared with, instructors, leaders, resource managers, and Army digital training management systems (Durlach, Washburn & Regan, 2015). Each of these instances supports the development of a competency, where learner data are transmitted and shared across the various training experiences, and builds upon prior interactions to guide and focus training. Key to this coordination across training applications and experiences is a shared understanding of learner progress toward shared competencies, as described in later sections.

In a similar vein, the Defense Language Institute (DLI) is a Department of Defense educational and research institution with the mission to rapidly instruct culture and language learning. DLI is often described as the world's largest language school, containing 4000 students which learn among 23 different languages. As with marksmanship training, learning a language entails classroom instruction on a wide range of topics including the rules of grammar, culture, vocabulary, etc., as well as practice pronouncing and understanding spoken words and phrases. A DLI near-future learning scenario might include a similar blend of classroom instruction, digital resources, and deliberate practice with virtual and live simulations supported by intelligent and human tutors, much like the marksmanship training scenario described above.

The ability to blend both foreign language and marksmanship training experiences across these different modes of instruction depends on the existence of a shared understanding of learner progress toward shared competencies, and a standardized high-level representation of domains. While some state-of-the-art systems like the Tactical Language and Culture Training System (TLCTS) provide a blend of experiences in a single system, the desired long-term learning is envisioned to be supported across different applications and systems. Both scenarios require that an adaptive training system like GIFT can understand learner progress toward relevant core competencies in these domains. These competencies are represented in the domain model of intelligent tutoring systems (ITSs). For a learner to be trained in a diverse training ecosystem comprised of multiple intelligent agents, competency definition data and learner profile data must be shared between applications to provide coherent training to the learner.

Shared Models of Competency

In order to support a technology progression of long-term learning integration, it is necessary to share a model of domain knowledge that represents the knowledge of learning resources used throughout the process. Sharing this model of domain knowledge requires enabling applications to have a common approach. Two different approaches to accomplish this goal are detailed below.

The first approach to a shared domain model is to directly create and expose a shared set of complete competencies – a competency framework. An example of this direct approach in K-12 mathematics edu-

cation is the Common Core competency framework (Initiative, 2011). Multiple applications can use the Common Core as high-level anchors to report learning progress. This direct approach is easy to understand and use, but is difficult to govern and is limited to high-level anchors whereas many intelligent applications want much more detailed domain models of skills. These high-level anchors, such as "Spanish 1" ensure that the generated models are difficult to directly assess.

A second, alternative approach, is to not share a common domain model, but instead share the problems and learners' answers to those problems between applications. This approach places a burden on applications to be able to infer competencies from these problems, but addresses the concern with the direct approach that shared competency frameworks are too high level for intelligent application needs. When a learner answers a problem in an intelligent application, that problem and answer can be shared with other applications by reporting the problem and answer using a shared protocol such as the Experience Application Programming Interface (xAPI) (Advanced Distributed Learning, 2016b). The intention with this technology solution is to use a "ground up" construction of competency models.

Work in marksmanship training has taken steps toward this second approach focused on the xAPI. Goodwin, Murphy, and Hruska (2015) illustrated how xAPI assessments derived from different periods of instruction for basic rifle marksmanship could be used to inform a persistent learner competency model. Work toward language training appears to focus more on the first direct approach. Cohn (2015) proposed modeling a language learning domain using at least two dimensions: (1) the concrete lexicon and grammar/linguistics associated with the language and (2) the more abstract cultural context which provides a set of meta-rules for applying the lexicon and grammar. These cultural contexts are ideally created in simulations focused on specific tasks and conditions.

Both language learners and marksmen have used virtual worlds as an opportunity to practice skills in task settings. Virtual worlds have resulted in added value on cultural, linguistic, interpersonal, and motivational issues (Canto, Jauregi & van den Bergh, 2013). Virtual worlds such as Second Life and Open Simulator have provided instructors and learners with tools used to create specific virtual environments. However, the complexity of installing and learning to use virtual world software has been a barrier to adoption, despite findings that teaching cultural sensitivity is found to be more effective than vocabulary and key phrases for the rank and file soldier (Abbe, 2009). In response to these installation and use barriers and leveraging modern web capabilities, the Advanced Distributed Learning Initiative has created the Virtual World (VW) Sandbox (Advance Distributed Learning, 2016). The VW Sandbox is an open-source collaborative authoring environment inspired by Second Life and Open Simulator that runs in a web browser with no install. Simulations created with the VW Sandbox can report learner performance using the xAPI. The VW Sandbox is a possible development tool and multiplayer engine for cultural scenarios that GIFT could launch or embed.

Still a third possibility is a combination of the above two approaches. That is, the organization defines the competencies that it believes are necessary for success. For example the Army has identified nine "21st century soldier" competencies. These are high-level competencies such as tactical and technical competence, adaptive and critical thinker, character, and accountability (Command, 2011). The organization would need to both define these competencies and identify outcomes indicative of individuals who exhibit these competencies. For example, the Army might consider certain awards, officer evaluations, or graduation rank to indicate that an individual has a particular competency. On the other hand, as adaptive intelligent training agents generate statements of learner activities and accomplishments, which are stored in an LRS, it becomes relatively easy for machine learning techniques to determine which learner actions, accomplishments, experiences, etc., are the best predictors of outcomes that indicate key competencies. In this way, the organization only needs to worry about defining the competency and associated outcomes, while machine learning algorithms are used to determine how to map course and lesson level experiences, with those competencies.

Competency Model Construction

Manual Ontologies for Manually Constructed Computational Traversal

The construction of all-encompassing ontologies was one of the early dreams of the artificial intelligence (AI) community. As part of this dream, the Cyc project sought to capture much of human knowledge inside of a single ontology. The project, in many ways, succeeded in its endeavor. However, part of the problem of building a large ontology is that that custom reasoning engines had to be constructed to deal with an encyclopedic volume of knowledge (Lenat, 1995).

Competency models are subject to similar problems and difficulties to those of creating large ontologies. One could create, by hand, a large and comprehensive set of all skills. The hand-creation of a competency model is complex and time consuming, but not impossible. The military currently models many job rates, the skills required for these job rates, and the skills required to advance in a given job classification area. The manual creation of competency models is further complicated by the fact that the competencies an organization values and therefore seeks to train are not static. For example, the Army is always revising its list of core competencies as military technologies and threats change. Even if that challenge can be met, such an ontological model can be difficult to implement.

The purpose of creating the ontology is to make use of it. In the competency modeling space, making use of it means the certification of skills, granting of promotions or badges, increased pay, selection for assignments, and other Human Resources (HR) functions. With manual ontology creation, there is the problem of creating customized queries. In addition to creating the ontology, one needs a database of the skills of all individuals in the organization against which queries can be made to identify individuals with desired attributes. For example, an HR representative might need to issue queries against a database to find people suited for individual jobs (e.g., individuals with qualifications on three requisite skills). An alternative to manual ontology creation is crowdsourcing ontology creation and automating the reasoning components (Map, 2016).

Crowd-Sourced Construction and Automated Reasoning (xAPI)

An example of the crowd-sourced creation of knowledge and skill ontologies is represented in the emerging xAPI standard (Advanced Distributed Learning, 2016b). xAPI is an enabling technology used across many different training systems. Each of the systems that make such of the xAPI standard ultimately produces a number of records that are stored for future use. In its current use, the training system and training developer make the assumption, at creation time, that there will be a consumer for statements that they opt to produce. A typical xAPI statement for declaring that someone has mastered a skill may look like the following:

```
{
 “actor”: {
 “mbox”: “mailto:email@domain.com”,
 “name”: “person”,
 “objectType”: “Agent”
 },
 “verb”: {
 “id”: “http://adlnet.gov/expapi/verbs/mastered”,
 “display”: {
 “en-US”: “mastered”
```

```
}
},
“object”: {
“id”: “http://domain.com/expapi/activities/Activity”,
“definition”: {
“name”: {
“en-US”: “Activity”
}
},
“objectType”: “Activity”
}
}
```


The ontology of objects is implicitly created by various system authors labeling activities that can be learned or mastered. An advantage of this semi-automated creation is that a single reasoning system is able to reason broadly across all systems that create this type of data. An HR representative no longer needs to create custom structured query language (SQL) queries against a database, but can instead simply search for the tags of interest across all xAPI-producing systems. These xAPI-producing systems, in turn, can validate against a common manner of representation, such as schema.org, or a common manner of interpretation, such as a credential registry. The emerging standard allows for automated reasoning machinery to access a wide swath of functions. The crowd-sourcing across training systems and simulators makes the automated analysis possible. Technologically speaking, these can be stored and shared via Resource Description Framework (RDF), web service, shared schema from schema.org, or other solutions.

Using this approach does require that different authors of different but related training systems are able to use a common vocabulary with common definitions to express learner actions and accomplishments. For example in the marksmanship realm, the Army does not distinguish between a “hit” and a “kill.” In other words, any bullet hitting the target anywhere is registered as a hit. The Marines do distinguish between hits and kills, with a kill being a hit that is either in the head or center chest and a hit being any other strike on the silhouette. It is easy to see that this could create some confusion if the Marines used a system that generated xAPI statements based on the Army’s standard and vice versa. In the world of language learning, this problem is similar to that of differing accents or dialogues in the same language. Other more subtle distinctions in the way that certain actions might be operationally defined by a system could also lead to confusion, or at worst, erroneous assessments of the competencies of learners. Currently, common vocabularies are established by communities of practice (CoP) that are centered on specific training domains and methods (Advanced Distributed Learning, 2016a).

Future Processes for Both Automated Construction and Reasoning

The manual creation manner of ontologies for training purposes is the manner in which business has been conducted for a number of years. The counterpoint to this trend is the emerging manner of describing the activity and storing it for the use of a future system. Tied into the xAPI standard is a description of the knowledge of skill to which authors are applying descriptive tags. The eventual goal is for systems to be able to seamlessly interoperate and share information, despite having different internal representations or languages.

One manner in which these systems would be able to interoperate would be through the aid of AI technologies. As an example, skills that are similar, or appear similar (learned together, similar people know both skills, etc.), can be represented as belonging to a shared class, which can be automatically identified by the system. As a concrete example, addition and subtraction are related skills which share a common

feature (mathematics). They may be described differently by different content authors (summing, adding, addition, additions, addición, etc.), but the underlying relationship is captured in the combination of the description and the association with the learners.

Furthermore, since these descriptors are presumably related to the content itself (i.e., addition content has the word "add" frequently), machine-understandable descriptors can be created from the content objects (Ray, Robson & Brawner, 2014). These descriptions further aid in the grouping of similar skills and activities, regardless of how they are described. Using automated reasoning systems to traverse the automatically created ontology of knowledge and skills allows for removal of the language barriers of the representations. These language barriers may exist as spoken language barriers (i.e., Spanish and English) or cultural language barriers (i.e., military and civilian).

The path to full automation of training systems that represent the skills of individuals and how they are related to the skill groupings of the population starts with the creation of ontologies. These ontologies can then be shared and traversed across training systems. Systems which have knowledge about the training content and population models can then traverse this information to build common representations. These representations can be used to identify critical skills, organizational strengths and weakness, valuable learning goals, and opportunities for learning – in other words, competencies.

Adaptive Training and Psychomotor Expansion

While it is proposed that GIFT support long-term technology integration through shared competency models, the primary use of a domain model for GIFT is to provide tutoring instruction directly to a learner within a specified learning interaction. Many different language tutors can be envisioned with GIFT. A language tutor that works with a VW Sandbox cultural simulation is one. Another kind could be created based on the use of sensors to collect objective performance data. In the case of marksmanship, rifle sensors are used (Chung, Nagashima, Espinosa, Berka & Baker, 2009). In the case of language learning, mobile phones could be used for speech recognition where improvements continue to emerge for computer-assisted language learning (Tong, Lim, Chen, Ma & Li, 2014).

Goldberg and Amburn (2015) are investigating how to apply GIFT to consume sensor data from a simulated firing range for basic rifle marksmanship training to trigger objective-based guidance and remediation. Sensors used to capture marksmanship performance are sent, filtered, and received through GIFT's gateway, sensor, and domain modules (p. 118). The tutoring behavior is based on comparing current behavior to expert behavior and then providing remedial content (e.g., video snippets of experts explaining and demonstrating) when deviations are found. This work can inform language tutors. Tutoring around the marksmanship fundamental of trigger squeeze using a trigger sensor may pave the way for tutoring around a fundamental of speaking using a speaking sensor. As the marksmanship models in GIFT are extended to support other marksmanship conditions (e.g., hitting moving targets, using different weapons), these task conditions could be considered models for cultural language task conditions.

Commonized Assessment Frameworks for Similar Skill Groupings

Sottilare et al. (2016) have divided domains of instruction up into several representative sections, including the cognitive, affective, and psychomotor. The cognitive portion of domain modeling involves a representation of student knowledge, problem-solving skills, and mental knowledge required to perform a task or master a skill. The affective portion of the domain modeling includes a mastery of emotions, feelings, and intuitions about a task. The psychomotor aspects of a task include the learning of physical movements of the task.

The cognitive, affective, and psychomotor portions of domain modeling are shared across many domains. Using the first example illustrated throughout this chapter, the marksmanship domain contains knowledge of marksmanship basics (aiming, breath control, etc.) as a cognitive component, emotional control during tense situations as an affective component, and the physical act of firing as a psychomotor component. Using the second example, language learning contains cognitive components such as vocabulary, conjugation, affective components such as negotiation and culture, and the psychomotor components of pronunciation.

Many languages share common components in the manner that domains share common components. Techniques for rapid learning across multiple languages, reported from polyglots, include items such as the 100 most frequently used words or memorizing grammatical structure through the use of 13 key sentences. Such representations are common and agnostic to the individual language of implementation. These can serve as the cognitive components of language learning.

As an example of these tendencies in practice, sentences such as "the apple is red", "I give John the apple", and "we give him the apple" can be used across multiple languages to show how descriptors are applied to objects, how possession is handled, how third person pronouns are treated, the use of indirect objects, and other items. Each of the sentences in this area can be used to generalize a host of other meanings. Vocabulary memorization for the most frequently used words (e.g., the, be, to, of, and etc. in English) can reveal the verb conjugations and nouns required to express basic thoughts. These two things, coupled with pattern-of-life situational settings, such as eating or negotiating, can provide the basis of a significant set of language learning.

On the affective side of language learning, the culture that speaks the language plays an important component. Culture learning can play itself out through day-to-day experiences such as going to the store, eating meals, conversing about past events and news, etc. Each of these experiences contributes a vocabulary domain in addition to having the student manage the differences between individual cultures. It has been shown that this cultural language learning component may be as effective as vocabulary learning for overall communication (Adair-Hauck et al., 2000).

The psychomotor components to language learning currently require human assessment and intervention. However, each of the cognitive/affective/psychomotor components can be taught simultaneously in specifically designed scenarios representative for each culture. As an example, a restaurant scene that attempts to use a majority of the common language phrases across the common grammatical categories while assessing pronunciation should be able to teach all three components of the domain seamlessly and simultaneously, while having the emphasis on pragmatic functionality known to contribute motivational factors (Martínez-Flor & Usó-Juan, 2006).

Conclusions

This chapter has painted a picture of a larger scale of learning. Learning takes place in areas that are both physically separate, in the example of the separation of classroom and "in the wild", and philosophically separate, in the example of cognitive psychomotor learning tasks. Using the examples of marksmanship and language learning, it is shown that the total training experience is simply larger than is encompassed in a single instance GIFT domain module. It seems clear that scalability requires that multiple instances be tied together through a common architecture.

While a larger representation than a single domain module is needed, scalability can be accomplished through common representations. These common representations, at the high level, break into cognitive, affective, and psychomotor components. Each of these components can then break into common domain

representations, such as psychomotor pronunciation across different languages or the cognitive aiming knowledge for different marksmanship tasks. These common representations are the key to scaling across many training tasks which are presented similarly.

The need for common representations of domain task breakdown requires technical solutions to the problems of representation, sharing, communication, storage, and other fundamental process building blocks. Components of these technical solutions, such as competency modeling from xAPI data, should be represented in GIFT in order to eventually automatically build and share models of learning across systems, instructional tasks, and real-world performance.

Acknowledgments

Acknowledgement is due to Benjamin Goldberg, who instigated many of the initial conversations guiding this work during the GIFT 2015 Expert Workshop.

References

Abbe, A. (2009). Transfer and generalizability of foreign language learning: DTIC Document.

Adair-Hauck, B., Willingham-McLain, L. & Youngs, B. E. (2000). Evaluating the integration of technology and second language learning. *CALICO journal*, 269–306.

Canto, S., Jauregi, K. & van den Bergh, H. (2013). Integrating cross-cultural interaction through video-communication and virtual worlds in foreign language teaching programs: is there an added value? *ReCALL*, *25*(01), 105–121.

Chung, G. K., Nagashima, S. O., Delacruz, G. C., Lee, J. J., Wainess, R. & Baker, E. L. (2011). Review of rifle marksmanship training research.

Chung, G. K., Nagashima, S. O., Espinosa, P. D., Berka, C. & Baker, E. L. (2009). An Exploratory Investigation of the Effect of Individualized Computer-Based Instruction on Rifle Marksmanship Performance and Skill. CRESST Report 754. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Command, D. (2011). The United States Army Learning Concept for 2015. *Fort Monroe, VA*.

Durlach, P., Washburn, N. & Regan, D. (2015). *Putting Live Firing Range Data to Work Using the xAPI*. Paper presented at the Interservice and Industry Training and Simulation and Education Conference, Orlando, FL.

Goldberg, B. & Amburn, C. (2015). *The Application of GIFT in a Psychomotor Domain of Instruction: A Marksmanship Use Case.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3).

Goodwin, G. A., Murphy, J. S., Hruska, M. & Consulting, Q. I. (2015). *Developing Persistent, Interoperable Learner Models in GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3).

Green, A. & Youngs, B. E. (2001). Using the web in elementary French and German courses: Quantitative and qualitative study results. *CALICO journal*, 89–123.

Initiative, C. C. S. S. (2011). *Common core state standards for mathematics*.

Learning, A. D. (2016). Virtual World Sandbox, 2016.

Learning, A. D. (2016a). xAPI Community of Practice (COP) .

Learning, A. D. (2016b). The xAPI Overview, 2016.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33–38.

Map, S. P. T. U. S. (2016). Skill Project.

Martínez-Flor, A. & Usó-Juan, E. (2006). A comprehensive pedagogical framework to develop pragmatics in the foreign language classroom: The 6R Approach. *Applied Language Learning*, *16*(2), 39.

Ray, F., Robson, R. & Brawner, K. (2014). *3QL: A query language for exchanging third tier metadata.* Paper presented at the Simulation Interoperability Standards Organization (SISO) Fall 2014 Workshop, Orlando, FL.

Sottilare, R. A. (2016). Domain Modeling Book Introduction. *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling* (Vol. 4): Eds. Graesser, Sottilare, Hu, Olney, Nye and Sinatra. US Army Research Laboratory.

Spain, R., Mulvaney, R. H., Cummings, P., Barnieu, J., Hyland, J., Lodato, M. & Zoellick, C. (2013). Enhancing Soldier-centered learning with emerging training technologies and integrated assessments: DTIC Document.

Tong, R., Lim, B. P., Chen, N. F., Ma, B. & Li, H. (2014). *Subspace gaussian mixture model for computer-assisted language learning.* Paper presented at the Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.

Zhao, Y. (2003). Recent developments in technology and language learning: A literature review and meta-analysis. *CALICO journal*, 7–27.

CHAPTER 9 – A Review of Self-Reference and Context Personalization in Different Computer-Based Educational Domains

Anne M. Sinatra
US Army Research Laboratory

Introduction

A goal of instruction is for a student to not only learn the material and be able to recall it immediately afterwards, but also to be able to retain it for future use. Additionally, it is preferable for the material to be taught in such a way that the student can use what was learned and then apply or transfer it to new situations. Human memory and learning have been extensively studied, resulting in mnemonics and general strategies that assist in memory across different domains of instruction. One such strategy is linking information to knowledge that the student already has and that is important to them. This strategy can include linking information to the self, to people that the individual knows, or simply customizing the materials to include mention of topics that are of interest to the individual or group of students. There appears to be significant benefits to employing all of these approaches; however, they are often lumped together in the literature, and the different domains in which they may or may not be beneficial have not been acknowledged.

Many different domains, topics, and areas of instruction include materials, examples, and practice problems with wording that can be adjusted to make it specific to different students or interest areas. In many classes, whether they are lecture-based, web-based, large or small, it is often difficult to retain student interest and motivation throughout. However, there are different approaches that can be used to draw student attention and personalize the materials that they receive in order to keep them more interested in the topic. Referencing the self and context personalization (changing the wording of student materials to be contextually consistent with student interests) has been shown to have positive impacts on student outcomes. Self-reference and context/interest personalization techniques offer benefits to memory and performance without requiring large adjustments to learning materials. This chapter reviews research and approaches to self-reference, context personalization, interest personalization, and the impacts that they have on learning and retention in different domains. It has been assumed that these strategies will work between learning domains; however, research has focused primarily on mathematics. The domains of instruction are examined and recommendations are made for using this strategy in intelligent tutoring systems (ITSs).

Related Research

While the positive impact of personalizing to self, context, and interest has been found consistently in the literature (Anand & Ross, 1987; Cordova & Lepper, 1996; Moreno & Mayer, 2000; Symons & Johnson, 1997), these manipulations have often been defined in different ways and applied differently between domains. Additionally, there are a number of theories on why this manipulation assists in learning. It has been suggested that interest or self-personalization increases student motivation by linking instruction to a topic that is already of intrinsic interest to the student and by providing a feeling that their self-selected interest preference was honored by the system (Cordova & Lepper, 1996; Ritter, Sinatra, Fancsali, 2014; Walkington, Sherman & Howell, 2014). Other suggestions include that providing a context results in grounding the concepts to be learned in a familiar example, which makes it easier for students to conceptualize and understand the material (Walkington, Sherman & Howell, 2014). Finally, it has been suggest-

ed that by linking information to interests or the self, it is using already existing schemata that the individual has for themselves others, and topic areas (Anand & Ross, 1987; Sinatra, Sims & Sottilare, 2014).

Personalization strategies have been used in computer-based training (Anand & Ross, 1987; Cordova & Lepper, 1996; Walkington, 2013), classroom/laboratory-based instruction and tests (Ku & Sullivan, 2002; Ross, 1983; Ross, McCormick & Krisak, 1986), and in-person student interview sessions (Walkington, Sherman & Petrosino, 2012). The approaches to personalization have varied between including the individual student name (Cordova & Lepper, 1996; Sinatra, Sims & Sottilare, 2014), using names of individuals that the student knows (Anand & Ross, 1987; Cordova & Lepper, 1996; Sinatra, Sims & Sottilare, 2014), using first-person phrasing or "you" in the text (Mayer, Fennell, Farmer & Campbell, 2004; Moreno & Mayer, 2000; Turk, et al., 2015), including topics of interest to the specific student (Anand & Ross, 1987; Ritter, Sinatra & Fancsali, 2014; Walkington, 2013), including topics that are familiar to the student (Sinatra, Sims & Sottilare, 2014), including topics that are of general interest to the class (Ku & Sullivan, 2002), and including examples that are consistent with the student's major (Ross, 1983; Ross, McCormick & Krisak, 1986).

The Impact of the Self-Reference Effect on Memory and Retention

One of the primary approaches to personalizing and linking information to be learned to the individual is through the Self-Reference Effect (SRE). The SRE has been examined at length in the cognitive psychology literature. According to the SRE, if information is tied to the self, or even someone that the person knows, it can assist in recall of information (Symons & Johnson, 1997). A traditional explanation for the SRE, which is consistent with other personalization literature, is that by referencing the self it is drawing on schemata that individuals have for themselves to assist in improving memory. While the majority of the research has found these positive effects, it has mostly been done using abstract approaches rather than in applied educational settings. Many of the studies involve presenting individuals with a series of words; when each word is presented the participant is asked a question such as is this word true of you or more general questions about the meaning of the word or if it fits in a sentence might be asked. It has been found that participants were able to recall the words that they were asked to link to themselves more than the other words (Rogers, Kuiper & Kirker, 1977; Klein & Kihlstrom, 1986). This suggests that thinking of the word in relationship to themselves improved their memory for it.

More applied educational approaches to the SRE have been explored in computer-based training. Cordova and Lepper (1996) included the names of the students as well as their interests in a computer-based game that taught mathematical order of operations concepts to 4^{th} and 5^{th} graders. It was found that those in the personalized conditions recalled and transferred the learned material significantly better than those who were in non-personalized conditions. With college-age students, the SRE was examined by including the participant's name and the name of his or her friends within the learning materials that provided instruction on successfully solving logic grid puzzles (Sinatra, Sims & Sottilare, 2014). In this study, the manipulations occurred during the learning phase, and participants were later assessed based on performance on similar and more difficult puzzles. This study included elements of both the SRE and context personalization, as names of popular culture characters were included in one condition and general names in another. Interestingly, while individuals with high need for cognition demonstrated transfer performance in the expected pattern, those with low need for cognition appeared to have significantly lower performance when provided with self-reference materials, and higher with popular culture materials. This example suggests that the effects of personalization may vary based on individual difference characteristics or in different domains.

Moreno and Mayer (2000) examined the SRE in the domain of science education. They examined student learning in an environmental science educational game and in explanations of lightning formation by pre-

senting instruction to students phrased in either the first-person (self-reference) or the third-person. They concluded that by using "I" or "you", it was asking the student to imagine themselves in relationship to the material, which then lead to increased understanding of the material and transfer. They further examined what they termed the "personalization effect" by including "you" as opposed to "the" in different versions of narration in a computer-based explanation of the respiratory system. They found that while there was no difference in retention between the two conditions, transfer performance was significantly better for those who received the personalized "you" version of the materials (Mayer, Fennell, Farmer & Campbell, 2004). Similarly, d'Ailly, Simpson, and MacKinnon (1997) employed the SRE through use of the word "you" in math problems and found it had a positive impact. Turk, et al. (2015) found that using the SRE by asking children to include "I" and themselves in sentences during learning was shown to lead to positive impacts of literacy such as vocabulary retention and spelling. Of particular note in Mayer's studies is the use of "you" instead of names, which assists in generalizing and reduces the need to make multiple edits to student materials. Additionally, the use of "you" or "I" was selected to minimize any interference or distraction to the student from the seductive details effect. The seductive details effect can result in students getting distracted by irrelevant information and not retaining material as expected (Harp & Mayer, 1998). Despite this, other studies have found positive effects of personalizing the context of example and learning materials with specific names and topics (Anand & Ross, 1987; Cordova & Lepper, 1996). These studies offer examples of the SRE being examined in varying domains: science, logic, and mathematics Therefore, the inclusion of names and asking students to think in terms of themselves appears to provide improvement over non-personalized conditions in regard to performance, transfer performance, and attitudes toward the tutoring.

The Impact of Context Personalization on Memory and Retention

While some studies have focused on linking information to the specific individual through names or the use of "you", others have examined personalization through linking to information that is of interest to the individual student by manipulating the context of the materials that are provided. In fact, there has been overlap in studies that have examined both elements of the SRE and context personalization through manipulations that reference the self, as well as that adapt to include specific interests or preferences of the student (Cordova & Lepper, 1996; Sinatra, Sims & Sottilare, 2014). Cordova and Lepper (1996) examined the impact of personalizing material to include names and interests of children, and adding an adventure narrative to a game designed to teach order of operations in math. In their study, elementary school students were either in a standard math game condition, a space game condition, or a treasure hunt condition. Additionally, there were personalization manipulations that resulted in the inclusion of the child's own name, names of their friends, their favorite toys, their favorite foods, etc., within the narrative of the story. It was found that students performed better, rated their experience as better, and retained more information from the conditions that were both narrative and personalized to the individual's interests/name. This suggests that by linking information to something that is intrinsically interesting to the learner, and including them in the narrative, it can result in better learning outcomes (Cordova & Lepper, 1996). This study is of particular interest as it overlaps conceptually with both the SRE (linking to the schemata for the self), as well as context personalization (including an interesting narrative, and personalized content that is of interest to the individual). While Sinatra, Sims and Sottilare (2014) included a self-reference condition, it also included a popular culture condition, which included names of *Harry Potter* series characters within the learning materials for instruction. In this case, it was expected that even if an individual was not a fan of *Harry Potter* their familiarity with the names would lead to schema activation and a context for participants to link the information to. Other studies have exclusively examined interests such as manipulating the instruction to include examples that were either consistent or inconsistent with the student's majors when teaching probability (Ross, 1983; Ross, McCormick & Krisak, 1986). Ross (1983) found that when preservice teachers received personalized examples (consistent) they performed better on posttest assessments than when they received medical examples (inconsistent). Additionally,

nurses also performed better when they received medical examples than teaching examples. This demonstrates that there appears to be a consistent benefit to matching the context of examples to a student's own background and major, which is important and relevant to them. A variety of additional approaches have been taken to examining personalization of materials to something that is familiar to the learner, including examining comprehension-based tasks, group discussions, and the influence of familiar or unfamiliar dialects.

Research has been conducted that tried to discern if the SRE and personalization effects were specifically associated with schemas, or alternate explanations. Guida, Tardieu, Le Bohec, and Nicolas (2013) provided personalized and non-personalized versions of lists and texts to be remembered by college-age participants. The information to be remembered was associated with locations that were familiar to students and unfamiliar, depending on the condition. Positive effects of personalization were only found for amount of time to recognition of words or objects, and only when the text based materials were used, not with list-based materials. Based on these results, they believe that long-term working memory is contributing to the effect, as opposed to a simple schema resulting in the performance improvements. They also propose that the positive impacts of personalization may increase as the complexity of the information to be processed increases. This study is important because it examines a new domain – comprehension-based tasks – and tries to determine the mechanisms behind the personalization effect.

The use of real-world contexts in mathematical instruction for bilingual high school students was examined in a case study (Zahner, 2012). Students were provided with algebra problems that were put in "real life" and story-based contexts. They were then prompted to engage in group discussions while solving the problems. It was found that the use of these contexts influenced the discussions that they had during the solving process. In some cases, unfamiliar words hindered their progress since it led to confusion and drew attention away from solving the problem itself, in other situations the context allowed students to better visualize the problem and assisted in reaching the solution. Interestingly, it was found that by using real-world contexts it also invited answers that were not necessarily mathematically sound, but more practical in the real world. Therefore, while there do appear to be benefits to grounding problems in real-world contexts, there are situations in which this could unexpectedly influence the answers provided, similar to the seductive details effect (Harp & Mayer, 1998), and lead to difficulties if terminology is not familiar to the students.

The influence of familiar dialects and personalization was investigated in an Austrian study of 11 to 14 year old children (Rey & Steib, 2013). Participants watched an instructional animated video about computer networks, which was either personalized or formal, and either included a familiar regional (Austrian) dialect that was shared by the participants or a standard (German) dialect that was not. Personalization was defined using the methodology used by Mayer and colleagues such that "your" and "you" were included instead of words like "the" (Moreno & Mayer, 2000). The use of the familiar or unfamiliar accent leads to an additional layer of personalization over previous studies. It was found that the personalized narrated animation led to significantly higher scores on both retention and transfer. Dialect had an effect only on retention performance, but not transfer. This study demonstrates the generalizability of personalization research to a new learning context (computer networks) and in a new direction (characteristics of instruction) that can have important implications for the use of personalization.

Adaptive computer based technology has been used to examine providing interest-specific examples instead of general materials (Anand & Ross, 1987; Bernacki & Walkington, 2014; Walkington, 2013). The ability for students to enter their interest preferences and information into the computer for later storage and retrieval assists in the personalization process. Names and information that are entered by the student can be saved as variables, and then automatically populated by the software.

Anand and Ross (1987) examined the impact of biographical personalization on computer-based instruction on how to solve fractions. Participants were fifth and sixth graders, and they completed a survey prior to beginning their lessons that asked information such as their favorite magazine, names of friends, favorite candy, hobbies, favorite television shows, etc. The information that was provided by the student via the survey was then incorporated into the learning materials by changing the themes and the names within the problems. Participants that received personalized materials performed better in solving general and transfer problems, and in their attitudes toward solving fractions problems.

Walkington and colleagues have conducted research examining the impact of context/interest personalization within the Cognitive Tutor ITS, which teaches Algebra (Bernacki & Walkington, 2014; Walkington, 2013). In their research, participants completed a personalization survey in which they ranked their interest areas in topics such as food, computers, TV, sports, and movies. In the personalization condition, the students received units of material that included word problems consistent with the interest survey. In the non-personalized condition general story problems were included in the same units. Students that received personalization rated their interest in their situational interest and interest in algebra as higher than those who did not receive personalization. Additionally, personalization resulted in a positive impact on the student's learning of algebra concepts (Bernacki & Walkington, 2014; Walkington, 2013).

Mathematics and closely related topics (i.e., probability) make up the majority of research on context specific personalization. Part of the reason for this is likely that it is traditional to include mathematical word problems in instruction and assessments. Since they are story and text based, these problems are easy to manipulate to include specific contexts and examples that might be familiar for students. Domain areas such as science or vocabulary may not have as many opportunities or as much available text in order to adjust the context of the materials to student interest. While initial research has begun to examine the importance of reading comprehension and personalization (Guida, Tardieu, Le Bohec & Nicolas, 2013), it is a domain that would be ideal to continue research in. Future research could be conducted on the impact of selected interest on training for reading comprehension, as it would be easy to manipulate passages to be consistent with selected interests of students.

Discussion

Domains That Have Been Investigated and Types of Personalization

There appears to be consistent positive effects from personalizing materials to encourage students to think of themselves while learning the material or relate the information to topics that are of interest to them. The SRE has been examined in varying domains such as science, mathematics, logic, and vocabulary. However, a closely tied concept, context, or interest personalization has been primarily examined in mathematics instruction. It would be advantageous for more research to be conducted into novel domains to see if consistent results are found and if it would be beneficial to include these strategies in tutors and instructional material in domains other than mathematics. It may additionally be of interest to examine the individual differences of students and how it may impact the influence of context personalization, as differences have been found in the effectiveness of these strategies between individuals with high and low need for cognition (Sinatra, Sims & Sottilare, 2014). Further, there has been variability of the age groups that research has been done with. Some studies have focused on the impact for elementary school age children (Anand & Ross, 1987; Cordova & Lepper, 1996), whereas other has examined college-level students (Ross, 1983; Sinatra, Sims & Sottilare, 2014). There may be differences in the impact of the manipulation when the difficulty level of the material is lower, as opposed to more advanced and conceptual.

There is a large amount of overlap between the context/interest personalization and self-reference literature, and additionally they have been a number of different approaches that have been taken to personali-

zation. Table 1 provides a visual representation of domains that have been examined in regard to personalization, and the variety of approaches to "personalization" that have been used in these cases. Some studies engage in both self-reference through inclusion of names and context personalization by including interests. Other studies engage in self-reference, or what is sometimes termed "personalization", by including "your" instead of "the" in the text to encourage individuals to think of themselves in relationship to it. For instance, when teaching about the respiratory system, Moreno, et al. (2004) included text such as "During inhaling, your diaphragm moves down…" for personalization, instead of the more general and non-personalized, "During inhaling, the diaphragm moves down…" (p.391). While all of these approaches appear to have positive impacts on motivation, learning, and particularly transfer, it is important to note that there are many different ways that "personalization" has been defined in the literature. It would be helpful for specific terms to be created for each type of common manipulation so that there is consistency in their discussion. It would also be of use for future studies to begin examining if the type of content that is manipulated (learning phase, as opposed to assessment phase) has an impact on performance. Also of note is that the majority of work has focused on the domain of mathematics. It is important for future work to examine self-reference and context/interest personalization in multiple domains to determine its generalizability. There may be specific domains that more readily lend themselves to these types of personalization or others that do not lead to as positive results.

Table 1. Overview of domains that have been studied and the approaches taken to personalization in those specific domains.

Domain	Studies	Personalization Methods Used
Computer networks	Rey & Steib, 2013	Using "you" instead of "the"
Mathematics (algebra; fractions; order of operations; probability; word problems)	Anand & Ross (1987); Bernacki & Walkington, 2014; Cordova & Lepper; d'Ailly, Simpson & MacKinnon, 1997; Davis-Dorsey, Ross & Morrison, 1991; Ku & Sullivan, 2002; Ross, 1983; Ross, S. M., McCormick, D. & Krisak, 1986; Walkington, 2013; Walkington, Sherman & Petrosino, 2012; Zahner, 2012	Inserting names into problems; inserting self-entered preferences into materials; inserting general interests into materials; using examples that are consistent with individual's major, using real life and story examples
Logic	Sinatra, Sims & Sottilare, 2014	Inserting own name and names of known others into materials; inserting familiar names into materials
Reading comprehension and vocabulary	Guida, Tardieu, Le Bohec & Nicolas (2013); Turk, et al., 2015	Including "me" in sentences while learning vocabulary; including lists and texts that are in familiar and unfamiliar locations
Science	Moreno & Mayer, 2000; Mayer, Fennell, Farmer & Campbell, 2004	Including "I" or "you" instead of "the"

Approaches to Personalizing Materials in ITS

A number of different studies have begun their personalization approach by giving the students a questionnaire in order to gather information (Anand & Ross, 1987; Ku & Sullivan, 2002; Bernacki & Walkington, 2014). In these studies, students may answer questions about family member names, friend names, favorite foods, favorite movies, etc. This information is then used by the instructor or computer-based system in order to customize learning materials and assessment questions to be consistent with the individual student answers. ITSs are advantageous because they can respond to individual student per-

formance and adapt based on whether or not the answers provided by the student are correct. However, they could also adapt by providing different questions based on student interests. There are a number of different ways to accomplish this. Among them would be for an ITS author to create general template questions that can be populated with words and context selections that have been indicated by the student's preferences. An additional approach would be for subsets of examples/instructions that have different contexts to be presented and for question banks to exist that provide questions that are specific to the preferred context for the student's interests. For example, the same question may be populated with contextual information about movies, music, or sports depending on the version of the question bank. The student's preference selection would then determine which question bank that questions were received from. A general approach that may be of benefit in an ITS is the authoring of questions with "you" or "I" included in them when appropriate. This would allow for initializing self-reference without needing to make any substitutions within the text of the questions themselves. One approach may be to use questions and materials that are personalized from the onset of tutoring. Another approach would be to use personalization as an adaptation for remediation in certain domains, and when certain individual differences are present as opposed to others. The Generalized Intelligent Framework for Tutoring (GIFT) is a domain-independent intelligent tutoring framework. GIFT provides authors with the ability to use their own learning content and create their own questions. GIFT offers a great amount of flexibility and the ability to approaches to adaptation. Based on the current functionality in GIFT, in order to use self-reference, at the best approach would be to include "you" or "I" in the materials. Self-reference experimentation has previously been performed in GIFT through adaptations that occurred in PowerPoint based materials (Sinatra, Sims & Sottilare, 2014). However, it would be advantageous for future versions of GIFT to support including self-entered words in questions or send participants down different paths or to different question banks based on selected areas of interest (e.g., movies, sports, music, books, etc.).

Recommendations and Future Research

Research has suggested that there are positive effects of personalizing learning materials, whether it is to the individual student including their name and first-person text or through providing material that is consistent with their interests. Research has recently been continuing into the examination of mechanisms that result in these effects (Guida, Tardieu, Le Bohec & Nicolas, 2013) and into the individual differences that may impact their effectiveness (Zahner, 2012; Sinatra, Sims & Sottilare, 2014). As has been determined in this chapter, while there seem to be consistent positive impacts from using personalized materials, the domains that have been selected for research are primarily mathematics. Additionally, there has been large variation in the ways that personalization has been defined. Future research should focus on determining if these benefits are domain-independent and what personalization methods lead to the best outcomes in different domains.

Adaptive tutoring systems have qualities that make them perfect candidates for integration of interest personalization and self-reference. As GIFT is a domain-independent framework, it would be advantageous to include flexibility within course authoring such that the topics or contexts of questions could be personalized to individuals. This could be done through creating additional authoring tools for personalized question banks or implementing the use of survey question answers as stored variables for incorporation into questions. An additional approach would be to allow a selection of interest preference to send the student to one version of the materials instead of another. In its current and future forms, GIFT will be an advantageous framework to use to continue examining the impact of self-reference and personalization on student performance in different domains.

References

Anand, P. G. & Ross, S. M. (1987). Using computer-assisted instruction to personalize arithmetic materials for elementary school children. *Journal of educational psychology*, *79*(1), 72.

Bernacki, M. & Walkington, C. (2014). The Impact of a Personalization Intervention for Mathematics on Learning and Non-Cognitive Factors. In *Submitted to the 2014 International Conference of Educational Data Mining, London*.

Cordova, D. I. & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of educational psychology*, *88*(4), 715.

d'Ailly, H. H., Simpson, J. & MacKinnon, G. E. (1997). Where should" you" go in a math compare problem?. *Journal of Educational Psychology*, *89*(3), 562–567.

Davis-Dorsey, J., Ross, S.M. & Morrison, G.R. (1991). The role of rewording and context personalization in solving of mathematical word problems. *Journal of Educational Psychology, 83* (1), 61–68.

Guida, A., Tardieu, H., Le Bohec, O. & Nicolas, S. (2013). Are schemas sufficient to interpret the personalization effect? Only if long-term working memory backs up. *Revue Europeenne Psychiologie Appliquee, 63*, 99–107.

Harp, S. F. & Mayer, R. E. (1998). How seductive details do their damage: A theory of cognitive interest in science learning. *Journal of educational psychology*, *90*(3), 414.

Klein, S. B. & Kihlstrom, J. F. (1986). Elaboration, organization, and the self-reference effect in memory. *Journal of Experimental Psychology: General*, *115*(1), 26.

Ku, H. Y. & Sullivan, H. J. (2002). Student performance and attitudes using personalized mathematics instruction. *Educational Technology Research and Development*, *50*(1), 21–34.

Mayer, R. E., Fennell, S., Farmer, L. & Campbell, J. (2004). A Personalization Effect in Multimedia Learning: Students Learn Better When Words Are in Conversational Style Rather Than Formal Style. *Journal of Educational Psychology*, *96*(2), 389.

Moreno, R. & Mayer, R. E. (2000). Engaging students in active learning: The case for personalized multimedia messages. *Journal of Educational Psychology*, *92*(4), 724.

Rey, G.D. & Steib, N. (2013). The personalization effect in multimedia learning: The influence of dialect. *Computers in Human Behavior, 29*, 2022–2028.

Ritter, S., Sinatra, A. M. & Fancsali, S. E. (2014). Personalized Content in Intelligent Tutoring Systems. *Design Recommendations for Intelligent Tutoring Systems*, 71.

Rogers, T. B., Kuiper, N. A. & Kirker, W. S. (1977). Self-reference and the encoding of personal information. *Journal of personality and social psychology*, *35*(9), 677.

Ross, S. M. (1983). Increasing the meaningfulness of quantitative material by adapting context to student background. *Journal of Educational Psychology*, *75*(4), 519.

Ross, S. M., McCormick, D. & Krisak, N. (1986). Adapting the thematic context of mathematical problems to student interests: Individualized versus group-based strategies. *Journal of Educational Research*, *79*(4), 245–252.

Sinatra, A. M., Sims, V. K. & Sottilare, R. A. (2014). *The Impact of Need for Cognition and Self-Reference on Tutoring a Deductive Reasoning Skill* (No. ARL-TR-6961). US Army Research Laboratory, Aberdeen Proving Ground MD.

Symons, C. S. & Johnson, B. T. (1997). The self-reference effect in memory: a meta-analysis. *Psychological bulletin*, *121*(3), 371.

Turk, D.J., Gillespie-Smith, K., Krigolson, O.E., Havard, C., Conway, M.A. & Cunningham, S.J. (2015). Selfish learning: The impact of self-referencial encoding on children's literacy attainment. *Learning and Instruction, 40*, 54–60.

Walkington, C. A. (2013). Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, *105*(4), 932.

Walkington, C., Sherman, M. & Howell, E. (2014). Tying algebra to students' interests—sports, music, and video games—helps students retain these concepts. *Mathematics Teacher*, *108*(4).

Walkington, C., Sherman, M. & Petrosino, A. (2012). "Playing the game" of story problems: Coordinating situation-based reasoning with algebraic representation. *The Journal of Mathematical Behavior*, *31*(2), 174–195.

Zahner, W. (2012). "Nobody can sit there": Two perspectives on how mathematics problems in context mediate group problem solving discussions. *Journal of Research in Mathematics Education*, *1*(2), 105–135.

CHAPTER 10 – Discovering Domain Models in Learning Curve Data

Ilya Goldin [1], Philip I. Pavlik, Jr. [2], and Steven Ritter [3]
[1] 2U, Inc., [2] University of Memphis, [3] Carnegie Learning, Inc.

A primary purpose of domain models is to define the scope of what students should learn and, in particular, determine for a given student which of the knowledge components (KCs) in a domain the student has and has not learned. Additionally, domain models can be viewed as predictions about knowledge transfer. If two activities or items are associated with the same KC in a domain model, it implies that successful performance in one activity will be associated with successful performance in the other. Recent work has taken this insight as a way to improve domain models based on data collected from students. Failures of such predictions may be taken as an indicator that the domain model needs to be revised; what was once considered to be a single element of knowledge may, in fact, be multiple separate elements. This chapter discusses the theory of domain modeling at the KC level and review current practice related to using data to test and refine such domain models.

Introduction

Historically, instructional design has aimed to define the scope of the subject matter to be taught to students, and the sequence of the topics and activities that make up instruction. The introduction of digital technologies into the learning process enables data collection on every attempt that a student makes to engage with a learning activity. In addition to facilitating ongoing assessment (DiCerbo & Behrens, 2012), the resulting data can be used to improve instruction itself.

In tutoring systems and adaptive learning technologies, the domain model describes the granularity and nature of the knowledge components (KCs) (Koedinger, Corbett & Perfetti, 2012). The domain model may also imply a sequence of learning objectives by describing dependencies in the domain. The domain model sits in relation to the learner model, which describes how the domain components are learned. The pedagogical model describes how the learner model is used for decision making about instruction for the domain contents. The pedagogical model imposes a sequence based on the domain dependencies and the current state of the learner model (Pavlik Jr., Brawner, Olney & Mitrovic, 2013). The Generalized Intelligent Framework for Tutoring (GIFT) community has discussed the topics of learner modeling, instructional management and authoring tools in this book series (Sottilare, Graesser, Hu & Brawner, 2015; Sottilare, Graesser, Hu & Goldberg, 2014; Sottilare, Graesser, Hu & Holden, 2013).

Even domain models designed by experts can be wrong. This may be because KCs and their relations are hypothesized and not directly observable, and because experts may forget the difficulties that novice learners face (Nathan, Koedinger & Alibali, 2001). A poor domain model means that instruction is inefficient and assessment is inaccurate. In designing instruction, we are obligated to consider that the learning experiences we design may be suboptimal and that we need to measure their efficacy.

The data we collect allow just such a measurement, which is based on two key ideas. One key idea is the learning curve, that is, as a population of students learns to perform some skill, the population error rate in the application of that skill will drop. This is remarkably distinct from data collected in assessment-only

settings, which assume that no learning takes place and students are unlikely to be assessed on a skill multiple times. The second key idea is that we want students to acquire knowledge that can be applied in multiple contexts. We want them not merely to memorize that 2+2 equals 4, but how to apply the plus operator to add arbitrary numbers. This is called transfer.

When we collect data on the performance of multiple students on a set of problems, we can measure which of several candidate domain models is a better description of actual student performance. This is because the data ought to follow from the key ideas of learning curves and transfer.

In this chapter, we first explain in detail what a domain model can look like, and give historical and psychological background. We then describe the issues pertaining to data-driven automation in domain modeling and give some recommendations on how to apply learning curve analysis in practice.

A Domain Model and its Discontents

One part of a domain model is a mapping between the activities in which the students engage and the skills required to perform these activities. For example, in the tradition of the Pump Algebra Tutor (PAT), the input elements in a user interface for solving a problem represent specific steps on the solution path (Koedinger, 2001; Koedinger & Anderson, 1997). Each step may require one or more skills, and it is sometimes difficult to determine what a skill is if "skills" are used in multiple contexts. From a theoretical perspective, tasks may be fully decomposable into independent KCs (Anderson, 2002). However, it is typically impractical in an educational context to fully represent KCs at this level. This illustrates a complexity. While a simple domain (one without much hierarchy or complex structure) may support the assumption that problems or items are somewhat independent (e.g., foreign vocabulary learning), KCs in a more complex domain may need to apply in different ways to different contexts, both alone and in combination with other knowledge components.

The idea of the independent KC is a first step in understanding how these independent skills combine to allow for the solution of problems, problem steps, and other tasks such as explanation or recall. In some areas of domain modeling, this assignment of KCs to items has become known as a Q-matrix, which is a simple table that describes how the collection of KCs apply to particular items of the domain (Barnes, 2005; Tatsuoka, 1983). This approach works well for many domains, but when the skills are very broad (e.g., legal reasoning), it becomes difficult to identify skill components that explain the different performances for similar items. Such domains are often called "ill-defined" (Goldberg, Sottilare, Brawner & Holden, 2011; Lynch, Ashley, Aleven & Pinkwart, 2006; Weerasinghe, Mitrovic & Martin, 2009).

Domain models take different forms, but these forms are typically in service of some instructional content that has been predetermined. As of now, there exist no systems that can freely describe generic experiences for a student. Rather, builders of educational software first must envision some target behaviors that the student must learn and present activities that exercise the KCs behind these behaviors. For example, if a student needs to learn addition, we must target the skills behind the addition behavior (perhaps including counting, carrying, and memory skills), and prepare activities that engage students in practicing these skills. This linking of the activity with a set of knowledge and skills is a *sine qua non* feature of a domain model.

This linking of KCs and activities is required for a system like GIFT, because this is basis for the pedagogical model to select activities, coupled with the status of the learner viz-à-viz the domain model. In other words, we need some set of skills that is monitored for learning to infer the proper pedagogy. This is relatively simple for an instructional activity with a one-to-one mapping of activity steps to knowledge components, but it becomes much more complex when issues of transfer are concerned. In the case of

possible transfer between tasks, the domain model must represent how that transfer occurs and the learning model must now make inferences based on the history of related task items, not just the history of the specific task at hand. An excellent example of how there may be multiple distinct assignments of skills to tasks is in the Geometry PSLC data set, where there are presently 37 posted domain models comparing alternative assignments of skills to geometry problems in the Geometry Area (1996–97) data set that can be accessed via DataShop (Koedinger et al., 2010).

Beyond determining what skills are relevant to accomplishing some activity step successfully, there are at least two key complexities that make such work challenging, which GIFT must account for. First, we must determine whether strengths on some relevant skills can compensate for weaknesses on other relevant skills, known as the compensatory issue. Second, we must determine a computational representation for how multiple skills combine to produce a response, known as a condensation rule (Rupp. & Templin, 2008). For example, we may propose a conjunctive (all skills needed) or disjunctive (any one skill needed) rule. These issues can affect credit assignment, namely, determining how to update the learner model after the student's completion of an activity with proper credit (blame) to skills that the student successfully (unsuccessfully) exercised. Such situations make transfer models difficult to formulate because as the number of possible models increases there is an explosion of possible ways to represent the domain.

If the wrong skill is credited with failure, a system may misdirect student practice, leading the student to practice a skill that is already mastered or neglecting to direct the student to practice a skill that has not yet been mastered (Koedinger, Pavlik Jr., Stamper, Nixon & Ritter, 2011). This credit assignment problem may be so severe that it explains why the ASSISTments project provides a useful alternative to standard problem solving. ASSISTments follows failed problems with additional simpler follow-up exercises and thereby allows for meticulous diagnosis in the case of failure. Such diagnosis may be critical to accurate updating of the learning model, which then allows the pedagogical model to function as it should (Razzaq, Heffernan, Feng & Pardos, 2007).

Another problem with the domain model can be when students work "off the model" forming KCs that are in error. These errant KCs can be called misconceptions, since they capture some aspects of correct knowledge but either overgeneralize or are too narrowly focused to account for all cases. For example, data on learning to compute least common multiples (LCMs) showed that many students provide the simple product, despite needing to further reduce the multiple, e.g., LCM of 4 and 6 is 12 and not 24. Prior practice with simple LCM problems actually appeared to support this error and poor transfer was observed from simple to reduced LCM problems. In contrast, the more difficult problems, which may begin with a simple multiplication, but are followed by a reduction and checking step, transferred well to the simple problems (González-Brenes & Mostow, 2013; Liu, Xu & Ying, 2012; Pavlik Jr., Yudelson & Koedinger, 2015). This sort of misconceived generalization can be problematic, since it is hard to anticipate without observing student examples, unlike the correct solution path. However, if a domain model of only correct task performance is applied, it certainly may face this issue of skills being learned incompletely or in ways applicable to only a subset of problems, which may lead to over- or under-generalization.

Automation in Domain Modeling from Learning Curve Analysis

The goal of domain modeling from learning curve analysis is to validate and improve the efficacy of instruction in a tutoring system or other learning environment. That is, learning curve analysis requires data as input. These methods cannot be applied for de novo design of tutoring systems in the absence of data.

Any given data set is the work product of students who engaged with some set of instructional activities. The activities that elicit student inputs are necessarily ordered in some way, such as according to some

theory of cognition and learning. For example, it might be proposed that some activities require students to engage in addition to arrive at a correct answer and other activities require subtraction. The theory provides the basis for a formal statement of the domain model and is the baseline domain model that the research may want to test and refine.

Domain Model Discovery

In some use cases, it may be impractical to describe existing activities through knowledge engineering, so one may wish to discover a Q-matrix automatically (González-Brenes & Mostow, 2013; Liu et al., 2012). Unfortunately, in this case, model interpretation is a challenge. First, there may be concerns with respect to identifiability, e.g., multiple Q-matrices may be equally likely to have generated a given data set. Even if we "establish sufficient conditions to ensure that the attributes required by each item are learnable from the data, we can "only expect to identify the Q-matrix up to some equivalence relation" (Liu, Xu & Ying, 2013).

Second, suppose it was possible to discover that some set of latent variables can partition a set of items based on responses to those items. It still may be unclear what skills the latent variables depict. It is critical to be able to label a latent variable as a particular skill or learning objective to ascribe instructional validity to a system. One strategy is to automate the analysis of the learning activities themselves to extract descriptors for the latent variables (Li, Cohen, Koedinger & Matsuda, 2011).

Domain Model Refinement

Rather than learning an entire domain model from data, we can start with a plausible, manually created Q-matrix and improve it. The key insight is that a particular domain model is merely a hypothesis about what makes for efficacious instruction and efficient student learning, a hypothesis that may be disproven based on the data we observe. Accordingly, we can treat domain model refinement as a search for a Q-matrix that is a better description of the data set at hand. If we start with a plausible Q-matrix and propose only incremental revisions, the search will lead us to a Q-matrix that can be interpreted with respect to its starting point.

Such a search is defined in Learning Factors Analysis (Cen, Koedinger & Junker, 2006), which consists of three components. First, *A** is a heuristic algorithm for exploring a search space efficiently. Second, Q-matrix change operators can be applied within the search to modify a Q-matrix. The split operator replaces a single latent skill variable in the Q-matrix with two distinct variables, each of which only codes for a subset of the activities that bear on the original latent variable. The merge operator does the reverse. A set of "difficulty factors" (Baker, Corbett & Koedinger, 2007) can provide a candidate list of new skills to incorporate in the Q-matrix. Third, an evaluation component defines how to measure the quality of a Q-matrix after a split or merge operation. The output of the evaluation signals whether the search is proceeding productively or if it should redirect or terminate.

Interestingly, the evaluation of a domain model can be automated in a data-driven manner by leveraging already collected data. To the degree that one admits causal inference from observational studies, it does not require novel experiments or data collection. Specifically, we can leverage statistical modeling to determine whether one Q-matrix is preferable to another for the purpose of describing a particular data set. The process is:

- Acquire a data set of student practice of skills in the domain of interest.

- Create or choose a statistical model appropriate for these data such that the model can incorporate a Q-matrix. Some existing models are described below.

- Fit the model to the data once for each Q-matrix.

- Confirm that the model is appropriate for the data.

- Compare the model fits from the different Q-matrices.

Two important evaluation considerations are which of two Q-matrices makes more accurate predictions about student performance and which is more parsimonious, i.e., uses relatively fewer parameters for its predictions. Akaike information criterion (AIC) and Bayesian information criterion (BIC) scores (Wasserman, 2004), which express predictive accuracy, parsimony, and their trade-off quantitatively, can be used for model selection.

A data set may include a number of skills. By definition, a split or merge operation only affects two skills and a small subset of student activities, while predictive accuracy is measured with respect to the entire data set. As a result, the difference in predictive accuracy may be small even if a Q-matrix modification reflects something that is inherently true about cognition and learning. A focused benefits investigation can identify which changes to the Q-matrix drove the greatest improvement in predictive accuracy (Koedinger, McLaughlin & Stamper, 2012).

The evaluation component of Learning Factors Analysis (Cen et al., 2006) was originally defined with respect to a statistical model called the Additive Factors Model (AFM), which posits that student success in practicing some activity is a function, in part, of the quantity of practice that the student has had on activities that rely on the same underlying skills. Informally, AFM computes a population-level learning curve for each skill defined in the Q-matrix and uses the curve to predict average performance after some amount of practice.

Other predictive models may also be used in place of AFM, including Performance Factors Analysis (PFA) (Pavlik Jr., Cen & Koedinger, 2009) and Recent-Performance Factors Analysis (R-PFA) (Galyardt & Goldin, 2015; Goldin & Galyardt, 2015a). The chief difference among these models is in how they represent the history of student practice on some skill for the purpose of predicting whether a subsequent attempt will be successful. Briefly, AFM considers the total quantity of practice so far; PFA distinguishes the total quantity of correct practice from incorrect practice; and R-PFA distinguishes correct and incorrect practice as well as gives greater weight to more recent evidence. These models make different predictions and their parameters differ in interpretation. Notably, the AFM model is more likely to fail to recognize the value of some domain model distinctions (a skill split) than the R-PFA model (Goldin & Galyardt, 2015b). The AFM and PFA models support domain models that tag each learning activity step with one or more compensatory skills, while the R-PFA model only supports domain models with one skill per activity step.

Shapes of Learning Curves

Once we have gathered data on student practice with some set of learning activities, we can generate a set of learning curves with respect to some specific domain model. Our goal is to generate one curve per KC, but this is not a trivial task. It is challenged by the issues of compensatory skill relationships, condensation rules, and credit assignment. Nonetheless, assuming that we have a data set and a domain model, for each skill defined in the domain model, for each practice opportunity, we can plot the average population probability of success on the skill (or equivalently, the error rate, flipping the vertical axis).

In general, we assume that our data set contains records of student learning. In an "ideal" curve (Figure 1 shape *a*), the initial probability of answering correctly is fairly low and after some sensible amount of practice the probability of success is high, indicating learning. Based on this, we can ascribe interpretations to some typical learning curve shapes (Koedinger, Stamper, McLaughlin & Nixon, 2013).

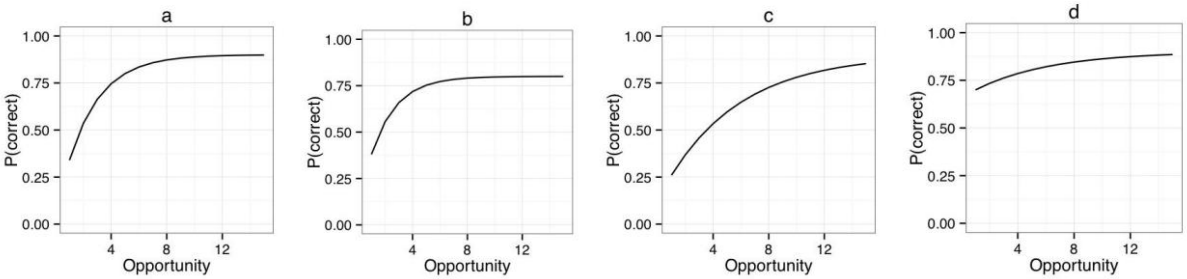


Figure 1. Typical learning curve shapes.

The population error rate may remain high even after some reasonable amount of practice (in shape *b*, the curve plateaus at 80% success). Possible causes of this are that students may not have sufficient practice opportunities to master the skill or that the curve in fact is a mixture of two skills, and averaging the skills together obscures the distinct error rates on the two skills. An example of the latter is given by Corbett and Anderson (1995, Figs. 5–6). Alternatively, the assessment activities available for this skill may have a high "slip" rate, where "slip" defines the probability of making an error despite correct knowledge, indicating a need to consider whether such a high slip rate is appropriate to teaching and assessing the skill.

If the population error rate declines very slowly (shape *c*), the skill may be especially challenging for students. This may indicate an opportunity to revise instruction on this skill or consider whether there may be sub-skills that can be called out for targeted instruction and assessment.

When the population error rate is low even at early practice opportunities (shape *d*), the data may indicate that by the time students in this data set encountered the activities that produced these observations, they have already mastered the skill in question. Alternatively, the assessment activities were very easy, perhaps inadvertently so, in the sense that students could arrive at correct answers by guessing or by applying a simple strategy.

If a learning curve is split into two distinct skills appropriately, the split ought to yield curves that are more sensible than the merged curve. For example, suppose two skills with ideal (shape *a*) curves are inappropriately merged into one in a domain model, and in the data set all practice of the two skills is blocked (all practice on one skill follows all practice on the second). The resulting shape will have a spike in the error rate when students begin to practice the second skill (Fig 2 shape *a Blocked*). If this curve is split correctly, the resulting curves will resemble shape *a* in Figure 1.

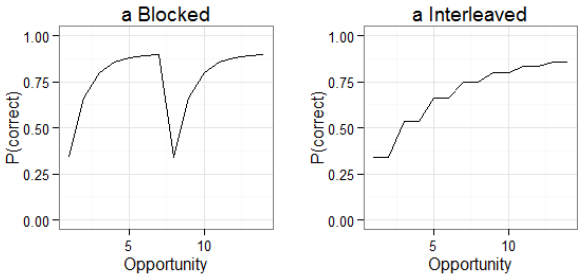


Figure 2. Learning curves when practice on two ideal skills is blocked (left) or interleaved (right).

However, it is also possible that when the two skills are merged into one, practice on the two skills is interleaved (i.e., students practice one activity from each of the skills, then a second activity from each of the skills, etc.). This would yield a merged curve depicted in Figure 2, *a Interleaved*. It is possible that splitting such a curve into its underlying two skills may not result in a statistically detectable difference; in this case, the R-PFA model is more likely to detect a difference than AFM (Goldin & Galyardt, 2015a).

In sum, the shape of learning curves may reveal opportunities for improving the domain model, the instructional activities, and their sequence.

Leveraging Learning Curve Analysis in Practice

An analysis of task data from Carnegie Learning's Cognitive Tutor (Koedinger, Corbett, Ritter & Shapiro, 2002) can serve to illustrate the way that learning curves are used to refine domain models. Figure 3 shows a word problem from the algebra tutor. The student's task is to complete a table corresponding to the word problem, and the table shown has been partially completed. One of the key steps in this problem is to complete the empty cell by entering an algebraic expression for the dependent variable in the word problem. The illustrated problem is one of a set of 41 word problems that could be presented to students in this section of the curriculum. For each student, problems are selected from this set of problems so as to maximize student exposure to KCs that have not been mastered and minimize work on KCs that have been mastered. If two or more problems are identical with respect to their match to the student's mastery, one will be selected at random.

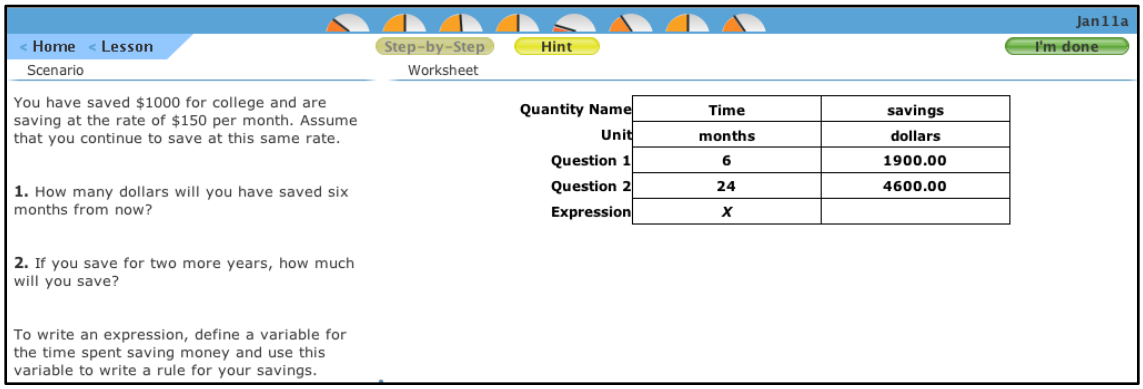


Figure 3. Cognitive Tutor word problem.

In the initial domain model for this section of the curriculum, a single KC represented the skill of writing an expression within a word problem. Figure 4 shows a learning curve for that KC, collected from 497 students who completed this section of the curriculum. The learning curve shows that the first time that students attempted to write an expression for a problem like this, 25% of the attempts were correct. On the second opportunity to write an expression (in the next problem presented), students averaged 23% correct. The third opportunity evidenced substantial improvement, reaching 62% correct. Students gradually improve with subsequent opportunities. Although this curve does illustrate learning (since students do improve at the task), the curve most closely resembles the "interleaved" curve illustrated earlier (Figure 2 *a Interleaved*). Since problems are randomly selected (given equality with respect to skills), this effect cannot be due to the educational value of specific problems. The shape of the curve indicates that there may be more than one KC being learned.

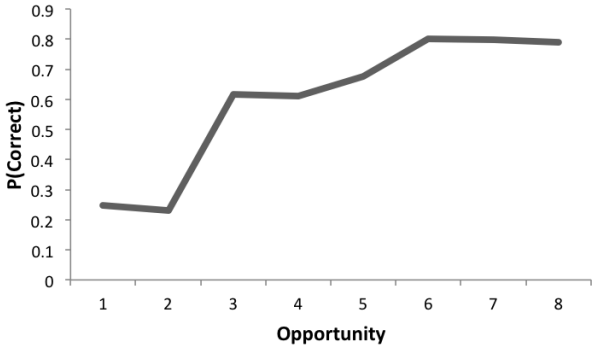


Figure 4. Learning curve for all word problems.

Figure 5 illustrates the same data as Figure 4, but split according to whether the word problem involved a positive or negative slope. A negative slope problem might, for example, involve calculating the altitude of an airplane that is landing, so that the altitude decreases over time. In the positive slope graph (left), the first opportunity represents the first positive slope problem presented to a student and the second opportunity represents the second positive slope problem presented. Similarly, the negative slope graph (right) displays opportunities with respect to problems of that subtype.

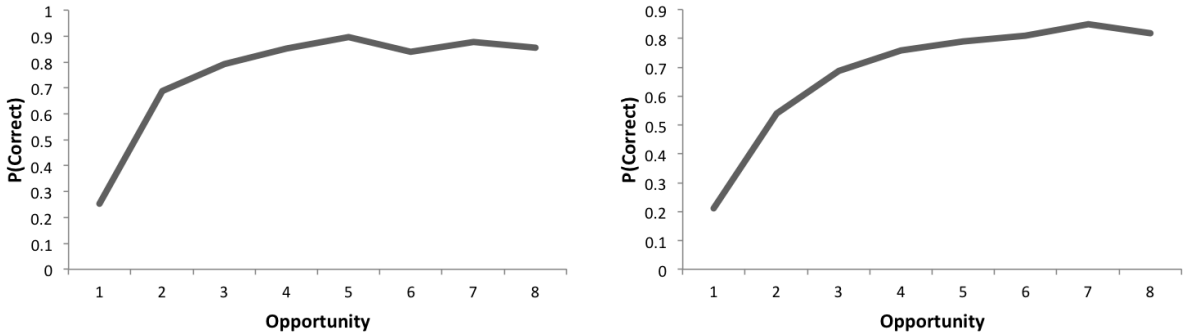


Figure 5. Learning curves for problems on positive slope (left) and negative slope (right).

Both positive and negative curves resemble the "ideal" curve shape illustrated earlier. This is evidence that there are, in fact, two KCs underlying performance of this task.[5] The pattern shown in these data suggests that there is in fact no transfer between writing an expression with a positive slope and writing one with a negative slope. This is a remarkable and non-obvious discovery. Students learning to solve problems of this type have good knowledge of positive and negative numbers and of positive and negative slopes in linear functions. Curricula and educational standards do not normally treat word problems with positive slopes as distinct from word problems with negative slopes. Nevertheless, this data pattern indicates that students think about increasing quantities and decreasing quantities very differently. Although the data were collected in the context of students using Cognitive Tutor, it is likely that this KC split is more generally applicable to students in beginning algebra. Many such students may understand how to model a linear relationship having a positive slope without knowing how to form such a model when the relationship has a negative slope.

This example illustrates an important point about domain modeling that is often underappreciated. The domain model is relative to students' level of knowledge. Once students come to a more advanced understanding of the linear functions underlying word problems, they will likely no longer show any distinction between positive and negative slope problems. A model of a more advanced student might not make this distinction.

Recommendations and Future Research

Domain models are traditionally constructed by subject-matter experts who are able to describe the basic distinctions made within a domain. Such distinctions, however, are often based on an expert view of the domain, rather than a view of the kinds of distinctions that students may need to make in order to develop a complete understanding of the domain. A fundamental assumption underlying the view presented in this chapter is that a domain model is primarily about transfer. We know a domain model is correct if it correctly predicts that knowledge required to solve one problem will (or will not) be sufficient to solve a different problem.

Many of the distinctions required to develop a sufficient theory of transfer in a particular domain can be delineated through cognitive task analysis (Clark, Feldon, vanMerrienboer, Yates & Early, 2008; Hess & Saxberg, 2013), and that process is an excellent first step toward developing a domain model. However, many important domain distinctions are not as obvious and are unlikely to be discovered without analysis of student performance. Current educational technologies are able to capture data in sufficient quantity and detail to enable data-driven refinements to domain models. Variants of learning factors analysis show great promise in automating these refinements, which we expect to greatly expand its utility.

Data-driven domain modeling is a process of refinement, starting with initial hypotheses about a domain (including a cognitive task analysis and specification of possible knowledge component distinctions). This cycle of refinements is necessarily incremental and proceeds fairly slowly. A major advance for domain model construction would be the development of standards and practices for documenting and sharing domain models found through these methods. Such sharing would allow instructional system designers to begin with more complete domain models and more accurately model the growth of student knowledge.

[5] Note that the initial curve (Figure 4) illustrates the "interleaved" pattern because the data were actually collected after the split between positive and negative slopes was implemented. The problem selection algorithm will tend to interleave skills.

Given the issues discussed in this chapter, below are some specific recommendations for domain modeling support within GIFT:

1. Treat each domain model as a hypothesis rather than a perfect final definition. Allow for a variety of domain models, similar to how the PSLC DataShop can flexibly treat domain models. Facilitate iterative improvement of domain models.

2. Create a separation between the activity user interface and the domain model, and guard against a domain model that addresses the quirks of your interface rather than actual aspects of domain knowledge. To do so, provide a variety of instructional activities for each target skill and leverage well-specified domain models in authoring of distinct and varied instructional activities.

3. Consider the sharing of domain models. A domain model that is validated in multiple instructional contexts is inherently more trustworthy than one that is only applied in one tutoring system.

4. Domain experts may overstate the significance of hypothetical difficulty factors. That said, if a difficulty factor does not improve a domain model, there are modeling and measurement issues worth checking. For example, is there sufficient statistical power to test for the presence of a difficulty factor? Is the model at hand falling prey to a false negative? Consider the validity of inferences from learning curves where student attrition and selection bias have differential effects at early and late practice opportunities.

5. In the pedagogical model, enable randomized ordering of activities for experimental validation of domain models. Even for adaptive activity selection in production-ready systems, when a domain model claims that some activities are interchangeable ways to acquire and practice a skill, there needs to be a way to validate this claim.

Acknowledgments

We thank April Galyardt for helping us clarify differences among different types of learning curves and for generating some figures.

References

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science*, *26*(1), 85–112.

Baker, R. S., Corbett, A. T. & Koedinger, K. R. (2007). The difficulty factors approach to the design of lessons in intelligent tutor curricula. *International Journal of Artificial Intelligence in Education*, *17*(4), 341–369.

Barnes, T. (2005). The Q-matrix Method: Mining Student Response Data for Knowledge. In J. Beck (Ed.), *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop* (pp. 39–46). Pittsburgh, PA: AAAI Press.

Cen, H., Koedinger, K. R. & Junker, B. (2006). Learning Factors Analysis - A general method for cognitive model evaluation and improvement *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 164–175): Springer Berlin / Heidelberg.

Clark, R. E., Feldon, D., vanMerrienboer, J., Yates, K. & Early, S. (2008). Cognitive Task Analysis. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology (3rd ed.)* (pp. 577–559). Mahwah, NJ: Lawrence Erlbaum Associates.

Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278.

DiCerbo, K. & Behrens, J. (2012). Implications of the digital ocean on current and future assessment. In R. W. Lissitz & H. Jiao (Eds.), *Computers and their impact on state assessment: Recent history and predictions for the future* (pp. 273–306). Charlotte, NC: Information Age.

Galyardt, A. & Goldin, I. (2015). Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining, 7*(2), 83–108.

Goldberg, B., Sottilare, R., Brawner, K. & Holden, H. (2011). Predicting Learner Engagement during Well-Defined and Ill-Defined Computer-Based Intercultural Interactions. In S. D'Mello, A. Graesser, B. Schuller & J.-C. Martin (Eds.), *Affective Computing and Intelligent Interaction* (Vol. 6974, pp. 538–547): Springer Berlin Heidelberg.

Goldin, I. & Galyardt, A. (2015a). Convergent validity of a student model: Recent-Performance Factors Analysis. In O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura & M. Desmarais (Eds.), *Proceedings of 8th International Conference on Educational Data Mining. Madrid, Spain* (pp. 548–551). Madrid, Spain.

Goldin, I. & Galyardt, A. (2015b). On feasibility of using learning curve analysis to refine domain models. *Invited presentation presented at the Generalized Intelligent Framework for Tutoring Domain Modeling Expert Workshop*. Orlando, FL.

González-Brenes, J. P. & Mostow, J. (2013). What and When do Students Learn? Fully Data-Driven Joint Estimation of Cognitive and Student Models. In S. K. D'Mello, R. A. Calvo & A. Olney (Eds.), *Proceedings of the 6th International Conference of Educational Datamining* (pp. 236–239). Memphis, TN.

Hess, F. M. & Saxberg, B. (2013). *Breakthrough Leadership in the Digital Age: Using Learning Science to Reboot Schooling*: Corwin Press.

Koedinger, K. R. (2001). Cognitive tutors as modeling tool and instructional model. In K. D. Forbus, Feltovich, P. J. (Ed.), *Smart Machines in Education: The Coming Revolutions in Educational Technology* (pp. 145–168). Menlo Park, CA: AAAI/MIT Press.

Koedinger, K. R. & Anderson, J. R. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30–43.

Koedinger, K. R., Baker, R. S., Cunningham, K., Skogsholm, A., Leber, B. & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. In C. Romero , S. Ventura & M. Pechenizkiy (Eds.), *Handbook of educational data mining* (Vol. 43). Boca Raton: CRC Press.

Koedinger, K. R., Corbett, A. T. & Perfetti, C. (2012). The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science, 36*(5), 757–798. doi:10.1111/j.1551-6709.2012.01245.x

Koedinger, K. R., Corbett, A. T., Ritter, S. & Shapiro, L. J. (2002). *Carnegie Learning's Cognitive Tutor™: Summary research results*. Retrieved from

Koedinger, K. R., McLaughlin, E. A. & Stamper, J. C. (2012). Automated Student Model Improvement. In K.Yacef, O. Zaïane, H. Hershkovitz, M. Yudelson & J. Stamper (Eds.), *Proceedings of the 5th International Conference on Educational Data Mining* (pp. 17–24). Chania, Greece.

Koedinger, K. R., Pavlik Jr., P. I., Stamper, J., Nixon, T. & Ritter, S. (2011). Fair blame assignment in student modeling. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero & J. Stamper (Eds.), *Proceedings of the 4th International Conference on Educational Data Mining* (pp. 91–100). Eindhoven, the Netherlands.

Koedinger, K. R., Stamper, J. C., McLaughlin, E. A. & Nixon, T. (2013). Using Data-Driven Discovery of Better Student Models to Improve Student Learning. In H. C. Lane, K. Yacef, J. Mostow & P. Pavlik (Eds.), *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9–13, 2013. Proceedings* (pp. 421–430). Berlin, Heidelberg: Springer Berlin Heidelberg.

Li, N., Cohen, W. W., Koedinger, K. R. & Matsuda, N. (2011). A Machine Learning Approach for Automatic Student Model Discovery. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero & J. Stamper (Eds.), *Proceedings of the 4th International Conference on Educational Data Mining* (pp. 31–40). Eindhoven, the Netherlands.

Liu, J., Xu, G. & Ying, Z. (2012). Data-Driven Learning of Q-Matrix. *Applied Psychological Measurement, 36*(7), 548–564. doi:10.1177/0146621612456591

Liu, J., Xu, G. & Ying, Z. (2013). Theory of the Self-learning Q-Matrix. *Bernoulli : official journal of the Bernoulli Society for Mathematical Statistics and Probability, 19*(5A), 1790–1817. doi:10.3150/12-BEJ430

Lynch, C., Ashley, K., Aleven, V. & Pinkwart, N. (2006). Defining ill-defined domains; a literature survey. In V. Aleven, K. Ashley, C. Lynch & N. Pinkwart (Eds.), *Proceedings of the workshop on intelligent tutoring*

systems for ill-defined domains at the 8th international conference on intelligent tutoring systems (pp. 1–10). Jhongli, Taiwan.

Nathan, M. J., Koedinger, K. R. & Alibali, M. W. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In L. Chen (Ed.), *Proceedings of the Third International Conference on Cognitive Science* (pp. 644–648). Beijing, China: USTC Press.

Pavlik Jr., P. I., Brawner, K. W., Olney, A. & Mitrovic, A. (2013). A Review of Learner Models Used in Intelligent Tutoring Systems In R. A. Sottilare, A. Graesser, X. Hu & H. K. Holden (Eds.), *Design Recommendations for Adaptive Intelligent Tutoring Systems: Learner Modeling* (Vol. 1, pp. 39–68): Army Research Labs/ University of Memphis.

Pavlik Jr., P. I., Cen, H. & Koedinger, K. R. (2009). Performance factors analysis -- A new alternative to knowledge tracing. In V. Dimitrova, R. Mizoguchi, B. d. Boulay & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 531–538). Brighton, England.

Pavlik Jr., P. I., Yudelson, M. & Koedinger, K. R. (2015). A Measurement Model of Microgenetic Transfer for Improving Instructional Outcomes. [Journal article]. *International Journal of Artificial Intelligence in Education, 25*, 346–379. doi:10.1007/s40593-015-0039-y

Razzaq, L., Heffernan, N., Feng, M. & Pardos, Z. (2007). Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning, 5*(3), 289–304.

Rupp, A. A. & Templin, J. L. (2008). Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art. *Measurement, 6*(4), 219–262.

Sottilare, R. A., Graesser, A., Hu, X. & Brawner, K. W. (2015). *Design Recommendations for Intelligent Tutoring Systems: Volume 3-Authoring Tools* (Vol. 3): US Army Research Laboratory.

Sottilare, R. A., Graesser, A., Hu, X. & Goldberg, B. (2014). *Design Recommendations for Intelligent Tutoring Systems: Volume 2-Instructional Management* (Vol. 2): US Army Research Laboratory.

Sottilare, R. A., Graesser, A., Hu, X. & Holden, H. (2013). *Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling* (Vol. 1): US Army Research Laboratory.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement, 20*(4), 345–354.

Wasserman, L. (2004). *All of statistics: a concise course in statistical inference*. New York: Springer.

Weerasinghe, A., Mitrovic, A. & Martin, B. (2009). Towards Individualized Dialogue Support for Ill-Defined Domains. *International Journal of Artificial Intelligence in Education, 19*(4), 357–379.

CHAPTER 11 – Making Static Lessons Adaptive through Crowdsourcing and Machine Learning

Joseph Jay Williams[1], Juho Kim[2], Elena Glassman[3], Anna Rafferty[4], and Walter S. Lasecki[5]
[1]Harvard University, [2]Korea Advanced Institute of Science and Technology,
[3]MIT Computer Science and Artificial Intelligence Laboratory,
[4]Carleton College, [5]University of Michigan

Text components of digital lessons and problems are often static: they are written once and too often not improved over time. This is true for both large text components like webpages and documents as well as the small components that form the building blocks of courses: explanations, hints, examples, discussion questions/answers, emails, study tips, and motivational messages. This represents a missed opportunity, since it should be technologically straightforward to enhance learning by improving text, as instructors get new ideas and data are collected about what helps learning. We describe how instructors can use recent work (Williams, Kim, Rafferty, Maldonado, Gajos, Lasecki & Heffernan, 2016a) to make text components into adaptive resources that semi-automatically improve over time, by combining crowdsourcing methods from human-computer interaction (HCI) with algorithms from statistical machine learning that use data for optimization.

Introduction

Many online education resources are arguably static and one size fits all. They provide significant content, but like a textbook, provide the same explanations and material to all users. Once the resource has been created, it generally remains constant, with text and pictures rarely changing, regardless of how helpful they are for learning. There is little technology support for instructors to do rapid content iteration or collect data that they can use to decide which versions are better than others.

An intelligent tutoring system (ITS) provides one alternative to this static instruction. ITSs deliver different versions of a learning experience to each student, based on their knowledge, attitudes, or behavior. However, this may require instructors to generate alternative versions of the content. Most instructors do not have the resources to make many different versions of lessons, such as alternative explanations and examples, and they may not be sure which explanations and examples will be most effective.

How can instructors turn static text into adaptive components that can be perpetually enhanced? This chapter discusses how the goals of instructors and researchers in the learning sciences can be advanced by integrating crowdsourcing and design ideas from human-computer interaction (HCI) with machine learning algorithms that provide automatic optimization. The approach is illustrated by describing a system that instructors can use as a plugin to improve text components of instruction. It is called the Adaptive Explanation Improvement System (AXIS) and was first reported in Williams, Kim, Rafferty, Maldonado, Gajos, Lasecki & Heffernan (2016a).

AXIS lets an instructor designate an existing static explanation to be turned into an adaptive, data-driven component that tests out different explanations and improves over time. Each component asks learners to explain why an answer to a problem is correct, which promotes reflection and understanding (Williams & Lombrozo, 2010), as is well established in education as the *self-explanation* effect (Chi et al., 1989; 1994; Lombrozo, 2006). At the same time, learners' explanations can be collected and then presented to help *future* students learn, given the importance of high quality explanations (Renkl, 1997). The instructor then indicates how explanations should be compared (e.g., based on learner ratings or on learner performance on a related problem), and AXIS applies a machine learning algorithm automatically analyze and use the

data being collected to choose better explanations for future students. For example, in a computer science class, one might want to determine what explanation of a sorting algorithm is most clear to learners, as measured by students' rating of the helpfulness of the explanation. If learners are provided with the opportunity to rate the explanation that was provided to them, their responses can be used to select better explanations for future learners.

The original paper (Williams et al., 2016a) provides the technical details of the system design and algorithm. The current chapter is intended to provide a higher level overview of how the approach is relevant to instructors and researchers outside of HCI and machine learning. In that vein, we begin with a brief overview of relevant work.

Related Work

Human Computation and Crowdsourcing for Education

Creating usable, understandable explanations for answers to problems is a task that current artificial intelligence (AI) approaches still struggle to achieve, although great success has been made in building intelligent conversational tutors for specific tasks (e.g., Graesser, Chipman, Haynes, Olney, 2005). Crowdsourcing for human computation has been used to create useful systems in a variety of settings where an AI still struggles (Doan, Ramakrishnan & Halevy, 2011), like assistance editing word documents (Bernstein et al., 2010). These systems achieve scalability by reducing the skill level needed by individuals in a crowd to complete components of a task (e.g., see Scribe, Lasecki et al., 2012).

Paulin and Haythornthwaite (2016) explore different facets of online education that can be crowdsourced. Yuan et al. (2016) specifically explore how rubric-guided crowd workers can produce helpful feedback on students' design submissions. Weir, Kim, Gajos, and Miller (2015) introduce the idea of "learnersourcing" as a way to elicit useful improvements to an educational system from "crowds" of learners, as a byproduct of their natural interaction with educational content (see also Kim, 2015; Kim et al., 2014).

Glassman & Miller (2016) describe a learnersourcing system that, through personalized prompts, collects hints for students from peers who have already acquired the expertise necessary to generate them. These hints helped students debug computer processors and design better transistor-based logic gates. The current chapter is similar in collecting learnersourced explanations and hints. Additionally, our work collects feedback from students, using machine learning to analyze this data and infer how popular explanations are, then actively change the system to present explanations more frequently as evidence accrues as to their effectiveness.

Tradeoffs between Experimentation and Optimization

Randomized experimental comparisons are a powerful method for quantitatively determining what is effective – from instructors and researchers identifying which of multiple explanations are satisfying, to doctors identifying which drug promotes health to product designers evaluating new interfaces. Experiments have brought great value to web analytics under the label of *A/B testing* (Kohavi, Longbotham, Sommerfield & Henne, 2009).

Typically, experimental comparisons assign people to conditions with equal probability (e.g., 50% to explanation A and 50% to explanation B) since this provides maximal statistical power to detect differences, relative to random assignment that uses non-equal probabilities (e.g., 30% to explanation A and 70% to

explanation B). Both approaches are random in the sense that it is not known *which* condition a participant will be assigned to (that is sampled), but not random in the more limited sense that the probability of being assigned to any condition is equal. Both approaches are statistically valid in allowing causal conclusions. The approach of random assignment with different probabilities provides the opportunity to reduce (but not eliminate) statistical power to discover differences, in order to maximize *actual benefit* to the participants in the experiment, since a greater proportion of them could receive the more beneficial condition.

This raises many instructional, statistical, and ethical questions, which instructors and researchers will themselves have to answer. In this chapter, we describe how this connects to a classic problem in reinforcement learning: Balancing exploitation of what has already been observed with exploration to learn more about different options.

Reinforcement Learning: Multi-Armed Bandits

Reinforcement learning (RL) is a type of machine learning that allows a system to learn through interaction with an environment (see Sutton & Barto, 1998 for an overview). RL algorithms try out different actions, and seek to determine what actions tend to be most effective. What is "effective" is defined by the algorithm designer as a particular parameter to optimize; in the case above, an effective website is one that generates mailing list signups.

In cases like testing different versions of websites or deciding what ads to show users, a class of algorithms known as *multi-armed bandit* algorithms is commonly used. The term multi-armed bandit comes from analogy to slot machines ("bandits" that "steal money" when you pay them) that have multiple arms to pull, with unknown and uncertain payoffs. A player may be trying to determine what arm to pull (what action to take) to get as large of a payoff as possible, which requires balancing exploration of arms against exploitation of arms that seem to be good.

While there are many different bandit algorithms (e.g., Auer, Cesa-Bianchi & Fischer, 2002; Chapelle & Li, 2011), all of them seek to optimize the total effectiveness of the selected options, and do so even when effectiveness is stochastic. In the case of educational applications, we define the payoff based on some measure we are trying to optimize, such as learning or motivation, and the "arms" (actions) are different versions of educational content. Multi-armed bandit algorithms provide a scalable, model free way of optimizing our choices so that we will end up being as effective as possible for as many students as possible (Clement et al., 2014; Liu, Mandel, Brunskill & Popovic, 2014).

Multi-armed bandit algorithms seek to maximize the total cumulative effectiveness of all actions over a period of time. For instance, if after one hour, 2% of users of version A and 5% of users of version B have signed up for the website, this type of algorithm will tend to assign more users to version B, since it has observed evidence that this version is more effective. However, it will still allocate some users to version A because it is not yet certain which is better: it may just be by chance that more people have signed up with version B, especially if only a small number of users have viewed the site. As the number of users increases, the algorithm's confidence in which version will be more effective increases, and it will tend to assign almost all users to the most effective option.

Making Static Text Components Adaptive

We illustrate our approach in a system for creating adaptive explanations for math problems, AXIS (Williams et al., 2016). In the following sections, we provide context about the value of explanations, then

describe how an instructor can set up AXIS for adding and adapting the explanations for how to solve math problems.

Explanations in Math Problems

To provide context for an instructor's goal of making explanations in math problems adaptive, consider the ubiquitous structure of problem activities in platforms like Khan Academy, ASSISTments, EdX, and Coursera. Students attempt a problem and are given feedback about whether their answer is correct. They may also be able to request an explanation or elaboration that explains why an answer is correct or how to solve the problem. The value of providing quality explanations is clear from the educational and psychological literature, although it is an active area of research as to which explanations are deemed satisfying and actually help learning (Lombrozo, 2006; Renkl, 1997).

Learnersourcing Interface for Eliciting Contributions

Typically, a student might see no explanation after solving a problem, or if one is given, it was written by the instructor who wrote the problem. To elicit explanations from *students* that can be used for future learners, we add a reflection prompt, asking the student to explain why the answer was correct in their own words. The prompt tells the learner that generating the explanation will help them solidify their knowledge, as shown in Figure 1. The design of this prompt is guided by decades of research in psychology and education on the benefits of generating explanations for learning (Chi et al., 1994; Williams & Lombrozo, 2010; Williams, Lombrozo, Hsu, Huber & Kim, 2016).

The prompt of course is designed to serve a second function, which is to generate explanations that future students might find helpful, in the spirit of crowdsourcing in HCI. These learner-generated explanations can be tested on future learners using reinforcement learning algorithms that evaluate the explanations using performance metrics designated by instructors.

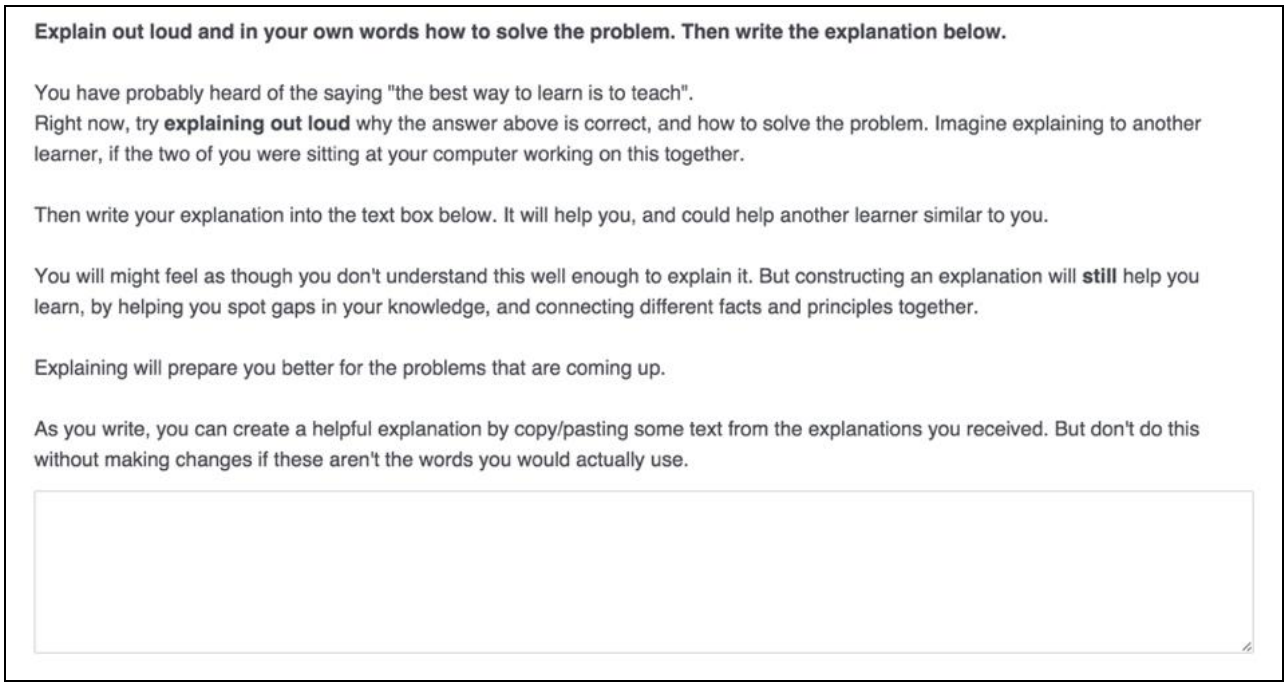


Figure 1. A prompt for a learner to explain how to solve a math problem.

Choosing Performance Metrics

The instructor must decide what metric the system should be maximizing via its choice of text component. For example, in the Williams et al. (2016a) deployment of AXIS, we decided to maximize learners' ratings of how helpful explanations were for learning. To maximize learners' ratings of helpfulness, an instructor can insert an additional question for learners who receive an explanation to answer: "How helpful do you think this explanation is for learning?" Students can respond on a Likert scale, such as 1 (Absolutely Unhelpful) to 10 (Perfect), as shown in Figure 2. In future versions of this system, instructors could choose to optimize a *combination* of performance measures, such as learners' accuracy on subsequent problems and the likelihood that they keep working.

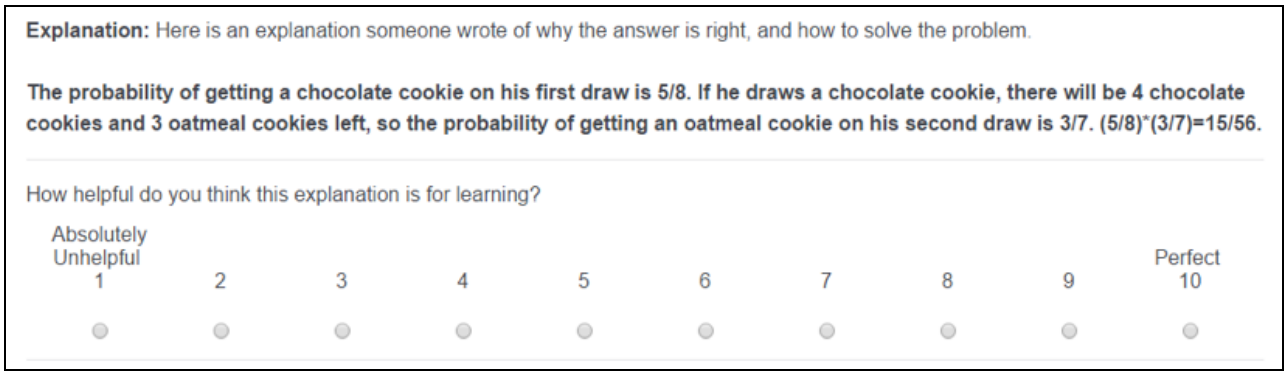


Figure 2. An example of an explanation of how to solve a particular math problem, with a prompt to rate the helpfulness of the explanation.

An instructor might wish to optimize a more direct measure of learning or progress toward the instructor's long-term goals. For instance, the value of each text option could be based on how many learners who saw that option got the next problem correct. We considered this measure but found that confounding factors, such as variations in knowledge across students, overwhelmed the relative differences in quality between explanations. Increasing the number of learners using the system would allow the machine learning algorithms to better cope with these confounding factors.

Deployment of AXIS

We (Williams et al., 2016a) deployed the AXIS system with 75 and then 150 learners, numbers comparable to multiple offerings of large residential introductory courses and sought to optimize the helpfulness of the explanations as rated by learners.

As learners solved problems, their construction of a deep understanding was guided by prompting them to explain why answers were correct. These explanations were captured and then provided to *other* learners, so that after receiving the correct answer, future learners could read an explanation of why the answer was correct. Learners were asked to indicate how helpful an explanation was on a scale from 1 to 10. A multi-armed bandit algorithm then automatically incorporated the rating from each new learner and used the ratings to alter the probabilities of which explanation it would present to the next learner. Those that tended to be highly rated were presented more frequently, and AXIS automatically incorporated new explanations that were given by users into the pool of explanations that it delivered. Figure 3 shows examples of explanations from several different categories: explanations discarded by AXIS, explanations identified as good by "Early Stage AXIS" (after just 75 learners), explanations identified as good by "Later Stage AX-

IS" (after 150 learners), and explanations written by an instructional designer on the ASSISTments platform.

We evaluated the explanations produced by the AXIS system after 75 learners ("Early Stage AXIS") and 150 learners ("Later Stage AXIS"). Relative to the original math problems that had no explanations, adding AXIS's crowdsourced and intelligently selected explanations to math problems significantly improved learners' subjective learning experience, as measured by their ratings of how much they learned (effect size of AXIS explanations impact on subjective ratings of learning, vs. no explanation: $d = 0.29$; vs. instructional designer: $d = 0.06$; vs. filtered explanation: $d = 0.22$).

In addition, these learner-generated explanations from AXIS also objectively increased learning, as measured by performance improvements on related problems that were presented in a *pre-test* (before studying problems) and then again in a *post-test* (after studying problems under different conditions, like receiving vs. not receiving AXIS explanations). Overall increases in accuracy solving problems were observed for AXIS explanations, vs. no explanation : $d = 0.19$; vs. instructional designer: $d = 0.09$; vs. filtered explanation $d = 0.01$. Breaking up accuracy into *isomorphic* and *transfer* problems: The increases were observed on *isomorphic* problems, that were identical except for changing the numbers in the problem (vs. no explanation: $d = 0.12$; vs. instructional designer: $d = 0.11$; vs. filtered explanation: $d = 0.07$), as well as on *transfer* problems, that were completely novel but tested generalization of the concept (vs. no explanation: $d = 0.22$; vs. instructional designer: $d = 0.04$; vs. filtered explanation: $d = 0.26$).

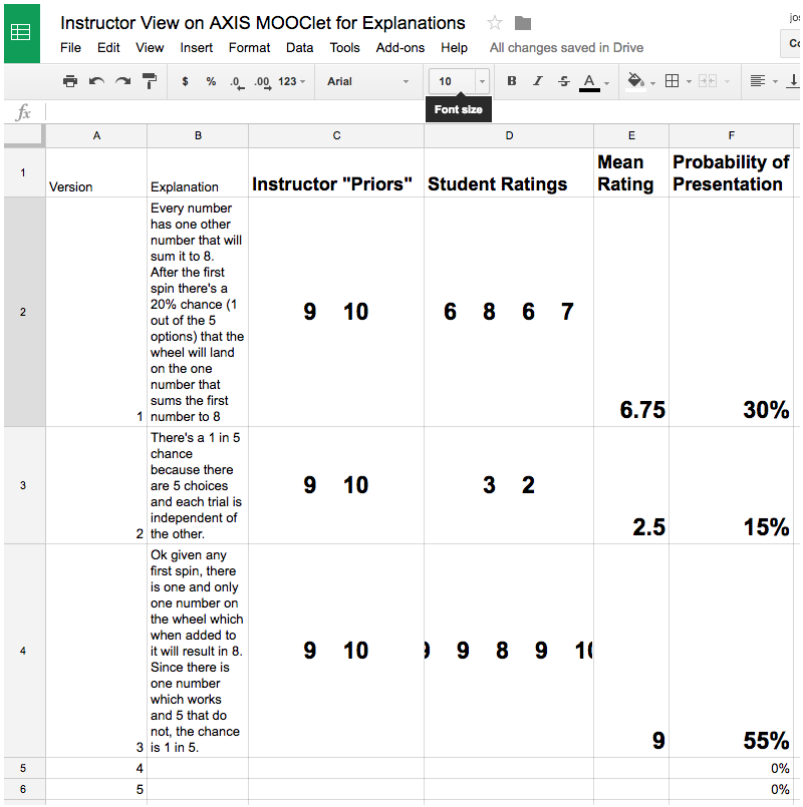
	Explanation	Explanation Rating
Learner Explanation AXIS Discarded via Filtering Rule	It is three over seven because after the chocolate cookie has been removed there are 7 cookies in the jar, leaving 3 oatmeal cookies remaining.	5.2
Early Stage AXIS	go based on the amount of cookies that are available and run a trial until the chocolate cookie is picked out, then do the same for oatmeal	4.2
Later Stage AXIS	When you have 8 cookies in the jar and 5 are chocolate you have a 5/8 chance of the cookie you draw being chocolate. When there are 7 cookies in the jar and 3 are oatmeal you have a 3/7 chance of drawing the oatmeal cookie. To get the overall probability you need to multiply 5/8 by 3/7 which results in overall probability of 15/56	6.8
Written by Instructional Designer	The total number of cookies in the jar is 8. Since there are 5 chocolate cookies the probability that Chris gets an chocolate cookie is 5/8 Since Chris removed 1 cookie from the jar and did not replace it or put it back there are now 7 cookies in the jar. So, the probability that Chris gets an oatmeal cookie from the jar is 3/7 5/8 x 3/7 = 15/56 So, the probability of Chris getting a chocolate cookie on the first draw, and an oatmeal cookie on the second draw is 15/56 Type in 15/56	7.7

Figure 3. Examples of learner-generated explanations collected and delivered by AXIS, as described in Williams et al., 2016.

Instructor Review in AXIS

At any point, the instructor can inspect the pool of explanations in the system, see the policy for how often each explanation is being presented, and also see the data about how explanations have been rated so

far by students. Figure 4 shows a screenshot from a prototype view we have built for instructors, which provides a rough illustration of the kind of information instructors can see and how they could change components of AXIS. This can be valuable in informing instructors about their "expert blind spots" about which explanations students will find helpful (Nathan, Koedinger, Alibali, 2001). The policy is the probability that each explanation will be presented and directly reflects the reinforcement learning algorithm's "judgment" of how much evidence suggests that this explanation is the best explanation, with respect to the instructor's performance metric.



	A	B	C	D	E	F
1	Version	Explanation	Instructor "Priors"	Student Ratings	Mean Rating	Probability of Presentation
2	1	Every number has one other number that will sum it to 8. After the first spin there's a 20% chance (1 out of the 5 options) that the wheel will land on the one number that sums the first number to 8	9 10	6 8 6 7	6.75	30%
3	2	There's a 1 in 5 chance because there are 5 choices and each trial is independent of the other.	9 10	3 2	2.5	15%
4	3	Ok given any first spin, there is one and only one number on the wheel which when added to it will result in 8. Since there is one number which works and 5 that do not, the chance is 1 in 5.	9 10	9 8 9 1(	9	55%
5	4					0%
6	5					0%

Figure 4. Illustration of a rough prototype for how instructors can view AXIS explanations, data, and policy.

The design of the system also allows the instructors to interact with AXIS, although Williams et al. (2016a) did not explore this. The AXIS frontend interface pulls the actual explanations from the Google sheet in Figure 4, computing the policy based on the data in that Google Sheet, using the probabilities displayed. Any modifications to that sheet directly impact the behavior of the AXIS system. This provides a range of promising functionality that needs to be carefully studied.

The instructor can directly add their own explanations at any point in time, by typing them into the sheet. AXIS will then start presenting it alongside student explanations. Instructors can also directly increase or decrease the probability of an explanation being presented. They could do this by directly overwriting the Probability of Presentation in column F, although this will override the AXIS algorithm's computation of a policy. A way for the instructor to *interact* with rather than *override* the AXIS algorithm would be by giving the explanation their *own* rating of how helpful it is for learning, entered into column C. Using a Bayesian approach that exploits the conjugate distributions in Thompson Sampling, the instructors could encode their beliefs about the quality of the explanation as giving a certain number of "fictional" observations of student ratings. In Williams et al. (2016a) this was not exploited, since we assumed that the "pri-

or" for every explanation was just two fictional observations, as if one student had rated it a 9 and another a 10.

Future works needs to explore how to elicit these judgments from instructors in a way that is effectively incorporated into the algorithm. For example, this requires specifying how much weight should be given to the instructor's rating of the explanation relative to an individual learner, which could be based on their confidence in their rating. Such capability can help instructors fine-tune the system when learner-generated explanations contain misconceptions or the instructor wants to direct learners to a particular explanation. The general approach we use aims to empower instructors to build adaptive educational systems, in the spirit of end-user programming that allows people to make changes to software systems without writing code (Myers, 1995).

Implementation and Use of AXIS: Available Resources and MOOClet Framework

While the AXIS components can be implemented using any technology of choice by the instructor team, we have provided an online tool to make it easy for instructors to use AXIS in their own classes. To find out more, sign up at the URL http://tiny.cc/useaxis. The tool allows instructors to submit their own problems and questions to get a version of AXIS for their own demo or classroom. Their version of AXIS can then be provided directly to students (e.g., by emailing a URL) or embedded into a range of Massive Open Online Course (MOOC) platforms and learning management systems, like EdX, Canvas, Moodle, since these and AXIS both use the Learning Technologies Interoperability (LTI) standard.

The crowdsourcing and machine learning approach used in AXIS can be applied to adapting a wide range of other resources. For example, an identical approach has been applied to adapting and personalizing which motivational emails students receive, which study tips appear above problems, and which lesson pages are presented. The technology and approach that support this broad range of adaptation is described by the MOOClet framework (Williams & Heffernan, 2015 and; Williams, Kim, Li, Whitehill, Maldonado, Pechenizky, Heffernan, 2014). This framework allows any existing website, app, or online resource to leverage crowdsourcing and machine learning to produce rapid adaptation, providing data in real time to both instructors and algorithms immediately after students interact with resources.

Summary, Recommendations, and Future Work

In this chapter, we have described a systematic approach for transforming static lessons into adaptive resources, in the context of self-improving explanations for how to solve mathematics problems. By drawing on research on explanation from psychology and education, we designed prompts that enhanced the learning of both the students who generated the explanations and the future students who receive the explanations. The system was designed by combining crowdsourcing principles to elicit learner responses at scale and statistical machine learning algorithms to optimize which explanations are presented, using target metrics defined by instructors. Instructors and researchers who wish to use this system or learn more can sign up at http://tiny.cc/useaxis. We briefly discuss future applications and extensions of this type of system.

Beyond Math Problems: Adapting Hints and Discussion Forum Q&A

Many features of a learning environment can be enhanced by iteratively improving written text or recommendations. MOOC discussion forums include many examples of questions asked and answered by learners and course staff. Our system could recommend answers, based on the objective performance or subjective ratings of the students who accessed them. In addition to providing explanations *after* an an-

swer is known, this system could adaptively provide *hints* for how to solve problems. The current system could be used to select and optimize the distribution of learner-generated hints from a learnersourcing system like Glassman and Miller's (2016) in real time.

Personalization

The current approach is limited by a one-size-fits-all assumption: that there is a single best-explanation for everyone, in every context, independent of their learning goal. An alternative is to collect data about differences between individuals' context and characteristics, and use this in learning *personalized* policies. When given individual's context and characteristics, reinforcement learning algorithms can also learn a policy for *whom* to deliver *which* content to. Learning personalized policies does not require without creating new explanations or modifying existing ones. The existing approach can be straightforwardly extended to create adaptive components that become personalized, once there is a metric for differences between learners.

Summary

Our system allows an instructor to launch with a single explanation, hint, or answer, and leave open the option to *improve* this over time, as data become available about what works for learners. The system is easily extendable to tailor explanations to individual learners based on their specific gaps in knowledge or misconceptions. Frameworks like our system provide the opportunity for an entire community of instructors and learners to be involved in instructional design, treating the activity as an ongoing resource to be improved and refined.

For example, if lessons, discussions, or problems in a single MOOC are used by a wide variety of classes, any instructor from these classes can contribute text components, as well as expert opinions on which text components are effective. Rather than making hard, permanent decisions about what explanation to present, the system probabilistically combines opinions and evidence from many different sources. Future ITSs can benefit from incorporating these adaptive text components into existing lessons. Future work focused on personalizing explanations based on the characteristics of individual learners and testing the effectiveness of these systems in a larger set of domains will enable ITS designers to better understand how these systems might be of the most benefit for increasing learning.

References

Aleven, V. A. and Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science*, 26(2):147–179.

Auer, P., Cesa-Bianchi, N. & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2–3), 235–256.

Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D. & Panovich, K. (2010). Soylent: a word processor with a crowd inside. In Proceedings of the 23nd annual ACM symposium on User interface software and technology (pp. 313–322). ACM.

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., and Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, 13(2):145–182.

Chi, M.T.H., de Leeuw, N., Chiu, M.H., LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive Science, 18, 439–477.

Doan, A., Ramakrishnan, R. & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. Communications of the ACM, 54(4), 86–96.

Clement, B., Oudeyer, P.-Y., Roy, D., and Lopes, M. (2014). Online optimization of teaching sequences with multi-armed bandits. In *Educational Data Mining 2014*.

Kim, J. (2015). *Learnersourcing: Improving Learning with Collective Learner Activity*. PhD thesis, Massachusetts Institute of Technology.

Kim, J., Guo, P. J., Cai, C. J., Li, S.-W. D., Gajos, K. Z., and Miller, R. C. (2014). Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 563–572.

Glassman, E. L. and Miller, R. C. (2016). Leveraging Learners for Teaching Programming and Hardware Design at Scale. In Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing (pp. 37–40). ACM.

Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005). Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Learning Technologies.* 48(4):612–618.

Weir, S., Kim, J., Gajos, K. Z. & Miller, R. C. (2015). Learnersourcing subgoal labels for how-to videos. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (pp. 405–416). ACM.

Kohavi, R., Longbotham, R., Sommerfield, D. & Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. Data mining and knowledge discovery, 18(1), 140–181.

Lasecki, W. S., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R. & Bigham, J. (2012, October). Real-time captioning by groups of non-experts. In Proceedings of the 25th annual ACM symposium on User interface software and technology (pp. 23–34).

Liu, Y. E., Mandel, T., Brunskill, E. & Popovic, Z. (2014, July). Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits. In Educational Data Mining.

Lombrozo, T. (2006). The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470.

Myers, B. A. (1995). User interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(1):64–103.

Nathan, M. J., Koedinger, K. R., and Alibali, M. W. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the Third International Conference on Cognitive Science*, pages 644–648. Citeseer.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive science*, 21(1):1–29.

Olney, A. M., Brawner, K., Pavlik, P. & Koedinger, K. R. (2015). Emerging Trends in Automated Authoring. Design recommendations for adaptive intelligent tutoring systems: Learner modeling, Vol. 3 of Adapive Tutoring (pp. 227–242). Orlando: US Army Research Laboratory.

Olney, A. M. & Cade, W. L. (2015). Authoring Intelligent Tutoring Systems Using Human Computation: Designing for Intrinsic Motivation. In D. D. Schmorrow & C. M. Fidopiastis (Eds.), Foundations of Augmented Cognition (Vol. 9183, pp. 628–639). Springer International Publishing.

Malone, T. W. & Bernstein, M. S. (2015). *Handbook of Collective Intelligence*. MIT Press.

Paulin, D. & Haythornthwaite, C. (2016). Crowdsourcing the curriculum: Redefining e-learning practices through peer-generated approaches. In *The Information Society*. 32(2), 130–142.

Sutton, R. S. & Barto, A. G. (1998). Reinforcement learning: An introduction. Cambridge: MIT press.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K., Lasecki, W. & Heffernan, N. (2016a). AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. Proceedings of the Third Annual ACM Conference on Learning at Scale.

Williams, J. J., Hsu, A., Huber, B., Kim, J. (2016b). Revising Learner Misconceptions Without Feedback: Prompting for Reflection on Anomalous Facts. In Proceedings of the ACM Conference on Human Factors in Computing Systems.

Yuan, A., Luther, K., Krause, M., ICSI, U., Vennix, S., Dow, S. P. & Hartmann, B. (2016). Almost an Expert: The Effects of Rubrics and Expertise on Perceived Value of Crowdsourced Design Critiques. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 1005–1017). ACM.

CHAPTER 12 – Data-Driven Domain Models for Problem Solving

Tiffany Barnes[1], Behrooz Mostafavi[1], and Michael J. Eagle[2]
[1]North Carolina State University, [2] Carnegie Mellon University

Problem solving is integral to learning in science, technology, engineering, math, and computer science (STEM+C) fields. Computer-aided instruction environments allow students to work open-ended problems, but often fall short on providing students with individualized feedback. Intelligent tutoring systems and cognitive tutors provide student feedback that approaches that of human tutors. We have created a data-driven method of producing problem-specific domain models as well as tutor-specific student models. We model each problem as an *interaction network*, a graph-based, problem-specific domain model that represents data collected during interactive problem solving. These domain models are naturally constrained by the specific problem-solving environment and the biases of students using it to solve problems. We use the interaction networks to mark problem steps as *correct* or *incorrect* by tracking each student's ability to recognize when to use specific tutor-actions and assign problems appropriate to each student's level. The combination of these methods allows tutor developers to individualize problem selection and provide next-step hints and worked examples that can improve student learning. In this chapter, we describe how we use interaction networks in place of formal domain models to provide intelligent support for learning, enabling the creation of intelligent tutors from existing problem-solving environments and their data. We also highlight how the process of constructing data-driven support for intelligent tutors can lead to understanding the problem-solving domains themselves.

Introduction

Problem solving is integral to learning. In science, technology, engineering, math, and computer science (STEM+C) fields, working open-ended problems is a common way for students to develop and demonstrate higher-level learning (Land, 2000). Most intelligent tutors improve learning in STEM+C by selecting problems for students and supporting problem solving with feedback that helps affirm or correct student work after each step. This individualized support is achieved by using a domain model to measure student knowledge over a set of tutor problems, i.e., knowledge tracing (Corbett & Anderson, 1994), and recognize and diagnose the work students demonstrate during step-level decisions within a problem, i.e., model tracing (Heffernan et al., 2008). Historically, these domain models have been expert systems that model how problems are solved in the given domain. In this chapter, we present our methods that leverage the design of the problem-solving interface as well as logs of student behavior to build data-driven model tracing (DMT) and data-driven knowledge tracing (DKT). Both of these techniques rely on our interaction networks that represent data-driven domain models for complex, open-ended problems, where the solver has to overcome barriers between a given problem state and a goal state using a set of multi-step activities, and there are complex interactions between the solver's used skills and abilities and the constantly changing requirements of the state-space (Wenke, Frensch & Funke, 2005).

In this chapter, we describe data-driven domain modeling with interaction networks for problem solving. It is straightforward to apply the processes presented here for problems such as those in logic and algebra, which use a set of defined rules, have multiple correct solutions, and have multiple paths to those solutions. We have also shown that it is possible to represent more general complex open-ended problems, such as writing programs, with our data-driven interaction networks (Hicks, 2014), and that is possible to

use the interaction networks to augment problem-solving environments with next-step hints (Stamper et al., 2013) and worked examples (Mostafavi et al., 2015c). Our methods build domain models from the trace, or transactional, data from human-computer interactions in problem-solving environments, leveraging the built-in system support for problem manipulation. This facilitates creating intelligent tutoring systems (ITSs) from instructional software without the need to develop formal domain models.

Interaction Networks

Interaction networks can be built using any system logs that can be mapped into *state*, *action*, and *resulting-state* tuples. Today, most learning environment logs record enough information to replay the sequence of recorded student behaviors (Baker, Corbett & Wagner, 2006), making it very likely that most interactive tutors can be mapped to these sequences. For an individual problem and student, we construct an attempt sequence, as shown in Figure 1, including edges for the student's actions and states for the pre- and post-conditions for each action. Actions correspond to rules and manipulations that can change the tutor interface, and states correspond to the values of interface elements. In problem-solving environments in STEM+C, actions are typified by rule applications, assignment of values to variables, and tying elements of a word problem to variable names, to name a few. Figure 1 uses a simple algebraic example showing a single attempt to solve the fraction addition problem of ¼+⅙ (the start state), where the first action is to find the common denominator of 12, the second is to find the numerator for ¼, the third is to find the numerator for ⅙, and the final step is to add the numerators. Each state, as illustrated here, represents enough information to reconstruct the tutor interface at a given step.

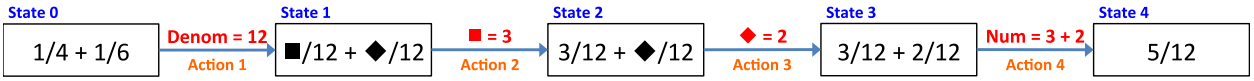


Figure 1. An attempt sequence for a fraction addition problem. The nodes are states, and edges are actions the student takes to solve the problem.

Once the tutor logs are directly converted to attempt sequences, we combine all attempts into a large, complex network representation called an interaction network, with states and edges annotated with frequencies, elapsed time, start states and goal states. This interaction network serves as the basis for our DMT and DMK, as described in the remaining sections.

Related Research

Using data-driven methods results in tutors that more accurately assess student performance and react to student behavior. Koedinger et al. (2013) give several recommendations for using data to adapt intelligent tutors, such as optimizing the cognitive model using learning factors analysis, fitting statistical models to individual students, and tracking student engagement by modeling off-task behaviors, careless errors, and mood.

Data-driven intelligent tutors have the potential to interpret why certain student behaviors occur, such as off-task behavior or errors. Elmadani et al. (2012) successfully used FP-Growth (Han et al., 2004) in order to build a set of frequent itemsets of errors to detect students' misconceptions. Fancsali (2014) used data-driven methods to detect behaviors that break the intended experience in an ITS, such as off-task behavior and gaming the system. These works show that researchers can analyze data gathered from data-driven intelligent tutors to find insights into what the students are doing and why, and use these insights to improve the curriculum and improve each data-driven tutor's ability to react to students.

Data-Driven Model Tracing

In our previous work, we have used interaction networks to automatically generate next-step hints using the Hint Factory (Stamper et al., 2013) and create worked examples of problems using the most efficient student solution (Mostafavi et al., 2015c). Initially, we calculate the expected value for each state in the interaction network using value iteration, which assigns a large positive reward to goal states and a small cost to actions, with the resulting expected values estimating the nearness each state is to a goal. Using these expected values, the Hint Factory generates a hint policy, which finds the recommended next-step action for each state in the interaction network. A hint template translates the information coded on the action into a human-readable hint. The Hint Factory can only work if each student's likely behavior is already encoded into the interaction network – and this depends on sufficient data collection. Interestingly, the Hint Factory has been shown to have good hint coverage for logic, where interaction networks built from small initial samples resulted in hints for a large percentage of students (Barnes & Stamper, 2007). This happens because there are subsets of tutor states that are particularly important to the problem. We call these key states, whose prominence in problem solving leads to scale-free networks – i.e., networks that exhibit a power-law degree distribution (Barabási & Albert, 1999) – that allow the Hint Factory to be effective with less data.

Key States in Interaction Networks and Scale-Free Properties

Key states are a subset of the interaction network states that have a much higher number of neighboring states than average. Key states represent points in the problem with a diverse set of steps leading to or away from it. These key states are the most highly connected in a long-tailed power-law distribution of network degree. Interaction networks represent an empirical sample of a group of students' problem-solving behavior. We have found evidence that interaction networks exhibit scale-free properties across the domains of propositional logic proofs, Cartesian coordinates, and programming puzzles (Eagle et al., 2015c). This means that, empirically, only a few of the observed states (key states) are highly connected to many other states, accounting for a majority of the actions taken while students solve problems. Even in domains where the potential state-space of the problem is infinite, in practice the observed state-space is much smaller. This is because there are humans exploring the state-space and they are biased toward actions that make sense for solving the problem. They may also be in a class, where the instructor was likely to provide them with strategies that would bias them to approach the problem solving in a particular way.

To explore connectivity across the interaction network, we calculate assortativity (Eagle et al., 2015c). Assortativity measures the correlation between a state's degree and the degree of its children, parents, or neighbors (depending on edge direction). Assortative (out degree) interaction networks represent problems where students make very diverse choices for several states in a row; this suggests that there may be several equally important but independent choices that students can make at once. Dissortative interaction networks represent problems where students have several choices of actions in one state, but consecutive states have lower degree. This may indicate a partial order associated with problem subgoals, or that some actions make the next few actions obvious to students. It is also possible for a network to be composed of both assortative regions and dissortative regions.

Part of the reason interaction networks are effective for modeling complex, open-ended problem solving is because they are scale-free and that even when the number of options a user has is unlimited, like-minded users are likely to perform similar actions. Users drawn from the same population (like a single class or students of the same ability) are likely to have similar behaviors, so their interaction networks produce a rather limited exploration of the problem state-space. The scale-free property of interaction networks means the likelihood of adequate state-space coverage is higher with smaller data sets. We have

shown that in logic, students taught different techniques for solving logic proofs explored quite different parts of the problem state-space (Eagle, 2014; Eagle et al., 2015b).

Estimating Coverage of Interaction Networks

The interaction network for a problem represents the empirically observed problem space. We treat the interaction network as a sample of behaviors generated by students from a population, and hence it makes sense to estimate the diversity and coverage of the behaviors observed as opposed to those that might exist in the larger population. We use Good-Turing frequency estimation to estimate size of the unobserved portions of the interaction network (Eagle et al., 2015b). Good-Turing frequency estimation (Gale, 1995) uses the distribution of previously observed data to predict the likelihood that a new observation will be something never before seen. P_0 is the estimation for the proportion of unobserved states and I_C is the network coverage. We have shown these estimators to be accurate in predicting the number of previously unobserved states encountered in a new sample, as well as in predicting the future size of the network (Eagle et al., 2015b).

In estimating the coverage of a network it is important to distinguish between the *hintable* states and the *unhintable* states. A *hintable* state is a state in which a path to a goal-state exists; therefore, it is a state in which it is possible to generate a next-step hint toward the solution. The induced sub-graph containing only the hintable states is referred to as the *hintable network*. Tracking the I_C and P_0 for both the full network and the hintable network allows researchers to monitor the effects of different state and action matching functions on the network.

For some domains, such as programming, particularly those with a large number of actions, or many similar but non-identical actions, using a straightforward state representation will result in a sparse state-space and a very small hintable network, making it difficult to derive a data-driven next step from any but the most common states (c.f. Rivers & Koedinger, 2015). For such domains, developing a state-matching function allows matching states to be merged in the interaction network, as one way to increase the number of hintable states. In this reduced interaction network, next steps can be derived from closely related observations in addition to exact matches. In choosing the state-matching function, we encounter a classic trade-off. Researchers can adjust the matching function to be more tolerant, which will make the network more dense, but this comes at a cost of potential loss of important contextual information. It is important to choose a state-matching function that retains the contextual information needed to provide feedback. In our prior work, we have represented states as the set of postconditions (results of performing actions in the tutor). We have proposed both ordered and unordered matching functions. Ordered matches mean that the two matching states have exactly the same steps executed in the same order. Unordered matches mean that the two matching states have all the same parts, but may not have been done in the same order. Hicks et al. (2014) matched states by the output of student code as well as by the direct code comparison. In this case, some contextual information is lost, so presenting the hint in terms of "next lines of code" may no longer be accurate for all students in the generalized state, who may have very different programs generating similar output. Instead, the hint is presented in terms of the expected changes to output, so it is still relevant from any of the generalized states.

Approach Maps for Comparing Problem-Solving Approaches

Interaction networks represent the problem-solving behavior of a sample of students drawn from a population. Hence, we would naturally expect that students drawn from different populations, say, from the same course with different instructors or from similar courses taught in different majors, would produce different interaction networks. To examine the differences between groups of students, we have derived *approach maps*, a high-level representation of student problem attempts. Approach maps are derived by

using an edge-betweenness network clustering method on interaction networks. The overall network is separated into *regions*, areas of the network where states have more internal connections than external connections. Each region roughly corresponds to a problem subgoal. As a visualization, approach maps greatly reduce the space needed to describe the student-tutor data.

For comparing the problem-solving behavior of two groups we look at the number of students from each group represented in each network *region.* If we find that one group is represented more than would be expected (by a chi-squared test that assumes equal probabilities of transitions between states across the groups) we will have found a difference in student behavior. We used Approach Maps to show differences in problem solving for students who received next-step hints and a control group and found that next-step hints had dramatic effects on behavior which persisted in later problems where neither group had hints (Eagle, 2014). This work also found evidence for unproductive (or buggy) regions, where students who entered would be highly unlikely to ever solve the problem.

Data-driven Domain Knowledge Models

In this section, we present our emerging methods to apply interaction networks to serve the purpose of domain knowledge models within data-driven tutors (Mostafavi & Barnes, 2013; Mostafavi et al., 2015a; Mostafavi et al., 2015b; Mostafavi et al., 2015c). In cognitive tutors, experts label knowledge components (KCs) that must be known to solve each problem, and in constraint-based tutors, experts specify constraints that valid solutions must satisfy (Mitrovic, Koedinger & Martin, 2003). Both of these represent expert-crafted domain knowledge models that can be used to model student knowledge. These student models facilitate pedagogical decision making, such as problem selection, the number of practice problems a student must solve, and often include expert crafted next-step hints. In this section, we describe our techniques to construct data-driven knowledge tracing and data-driven pedagogies to promote tutor completion (Mostafavi et al., 2015a), that circumvent the need to build explicit domain knowledge models into the tutor. Further, the derived models can be inspected to reveal the structure of domain knowledge as demonstrated by students. Johnson et al. (2013) created a visualization tool for exploring student behavior, and Eagle et al. (2015a) used visualizations of interaction networks to derive insight into how students solved puzzles in an optics game.

Data-Driven Knowledge Tracing (DKT)

Our approach for DKT assigns knowledge scores to student actions according to how they compare with students in the interaction network. In classical Bayesian knowledge tracing (BKT), student models consist of probabilities that students have learned each KC, with updates to these scores after a student interacts with an item related to the KC, called a skill opportunity (Desmerais & Baker, 2011). In a data-driven tutor, there are no expert-defined KCs, so we define KCs for each action (action-KCs) and update the action-KCs based on whether students' best use opportunities to apply an action (action-opportunities). This results in a model that represents whether students know both how and when to apply actions (Eagle, 2012).

In problem-solving environments, it is relatively simple to identify correct problem solutions, but it is harder to automatically identify the correctness and appropriateness of individual actions. For example, an individual action (like adding 3+3 to get 6) may be a correct application of a rule (addition), but might not be useful for solving the specified problem (like multiplying 3x5). Instead of expert evaluation of each action, we use interaction networks to determine the relative value of each action in moving the student toward the goal. We consider each problem-solving step to be an action-opportunity: a place where many potential actions could have been applied. At each action-opportunity, we update action-KCs according to (1) whether the student applied the chosen action correctly and (2) whether other actions led more quickly

to the goal. The first update is the same as that used in BKT, assigning credit for correct applications and subtracting for incorrect ones. In the second update, we subtract credit from every action-KC whose action leads more quickly to a goal. This is determined using the expected values for each state as calculated for choosing hints in the Hint Factory (Stamper et al., 2013).

Data-Driven Problem Selection

Once we have built the DKT system, we use it to create data-driven profiles of successful strategies for completing a problem-solving tutor and for our data-driven problem selection (DDPS). To do this, we identify *exemplars* that have completed the given tutor – and use their data to build target performance models. At the end of each problem set, we compute set-scores as weighted sums of action-KCs, comparing the set-score to target models to decide what problems the student should solve next. Initially, we use expert-set weights for action-KC scores, and once we collect data, we cluster exemplar students to build data-driven set-score weights.

We now describe how to extract set-score weights. We first perform standard cluster analysis on exemplar action-KC vectors for each problem set. The resulting clusters classify exemplars according to the groups of actions they used to complete the problem set. This allows our data-driven system to learn student strategies, rather than depending on expert classifications, which may not capture the full range of student performance (Perez, 2012; van de Watering, 2006). We then use principal component analysis on the exemplar action-KC vectors to obtain the weights for the cluster set-score (Mostafavi et al., 2015b), but other methods could also be used to describe the primary action-KCs used in each cluster (such as setting weights to 1 for the top three action-KC scores and 0 to the others). The final result is a weighted vector for action-KCs for each cluster and problem set, and a final weighted average of action-KC scores for the exemplars in the cluster.

Once cluster-set-scores are computed, we match students to the nearest cluster at the end of the problem set. We then use that cluster's set-score weights to compute a proficiency score for the student and compare it to the cluster exemplar average. The DDPS assigns students with a set-score above the cluster exemplar average to a more difficult problem set and assigns students with lower set-scores to a simpler problem set. Once it is created, the DDPS can be examined to discover student problem-solving strategies, and it automatically assesses the proficiency of new students in applying the actions in those strategies. The DDPS can be customized to select problems more or less frequently, by altering the number of problems in the problem sets, and assessing proficiency more often. The DDPS could also be modified to choose problems that target particular actions where student set-scores are below those for others in their assigned cluster.

Our work has shown that students using our logic tutor with data-driven problem selection system complete more of the tutor, are more likely to complete the *entire* tutor, and have higher action-KC scores than students using the same tutor without DDPS (Mostafavi et al., 2015a). We have also shown that clustering scores to match students to exemplars within a domain can approximate expert-decision-based proficiency assessment and can create a domain model that refines itself as new users enter the system without the need for regular expert intervention (Mostafavi et al., 2015b).

Discussion

Research has shown that individualized support for learning, through feedback, hints, and adaptive problem selection can significantly improve student learning. However, ITSs require experts to create complex domain models to support this individualization, and this work cannot be reused to support learning in new domains. On the other hand, there are many existing computer-aided instructional environments in

use, and emerging platforms for massive, open online courses (MOOCs) are providing new opportunities for data collection around problem-solving. Data-driven domain modeling can harness these rich and abundant data sets to augment existing and emerging problem-solving environments with the individualized support that can improve learning. The methods proposed here can be adopted quickly and cheaply into existing systems, as we have demonstrated in our work in logic, puzzle games and as we are now applying for novice programming environments. In addition to the benefit of reduced development time, data-driven systems naturally avoid the expert blind spot problem, and can be adapted to specific populations.

In this chapter, we have described data-driven methods to build both the outer loop of problem selection and the inner loop of step-level support for intelligent tutors, providing the central pieces of intelligent tutors (VanLehn, 2006) with minimal domain expert involvement. Our interaction networks and action-KCs allow us to leverage student data to directly inform tutor pedagogical decisions – bypassing the need to build explicit domain models linking actions to KCs, production rules, or constraints. Our novel methods for action-KCs and action-opportunities allow us to assess whether students recognize when it is appropriate to apply certain actions in the environment. Interaction networks provide an input to our action-opportunity algorithm which provides *correct* or *incorrect* labels to each action, when *correctness* might otherwise be ambiguous. We have demonstrated that using data-driven knowledge tracing based on action-opportunities as part of a larger DDPS system results in greater retention and learning for students. This is particularly useful, as it allows us to use BKT-like methods, one of the most popular student modeling techniques, in open-ended problem-solving environments, with a relatively low development cost.

Recommendations and Future Research

The methods presented in this chapter use a corpus of historical student data in place of a formal domain model to provide intelligent feedback and support for problem solving. This means that simulation software, computer-aided instruction, and other problem-solving environments can be transformed into ITSs. The broader impacts of this type of work are in learning how humans solve problems (since our models are inspectable), and what computers can do to assist them. This work also makes strides for the development of tutors that are more open-ended and closer to real-world problem solving.

To apply these methods to existing problem-solving environments without intelligent feedback, we point readers to the work from Stamper et al. (2013) and Hicks et al. (2014), which can serve as case studies for applying our methods. The following is the basic work-flow to build data-driven domain modeling using logs from a computer-aided instructional system for problem solving. First, transform the log files into *prestate-action-poststate* tuples to build an interaction network, check scale-free metrics and network coverage, and iteratively refine the state-space representation and matching functions until network coverage is at least 80 percent. Second, build a Hint Factory by creating a state matching function that maps the current student state to a lookup table of interaction network states and their corresponding hints and add a hint button to the tutor interface. Third, build a DKT system by creating action-KCs for each available tutor action and writing methods to use the interaction network to assign credit for action-opportunities during each problem-solving step. Fourth, cluster exemplar action-KC scores to identify problem-solving strategies and proficiency thresholds, and add problem selection code to the system. Last, as a new student uses the system, compare the student's states and tracked action-KC scores to generate hints and worked examples, select problems based on identified problem-solving strategy, and assess student performance to select problems.

We are continuing to refine these data-driven methods, and applying them to the challenging domain of open-ended programming problems for novices. We also plan to further explore the representation and impact of in-problem step- and total tutor-time on interaction networks, hint generation, and student learn-

ing. Another important future direction for research is on how data-driven domain models should be used to direct learning, especially when combined with information on affect, and other information that might be external to a particular tutor, such as prior knowledge. Collaboration is also becoming increasingly important for learning environments, and the models presented here are built for individual learners; however, the methods here can be scaled by expanding the set of action-KCs to include group activity.

References

Baker, R., Corbett, A. & Wagner, A. (2006). Human Classification of Low-Fidelity Replays of Student Actions. In *8th International Conference on Intelligent Tutoring Systems, Educational Data Mining Workshop* (pp. 29–36). Taiwan.

Barabási, A. L. & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512.

Barnes, T. & Stamper, J. (2007). Toward the extraction of production rules for solving logic proofs. In *13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop* (pp. 11–20). Los Angeles, United States.

Ben-Naim, D., Bain, M. & Marcus, N. (2015). A User-Driven and Data-Driven Approach for Supporting Teachers in Reflection and Adaptation of Adaptive Tutorials. In *2nd International Conference on Educational Data Mining* (pp. 21–30). Cordoba, Spain.

Clauset, A., Shalizi, C. & Newman, M. (2009). Power-Law Distributions in Empirical Data. *SIAM Rev.*, *51*(4), 661–703.

Corbett, A. T. & Anderson, J.R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4), 253–278.

Desmarais, M. & Baker, R. (2011). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, *22*(1–2), 9–38.

Eagle, M. & Barnes, T. (2012). Data-driven methods for assessing skill-opportunity recognition in open procedural problem solving environments. In *11th International Conference on Intelligent Tutoring System* (pp. 615–617). Chania, Crete, Greece.

Eagle, M. & Barnes, T. (2014). Exploring differences in problem solving with data-driven approach maps. In *7th International Conference on Educational Data Mining* (pp. 76–83). London, United Kingdom.

Eagle, M., Brown, R., Rowe, E., Barnes, T., Asbell-Clarke, J. & Edwards, T. (2015a). Exploring Problem-Solving Behavior in an Optics Game. In *8th International Conference on Educational Data Mining* (pp. 25–30). Madrid, Spain.

Eagle, M., Hicks, A. & Barnes, T. (2015b). Interaction Network Estimation: Predicting Problem-Solving Diversity in Interactive Environments. In *8th International Conference on Educational Data Mining* (pp. 342–349). Madrid, Spain.

Eagle, M., Hicks, A., Peddycord III, B. & Barnes, T. (2015c). Exploring networks of problem-solving interactions. In *5th International Conference on Learning Analytics and Knowledge* (pp. 21–30). Poughkeepsie, United States: ACM.

Elmadani, M., Mathews, M. & Mitrovic, A. (2012). Data-Driven Misconception Discovery in Constraint-based Intelligent Tutoring Systems. In *Workshop Proceedings of the 20th International Conference on Computers in Education*. Singapore.

Fancsali, S. (2014). Causal Discovery with Models: Behavior, Affect, and Learning in Cognitive Tutor Algebra. In *7th International Conference on Educational Data Mining* (pp. 28–35). London, United Kingdom.

Gale, W. & Sampson, G. (1995). Good-turing frequency estimation without tears*. *Journal Of Quantitative Linguistics*, *2*(3), 217–237.

Heffernan, N. T., Koedinger, K. R. & Razzaq, L. (2008). Expanding the model-tracing architecture: A 3rd generation intelligent tutor for Algebra symbolization. *International Journal of Artificial Intelligence in Education*, *18*(2), 153–178.

Hicks, A., Peddycord III, B. & Barnes, T. (2014). Building games to learn from their players: Generating hints in a serious game. In *12th International Conference on Intelligent Tutoring Systems* (pp. 312–317). Honolulu, United States.

Johnson, M., Eagle, M. & Barnes, T. (2013). Invis: An interactive visualization tool for exploring interaction networks. In *6th International Conference on Educational Data Mining* (pp. 82–89). Memphis, United States.

Jonassen, D. (2000). Toward a design theory of problem solving. Educational Technology Research and Development, 48(4), 63–85.
Koedinger, K. R. (2013). New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization. *AI Magazine*. 34(3), 27–41.
Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*. 48(3), 61–78
Mitrovic, A., Koedinger, K. & Martin, B. (2003). A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling. In *9th International Conference on User Modeling* (pp. 313–322). Johnstown, Pennsylvania, USA.
Mostafavi, B. & Barnes, T. (2013). Determining Problem Selection for a Logic Proof Tutor. In *6th International Conference on Educational Data Mining* (pp. 387–389). Memphis, United States.
Mostafavi, B., Eagle, M. & Barnes, T. (2015a). Toward data-driven mastery learning. In *5th International Conference on Learning Analytics and Knowledge* (pp. 270–274). Poughkeepsie, New York, United States: ACM.
Mostafavi, B., Liu, Z. & Barnes, T. (2015b). Data-driven Proficiency Profiling. In *8th International Conference on Educational Data Mining* (pp. 335–341). Madrid, Spain.
Mostafavi, B., Zhou, G., Lynch, C., Chi, M. & Barnes, T. (2015c). Data-driven Worked Examples Improve Retention and Completion in a Logic Tutor. In 17th International Conference on Artificial Intelligence in Education (pp. 726–729). Madrid, Spain.
Perez, E. V., Santos, L. M. R., Perez, M. J. V., de Castro Fernandez, J. P. & Martin, R. G. (2012). Automatic classification of question difficulty level: Teachers' estimation vs. students' perception. In *Frontiers in Education Conference* (pp. 1–5). Seattle, United States: IEEE.
Rivers, K. & Koedinger, K. R. (2015). Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor. *International Journal of Artificial Intelligence in Education*, 1–28.
Stamper, J., Eagle, M. J., Barnes, T., and Croy, M.. (2013). Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education*, 22(1), 3–18.
VanLehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education.* 16(2), 227–265.
Watering, G. van de & Rijt, J. van der. (2006). Teachers and students perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. *Educational Research Review*. 1(2), 133–147.
Wenke D., P. A. Frensch, and J. Funke. (2005). Complex problem solving and intelligence: Empirical relation and causal direction. In *Cognition and intelligence: Identifying the mechanisms of the mind* (pp. 160–187). Cambridge University Press.

CHAPTER 13 – Mining Expertise: Learning New Tricks from an Old Dog

Brandt Dargue and Elizabeth Biddle
Boeing Research & Technology

Introduction

Knowing what an expert does is a key requirement for a tutor to help a learner develop and acquire expertise. However, "what an expert does" is so much more than just performing a sequence of tasks. Expertise requires intimate knowledge about the subject, automation of the fine skills to perform the task, and a particular attitude that comes from an understanding of safety, priorities, goals, and how to adapt/optimize techniques for the particular task (e.g., Ericsson and Smith, 1991). Most of this comes from experience and is performed unconsciously by experts, even they often do not realize or cannot explain what they do above and beyond the normal practice (Anderson, 1992). The art and science of cognitive task analysis (CTA) was born to capture those mental processes behind the tasks.

The CTA that the authors have used most to build domain models was developed for one of the most researched and successful deployments of intelligent tutoring. Created for discovering the expert model of maintenance technicians, we have successfully applied the technique to soft skills such as Export Compliance training. The Precursor, Action, Results, Interpretation (PARI) (Gott, 1987; Gott, Bennett & Gillet, 1986; Hall, Gott & Pokorny, 1995; Means & Gott, 1988) knowledge acquisition, or CTA process, focuses on obtaining the mental model of the experts as they perform the task. It was originally developed as a process structured around one subject-matter expert (SME) interviewing another SME. However, we have shown that PARI can be completed by a single SME using simple tools such as a spreadsheet. With this method, the spreadsheet essentially "interviews" the SME.

This chapter provides an overview of the PARI process. It then discusses a spreadsheet we have developed and used as well as a schema that enables the domain model to be built "automatically" by an expert or group of experts. While not a panacea, the method is simple to perform and has proven to be effective in a variety of domains. Therefore, the method, tools, and schema, if incorporated into the Generalized Intelligent Framework for Tutoring (GIFT), present a domain modeling capability.

Surpassing Task Analysis

An intelligent tutor provides an interactive learning environment with the purpose of developing expertise in a particular domain. Although expertise can be developed over years of experiences, a more practical process transfers the knowledge, skills, and attitudes (KSAs) of an expert to the learner. The tutor, therefore, is the conduit, the enabler, and the facilitator of that transfer. While the tutor does not need to be an expert, it needs to be a good facilitator. Of course, the tutor needs to have access to the information it is responsible for transferring to the learner. Additionally, the learner must trust both the tutor and the information. These requirements add to the criticality of the domain expertise model.

Transferring Mental Models and Intuition

Two of the main elements of the domain knowledge that novices need to learn are the mental model that the expert has of the domain and the intuition exhibited by the expert. The mental model is often a high-level functional understanding that helps the expert imagine how to respond to the situation and perform

the appropriate tasks and visualize the situation and status. For example, a mental model for automotive troubleshooting includes how the components are related and connected. The mental models can provide structure to the process and can change based on the state/status. Continuing the automotive troubleshooting example, the mental model can show how the symptoms tell the likely causes for a fault, which may change with every diagnostic test. Each expert has unique mental models of a system, but they are formed from a similar set of realities and convey the same formation.

Intuition is a term often used to describe how an expert understands the situation and knows what to do without thinking. The key here is that the reasoning is accomplished unconsciously. This is the system 1, "fast" thinking popularized by Daniel Kahneman in his book *Thinking Fast and Slow* (Kahneman, 2011). It happens quickly without our awareness based on rules the brain has formed both consciously and unconsciously. Although an important capability of the expert, intuition is difficult for experts to explain in detail for transfer to novices. Both mental models and intuition are developed over time through repeated exposure to the conditions that trigger the knowledge components and rules. In time, they become automated.

Optimizing the Learning Experience in Cognitively Realistic Environments

Klein (1993) postulated that the unconscious reasoning exhibited by experts was the result of an expert quickly matching salient perceptual cues to a recommended course of action. He coined this phenomenon Recognition Primed Decision Making. Therefore, ensuring the domain model provides the student with a variety of situations and associated cues is essential to develop automatized associations between cues and optimal responses. Studies of relevance such as those by Dewey have similarly demonstrated that "experiencing in purposeful activity is a way of understanding" (Dykhuizen, 1973, p. 272). This often drives a requirement for higher fidelity in realism. However, the training, entertainment, robotics, and modeling and simulation industries have noted a phenomenon called "The Uncanny Valley" (Mori, MacDorman & Kageki, 2012), where realism has not only a point of diminished returns but can actually be detrimental to engagement. In terms of simulations, the effect is that when a simulation is sufficiently non-realistic, factors that are realistic are easily noticed/identified. When a simulation is almost 100% realistic, non-realistic factors stand out. In training, where engagement, relevancy, and salient perceptual cues are critical to the learning process, we need to focus the realism to the salient perceptual cues that an expert uses. Therefore, it is important for cognitive transfer that the domain model enables the learner to see the relevance and think they are having the same experience that an expert would experience. To ensure the desired learning outcomes, it may be more important to model the expert's description of the salient perceptual cues the expert would use to make decisions rather than attempting to model everything.

Related Research

In the context of this chapter, the primary function of the domain model is to describe the expert's knowledge, skills, and attitudes in order to show a novice how to think and behave like an expert for the targeted domain. The domain model is the substance of the knowledge and attitude transferred from the expert, which enables the novice to acquire expertise and perform like the expert in similar situations. When obtained from the process described in this chapter, the domain model can be used to simulate the expert. However, we limit the discussion to using the domain model to enable the transfer of the expertise to the learner.

The domain model obtained as described here includes not only the expertise, but also the portions of what is often called the system model as it models systems included in the domain. For example, a household electrical wiring tutor domain model may include the logic that the light will turn on only when the switch is in the on position and the fuse, switch, wires, and bulb are good. During the CTA described

here, the expert describes his or her vision, or mental model, as the situation evolves. The expert performing the analysis also records the status of the systems in the domain. For the household electrical tutor, the analyst would record whether or not the light turns on based on the current malfunction and the actions by the expert. This is discussed further in this chapter.

A variety of methods have successfully been used to build domain models for ITSs. Each has its merits and may indeed be the best method for a particular domain or even for a particular task/activity. This chapter focuses on a particular method of cognitive task analysis originally designed and implemented for the highly complex, ill-structured job of fighter jet avionics troubleshooting. While not intended to be a complete list of methods, this section starts by briefly reviewing a few classes of alternatives.

There are two primary challenges that experts have when describing their expertise.

1. Stating for certain what they will do in a hypothetical situation. Even highly experienced people tend to behave differently when system 1 thinking is used in real life compared to how they describe they would behave when given time to think it through and use conscious system 2 thinking.

2. The devil is in the details. Experts often omit steps or details when they describe what they do. When knowledge components (KCs) are automatized, they become difficult to unpack. This phenomenon is what some people call the pathology of expertise. They primarily fail to include routine steps and they often fail to provide sufficient details for the steps that are performed on intuition.

Cognitive Task Analysis (CTA)

As there have been well over 100 CTA methods described and applied, they have been categorized into a few different classes. One class of methods has been referred to as observation and interviews (Cooke, 1994). These methods are unstructured. They can be highly informative, but they are difficult to make repeatable, and analysis of the interviews and observations takes considerable time and effort. An example of this technique is talk-aloud protocols. A second class is referred to as conceptual techniques. Rather than eliciting information from experts as they solve a problem, experts are asked questions that are intended to elicit the structure of their knowledge. An example of this is a knowledge audit (Crandall, Klein & Hoffman, 2006). A third class is referred to as process tracing. This class generally observes a process (such as a troubleshooting process) and associates data with steps of the process. PARI is an example of this class. While PARI collects information in real time as experts solve problems, related methods involve collecting information from real cases rather than simulated cases or collecting information from experts after a solution process has been completed. We describe our efforts as PARI; we think of PARI as an excellent structure by which to collect and represent all process tracing classes of CTA.

Goals, Operators, Methods, Selections (GOMS) Task Analysis

GOMS is a popular user interface design technique developed by Card, Moran, and Newell (1983) to predictively model a user's interactions with the interface to accomplish specific goals and is another example of a process tracing CTA method. Starting with the specific goal, it describes the operators (interactions) used to accomplish the goal, methods of performing a sequence of operators, and selection rules to determine which method to perform. GOMS assumes system and domain knowledge are not relevant. For example, the user interface of a word processor should not require that the user knows how the software code works to accomplish their task of editing a document. Hall, Gott, and Pokorny (1995) describe GOMS with examples and discuss its limitations and similarities to PARI. Sternberg and Gitomer (1992)

describe how they performed a PARI analysis and then used GOMS to analyze the user interface of HYDRIVE intelligent tutor. Jonassen, Tessmer, and Hannum (1999) discuss how to apply GOMS and PARI for instructional design.

For a discussion of CTA approaches used for training and ITS development, see Clark and Estes (1996). In addition to their discussions, they recommend Gott's (1989) reviews of PARI and other CTA approaches used by "Anderson's LISP tutor, Soloway's PROUST de-bugging tutor, and the technique used by Morris and Rouse (1985) in their study of trouble shooting characteristics" (Clark & Estes, 1996, pp. 5–6).

Fly, Then Review

An approach that follows typical mentor/apprentice methods has the expert perform in a simulation or simulator as they would in real life. A recording of the session is played back and paused at certain events, especially where an action was taken or choice was made. The expert is asked to describe what she was thinking at that point in time, what the alternatives would have been and why the particular choice or action was made. The expert is also asked about tolerances of acceptance for parameters of the behavior such as time. Lesgold and Nahemow (2001) describe this as "stimulated recall." In addition, the expert may record a continuous narrative during the original demonstration or during a playback. The result is a simulation or video that demonstrates what the expert would do along with clear descriptions. From those sessions, a software engineer or developer can create assessment rules for the behaviors and triggers for the key events. The assessment rules are integrated with the simulation and enable the learner to be assessed while performing the same or similar task the expert demonstrated.

We used that method for a couple of prototype adaptive training simulations for pilots. One demonstration was comprised of four rungs. The first rung was a replay with the expert's narrative. The second rung allowed the learner to try the simulation with the expert's narration cued on the key events. The third rung had the learner perform with the assessment rules enabled and without narration except at critical events at which the learner had failed to meet the acceptable tolerances of behavior. The fourth rung replayed a recording of the learner's third rung with the narration enabled.

Simulation-based adaptive training built in this manner demonstrates modes of training commonly known as show me, let me try, test me, and what we call "help me improve." These modes provide graduated higher levels of learning similar to the US Army's Crawl-Walk-Run Training Process (TC 25-10, 1996). The results are memorable learning experiences that are close to real-life apprenticeship. This works well for very complex environments but requires a simulation for the walk and run levels. Without sufficient instructional assistance and robust rules and restrictions captured by the CTA, negative transfer can be likely because the learner can make incorrect assumptions and cause/effect relations during training.

Discussion

The strategy presented in this chapter is to combine task analysis, instructional design, and ITS programming into a streamlined process with automaticity. Our goal is to provide a structured method to perform an analysis of expertise for a domain; organize the results into a format that is understood by experts, novices, and the GIFT software; and automatically create tutors. The first stage uses a CTA approach that has been used to create tutors for a variety of domains from aircraft avionics troubleshooting to weather forecasting. This chapter suggests using a spreadsheet to organize the domain model and a schema for exporting from the analysis software to GIFT.

PARI

PARI was inspired by reviewing think-aloud protocols (the form of protocols used in early cognitive science experiments) of expert technicians solving difficult maintenance problems. The protocols used by those early experiments always included an expert's action and the results of that action. Some of the time the protocol included an interpretation of the results of an action, in terms of what equipment had been removed from suspicions and what equipment sections the expert thought the fault was more likely to be found. Additionally, the protocol sometimes included the reason underlying an answer, though not always. This review led to a structured protocol effort, where the expert's reason, or cognitive precursor, for the action and the interpretation of the result are captured as well. These four types of data captured are called the precursor, action, result, and interpretation terms, which form the acronym PARI and are described in the following sections of this chapter.

Precursor

A precursor is the expert's goal or reason for a particular action. A sequence of precursors describes the overall strategy. For tasks such as troubleshooting, a precursor is often a hypothesis that will be tested by the action. During the interview or post analysis, the analyst might ask the expert "why are you taking this step" or "why did you perform that action?" For example, a typical troubleshooting precursor might be "I need to see if the fault still exists."

Action

The action is a record or description of a particular step taken by the expert to attempt meeting the precursor's goal. The action may include several sub-steps at the appropriate level for the describing the action to a learner or to assess a learner. For example, if the fault is that a car won't start, the first action might be as simple as "Try starting the car." If the level of understanding for the target learners is lower, the Actilon might include more details such as "enter the car, insert the key, and turn the ignition switch all the way clockwise for at most 3 seconds."

Result

The result records the outcome of the action. It usually describes how the domain or system has changed as the result of the action. For the PARI process in the form of a dyad, the result is presented by the analyst or interviewer. To continue the example from above, the result might be "The engine rotates with full power from the starter, but does not start."

Interpretation

The interpretation records the expert's analysis of the results. It contains what the expert learned from the results of the action. It often contains how the expert's mental model has changed. For example, "Since the engine rotated, the battery, ignition switch, and starter motor are functioning properly."

Figure 1 shows an example schema that was developed for structuring the PARI data for use in an ITS. Comments in the schema demonstrate guidance for the nodes. It also depicts the dyadic interview between the analyst providing the symptoms, asking the questions, and providing the results on the left and the expert being interviewed on the right. The top of the schema provides a structure to initialize the system model or provide the learner with the problem statement, the situation, or context and overall goal as well as the expert's initial precursor or interpretation of provided symptoms. The dialogue represented in the figure and the examples above is greatly simplified. An actual PARI session would include more de-

tail for each step. For example, the "results" would capture all the perceptual cues that the interviewer believes would be important. If the interviewee believes there are cues that are missing from the results description, he or she can ask for details such as "were there any indication of ignition during the engine starting attempt and was there an odor of gasoline?" The interviewer may provide the details or, for example, indicate that the detail could not be sensed because of certain conditions.

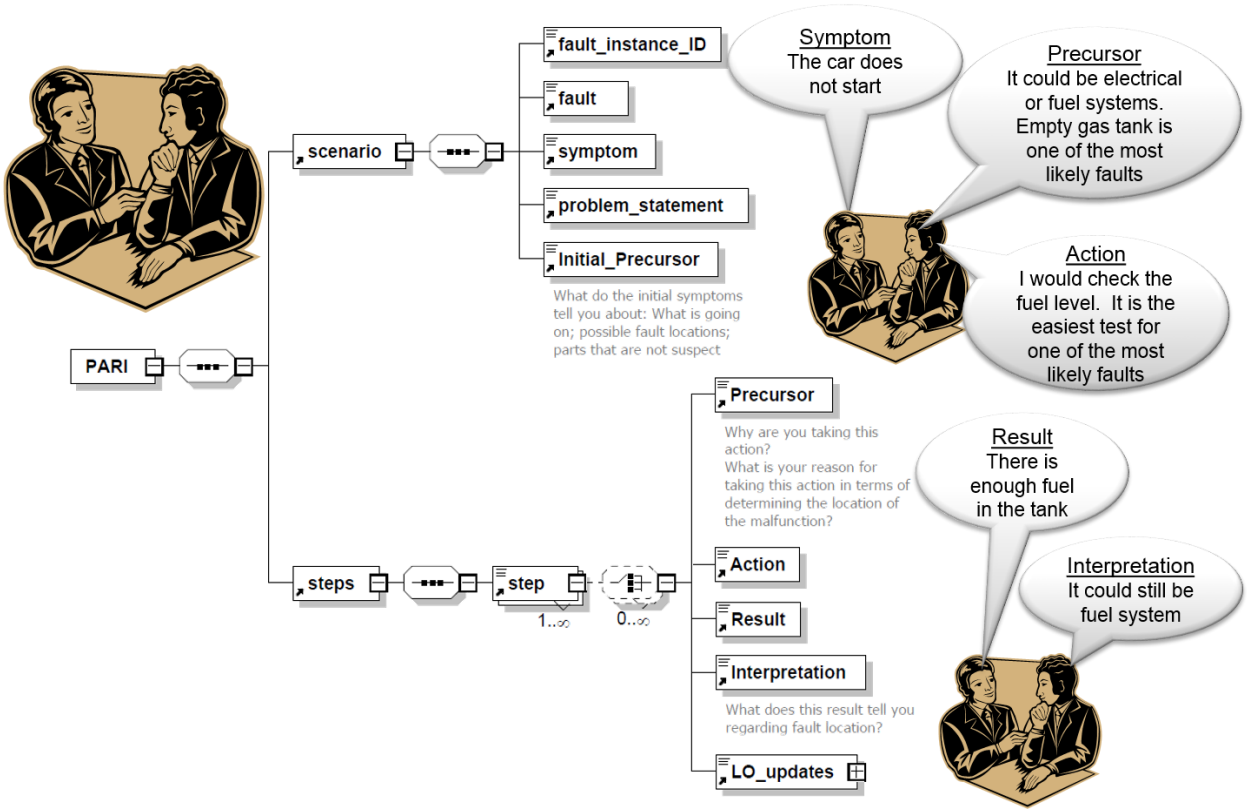


Figure 3 - PARI schema and interview example.

Both GOMS and PARI define steps and the rules a person uses to decide what to do. However, GOMS is focused on understanding the procedures of typical users and the conditions by which the different procedures might be selected expressed as simple if-then rules to accomplish a specified goal. PARI focuses on the components of expertise. The expert's conceptual knowledge; how that knowledge is used to select goals; the strategies used to select actions; and how the conceptual knowledge dynamically changes based on the results. Jonassen, Tessmer, and Hannum (1999) compare and contrast GOMS and PARI for instructional design where they state that GOMS is not designed for high level cognitive tasks such as problem solving; PARI was designed for more conceptual and strategic activities; and "GOMS works only for goal-directed tasks" (p. 118). Hall, Gott, and Pokorny (1995) suggest PARI methodology extends the GOMS approach with "procedural skills… and conceptual support knowledge consisting of system and strategic knowledge structures" (p. 23).

PARI Spreadsheet

The PARI process was developed and refined as a "structured, thinking-aloud dialogue" (Hall, Gott & Pokorny, 1995, p. 1) between two experts. The PARI process leverages the structured interview to elicit the cognitive as well as the procedural elements of the task. In order to record the information in a way

that can be easily interpreted by the ITS engine and provided to the learner, we created an electronic form that the interviewer fills out. When we used the tool to build a basic troubleshooting skills ITS for the US Marine Corps (USMC), the experts wanted to work independently. These experts filled out the form by themselves. Since the form guided the SME through a series of questions, each expanding upon the previous question, the form was, in essence, interviewing the expert. Although it still required multiple iterations and reviews by other SMEs, this technique worked well. The same tool was then used on the project by other experts to create a tactical logistics ITS and a mission planning ITS for the USMC. The PARI spreadsheet was also used on projects for adaptive training for leaders with small unmanned aerial systems (SUAS), as well as prototype adaptive training for factory employees, commercial air pilots, and tactical air pilots. For these ITSs, the single SME approach enabled the expert to develop the scenario storyline at the same time resulting in a continuous, engaging narrative for the scenario.

Hall, Gott, and Pokorny (1995) found that having a dyadic interaction of pairs of experts provided for a much more realistic situation and consequently recorded expert performance that was more authentic. "When [experts] are naive to the problem source, they consider a much wider range of hypotheses and correspondingly search and access richer knowledge structures" (Hall, Gott & Pokorny, 1995, p. 4). This is very important in ill-defined domains. When the domain is well defined, though, we have found that experts can independently fill out the PARI spreadsheet. We have also found that two SMEs could create two scenarios independently in less time than two SMEs working together.

Relevant Attributes of the Domain Model

The content within the PARI domain model contains details on what the expert does as well as what the expert thinks about at steps in the task. It describes the expert's plan and each sub-goal along the path to the end goal. It describes each action the expert performs, what the expert learns from each action, and how the mental model changes with the expert's interpretation of the results observed after each action. These attributes are key points to make salient to the learners as they develop expertise. The PARI domain model also describes how the system changes and how that affects the situation. Thus, it also records the dynamics of the system model that can be used to simulate the system to provide a realistic learning and practice environment. For many domains, this record is just the right level of fidelity to model the "entire" system model simulation used for the ITS.

PARI was used for one of the most-studied ITSs called Sherlock, which showed to be very effective at teaching troubleshooting in a complex, ill-defined domain. A rich hinting mechanism provided the primary mode of individualization for learners in Sherlock (Lesgold, Lajoie, Bunzo & Eggan, 1988, p16). The mechanism provided four categories of hints and five levels of explicitness. The instructional design of Sherlock defined hint categories that aligned with the four types of activity required for the current step: option (precursor), action (action), outcome (result), and conclusion (interpretation). the alignment provides a straightforward path from the PARI analysis to develop hints to help novices think like an expert. Option refers to the activity for deciding what approach should be taken next (precursor). Action refers to knowing what specific decision, test, or action to make (action). Outcome refers to seeing if the action has positive results (results), and conclusion interprets the results (interpretation). The levels of explicitness were
(1) recapitulation of what was done, (2) general target area, (3) specific target area, (4) specific goal, and (5) specific action. When asking for help, a learner was first presented with the first level of hint, which was a review of what the leaner had done so far. More than half of the time, this first-level hint was enough to help a learner when needed. If the learner needed more help, the system would select from the category (option, action, outcome, and conclusion) that is relevant to the specific step and select the appropriate level of explicitness for the individual learner at that step. The purpose of the hints is to help the learner form a mental model similar to the expert's and encourage deep learning, not to provide answers.

For Sherlock and Sherlock II, these categories and levels of hints were crafted by instructional design methods. However, since the outcomes of the PARI map very closely to the hint categories, and can be arranged in order from abstract to explicit, we have found that the hint mechanism can be automated directly from the PARI. The first level of hint is still a recapitulation of the current situation. For any step, the first-level hint could be a reminder of what they learned (the interpretation) from the previous action. For some domains, the interpretation could be broken into two levels of hints – the narrative description and the mental model of the system. The second-level hint would suggest a strategy, which is the precursor for the current action. The third level of hint would suggest the specific action. Since the PARI method is based on a structured, thinking-aloud dialogue, basing the hints on the PARI analysis provides hints that encourage deep learning by using a dialogue of questions rather than answers. Chi et al. (2001, 2008) shows that "a learner in guided-construction learns from being constructive." (Chi, 2009, p15).

For the PARI example in Figure 1, after the learner checked the fuel level, the first-level hint would be "You tested the fuel level. Although there is fuel in the tank, it still could be a fuel problem." Prior to that action, the second-level hint would be "It could be electrical or fuel systems." The third-level hint for the step shown would be "I would check the fuel level, it is the easiest test for one of the most likely faults." To further approximate the learning efficacy of interactive constructive dialogues (Chi 2009), the hint would be "I would check _____, it is the easiest test for one of the most likely faults."

In some of our tutors, we penalized the learner for asking second- and third-level hints. The first-level hint was "free" as it was simply restating what they had already learned. For one ITS, we built for the US Army (Durlach & Dargue, 2010), asking for a third-level hint counted as a "strike" in the student model against the learning objectives associated with the particular step and/or action. If the learner got three strikes against any learning objective, then the ITS would "pause" the scenario and provide remedial information about that learning objective. Penalizing the higher level of hints worked well in that and other studies. However, some instructional designers are philosophically opposed to penalizing learners for asking for help, so we make the penalty mechanism configurable.

Subject Matter Domains and PARI

Subject-matter domains that are well suited for the PARI method consist of a series of decisions such as critical thinking, troubleshooting, tactical logistics, and mission planning and preparing. PARI is also ideal for developing skills for non-normal procedures. PARI also works well in normal procedures, but can be overly time consuming in this case as there is less of a need for capturing a variety of situations and cues that are associated with a specific response. PARI should be used with other methods of capturing expertise or data for some environments and real-time performance tasks. For example, PARI can be used along with an after-action report (AAR) system as a tool to capture the cognitive reasoning and indicate the important perceptual cues during an AAR of a recorded event, simulation, or demonstration. This is because PARI may not capture all the data inputs and analysis used by experts in a dynamic situation, or for perceptual tasks or tactile tasks. PARI captures the narrative description of the perceptual cues and tactile information, but possibly not in enough detail or in the correct format to be used to develop the required perceptual or tactile skills.

Recommendations and Future Research

An ITS built to develop expertise must help the learner form the mental models and intuition of an expert. While they must have the knowledge and skills of an expert, they must also hone the attitudes of an expert. When GIFT or other generalized tutoring systems are used to develop and deliver the KSA of expertise to a learner, it needs to have access to the explicit models of those attitudes, mental models, and intui-

tion. As described by Gott and Pokorny (1987), expert performance requires procedural (or how to-do-it), declarative (what-is-it), and strategic (or how to-decide-what-to-do-and-when) knowledge.

The PARI process records the procedural, declarative, and strategic knowledge as well as dynamic mental models in sufficient detail to transfer that expert knowledge to a novice. It structures those knowledge structures in context of job performance in achieving goals of a task to enable learning by doing. An ITS that uses the PARI information can guide the learner as needed to help facilitate learning that lasts. These qualities of PARI as well as methods to transfer knowledge from (1) the expert to the PARI data, (2) the PARI data to the ITS, and (3) the ITS to the learner can be implemented in a well-defined structured process.

PARI provides a machine-understandable structured format of that information to enable efficient and effective transfer and comparison. In GIFT, or other ITS development tools, our implementation of PARI has a goal of facilitating, and even automating, parts of the CTA and domain modeling tasks along with instructional systems design (ISD) tasks. First, our PARI spreadsheet implementation captures expert performance and their related thoughts and strategies by walking the expert through a series of questions in the context of the skill or procedure of interest. Whereas it is fairly straightforward to capture the expert's solution, our approach's primary value is its ability to codify the expert's goals, their interpretation of the situation, the strategies that they dynamically apply to the specific situation, and reasoning in solving the problem. Because of the way it is structured, our approach also enables the knowledge to be used by an ITS like GIFT with little additional programming to provide very effective adaptive automated tutoring that follows the US Army's Crawl-Walk-Run Training Process.

Conclusions

Sherlock was one of the most successful implementation of an ITS for the military and successfully transitioned to industry (Lesgold, 2001). Much of its success is attributed to the PARI CTA that was used to design it. PARI has been effective in several domains other than troubleshooting. Although designed to be a structured interview between SMEs, the PARI interview can be administered by software that walks the SME through the decision-making process throughout each step of a specific problem. This approach can be likened to software applications such as TurboTax®, which gather the background and contextual information, and walk the user through a series of focused questions. Since the PARI authoring tool "interviews" without the involvement of a human, the risk of introducing bias or an irrelevant cue is minimized through the scripted interview process. SMEs often have difficulty defining the right level of detail, so the PARI spreadsheet approach keeps the SME focused on providing information at the level of detail required. PARI aligns with US Army AAR and can be incorporated to automatically create training from debriefs of real events while everything is fresh in the minds of the interviewees.

Recommendations

Good, consistent knowledge acquisition provides better tutoring. The PARI software can be built for use by the interviewer with the expert or the interviewee with no interviewer (solo mode). Since the PARI process provides a standard means of walking through the steps of a variety of procedures or troubleshooting activities, it can be applied to a variety of domains. If required, it can be tailored for a specific domain, such as having probe questions regarding cues or thought process related to a specific task environment. It is possible to provide interfaces to simulations and simulation records to help the expert use the tool in solo mode. However, caution is recommended to make sure that the perceptual cues that the expert uses in the real-world environment are adequately modeled in the domain (e.g., Munro et al., 2015). Tools can be created to enable the structured output from PARI to be imported to or used directly by GIFT tools and modules such as the domain knowledge file (DKF) and for use by AutoTutor (Nye,

Graesser & Hu 2014). In addition, tools to import mission debriefs and AARs can enable the ITS to train lessons learned quickly after events. Research might also consider defining and building a PARI repository/library that is searchable and allows the scenarios and lessons learned modules to be discoverable and easily imported into training.

References

Anderson, J. R. (1992). Automaticity and the ACT* theory. *The American Journal of Psychology, Vol. 105,* pp. 165–180.

Card, S., Moran, T. P. & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum Publishing

Clark, R. E. and Estes, F. (1996). Cognitive task analysis, *International Journal of Educational Research*. 25(5), 403–417.

Crandall, B., Klein, G. & Hoffman, R. R. (2006). Chapter 5: Incident based CTA: Helping practitioners tell stories. In, *Working Minds: A Practitioner's Guide to Cognitive Task Analysis* (pp. 69–90.). Cambridge, MA: MIT Press.

Durlach, P. & Dargue, B. (2010). An Adaptive Training Prototype for Small Unmanned Aerial System Employment. *Proceedings of The 23rd Florida Artificial Intelligence Research Society Conference (FLAIRS–23)* (pp. 530–531), Menlo Park, CA:AAAI Press.

Dykhuizen, G. (1973). *The Life and Mind of John Dewey*. Chicago: Southern Illinois University Press.

Ericsson, K. A. & Smith, J. (1991). *Towards a general theory of expertise: Prospects and limits*. New York: Cambridge University Press.

Gott, S. P. (1987). Assessing technical expertise in today's work environments. *Proceedings of the 1987 ETS Invitational Conference* (pp. 89–101). Princeton, NJ: Educational Testing Service.

Gott, S. (1989). Apprenticeship Instruction for Real-World Tasks: The Coordination of Procedures, Mental Models, and Strategies. *Review of Research in Education*, 15 (pp. 97–169). Washington D.C.: American Educational Research Association

Gott, S. P., Bennett, W. & Gillet, A. (1986). Models of technical competence for intelligent tutoring systems. *Journal of Computer-Based Instruction. 13(2)*, 43–46.

Gott, S. P. & Pokorny, R. (1987). The training of experts for high-tech work environments. *Proceedings of the Ninth Interservice/Industry Training Systems Conference* (pp. 184–190). Washington, DC: American Defense Preparedness Association.

Hall, E. P., Gott, S. P. & Pokorny, R. A. (1995*). A procedural guide to cognitive task analysis: The PARI methodology* (Technical Report 1995-0108). Brooks AFB TX: Armstrong Laboratory, Human Resources Directorate.

Jonassen, D. H., Tessmer, M. & Hannum, W. H. (1999). *Task Analysis Methods for Instructional Design*. Mahwah, NJ: Lawrence Erlbaum Associates.

Klein, G. (1993). A recognition primed decision (RPD) model of rapid decision making. In G. A. Klein, J., Orasanu, R. Calderwood & C. Zsambok (Eds.), *Decision making in action: Models and methods* (pp. 138–147). Norwood, NJ: Ablex.

Lesgold, A., Lajoie, S., Bunzo, M. & Eggan, G. (1988). SHERLOCK: *A coached practice environment for an electronics troubleshooting job.* Pittsburgh: University of Pittsburgh, Learning Research and Development Center.

Lesgold, A. (2001). The nature and methods of learning by doing. *American Psychologist, 56(11)*, 964–973.

Lesgold, A. & Nahemow, M. (2001). Tools to assist learning by doing: Achieving and assessing efficient technology for learning. In D. Klahr & S. Carver (Eds.), *Cognition and instruction: Twenty-five years of progress* (pp. 307–346). Mahwah, NJ: Erlbaum.

Means, B. & Gott, S. P. (1988). Cognitive task analysis as a basis for tutor development: Articulating abstract knowledge representations. In J. Psotka, D. Massey & S. Mutter (Eds.). *Intelligent tutoring systems: Lessons learned* (pp. 35–47). Hillsdale, NJ: Erlbaum.

Mori, M., MacDorman, K. F. & Kageki, N. (2012). *The uncanny valley [From the field]*. In *IEEE Robotics & Automation Magazine* 19 (2): 98–100. doi: 10.1109/MRA.2012.2192811 or http://spectrum.ieee.org/automaton/robotics/humanoids/the-uncanny-valley

Morris, N. M., and Rouse, W. B. (1985). Review and evaluation of empirical research on troubleshooting. *Human Factors*, 27(5), 503–530.

Munro, A., Patrey, J., Biddle, E. S. & Carroll, M. (2015). Chapter 16: Cognitive aspects of virtual environment design. In, Hale, K.S. & Stanney, K.M (Eds.), *Second Edition; Handbook of Virtual Environments: Design, Implementation and Application* (pp. 391–410). Boca Raton, FL: CRC Press.

Nye, B. D., Graesser, A. C. & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education, 24*, 427–469.

Steinberg, L. S., and Gitomer, D. H. (1992). *Cognitive Task Analysis, Interface Design, and Technical Troubleshooting*. Princeton, N.J.: Educational Testing Service.

HQ TRADOC Deputy Chief of Staff for Operations and Training (1996). *TC 25-10 - A leader's guide to lane training*. Retrieved from https://rdl.train.army.mil/catalog-ws/view/100.ATSC/2F5E29AA-DABF-4BE4-9F6F-2C804BE46EE7-1274436620178/25-10/CH1.htm#s5.

SECTION III

APPLICATIONS OF DOMAIN MODELING

Anne M. Sinatra and Robert A. Sottilare, Eds.

CHAPTER 14 – Exploring the Diversity of Domain Modeling for Training and Educational Applications

Anne M. Sinatra and Robert A. Sottilare
US Army Research Laboratory

Introduction

Domain models represent the knowledge within a task domain in order to facilitate instruction. This knowledge may be visual (e.g., text, graphic) or verbal and represents domain content to be presented to the learner during instruction, feedback provided by the tutor, and other communications initiated by the tutor (e.g., hints, prompts, questions, assertions). The goal of this chapter is to explore assorted domain models as they are applied within intelligent tutoring systems (ITSs) during adaptive instruction. Domains may be defined per the various taxonomies in the literature: cognitive (Bloom, 1956), affective (Krathwohl, Bloom & Masia, 1964), psychomotor (Simpson, 1972), or social (Soller, 2001), but most instructional domains are a mixture of two or more of these taxonomies.

Within the Generalized Intelligent Framework for Tutoring (GIFT), the domain model specifies concepts (also known as learning objectives) and their associated methods of measurement and sources for data to determine whether learners have met the standards for a concept. The GIFT domain model also has data structures that allow the author to specify the tutor actions (e.g., ask the learner a question or change the challenge level of the environment) in response to conditions identified within the environment and the learner. The authored tutor actions are linked to strategies (plans) recommended by the pedagogical model.

As GIFT is largely a domain-independent framework, the authoring of all domain-specific content occurs only in a domain knowledge file (DKF). The GIFT Authoring Tool provides an opportunity to create a DKF which is specifically associated with different media or simulations which can be of varying types (e.g., PowerPoint, Virtual BattleSpace, Virtual Medic, Physics Playground). A screenshot of the DKF authoring process as of GIFT 2015-2X can be seen in Figure 1.

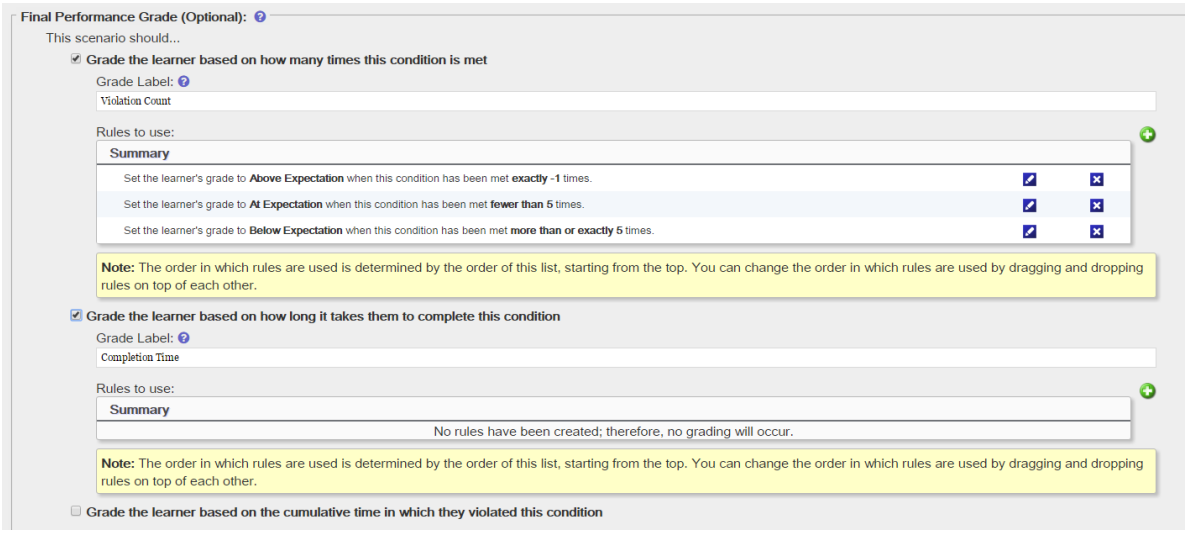


Figure 4. Screenshot of DKF authoring in the GIFT Authoring Tool. Rules have been authored to determine the learner's state based on actions taken by the individual.

Tasks and associate concepts (also known as learning objectives) are defined by the author, as well as what actions should be taken when the state of the individual or the training environment changes. In GIFT, the learner's performance state can be unknown, below expectation, at expectation, or above expectation. In keeping with GIFT's architectural principles, a generalized approach for structuring and modeling domains was specifically selected to support a broad variety of domains and tasks. ITSs that are built to support only a single domain use unique approaches to domain modeling and authoring.

Characterizing Domain Attributes

To apply ITSs to various domains, we must first understand the characteristics of each domain and compare and contrast their primary characteristics. We have chosen to examine task domains in terms of their definition and complexity. Definition represents how well the level of organizing principles in that domain are generally understood with well-defined tasks being highly structured and ill-defined tasks being unstructured. Complexity represents the level of difficulty or intricacy for domain tasks with low complexity tasks represented by simple, concise processes only a few steps in length and high complexity tasks represented by difficult, long processes with many steps and branches.

We chose to examine domain definition and complexity in the context of four primary task taxonomies (**Figure 5**): cognitive, affective, psychomotor, and social. For example, affective tasks, which are well defined and of high complexity, are labeled as *complex structured judgment processes*. Examples are provided for each one of 12 domain types. We have assumed that all ill-defined tasks are of high complexity and have eliminated these domain types from our taxonomy of tutoring domains. The colors (green, yellow, red) in **Figure 5** represent the relative frequency of ITSs built for each domain type. Green domain types are frequently the subject of ITS development. Yellow domain types are less frequent and red domains are under-represented or not represented at all.

	Cognitive mental skills (*knowledge*)	**Affective** emotional growth (*attitude*)	**Psychomotor** physical ability (*skills*)	**Social** cooperative ability (*team skills*)
Well-Defined/ Low Complexity	Simple Structured Mental Processes	Simple Structured Judgment Processes	Simple Structured Physical Processes	Simple Structured Team Processes
	Recalling a list; Decision-making; Problem-solving	Selecting options by preference	Riding a Bicycle; Typing an Email	Brainstorming
Well-Defined/ High Complexity	Complex Structured Mental Processes	Complex Structured Judgment Processes	Complex Structured Physical Processes	Complex Structured Team Processes
	Mathematics; Classical Physics	Managing Emotions; Influencing Others	Playing Music; Marksmanship; Driving a car	Collaborative Problem Solving; Team Sports
Ill-Defined/ High Complexity	Complex Unstructured Mental Processes	Complex Unstructured Judgment Processes	Complex Unstructured Physical Processes	Complex Unstructured Team Processes
	Creative Thinking ; Creative Writing	Moral Reasoning; Leadership	Improvisational Music; Sculpting or Painting	Art of War; Dramatic Improvisation

Figure 5. Taxonomy of tutoring domains.

Summary of Chapters in this Section

The chapters in this section cover a wide range of topics, including overviews of domain modeling, recommendations for domain modeling in applied psychomotor domains, specific examples of psychomotor domain modeling, examples of domain modeling in specific systems, conceptual techniques that can enhance domain modeling, and approaches that could be useful in enhancing learning in multiple domains. The focus and description of the individual chapters included in this section are below.

The chapter by Brusilovsky provides an overview of methods and techniques for personalizing guidance within adaptive educational systems (AESs). The methods reviewed can be applied to varying domains, and different approaches to personalized guidance are discussed (e.g., algorithms, student decisions, etc.). The author notes that domain models are critical, and they may vary based on the types of decisions that are to be made in regard to guidance. The author breaks down the components of domain models to three dimensions, nature, structure, and usage, and provides a review of approaches that have been used for domain modeling. The techniques described in this chapter may be applied across domains represented in **Figure 5**.

The chapter by Sottilare and LaViola describes a process for adaptive instruction of training tasks in the psychomotor domain. The authors recommend closely aligning the training and work environments to encourage the transfer of skills in the real world. The importance of considering both cognitive functions and physical skills in psychomotor tutoring is highlighted, and recommendations of psychomotor measures are provided. The learning effect model is discussed in relationship to psychomotor instruction, and a process model of adaptive training of psychomotor tasks is presented. The described techniques may be applied across psychomotor tasks and are intended to maximize transfer.

The chapter by Goldberg and Amburn specifically highlights a psychomotor domain use case of an adaptive marksmanship task. Their described task has been implemented using GIFT, and the chapter details different considerations and decisions that were made in the design of this specific ITS. The methodology presented is specific to the task of adaptive marksmanship, but may transfer to other psychomotor tasks. Further, this chapter highlights the adaptability of GIFT to different domains and modes of input, as well as provides an example of a GIFT tutor that has been developed. The chapter concludes with future recommendations for additional GIFT features that would enhance the authoring of psychomotor tutoring in GIFT. The described techniques may be applied to psychomotor tasks, and particularly focus on the well-defined, highly complex task of marksmanship.

The chapter by Cai, Graesser, and Hu begins with a review of ways that domain modeling have been defined and then describes the specific domain modeling process within AutoTutor, a dialogue-based ITS. The different ways that AutoTutor has been used are described, and the authoring of feedback, prompts, hints, and trialogs are discussed. While the described domain model and authoring approaches are actions specific to AutoTutor, these methods may be applied across the range of domains supported by AutoTutor and provide an example of an implemented domain modeling approach. This chapter additionally includes suggestions of ways that GIFT and AutoTutor could be integrated, and how lessons learned from AutoTutor can be leveraged to improve GIFT. The described techniques may be applied across domain tasks, but may be relevant to social domains due to AutoTutor's dialogue-based nature.

A more specific approach to domain modeling is outlined in the chapter by Goldstone, Weitnauer, Ottmar, Marghetis and Landy. This chapter describes applying a theoretical and conceptual approach in the defined domain of algebra. The authors outline their work which has focused on perceptual processing as it relates to mathematical reasoning. They discuss their hypothesis, called Rigged Up Perception-Action Systems (RUPAS), which involves the conversion of high level cognitive tasks to automatically processed perceptions and actions. They propose that training our perceptual processes to efficiently pro-

cess stimuli will lead to better learning outcomes. The authors describe their software, Graspable Math, which is designed to allow students to interact with equations using their proposed perception-action processes. The concepts and approaches that are presented in this chapter highlight the power of using the perceptual system to enhance training and reduce errors, which is applicable to a number of additional domains. The described techniques are highly relevant to well-defined cognitive tasks such as problem solving and mathematics.

Finally, the chapter by Forbus outlines sketch capabilities for ITSs in order to promote spatial learning. The chapter discusses the advantages of using spatial learning and points out that this type of learning is important to multiple domains including mechanics and chemistry. The author discusses how incorporating sketching into ITSs could have a positive impact on learning outcomes, and the different approaches to sketch understanding/recognition. In particular the elements of CogSketch are described, and examples are provided on how it has been leveraged to provide adaptive feedback in educational contexts as well as how sketching has been implemented at multiple universities. The chapter additionally provides examples of how to leverage CogSketch and its functionalities in an ITS for learning, which can be applied in multiple different domains. The described techniques are highly relevant to well-defined cognitive domains, and domains that can be represented spatially.

Recommendations and Future Research

The chapters in this section cover a wide range of topics in domain modeling. While approaches to domain modeling in algebra and marksmanship were specifically described, there are many other domains that may require specific considerations or content in their domains models. The methodology and techniques recommended in this section can be leveraged to support the creation of domain modeling authoring tools and assist in making decisions about what elements are important in domains other than the ones that were specifically covered. As a largely domain-independent architecture, decisions about GIFT's domain model require careful consideration. Both theory and practical tools/input devices such as sensors have been highlighted in the chapters of this section as being important elements to assess in certain domains. Future research can examine flexible approaches to authoring feedback for multiple domains and specific adaptations that are necessary for more active psychomotor domains.

References

Bloom, B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co. Inc.

Krathwohl, D.R., Bloom, B.S. & Masia, B.B. (1964). *Taxonomy of Educational Objectives: Handbook II: Affective Domain*. New York: David McKay Co.

Simpson, E. (1972). *The classification of educational objectives in the psychomotor domain: The psychomotor domain.* Vol. 3. Washington, DC: Gryphon House.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, 12(1), 40–62.

CHAPTER 15 – Domain Modeling for Personalized Guidance

Peter Brusilovsky
University of Pittsburgh

Introduction: Defining the Problem of Personalized Guidance

This chapter attempts to untangle the relationships between personalized guidance and domain modeling, as well as explain how domain modeling could be used to provide personalized guidance. The problem of personalized guidance has a long history in the area of adaptive educational systems (AESs). In fact, the very first recognized AES SCHOLAR (Carbonell, 1970) focused on guiding students to the most relevant facts and questions about the geography of South America. The SCHOLAR functionality was based on a domain model in the form of a semantic network and an overlay student model. Since that time, a considerable share of research in the field of AESs has focused on different kinds of personalized guidance, and the majority of this work relied heavily on domain modeling—which makes these two research directions heavily interconnected.

In this chapter, personalized guidance is defined as any approach used to guide learners to the most appropriate learning content[1] that accounts for any unique features of the learner, including current knowledge level, interest, and motivation, among others. Selecting the learning activity to work on is known as the *outer loop* problem in the field of AESs (VanLehn, 2006) (with the inner loop referring to assisting students *inside* the selected activity). The goal of personalized guidance in the outer loop is to provide the best educational experience for every learner by guiding the learner through the optimal sequence of learning activities that lead to the best educational outcome, such as faster learning, more reliable knowledge, or a more enjoyable process.

Early attempts to create an adaptive learning process were made in the field of classic computer-adaptive instruction (CAI) where different answers on a multiple-choice test could lead to transferring the learner to different content elements. However, this approach has not been considered to be truly adaptive, since decisions were made on the basis of the last answer, rather than on the full picture of student knowledge. To do better—namely, to make decisions on the whole state of a student's knowledge—a personalized system needs more advanced infrastructure. It needs to understand how student knowledge can be represented, how student interactions with learning content could be used to update this knowledge, how missing knowledge could be identified, and how it would be possible to learn content that can help with filling a particular knowledge gap. As demonstrated by many AESs starting from SCHOLAR, a domain model can assist with performing these functions.

The scope of domain modeling is broader than the needs of personalized guidance. A domain model in the AES field is traditionally understood to be any formal representation of knowledge about the domain. This model can serve various needs, not just personalized guidance. However, due to the special needs of personalized guidance, domain models used for personalized guidance have some common features, which enable different types of personalization, some of which are reviewed in this chapter. To examine the specific needs of personalized guidance, we need to understand how educational content is organized and which decisions have to be made to guide students to the most appropriate elements of this content.

[1] In a more general sense, we should talk about learning as a sequence of learning activities or tasks, not just content; however, in modern e-learning, these activities are triggered by interacting with specific types of content—problems, examples, animations, collaborative tasks, and so forth.

Early AES studies focused on a relatively narrow domain (usually a small part of a course) and have considerably low volumes of content (Brusilovsky, 1992; McArthur, Stasz, Hotta, Peter & Burdorf, 1988; Wescourt, Beard & Gould, 1977). In these AES studies, content has no additional structure; in other words, it was an unstructured pool (usually a pool or problems), and the goal of adaptive sequencing, which is the oldest kind of personal guidance, was to select the best item from this pool to offer to the student. Nowadays, however, many educational systems cover broader domains (such as whole-semester or year-long courses) and have much larger volumes of content. To better manage this content, the domain is usually subdivided into several *topics* (sometimes also called units), and every learning content item is assigned to one of these topics. The topics could be relatively independent from each other, as in an *inq-its.org* learning environment (Sao Pedro, Baker & Gobert, 2013); form a fixed pedagogical sequence, as in Deep Tutor (Rus et al., 2013); or form a partial order with some topics depending on some other earlier topics, as in QuizPACK (Sosnovsky & Brusilovsky, 2015). In addition, modern systems routinely operate with different kinds of learning content, such as book-style presentations, video fragments, animated examples, simple questions, and problems, among others (Figure 6). Some good examples of this two-level "topic-content" organization are college courses with lectures structured by topics, books organized into chapters, and folder structures in learning management systems (LMSs). Since this organization is most typical, we use it to discuss decisions that have to be made by personalized guidance in this general context. While content organization might be more complex in some cases (such as a hierarchy of topics, book subsections, or LMS subfolders), a two-level organization of "topics-content" is still sufficient to analyze key decisions. On the other hand, it is easy to see that earlier simpler organization with a single pool and one type of learning content is just a special simple case of the two-level organization that includes only a single topic.

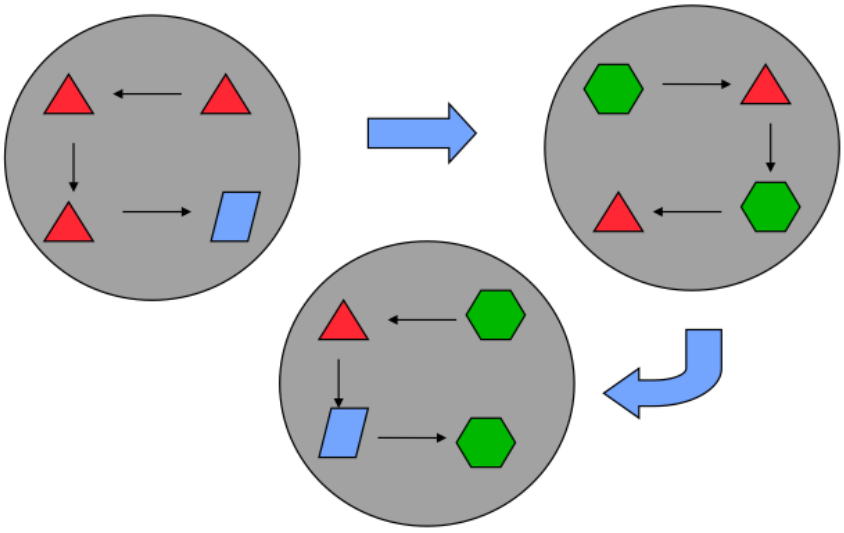


Figure 6. Two level "topic-content" organization of learning content in modern educational systems.

Within this two-level organization, the decisions about continuing the educational process must usually be made on two levels. On each level, someone should decide (1) what to do next and (2) when to stop. On the course level, it means selecting the next best topic to study and deciding when the learning process is over (everything has been learned). On the topic level, it means selecting the next content item and deciding when the topic has been mastered. Each of these decisions could be made adaptively with an understanding of the whole picture of student knowledge, but this is usually not the case even in AESs. In

general, there are four distinctive alternatives for making each of the four decisions (two decisions on each of two levels):

- The decisions could be made in advance by the system developer or instructor. A classic example of it is programmed learning. A more current example of this model is the developer-fixed order of topics and tasks in many intelligent tutoring systems (ITSs).

- The decisions could be left for students to make, usually by selecting a link to the desired topic or content. Hypertext-based learning introduced as an alternative to programmed learning (Hammond, 1989) provides a classic example of this approach, while modern LMSs and learning content repositories (Cafolla, 2006) that provide unrestricted access to all learning content offer a modern example.

- An adaptive algorithm could make the decision for the student. This approach was introduced in the very first AES SCHOLAR (Carbonell, 1970) and has become one of the signature technologies for AESs and ITSs in the form of content or topic sequencing, as mentioned above (Brusilovsky, 1992; McArthur et al., 1988; Wescourt et al., 1977), mastery learning (Corbett & Anderson, 1995), and adaptive course generation (Diessel, Lehmann & Vassileva, 1994).

- An adaptive system can *assist* the student in making the decision. The classic example of this combined approach is adaptive navigation support (Brusilovsky, 2007) in educational adaptive hypermedia, which used intelligent techniques to annotate, re-order, or remove links to content and topics, while ultimately leaving the final choice to the student. A more recent example is provided by educational recommender systems (Manouselis, Drachsler, Verbert & Duval, 2013) that generate a ranked list of the most useful learning resources for the student to choose from.

The last two cases present two different approaches to *personalized* guidance. To be considered as adaptive, an educational system should use personalized guidance to make at least one of the four decisions; however, the remaining decisions could still be left to the instructor or students. Historically, researchers have explored almost the whole range of possible combinations of fixed, free, and guided choices on two levels. For example, the well-known *mastery learning* approach (Corbett & Anderson, 1995) popular in ITSs usually has a fixed order of topics, as well as a fixed order of problems within a topic; however, a decision on when to stop working on a topic is adaptively made on the basis of student knowledge modeling. Topic-based adaptive navigation support approaches (Sosnovsky & Brusilovsky, 2015) provide personalized guidance to help students in selecting a topic to work on, while leaving the choice of content items within topic to the students. In contrast, a system like Problets (Kumar, 2006) can leave the choice of topic to work on to the student, but once the topic is selected, can provide an adaptive sequence of exercises. An adaptive testing system like SIETTE (Conejo, Guzman & Millán, 2004) can leave the choice of topics to test to the instructor while simultaneously generating an adaptive sequence of questions within these topics. Finally, even if an adaptive system offers personalized guidance on both levels, it might use different approaches on different levels. For example, the INSPIRE system (Papanikolaou, Grigoriadou, Kornilakis & Magoulas, 2003) uses knowledge-based personalization to provide personalized guidance on the topic level while learning style-based personalization for content selection within a topic. In most of the cases mentioned above, the use of domain models is critical for providing personalized guidance; however, depending on the level and the kind of decisions to be made, AESs could use considerably different domain models and personalization approaches. The next section reviews several types of domain models that are used for personalized guidance, and explains how student models and content models should be built on the top of these models to support different kinds of guidance.

Related Research: Domain, Student, and Content Modeling

Domain Modeling

Domain models can be classified in several different ways. In this chapter, we follow Sleeman (1985), who suggested classifying models by the nature and form of information contained in the model, as well as the methods of working with it. Following this suggestion, we analyze domain models along three dimensions: what is being modeled (nature), how this information is represented (structure), and how different kinds of models are used for personalized guidance (usage).

The *nature* of domain models is primarily defined by the kind of domain knowledge (which is sometimes also called expert knowledge) represented in the model, which, in turn, might depend on the type of personalization that these models have to support. The majority of AESs have focused on representing two types of domain knowledge: conceptual knowledge (or *about* knowledge) and procedural knowledge (or *how* knowledge). Both kinds of knowledge have been used since the early days of AESs. For example, SCHOLAR (Carbonell, 1970) represented conceptual knowledge about South America in the form of a network of concepts, while Lisp-Tutor (Anderson, Farrell & Sauers, 1984) represented procedural problem-solving knowledge of LISP as a set of problem-solving rules. More recently, Stellan Ohlsson has suggested modeling a different kind of procedural knowledge: not the knowledge that allows the user to solve the problem, but the knowledge that allows the user to evaluate the correctness of the solution (Ohlsson, 1992). This knowledge is typically represented as a set of constraints as in SQL-Tutor (Mitrovic, 2003).

The nature of knowledge represented in a specific AES is usually determined by the kind of personalized support that it provides. AESs that focus on helping users to solve educational problems usually rely on procedural problem-solving knowledge. Systems that focus on assessing student problem solutions prefer to use constraints. AESs that focus on presenting declarative educational content (such as textbook or manual pages) and assessing knowledge of them usually rely on conceptual knowledge about the domain.

In addition, the nature of represented knowledge depends on the *granularity* of representation. The majority of domain models represent domain knowledge in a structural form. A structured domain model is composed of a set of domain knowledge components (KCs). Each KC represents a fragment of knowledge for the given domain. KCs could represent knowledge fragments of different types and granularity; see Koedinger, Corbett, and Perfetti (2012) for an excellent review of various KC types. The type of knowledge represented by a KC is defined by the nature of the domain model (conceptual, problem-solving, evaluation), while the granularity is typically defined by domain experts who decide the level of detail that a domain knowledge representation should have. In different systems, the same kind of knowledge could be represented at different levels of granularity. For example, the KC of procedural knowledge could represent fine-grained rules or coarse-grained competencies, while the KC of conceptual knowledge can represent fine-grained concepts or coarse-grained topics. As with many other design decisions, the granularity of knowledge representation depends on the intended use for such domain models. In particular, advanced problem-solving feedback that is usually critical to support problem solving usually requires finer-grained representation; in contrast, many kinds of personalized guidance can work successfully with relatively coarse-grained models.

By their *structure*, domain models may be divided into *set models* and *network models*. Set models, also called *vector models* (Brusilovsky, 2003), are formed by sets of independent KCs that have no internal structure. In more advanced network domain models, KCs are connected to each other by different kinds of relationships that form a semantic network. This network represents the structure of the domain that is covered by an AES. The most popular kinds of links in AESs are *prerequisite links* between the KCs. A

prerequisite link represents the fact that one of the related KCs has to be learned before another. Prerequisite links can support several personalized guidance approaches. In many AES systems, prerequisite links are the only kind of links between KCs (Davidovic, Warren & Trichina, 2003; Farrell et al., 2003; Henze & Nejdl, 2001; Papanikolaou et al., 2003). For example, the QuizGuide domain model is a network of Java programming topics that are connected by prerequisite links (Figure 7a). Other types of popular links are the classic ontological links "is-a" and "part-of" (Brusilovsky & Cooper, 2002; De Bra, Aerts & Rousseau, 2002; Hoog et al., 2002; Sangineto, Capuano, Gaeta & Micarelli, 2008; Steinacker et al., 2001; Trella, Conejo & Bueno, 2002; Vassileva, 1998). An example of the ontological part-of concept hierarchy for the SH-60 helicopter IETM is shown in Figure 7b, and an example of is-a taxonomy of educational objectives and concepts for a Java problet (Kumar, 2006) is shown in Figure 8. The popularity of these links have gradually increased, following the progress of Semantic Web research and the increased use of more formal ontologies as domain models (Dagger, Wade & Conlan, 2004; Mitrovic & Devedzic, 2004; Sangineto et al., 2008; Sosnovsky & Dicheva, 2010; Trausan-Matu, Maraschi & Cerri, 2002).

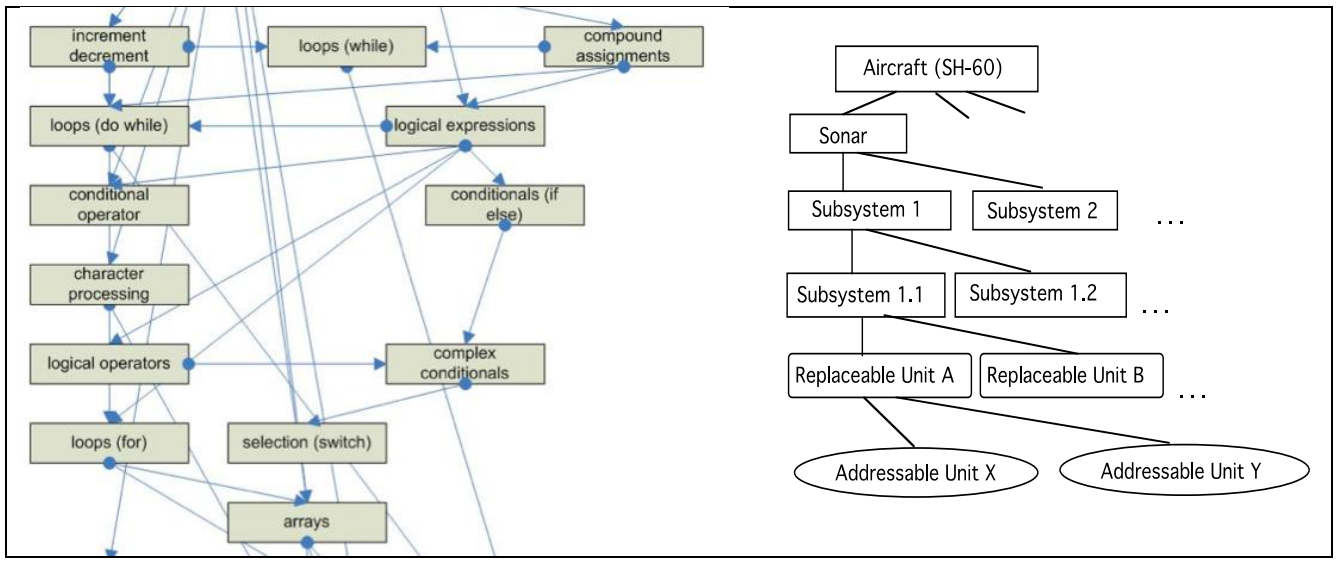


Figure 7. Network domain models: (a) A fragment of domain model for C programming with prerequisite links (Sosnovsky & Brusilovsky, 2015); (b) Domain model for SH-60 IETM with part-of links (Brusilovsky & Cooper, 2002).

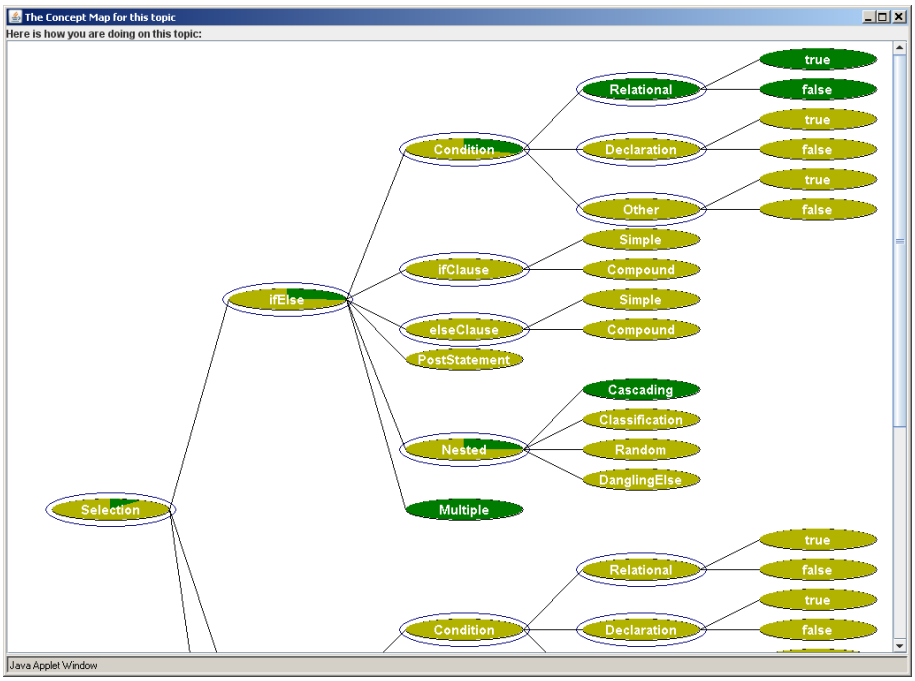


Figure 8. A part of the domain hierarchy for practicing *if/if-else* statements in Java (Kumar, 2006). The hierarchy is formed by learning objectives and concepts, connected by is-a relationships.

Domain Models for Student and Content Modeling

Structured domain models are critical for personalized guidance, because they provide a bridge between individual student models that represent the current state of knowledge of a particular learner and content that could be recommended for that learner. In other words, both the knowledge (or interest) of each learner and the units of learning content are described in terms of the domain model. The presence of this "common language" allows for tracing a specific lack of knowledge or presence of misconceptions to the learning content that could directly address this problem.

One of the most important functions of the domain model is to provide a framework for long-term modeling and representation of the learner's domain knowledge. Long-term modeling is performed by following the student in the learning process, observing her actions and learning progress, and using these observations to maintain a dynamic model of student knowledge about the domain.

The majority of AESs use an overlay model of student domain knowledge, which is also known as an overlay *student model* (Brusilovsky & Millán, 2007). The key principle of the overlay model is that for each domain model KC, an individual user knowledge model stores some data that is an estimation of the user knowledge level on this KC. In its simplest form, it is a binary value (known—not known) that enables the model to represent a user's knowledge as an overlay of domain knowledge. This form was popular in early ITSs; for example, it was the type of model used by the SCHOLAR system to store student knowledge about South America. Today, almost all adaptive systems use a *weighted* overlay model that can distinguish several levels of knowledge for each KC by using a qualitative value (Grigoriadou, Papanikolaou, Kornilakis & Magoulas, 2001) (i.e., good-average-poor), an integer quantitative value (for example, from 0 to 100) (Brusilovsky, Eklund & Schwarz, 1998; De Bra & Calvi, 1998), or a probability that the user knows the concept (Corbett & Anderson, 1995; Conati, Gertner & Vanlehn,, 2002; Henze & Nejdl, 2001). A weighted overlay model of user knowledge can be represented as a set of "concept–

value" pairs, one pair for each KC. The overlay model is powerful and flexible because it can independently measure the learner's knowledge of different KCs.

The domain model also provides a basis for describing the learning content in terms of domain knowledge. To enable personalized guidance, every fragment of learning content should be connected with one or more domain KCs that are presented or assessed by that fragment. These connections define the organization of learning content and its relationships to the domain knowledge. From the prospect of personalization, there are two principal approaches to structure content in respect to the domain model, which are usually called topic-based and concept-based organization.

Topic-based organization assumes that each fragment of educational content is related to one and only one domain model KC. In most cases, this is a relatively coarse-grained KC that corresponds to a relatively large course topic (which is why it is usually called topic-based). Topic-based organization is simpler and more intuitive for both authors and users. It is also easier from the prospect of student modeling: both successes and failures when working with content can be clearly attributed to a single connected topic. At the same time, this model is relatively weak from the prospect of personalization, as will be explained below. A simple example of topic-based organization is the adaptive testing system SIETTE (Conejo et al., 2004), where the domain model is represented as a hierarchy of topics and the content space is formed by a large set of questions. Each SIETTE question here is associated with exactly one of the topics (Figure 9). Problets (Kumar, 2006) offer a similar example of topic-based organization: its KCs form a hierarchy (see Figure 8) and each problem offered by the system is associated with exactly one lower-level educational objective. QuizGuide (Sosnovsky & Brusilovsky, 2015) with its prerequisite-based network of topics (see Figure 7a) also uses topic-based organization, where each domain topic has several quizzes. Due to its transparency, topic-based content organization has been used as an organizing principle in several authoring frameworks for personalized systems (Specht & Oppermann, 1998; Vassileva & Deters, 1998; Weber, Kuhl & Weibelzahl, 2002).

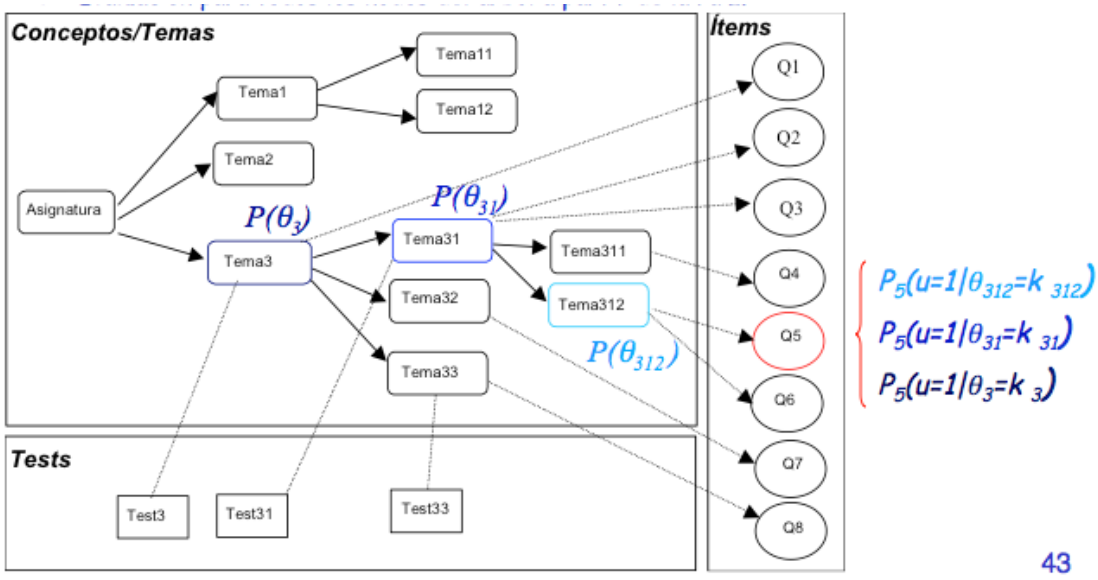


Figure 9. Content organization in SIETTE. Each question is associated to one of the topics in the hierarchy of topics.

Concept-based organization is a more general case and is used when each fragment of content is related to several KCs. In most cases, these are finer-grained KCs that are frequently referred as "concepts".

Concept-based organization is more powerful from the prospect of personalization, but it is more difficult to visualize. It also makes the process of content connection to the domain model more complex and requires a more experienced authoring team. In many cases, however, the nature of the domain demands concept-based content organization. For example, in programming and mathematics, elementary constructs and operators are often selected as domain model concepts or rules. As a result, almost any meaningful problem or example is associated with several KCs. Even in a relatively small domain, such as the Simplex algorithm (Millán, Loboda & Pérez de la Cruz, 2010), this organization might create a complex domain structure, such as the network of skills and problems shown in Figure 10.

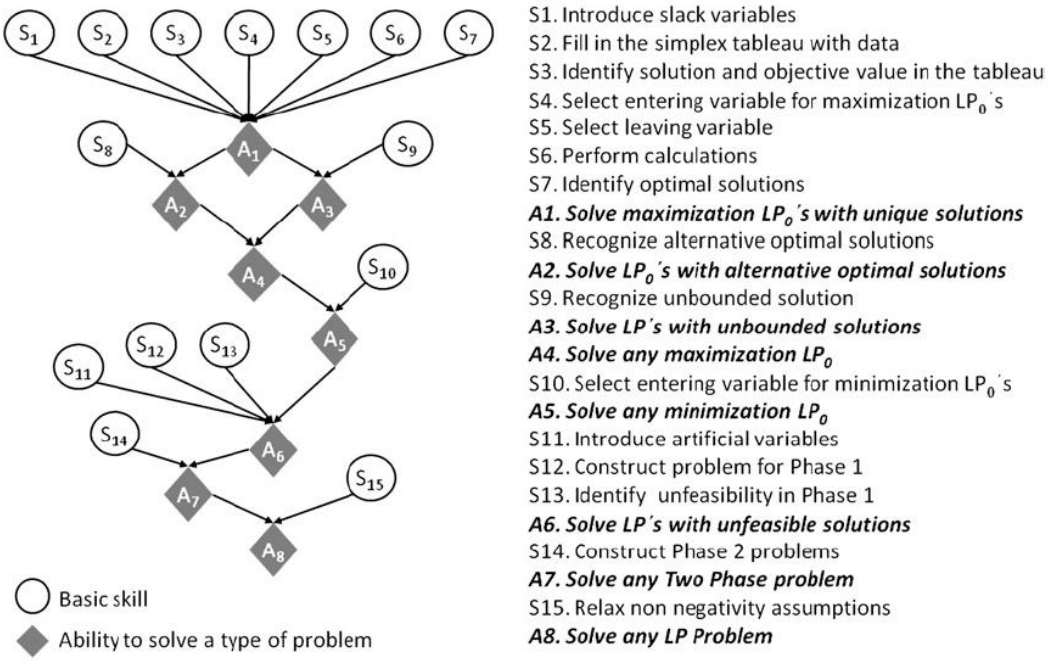


Figure 10. Domain model and content organization in an ITS for the Simplex algorithm (Millán et al., 2010).

From the perspective of student modeling, a concept-based approach presents the problem of *blame allocation*: in case of student failure to solve a problem that is connected to several KCs, as in Figure 10, a student modeling approach needs to decide how to attribute this problem to the lack of knowledge in different associated KCs. To resolve this problem, ITSs use step-by-step problem-solving support with model tracing (Anderson, Boyle, Corbett & Lewis, 1990), where student performance on each step is attributed to a single domain model KC. If step-by-step tracing is not possible, reliable modeling might require complex Bayesian networks (Millán et al., 2010) for blame allocation, or apply advanced solution analysis that can reliably recognize correctly and incorrectly used KCs in student solutions (Weber & Brusilovsky, 2001). An example of a problem-based system with concept-based indexing is SQL-Tutor (Mitrovic, 2003), where each structured query language (SQL) problem is associated with multiple domain *constraints*. Advanced diagnostic capabilities allow this system to recognize which constraints were satisfied and which were not satisfied in each student solution.

Some systems with concept-based content organization and a large content space use different types of links to connect a fragment of content with domain concepts. For example, the adaptive hypermedia system InterBook (Brusilovsky et al., 1998) distinguishes *outcome* concepts for a content page (those explained by this page) and *background* concepts (those not presented on a page, but that are required for

understanding it). This organization supports both complex hyperspace and several types of personalized guidance (Figure 11).

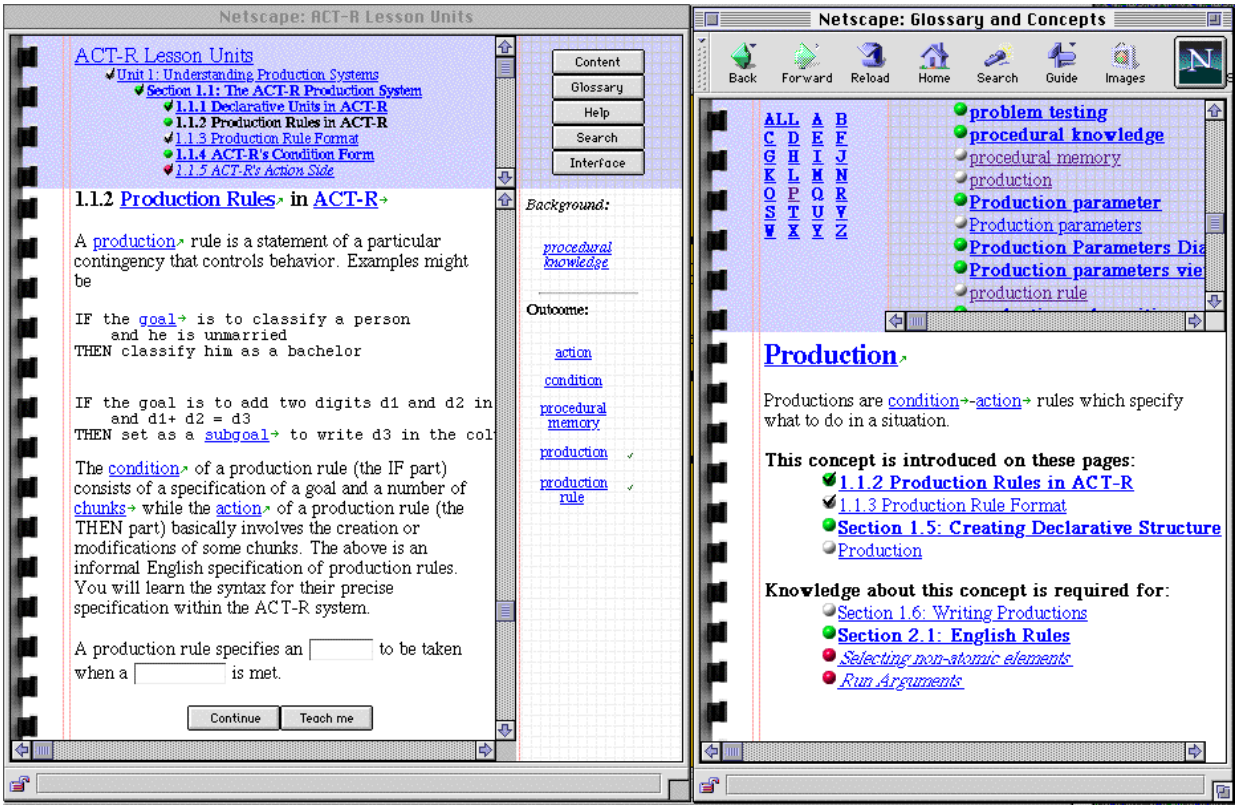


Figure 11. InterBook content organization: Each content page (left) is connected to several domain concepts that could be either background or outcome concepts. Each concept (right) is connected to multiple content pages as outcome or background.

Personalized Guidance Based on Domain Models

As explained in the introduction, personalized guidance in a general case can be provided on two levels: helping students to select the most appropriate course topic and the next educational content fragment that will engage the user into the most appropriate activity. It has been also mentioned that on each level the decision could be prescribed by the instructor or course author, left to the student, or offered by the personalized guidance mechanism with or without student input. The ability to support personalization on each level and the kind of personalization support that the system can provide, to a large extent, is determined by the nature of the domain model and content organization in the system. This section briefly reviews most typical cases of personalized guidance and explains how the domain models could support it.

Topic-Based Personalized Guidance

Topic-based organization of a course is frequently used as a basis for domain modeling. In other words, a topic in the domain model corresponds to a course topic. This could be considered to be a relatively simple and coarse-grained domain model; however, it could be still used to guide users to the most appropriate topic. An example of topic-based guidance is provided by the QuizGuide system (Sosnovsky & Brusilovsky, 2015). The QuizGuide domain model is a network of course topics that are connected by

prerequisite relationships (see Figure 7a). For each topic, the overlay student model stores the achieved level of student knowledge. C programming quizzes form the content, with each quiz belonging to one of the topics. QuizGuide provides personalized topic-level guidance by annotation-based adaptive navigation support. Each topic is annotated with a "target" icon that expresses two important aspects. The color of the target shows how timely the work on this topic is in the current course context (Figure 12). Recommended topics are bright blue, their immediate prerequisites are light blue, older topics are grey, and forthcoming topics for which the student might not yet be ready are crossed out. The number of arrows in the target shows the estimated level of knowledge: no arrows mean little to no knowledge, while three arrows indicate good knowledge. These icons could be easily generated using the current state of the student model and the topic-based domain model. This approach does not explicitly tell the student what to do next, but provides personalized guidance for selecting the right topic, depending on student intentions. For example, a student who is interested in advancing through the course should focus on current but not yet learned topics, while a student who wants to prepare for an exam could work with content from current and immediate past topics that are not yet fully mastered. While this approach is relatively easy to organize, it is surprisingly efficient (Sosnovsky & Brusilovsky, 2015). Simpler versions of topic-based guidance are now used in several practical systems, such as Khan Academy (https://www.khanacademy.org/).

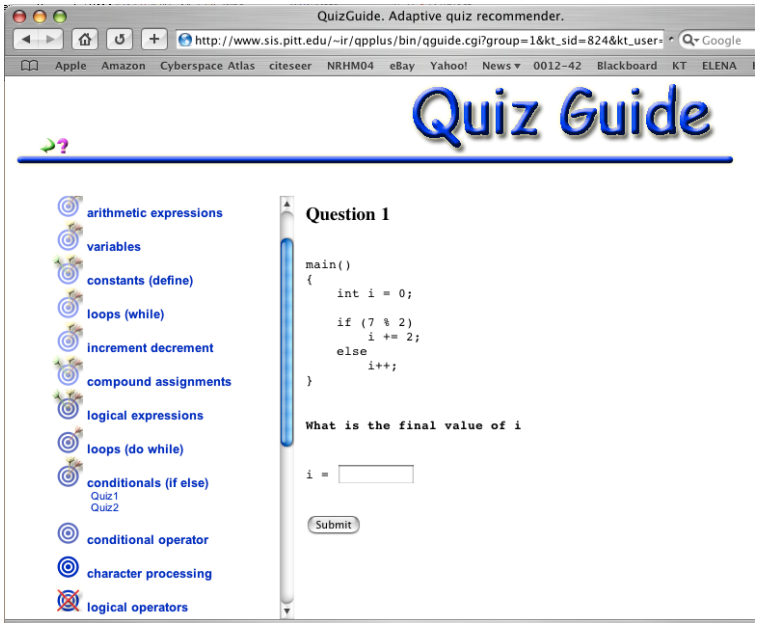


Figure 12. Topic-based adaptive navigation support in QuizGuide with dynamic "target" icons.

The problem of coarse-grained, topic-based guidance is its weakness in guiding users *within* a topic. The first reason is that the quality of student modeling provided by traditional modeling approaches, such as Bayesian knowledge tracing (Corbett & Anderson, 1995) over coarse-grained topics is relatively low, as shown in Sosnovsky and Brusilovsky (2015). It is not critical for navigation support when students are engaged in the decision of what to do next, but is important for all cases where a decision is made for the student. In particular, mastery learning over coarse topics (deciding when the topic is sufficiently learned in order to stop working with in-topic content) is technically feasible, given that knowledge modeling is done on the topic level, but might not be very reliable. The second reason is the inability to apply a topic-level model of student knowledge to distinguish and recommend a specific content within a topic. Indeed, from the prospects of topic-based content organization, all content items that belong to the same topic are equal: they allow users to practice the same knowledge and are equally ready to be applied. As a result,

systems with coarse-grain topic-based models have to use additional information to guide users to a specific content within the selected topic. On the modeling side, it calls for more advanced approaches, such as FAST, that could use various additional features (González-Brenes, Huang & Brusilovsky, 2014). On the navigation support and sequencing side, these models demand richer content-based or usage-based knowledge about every piece of content. For example, INSPIRE (Papanikolaou et al., 2003) uses coarse-grained, topic-based navigation support to guide users to the most appropriate topics. The support is provided in the form of adaptive annotation in the form of an empty, full, or partially filled glass next to the topic link, and reflects the current level of knowledge on the topic (Figure 13). In addition, it classifies fragments that belong to the same topic into different content types and levels and uses information about the current level of knowledge on the topic and the student's learning style to offer the most relevant content for the selected topic.

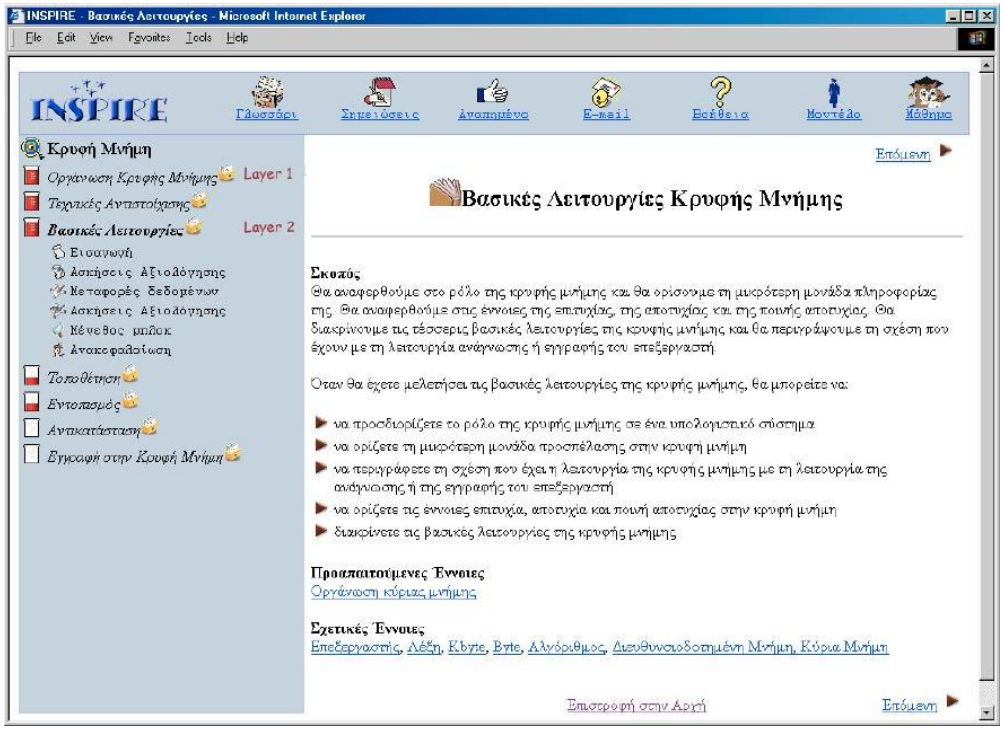


Figure 13. Topic-based content organization in INSPIRE (Papanikolaou et al., 2003). Each concept (annotated with a glass icon on the right) includes a number of content fragments of different types that are selected for the user on the basis of current topic knowledge and learning style.

Systems with finer-grained, topic-based guidance usually have their topics organized in a topic-subtopic hierarchy, just like domain models in Problets (see Figure 8) and SIETTE (see Figure 9). In this organization, the upper levels of this hierarchy typically correspond to the main course topics, while lower levels define a tree of subtopics within topics in a course. This finer-grain organization supports some reasonable approaches to content sequencing within topics. Since student knowledge is now independently tracked for each low-level topic, sequencing algorithms can distinguish content items associated with different subtopics and may have a reason to prefer one or another, depending on the state of the student model. Even reasonably simple sequencing algorithms based on ensuring subtopic coverage (Kumar, 2006) can now provide useful personalized guidance. More sophisticated IRT-based approaches, like the one used in SIETTE (Conejo et al., 2004) can produce highly personalized sequences of content items. Similarly, a finer-grained model could serve as a basis for more precise mastery learning. The weak side of finer-grained, topic-based approaches is the complexity of the domain model, which makes it harder to

visualize it as a whole to offer adaptive navigation support (in topic-based organization, each content fragment is related to exactly one topic; as a result, navigation support has to be organized on the level of the topics). While coarse-grained flat domain models, such as those used in INSPIRE (Papanikolaou et al., 2003), QuizGuide (Sosnovsky & Brusilovsky, 2015), and MasteryGrids (Loboda, Guerra, Hosseini & Brusilovsky, 2014) have straightforward linear visualization, a deep fine-grained hierarchy is usually too large to visualize. One solution to this problem is to visualize only a fragment of the whole model for each top-level topic, as shown in Figure 8, or use a zoomable hierarchical domain model to provide navigation support, such as a zoomable treemap used in the KnowledgeZoom system (Brusilovsky, Baishya, Hosseini, Guerra & Liang, 2013).

Concept-Based Personalized Guidance

Concept-based personalized guidance must work on the top of a more sophisticated content organization, where every content item could be related to many domain model KCs (such as concepts or rules, among others). This context introduces opportunities for sophisticated guidance approaches. When deciding whether to recommend a problem or another content item to a student, a guidance algorithm usually needs to balance two aspects: whether the item is useful (namely, does it introduce missing or insufficiently learned concepts) and whether its difficulty is appropriate (namely, to what extent the student is familiar with concepts that are required to understand the content or solve the problem). In the case of topic-based guidance, these decisions are simply made by examining the current knowledge level of the topic to which the content item belongs, as well as prerequisites for this topic. In the more advanced case of concept-based guidance, knowledge of all concepts related to the item should be considered when introducing a large number of possible situations. The situation becomes even more complicated if a content item has separate sets of *prerequisite* concepts (which are required to work with the item) and *outcome* concepts, which are practiced while working with the item (Bieliková et al., 2014; Brusilovsky et al., 1998; De Bra et al., 2003; De Bra et al., 2013), as shown in Figure 11. In this case, a personalized guidance algorithm should separately assess the knowledge of prerequisite and outcome concepts to balance both the readiness and usefulness of each item.

The classic approach to provide personalized guidance in domains with concept-based content organization is sequencing, namely, the selection of the generation of the single "next best" activity. A range of sequencing approaches were suggested and explored relatively early in domains with complex problems, such as programming and mathematics (Barr, Beard & Atkinson, 1976; Brecht, McCalla & Greer, 1989; Brusilovsky, 1992; McArthur et al., 1988; Wescourt et al., 1977). These approaches usually apply some type of scoring function to determine the "goodness" of each candidate item, and then select the item with the best score. However, due to the large number of decisions to be made using relatively noisy student models and imperfect knowledge engineering, the quality of concept-based sequencing has never been sufficiently good. As a result, sequencing has gradually been overshadowed by safer approaches. In problem-oriented ITSs, it was replaced by more reliable mastery learning, where all content is carefully sequenced in advance and personalized guidance is only used to make a more reliable decision that a topic is mastered. In AESs with more diverse content, sequencing was overtaken by adaptive navigation support and recommendation. As mentioned above, adaptive navigation support modifies existing links in a hyperspace of learning content using link annotation, ranking, or removal. For example, the popular "traffic light" link annotation approach (De Bra & Calvi, 1998; Eklund & Brusilovsky, 1998; Henze & Nejdl, 2001; Weber & Brusilovsky, 2001) marks links to content that is not yet ready to be learned with red bullets, content that is not useful anymore with white bullets, and content that is both ready and useful with green bullets (see Figure 11). Content recommendation creates a new ranked list of recommended content items (Figure 14). In both cases, annotation or ranking decisions are usually based on content scoring functions that are similar to those used in classic sequencing. However, both, navigation support and recommendation approaches allow the student to choose from several opportunities, while explaining which

of them are considered as good and why. This introduces "a human in the loop", correcting possible mistakes that could be made when a sequencing approach bets on one presumably best option. It is important to acknowledge, however, that over the last few years, a new generation of data-driven sequencing algorithms (Doroudi, Holstein, Aleven & Brunskill, 2015; Rowe & Lester, 2015; Tang, Gogel, McBride & Pardos, 2015) brought the ideas of sequencing back into the focus of the research community. These algorithms are based on a large volume of learning data rather than on imperfect content engineering produced by domain experts, and can potentially deliver more reliable content suggestions.

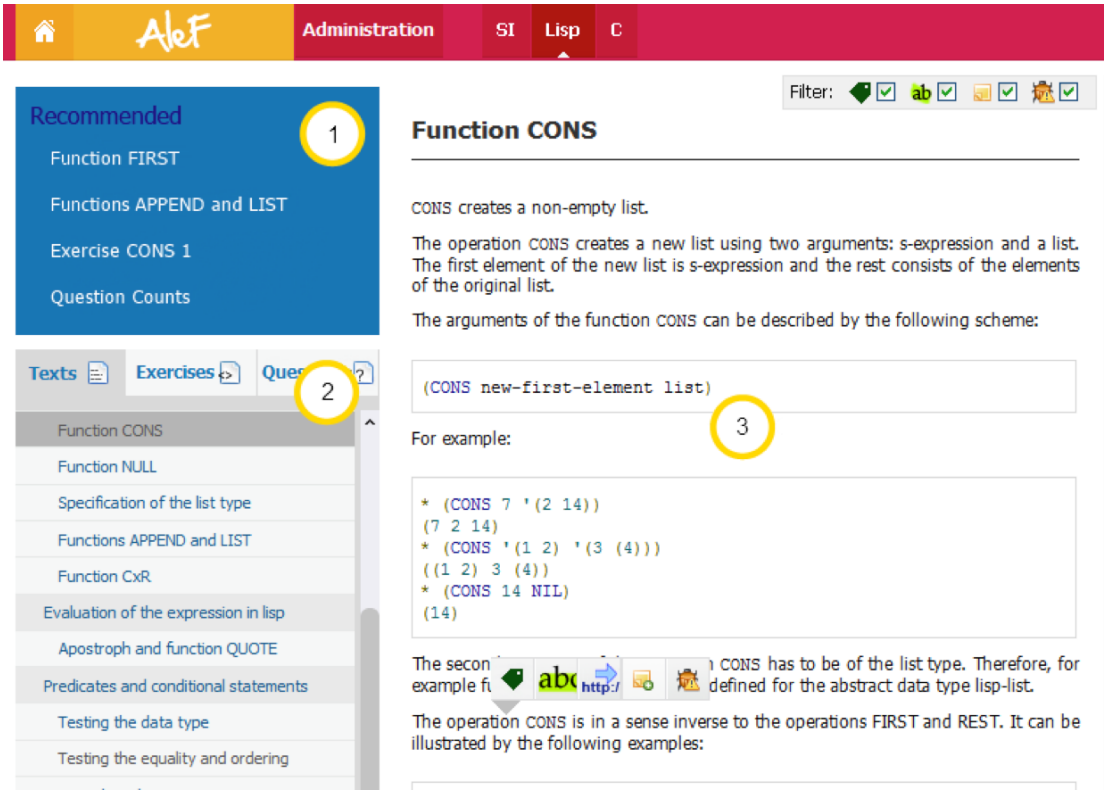


Figure 14. Learning content recommendations in ALEF (Bieliková et al., 2014): (1) list of recommended learning objects, (2) full list of learning objects organized by topic, and (3) selected learning object.

Just as in the case of finer-grained, topic-based guidance, the positive sides of concept-based content organization are somewhat balanced by the difficulty of presenting the concept-based content space to the students. As explained above, in coarse-grained topic organization, all content could be presented as a simple hierarchy organized along the sequence of topics (see Figure 12 and Figure 13), however in concept-based organization it is not feasible: the number of KCs is usually quite large and each content item is connected to many KCs. While in very small domains, the resulting network of concepts and content items could be simply presented to the user (see Figure 10), in regular cases, it is not feasible.

A recommended approach for this situation is combining the straightforward presentation of topic-based organization and the personalization power of concept-based organization. The combination could be produced by grouping fine-level domain concepts into coarse-level topics, preferably corresponding to the sequence of course lectures or textbook chapters. Once this grouping is done, course content could also be grouped into topics according to its outcome concepts and presented within an easy-to-understand topic sequence. The SQL QuizGuide system (Brusilovsky et al., 2010) shows an example of this approach (Figure 15). While the whole organization looks similar to a topic-based course (see Figure 12), the finer-

grained concept layer allows for independently assessing the usefulness of each problem for the target students and generating link annotations that show the estimated difficulty and fraction of already mastered concepts for every problem. As a result, a single type of content organization is used to provide adaptive navigation support on both the topic level and the concept level.

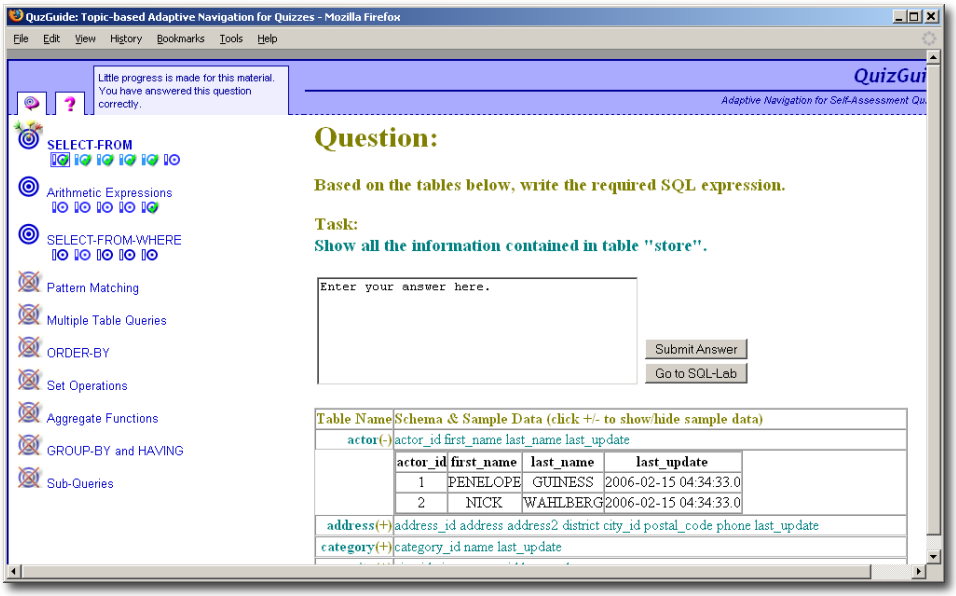


Figure 15. Content space in SQL QuizGuide was organized as a list of topics with several problems associated with each topic.

A more complex case of presenting several kinds of personalized guidance on the basis of the transparent topic-based organization is provided by the Mastery Grids system (Loboda et al., 2014) and is shown in Figure 16. Mastery Grids uses traditional topic-level navigation support that displays current user knowledge for course topics (top row). To complement that, it uses social navigation support to help students in selecting both the topic and the content items within a topic. Social navigation support integrates the progress of all students in class for each content item and for each topic, and displays class progress in comparison with the student's own progress. Progress is shown using color density: darker blue levels show topics and content extensively used by students in class, while a lighter color marks less explored content and topics. By comparing personal progress with the class progress, a student can determine the most appropriate topics, problems, or examples to study. In addition, MasteryGrids uses a content recommendation algorithm; however, instead of presenting recommended content as a ranked list, it marks recommended topics and content items with start icons of different size. This allows for the combination of several levels of suggestions in helping the student to select the most appropriate content item to work with.

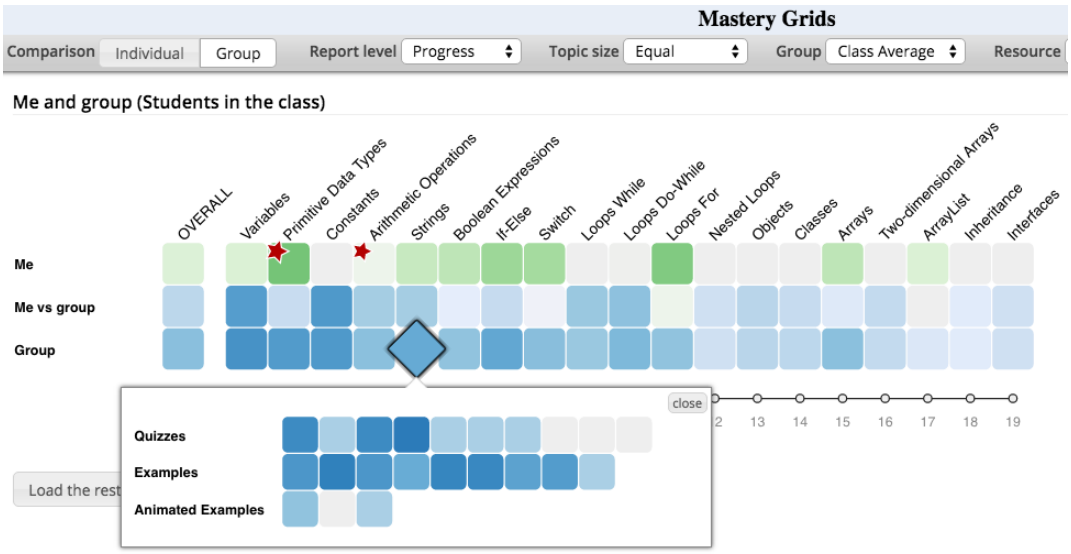


Figure 16. A combination of knowledge-based navigation support (green color), social navigation support, (blue color) and content recommendations (red stars) in the Mastery Grids system.

Recommendations and Future Research

This chapter examined the use of domain models for providing personalized guidance in AESs. We reviewed approaches for building domain models, structuring student models, and learning content on the basis of domain models, and providing several kinds of personalized guidance within this content. This review offers several recommendations to the authors of authoring systems and frameworks for building AESs. Most importantly, it shows that to meet the needs of the possible diversity of the domains and the related diversity of learning content, an authoring system should support a range of domain model organizations, including different levels of KC granularity and diverse links between KCs. It should also support several kinds of learning content organization, as reviewed in the chapter. More specifically, an authoring system should offer rich tools to define a set of KC for the target domain and structure then into a rich network a range of links. It should allow adding a variety of learning content types and connecting each content fragment with domain model KCs. Domain model should serve as a basis for the overlay student model, i.e., for each domain model KC, a system should maintain a variable that represent current level of knowledge.

On the top of this infrastructure, the authoring system should offer a selection of computational student modeling approaches such as Bayesian knowledge tracing (BKT). Depending on a specific approach, the system should also allow the author to specific student model parameters (such as transfer, guess, and slip probability for BKT).

Finally, it should allow authors to choose among several ways of guiding the user through the learning content, from predefined order and free hypertext to mastery learning, sequencing, navigation support, and recommendations. It should also allow for different mechanisms and algorithms to fuel personalized guidance approaches. Without supporting this diversity, an authoring framework will be limited to a subset of domains and application contexts.

The necessity to provide rich tools and flexibility for authors of AESs has been apparently learned by the developers of authoring systems and frameworks. While early authoring frameworks, such as InterBook (Brusilovsky et al., 1998), ACE (Specht & Oppermann, 1998), MetaLinks (Murray, 2003), ECSAIWeb

(Sanrach & Grandbastien, 2000), NetCoach (Weber et al., 2002), and SUGUE (Carmona, Bueno, Guzmán & Conejo, 2002) usually supported one specific approach to domain modeling, content organization, and personalized guidance, more recent systems, such as Diogene (Sangineto et al., 2008), GRAPPLE (De Bra et al., 2013), and ALEF (Bieliková et al., 2014) gradually integrated many successful design features of earlier systems. These systems offer opportunities for rich domain modeling with multiple kinds of domain links, flexible content organization, and pluggable personalized guidance (Figure 17).

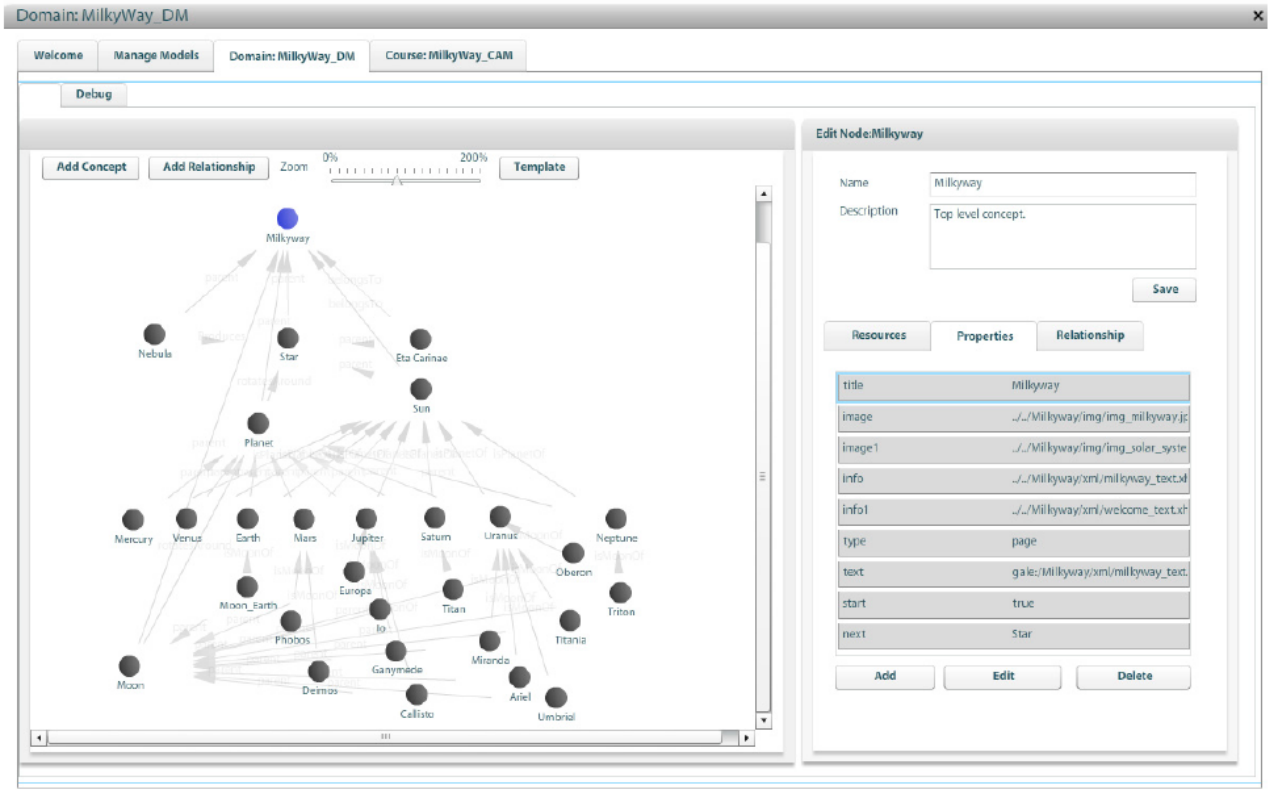


Figure 17. Extensive opportunities for defining domain models in GRAPPLE (De Bra et al., 2013).

References

Anderson, J. R., Boyle, C. F., Corbett, A. T. & Lewis, M. W. (1990). Cognitive modeling and intelligent tutoring. *Artificial Intelligence 42* (1), 7–49.

Anderson, J. R., Farrell, R. & Sauers, R. (1984). Learning to program in LISP. *Cognitive Science 8*, 87–129.

Barr, A., Beard, M. & Atkinson, R. C. (1976). The computer as tutorial laboratory: the Stanford BIP project. *International Journal on the Man-Machine Studies 8* (5), 567–596.

Bieliková, M., Šimko, M., Barla, M., Tvarožek, J., Labaj, M., Móro, R., Srba, I. & Ševcech, J. (2014). ALEF: From Application to Platform for Adaptive Collaborative Learning. In: N. Manouselis, H. Drachsler, K. Verbert and O. Santos (eds.): *Recommender Systems for Technology Enhanced Learning*. Springer New York, pp. 195–225.

Brecht, B. J., McCalla, G. & Greer, J. (1989). Planning the content of instruction. In: D. Bierman, J. Breuker and J. Sandberg (eds.) *Proceedings of 4th International Conference on AI and Education*, Amsterdam, 24–26 May 1989, Amsterdam, IOS, pp. 32–41.

Brusilovsky, P. (1992). A framework for intelligent knowledge sequencing and task sequencing. In: C. Frasson, G. Gauthier and G. McCalla (eds.) *Proceedings of Second International Conference on Intelligent Tutoring Systems*, ITS'92, Montreal, Canada, June 10–12, 1992, Springer-Verlag, pp. 499–506.

Brusilovsky, P. (2003). Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: T. Murray, S. Blessing and S. Ainsworth (eds.): *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software.* Kluwer: Dordrecht, pp. 377–409.

Brusilovsky, P. (2007). Adaptive navigation support. In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. *Lecture Notes in Computer Science*, Vol. 4321, Berlin Heidelberg New York: Springer-Verlag, pp. 263–290.

Brusilovsky, P., Baishya, D., Hosseini, R., Guerra, J. & Liang, M. (2013). KnowledgeZoom for Java: A Concept-Based Exam Study Tool with a Zoomable Open Student Model. In: *Proceedings of 2013 IEEE 13th International Conference on Advanced Learning Technologies*, Beijing, China, July 15–18, 2013, pp. 275–279.

Brusilovsky, P. & Cooper, D. W. (2002). Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. In: Y. Gil and D. B. Leake (eds.) *Proceedings of 2002 International Conference on Intelligent User Interfaces*, San Francisco, CA, January 13–16, 2002, ACM Press, pp. 23–30.

Brusilovsky, P., Eklund, J. & Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. In: H. Ashman and P. Thistewaite (eds.) *Proceedings of Seventh International World Wide Web Conference*, Brisbane, Australia, 14–18 April 1998, Elsevier Science B. V., pp. 291–300.

Brusilovsky, P. & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. *Lecture Notes in Computer Science*, Vol. 4321, Berlin Heidelberg New York: Springer-Verlag, pp. 3–53.

Brusilovsky, P., Sosnovsky, S., Lee, D., Yudelson, M., Zadorozhny, V. & Zhou, X. (2010). Learning SQL programming with interactive tools: from integration to personalization. *ACM Transactions on Computing Education, 9* (4), Article No. 19, pp. 1–15.

Cafolla, R. (2006). Project MERLOT: Bringing Peer Review to Web-Based Educational Resources. *Journal of Technology and Teacher Education, 14* (2), 313–323.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer aided instruction. *IEEE Transactions on Man-Machine Systems MMS, 11* (4), 190–202.

Carmona, C., Bueno, D., Guzmán, E. & Conejo, R. (2002). SIGUE: Making Web Courses Adaptive. In: P. De Bra, P. Brusilovsky and R. Conejo (eds.) *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, Málaga, Spain, May 29–31, 2002, Springer-Verlag, pp. 376–379.

Conati, C., Gertner, A. & Vanlehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction, 12* (4), 371–417.

Conejo, R., Guzman, E. & Millán, E. (2004). SIETTE: A Web-based tool for adaptive teaching. *International Journal of Artificial Intelligence in Education, 14* (1), 29–61.

Corbett, A. T. & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4* (4), 253–278.

Dagger, D., Wade, V. & Conlan, O. (2004). A Framework for Developing Adaptive Personalized eLearning. In: J. Nall and R. Robson (eds.) *Proceedings of World Conference on E-Learning, E-Learn 2004*, Washington, DC, USA, November 1–5, 2004, AACE, pp. 2579–2587.

Davidovic, A., Warren, J. & Trichina, E. (2003). Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education, 46* (2), 241–251.

De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D. & Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. In: *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, Nottingham, UK, ACM, pp. 81–84.

De Bra, P., Aerts, A. & Rousseau, B. (2002). Concept Relationship Types for AHA! 2.0. In: M. Driscoll and T. C. Reeves (eds.) *Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, October 15–19, 2002, AACE, pp. 1386–1389.

De Bra, P. and Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia, 4*, 115–139.

De Bra, P., Smits, D., van der Sluijs, K., Cristea, A., Foss, J., Glahn, C. & Steiner, C. M. (2013). GRAPPLE: Learning Management Systems Meet Adaptive Learning Environments. In: A. Peña-Ayala (ed.) *Intelligent and Adaptive Educational Learning Systems: Achievements and Trends*.

Diessel, T., Lehmann, A. & Vassileva, J. (1994). Individualised course generation: A marriage between CAL and ICAL. *Computers and Education, 22* (1/2), 57–64.

Doroudi, S., Holstein, K., Aleven, V. & Brunskill, E. (2015). Toward Understanding How to Leverage Sense-making, Induction/Refinement and Fluency to Improve Robust Learning. In: O. Santos, et al. (eds.) *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, Madrid, Spain, June 26–29, 2015.

Eklund, J. & Brusilovsky, P. (1998). Individualising Interaction in Web-based Instructional Systems in Higher Education. In: *Proceedings of The Apple University Consortium's Academic Conference*, Melbourne, Australia, September 27–30, 1998, pp. 27–30.

Farrell, R., Thomas, J. C., Dooley, S., Rubin, W., Levy, S., O'Donnell, R. & Fuller, E. (2003). Learner-driven assembly of Web-based courseware. In: A. Rossett (ed.) *Proceedings of World Conference on E-Learning, E-Learn 2003*, Phoenix, AZ, USA, November 7–11, 2003, AACE, pp. 1052–1059.

González-Brenes, J. P., Huang, Y. & Brusilovsky, P. (2014). General Features in Knowledge Tracing to Model Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. In: J. Stamper, Z. Pardos, M. Mavrikis and B. M. McLaren (eds.) *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, London, UK, July 4–7, 2014, pp. 84–91.

Grigoriadou, M., Papanikolaou, K., Kornilakis, H. & Magoulas, G. (2001). INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In: P. D. Bra, P. Brusilovsky and A. Kobsa (eds.) *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia*, Sonthofen, Germany, July 14, 2001, Technical University Eindhoven, pp. 13–24.

Hammond, N. (1989). Hypermedia and learning: Who guides whom? In: H. Maurer (ed.) *Proceedings of 2nd International Conference on Computer Assisted Learning*, ICCAL'89, Berlin, May 9–11, 1989, Springer-Verlag, pp. 167–181.

Henze, N. & Nejdl, W. (2001). Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education, 12* (4), 325–350.

Hoog, R. d., Wielinga, B., Kabel, S., Anjewierden, A., Verster, F., Barnard, Y., PaoloDeLuca, Desmoulins, C. & Riemersma, J. (2002). Re-using technical manuals for instruction: document analysis in the IMAT project. In: Y. Barnard (ed.) *Proceedings of Workshop on integrating technical and training documentation held in conjuction with ITS'02 conference*, San Sebastian, Spain, June 3, 2002, pp. 15–25.

Koedinger, K. R., Corbett, A. T. & Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36* (5), 757–798.

Kumar, A. (2006). The Effect of Using Problem-Solving Tutors on the Self-Confidence of Students. In: *Proceedings of 18th Annual Psychology of Programming Workshop (PPIG 06)*, Brighton, U.K., September 7–8, 2006, pp. 275–283.

Loboda, T., Guerra, J., Hosseini, R. & Brusilovsky, P. (2014). Mastery Grids: An Open Source Social Educational Progress Visualization. In: S. de Freitas, C. Rensing, P. J. Muñoz Merino and T. Ley (eds.) *Proceedings of 9th European Conference on Technology Enhanced Learning (EC-TEL 2014)*, Graz, Austria, September 16–19, 2014, pp. 235–248.

Manouselis, N., Drachsler, H., Verbert, K. & Duval, E. (eds.) (2013). *Recommender Systems for Learning*. Berlin: Springer.

McArthur, D., Stasz, C., Hotta, J., Peter, O. & Burdorf, C. (1988). Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science, 17* (4), 281–307.

Millán, E., Loboda, T. & Pérez de la Cruz, J. L. (2010) Bayesian networks for student model engineering. *Computers & Education ,55*, 1663–1683.

Mitrovic, A. (2003) An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education, 13* (2–4), 173–197.

Mitrovic, A. & Devedzic, V. (2004) A Model of Multitutor Ontology-based Learning Environments. *Continuing Engineering Education and Life-Long Learning, 14* (3), 229–245.

Murray, T. (2003) MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. I*nternational Journal of Artificial Intelligence in Education, 13* (2–4), 199–233.

Ohlsson, S. (1992) Constraint-based student modeling. *Journal of Artificial Intelligence in Education, 3* (4), 429–447.

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H. & Magoulas, G. D. (2003) Personalising the interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. *User Modeling and User Adapted Interaction, 13* (3), 213–267.

Rowe, J. P. & Lester, J. C. (2015). Improving Student Problem Solving in Narrative-Centered Learning Environments: A Modular Reinforcement Learning Framework. In: C. Conati, N. Heffernan, A. Mitrovic and M. F.

Verdejo (eds.): 17th International Conference on Artificial Intelligence in Education, AIED 2015. *Lecture Notes in Computer Science*, Madrid, Spain, pp. 419–428.
Rus, V., Baggett, W., Gire, E., Franceschetti, D., Conley, M. & Graesser, A. (2013). Towards Learner Models based on Learning Progressions in DeepTutor. In: R. Sottilare (ed.) *Learner Models*. Army Research Lab.
Sangineto, E., Capuano, N., Gaeta, M. & Micarelli, A. (2008). Adaptive course generation through learning styles representation. *Universal Access in the Information Society, 7* (1–2), 1–23.
Sanrach, C. & Grandbastien, M. (2000). ECSAIWeb: A Web-based authoring system to create adaptive learning systems. In: P. Brusilovsky, O. Stock and C. Strapparava (eds.) *Proceedings of Adaptive Hypermedia and Adaptive Web-based Systems*, AH2000, Trento, Italy, August 28–30, 2000, Springer-Verlag, pp. 214–226.
Sao Pedro, M. A., Baker, R. S. & Gobert, J. D. (2013). Incorporating Scaffolding and Tutor Context into Bayesian Knowledge Tracing to Predict Inquiry Skill Acquisition. In: *Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013)*, Memphis, Tennessee, July 6 - 9, 2013, pp. 185–192.
Sleeman, D. H. (1985). UMFE: a user modeling front end system. International Journal on the Man-*Machine Studies*, *23*, 71–88.
Sosnovsky, S. & Brusilovsky, P. (2015). Evaluation of Topic-based Adaptation and Student Modeling in QuizGuide. *User Modeling and User-Adapted Interaction, 25* (4), 371–424.Sosnovsky, S. and Dicheva, D. (2010). Ontological technologies for user modelling. I*nternational Journal of Metadata, Semantics and Ontologies, 5* (5), 32–71.
Specht, M. & Oppermann, R. (1998). ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia, 4*, 141–161.
Steinacker, A., Faatz, A., Seeberg, C., Rimac, I., Hörmann, S., Saddik, A. E. & Steinmetz, R. (2001). MediBook: Combining semantic networks with metadata for learning resources to build a Web based learning system. In: *Proceedings of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Tampere, Finland, June 25–30, 2001, AACE, pp. 1790–1795.
Tang, S., Gogel, H., McBride, E. & Pardos, Z. (2015). Item Ordering Effects using Online Tutoring Data. In: O. Santos, et al. (eds.) *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, Madrid, Spain, June 26–29, 2015.
Trausan-Matu, S., Maraschi, D. & Cerri, S. A. (2002). Ontology-centered personalized presentation for knowledge extracted from the Web. In: S. A. Cerri, G. Gouardères and F. Paraguaçu (eds.) *Proceedings of 6th International Conference on Intelligent Tutoring Systems (ITS'2002)*, Berlin, June 2–7, 2002, Springer-Verlag, pp. 259–269.
Trella, M., Conejo, R. & Bueno, D. (2002). An autonomous component architecture to develop WWW-ITS. In: P. Brusilovsky, N. Henze and E. Millán (eds.) P*roceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, Málaga, Spain, May 28, 2002, pp. 69–80.
VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16* (3), 227–265.
Vassileva, J. (1998). DCG + GTE: Dynamic Courseware Generation with Teaching Expertise. *Instructional Science, 26* (3/4), 317–332.
Vassileva, J. & Deters, R. (1998). Dynamic courseware generation on the WWW. *British Journal of Educational Technology, 29* (1), 5–14.
Weber, G. and Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education, 12* (4), 351–384.
Weber, G., Kuhl, H.-C. & Weibelzahl, S. (2002). Developing adaptive internet based courses with the authoring system NetCoach. In: S. Reich, M. M. Tzagarakis and P. M. E. De Bra (eds.): *Hypermedia: Openness, Structural Awareness, and aptivity*. Berlin: Springer-Verlag, pp. 226–238.
Wescourt, K. T., Beard, M. & Gould, L. (1977). Knowledge-based adaptive curriculum sequencing for CAI: application of a network representation. In: *Proceedings of 1977 annual ACM conference*, Seattle, October 1977, pp. 234–240.

CHAPTER 16 – A Process for Adaptive Instruction of Tasks in the Psychomotor Domain

Robert A. Sottilare[1] and Joseph LaViola[2]
[1] US Army Research Laboratory, [2] University of Central Florida

Introduction

Intelligent tutoring systems (ITSs) have been authored to support desktop applications with the most common domains involving cognitive problem solving (e.g., mathematics and physics) in which sets of problems are presented to the learner with increasing complexity and decreasing tutor support as learning progresses. In recent years, implementations of game-based tutors based on the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg & Holden, 2012), an open-source tutoring architecture, have provided tailored training experiences in desktop applications (e.g., Virtual Battlespace and Virtual Medic) for cognitive elements of military tasks (e.g., building clearing and medical triage). The effectiveness of training systems is largely evaluated by the transfer of skills learned during training into the operational or work environment.

In order to maximize transfer of skills from the training environment to the work environment, we advocate closer alignment of learner experiences in the training environment with those expected in the work environment. This implies that psychomotor tasks will benefit from greater transfer if the training includes physical as well as cognitive elements represented in the work environment. Toward this end, cognitive and physical measures should be modeled after successful or expert performers and essential stressors should be represented in the training environment to promote realism and learning. Adequate measures for assessment and appropriate feedback mechanisms to support psychomotor tasks should be represented in ITS domain models. By doing this, the ITS can then more effectively align training and optimize transfer.

Sottilare and LaViola (2015) described an adaptive system prototype to extend training beyond the desktop in support of a land navigation psychomotor task using smart glasses to assess progress, augment reality, and provide feedback to the learner. Sottilare, Hackett, Pike, and LaViola (2016, in review) described a similar adaptive system for training hemorrhage control tasks using smart glasses and pressure sensors to assess the appropriate application of tourniquets and pressure bandages. This chapter expands on these concepts to examine interaction design for GIFT-based tutors to provide generalized adaptive instruction across the breadth of tasks in the psychomotor domain while maximizing opportunities for transfer.

Discussion

This section provides background for an adaptive instructional model, the learning effect model (LEM), used in GIFT, then defines, compares, and contrasts the characteristics of task domains, and explains interaction design as it relates to psychomotor tasks. Next, notional requirements to support adaptive instruction in psychomotor domains are discussed. Finally, lessons learned from usability studies and notional interaction designs of ITSs for psychomotor tasks are shared.

Adaptive Instruction

ITSs provide adaptive instruction by observing the learner and the training environment, and then acting upon them as needed to maintain learner engagement and progress toward learning objectives. Interaction

between the ITS and the learner is largely domain-independent and focused on understanding the learner's cognitive states (e.g., engagement), affective states (e.g., emotions), and physical limits in order to manage the learner's performance (at, below, or above expectation) and progress toward goals. The interaction between the ITS and the training environment is domain-dependent in that actions are tailored to change the environment's complexity to more closely match the capabilities of the learner.

Per Vygotsky (1978), failing to match the domain competence of the learner with the complexity of the training environment might result in anxiety, frustration, confusion, boredom, or even withdrawal. If the learner is struggling to meet expectations, the ITS might choose to decrease the difficulty of problems or scenarios in the environment in order to maintain the learner's engagement or it might decide to provide additional support or scaffolding to the learner to keep him engaged in the learning process.

Differentiating Task Domains

For individual instruction, learning is identified as occurring in three primary task domains: *cognitive, affective*, and *psychomotor* with a fourth group domain, called the *collective, social, collaborative, or team* instructional domain.

Learning in cognitive task domains is measured by increasingly complex behaviors and abstract mental capabilities that range from least complex to most complex and include remembering, understanding, applying, analyzing, evaluating, and creating (Anderson & Krathwohl, 2000). *Remembering* is the simple recall of facts and basic concepts, and includes behaviors like defining, listing, memorizing, and repeating. *Understanding* is demonstrated by the learner's explanation of ideas or concepts, and includes behaviors like describing, explaining, recognizing, and classifying. *Applying* is the ability to use previously acquired information in new situations, and is exhibited through behaviors like demonstrating, implementing, operating, and solving. *Analyzing* is demonstrated by drawing connections among different ideas, and is the ability to differentiate, organize, relate, compare, contrast, and question. *Evaluating* is the ability to appraise situations and justify decisions. Finally, *creating* is the skill used in producing new or original work through design, authoring, and investigation.

Learning in *affective* domains is measured by behaviors indicating emotional growth and maturity. These measures include from least complex to most complex: receiving, responding, valuing, organizing, and characterizing by values (Krathwohl, Bloom & Masia, 1964). *Receiving* includes awareness, the willingness to listen and be courteous, and the ability to focus attention. *Responding* is active participation by the learner who demonstrates a willingness to respond and is motivated to learn. *Valuing* is demonstrated by the belief in the worth attached to a person, object, concept, or behavior. *Organizing* is the ability to rank and contrast values, and resolve conflicts between important values. Finally, *characterizing* is revealed by people who have a value system that drives their behavior. It is difficult to separate learning in affective domains from cognitive domains since affect involves cognitive processes. Graesser and D'Mello (2012) assert that learners encounter a state of cognitive disequilibrium when confronting difficult situations or scenarios that cause conflict with their values. This usually results in affective behavior. Cognitive-affective processes interact until equilibrium is restored. This disequilibrium may block learning for its duration.

Learning in *psychomotor* domains is measured by examining the relationship between the learner's cognitive functions and their physical skills (e.g., coordination, strength, or speed). Simpson (1972) identified psychomotor behaviors which include perceptions (awareness), sets (readiness), guided responses (attempts), mechanisms (basic proficiency), complex overt responses (expert proficiency), adaptation (adaptive proficiency), and origination (creative proficiency).

Social or team domains are another level of complexity from individual training domains. Soller (2001) noted that a team's learning potential is maximized when each individual actively participates in the learning task, thereby, increasing the probability that all trainees understand the learning material and no one is left behind. While participation is a prerequisite to success in team learning, it is not the only antecedent. Sottilare, Holden, Brawner, and Goldberg (2011) note the assessment of team performance is likely to include a model of the interdependency of the roles and responsibilities of the team members, leadership roles and communication, understanding of roles by the team, the domain competency of team members, trust within the team (credibility and reliability), and finally, collective models of team cognition (e.g., shared mental models, workload, and engagement) and affect (e.g., emotions, motivation). Toward this end, Sottilare, Burke, Johnston, Sinatra, Salas, and Holden (2015) conducted a team tutoring meta-analysis to examine the relationship between team behaviors and team learning. A similar analysis is also being conducted to analyze the relationships between team behaviors and team performance, team satisfaction, and team viability. Understanding antecedents of successful team outcomes will help determine what is important to measure and to some degree how to measure variables of interest.

Each of these domains discussed above differs in the behaviors that are externalized by individuals and team members and vary in many dimensions including complexity, definition, and alignment between training and work environments. *Complexity* refers to the range of difficulty in progressing through the various levels in each of the taxonomies for cognitive, affective, psychomotor, and social tasks. In other words, some domains, regardless of whether they are cognitive, affective, psychomotor, or social, require significantly more or less preparation and deliberate practice to master than others.

Definition refers to the level of understanding of how success is measured at each of the levels in each of the domain taxonomies. Well-defined tasks (e.g., mathematics, physics) usually have one or a small number of paths to success and this success is easily measured with respect to an outcome (e.g., right answer to a problem) or interim steps (e.g., demonstrating understanding of a process or series of steps in solving a problem). Ill-defined tasks (e.g., hitting a baseball, practicing law) may have many more paths to success and success may not be easily measured and understood. In other words, the connection between antecedent behaviors and successful outcomes is less clear.

Alignment refers to the level of compatibility between task execution during training and task execution in the actual work environment. In previous works (Sottilare, 2013; Sottilare, 2015), we referred to this as dynamics or physical dynamics, but additional research has led us to this broader term of alignment and a focus on transferring skills learned during training to other environments. Behaviors may differ between training and work environments depending upon the goals of the training. There is a vast difference between golf practice in a desktop game environment used for training and the physicality of the same task executed in the real world. In Figure 18, the game platform (left) is being used to train the cognitive process and strategies associated with golf (e.g., club selection) whereas the physical training (right) of swinging the golf club provides better alignment between training and work environments.

Figure 18. Comparative illustration of the concept of alignment.

There may be tighter alignment of the nature and behaviors of a golfing task in training and work environments where the learner practices swings with various clubs on the golf range and then attempts to transfer those skills to a scored round of golf.

Complexity, definition, and alignment also have an impact on the interaction design of adaptive training systems and on efforts to provide a generalized design to support adaptive instruction across the wide variety of psychomotor tasks. The following sections of this chapter examine interaction design for training psychomotor tasks leading to requirements for adaptive instruction in psychomotor domains.

Interaction Design for Training Psychomotor Tasks

Psychomotor tasks are also referred to as sensory-motor or perceptual-motor tasks. According to Britannica.com (2016), the learning of psychomotor skills involves the "development of organized patterns of muscular activities guided by signals from the environment" (Figure 2). Psychomotor skills could involve very simple one-step tasks like touching your finger to your nose or more complex activities like sewing a quilt or flying a plane. In general psychomotor tasks involve coordinated use of the arms, hands, fingers, and feet. Vision is usually a critical element in successful execution of psychomotor task where the tasks require accurate movement or placement within boundaries, involve depth perception, or involve discrimination of objects.

As an example, we use a student pilot learning to land a plane. According to Jung (1971), the learner perceives or takes in information about the environment (e.g., cockpit instrument readings). Next the learner assesses the quality of what is perceived. In the case of a psychomotor task, the judgment might be that they are slightly above the glide path (negative assessment) and an adjustment to the pitch is needed to bring the plane back into alignment. The pilot acts to push the pitch of the plane down to intercept the glide slope and stops when the pilot perceives it matches the glide slope for a safe landing.

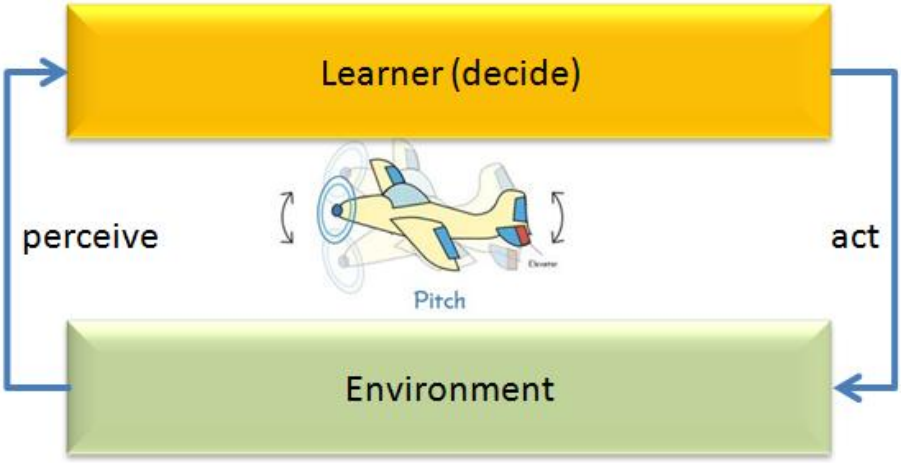


Figure 19. Interaction between the learner and the environment (derived from Jung, 1971).

Methods to acquire measures are essential to understanding successful behaviors (antecedents) and outcomes for psychomotor tasks. Measures include, but are not limited to, reaction and response times, amplitude and speed of movement, pressure exerted, accuracy of movement, and associated error rates or deviations from expected behavior. The ability to capture measures unobtrusively is important to the goal of reducing negative impact to learning. It is not only important to capture measures without altering the learning process, but it is also important to provide timely feedback. These two goals may work in oppo-

sition to each other if the feedback is not explicit. Vague feedback may be distracting or use cognitive resources needed to focus on the execution of the task. An alternate to real-time feedback is the after-action review (AAR). The AAR is usually a more effective alternative when there are sufficient methods in place to capture the learner's performance and a mechanism to play it back soon after completion of the training.

When the measurement acquisition methods in the training environment are insufficient to provide essential elements of an AAR, alternative methods of data capture must be applied. In instances where the training is conducted in the wild (or in areas without instrumentation infrastructure; LaViola, et al., 2015) effective training may rely on portable technologies (e.g., smart glasses or lightweight physiological sensors; Sottilare & LaViola, 2015) to acquire needed measures. However, sensors to detect fine motor movements may not be practical in these environments in the near-term.

In the meantime, other sensor data may be used to indirectly detect learner behaviors. For example, unobtrusively detecting arm, wrist, and hand movements during basketball free throw shooting in a gym may not be practical now, but the resulting location of the ball as it leaves the shooter's hand and approaches the rim might be easier to measure. The ball location data might indirectly indicate whether the shooter's arm position and follow-through are accurate models to ensure consistent success. Likewise, it may not be convenient to instrument a weapon for marksmanship training since it could change the balance of the weapon, but the accuracy and pattern of the rounds at the target might be used to indirectly diagnose common errors (e.g., unsteady position).

Adaptive Instruction for Psychomotor Domains

Adaptive instruction differs from standard computer-based training in that feedback and support are tailored to the needs of each individual learner. There are two primary drivers for adaptation by the tutor: (1) changes in the learner's states (e.g., performance, learning) or (2) changes in the environment. Based on the LEM (Figure 20), learner states including knowledge and skill acquisition are part of the set of conditions the ITS must use to optimize instructional options. Changes in the environment may also be driven by the tutor in response to learner's performance. It is essential to maintain open channels of communication between the ITS and the learner(s) regardless of the domain. The ability to identify and measure behaviors related to psychomotor tasks is critical to understanding where the learner fits in the continuum of competency.

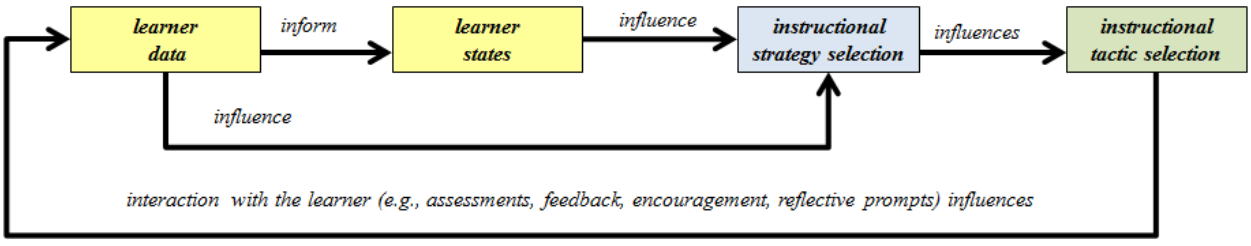


Figure 20. LEM for real-time individual instruction.

To provide effective adaptive instruction, the ITS must be able to understand the learner's states, recognize the difference between ideal performance and the learner's performance, and be able to assess what is driving performance differences. Finally, the ITS must provide actionable feedback to the learner to aid in modifying behavior or acting on the environment to adapt the scenario to make it easier or more difficult.

As noted previously, measures are also critical to understanding the difference between the learner's actual performance and expected performance. Several factors may act as moderators of physical performance: physical fatigue or injury, mental fatigue, motivation and goal alignment, goal-orientation, and emotions are just a few. Each of these moderators should be detectable by an ITS in order to understand the learner's capacity for learning at any given time during training.

Based on the LEM and moderators of physical tasks, Table 1 was developed to define a set of variables of interest required to support adaptive instruction of psychomotor tasks. This table was developed without respect to whether the tasks are conducted in the wild or within instrumented spaces. The information in the description[1] and associated behaviors[1] columns of Table 1 were derived from Simpson's (1972) psychomotor taxonomy.

Table 1. Variables of interest for adaptive instruction of psychomotor tasks.

	Description[1]	Associated Behaviors[1]	Example Psychomotor Measures
Learner Variables			
Perceptions (awareness)	able to use sensory cues to guide motor activity	chooses, describes, detects, differentiates, distinguishes, identifies, isolates, relates, selects	* chooses best option based on selection criteria * accurately estimates location of an object based on senses (vision, smell, sound, vibration) * ability to identify differences or shared attributes
Sets (Readiness)	is mentally, physically, and emotionally ready to act	begins, displays, explains, moves, proceeds, reacts, shows, states, volunteers.	* ability to recall steps in a process * ability to initiate steps in a process * demonstrates readiness by volunteering
Guided Responses (attempts)	able to imitate and follow instructions to accomplish tasks	copies, traces, follows, react, reproduce, responds	* ability to follow to reproduce a set of steps in a process
Mechanisms (basic proficiency)	learned responses are habitual and movements are performed with some confidence	assembles, calibrates, constructs, dismantles, displays, fastens, fixes, grinds, heats, manipulates, measures, mends, mixes, organizes, sketches	* demonstrates the ability to complete tasks with moderate accuracy and speed
Complex Overt Responses (expert proficiency)	able to perform complex motor movements	quicker, better, more accurate behaviors than mechanisms above	* demonstrates the ability to complete tasks with high accuracy and speed
Adaptation (adaptive proficiency)	ability to modify movement patterns to fit new or special requirements	adapts, alters, changes, rearranges, reorganizes, revises, varies	* ability to apply/adapt skills learned in one domain in another domain * adapt dance steps learned for one type of music to another type of music
Origination (creative proficiency)	ability to creating new movement patterns to fit a particular situation or overcome a specific problem	arranges, builds, combines, composes, constructs, creates, designs, initiate, makes, originates	* ability to create new movements extending the traditional movements in a domain * see Olympic high jumping and the Fosbury Flop

Six general skills appear frequently in the literature relative to training and measuring performance during psychomotor tasks: agility, balance, coordination, speed, power, and reaction time. Each of these skills and associate measures relate to variables noted in Simpson's taxonomy, and range from low skill (perceptions) to high skill (origination). Understanding the range of each of these skills and how they are measured is important to delivering effective adaptive training.

Agility is the "ability to move and change the direction and position of your body quickly and effectively while under control" (About Health, 2016). Running obstacle courses are a common practice for assessing agility. Since agility requires quick reflexes, coordination, balance, speed, and correct responses to the

changing conditions, measures of these prerequisite skills may also indirectly indicate agility. *Balance* is usually assessed as "static" or "dynamic" balance. Static balance involves maintaining a steady position while dynamic balance involves the learner's ability to move through an area in a stable manner (e.g., walking across a balance beam). *Coordination* is the ability to integrate multiple gross motor skills into a smooth movement pattern (e.g., dance steps, juggling, hitting a baseball). *Speed* is the rate at which someone or something is able to move or operate, and can be specified by time and distance. *Power* is physical might or the ability to move with great force (Merriam Webster, 2016). Power can be displayed through running, jumping, lifting, and moving (e.g., throwing a ball). Power can be specified by force, which includes three primary variables: mass, acceleration, and distance. Finally, reaction or reflexes can be measured by the time it takes a participant to react to a stimulus (e.g., green light).

A Process Model for Adaptive Training of Psychomotor Tasks

The LEM was used as a basis for developing a process model for adaptive training of psychomotor tasks, but consideration was also given to the unique characteristics, measures, and dimensions of the psychomotor domain. Figure 21 provides real-time consideration for this process model.

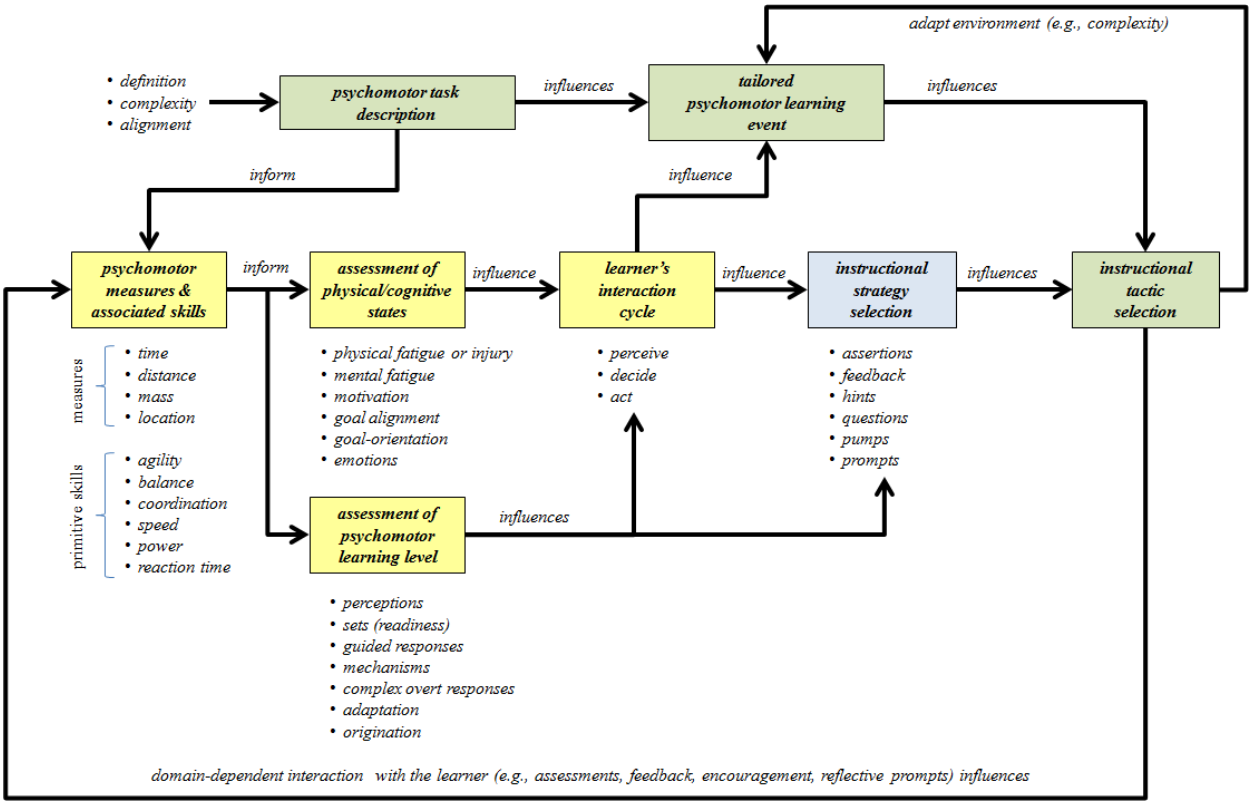


Figure 21. Process model for adaptive training of psychomotor tasks.

As discussed, definition, complexity, and alignment shape the scope of a psychomotor task under adaptive instruction and the psychomotor task description informs the need for specific measures. From these measures, associated primitive skills (e.g., agility, balance, coordination, speed, power, or reaction time) may be derived. This set of primitive skills is important in predicting success in various psychomotor tasks and in understanding how skills might be transferred from one psychomotor task to another. In our

process model, measures and primitive skills aid in the assessment of both the learner's physical/cognitive states and psychomotor learning level defined by Simpson (1972).

A learner's physical states are moderators of the ability to interact (perceive, decide, and act) during a psychomotor task in that they may enhance or limit the learner's performance. Measures and primitive skills may also be used to assess progress along the psychomotor continuum from perception to origination. All of the learner's states and traits (yellow boxes) are used by the adaptive tutor to select appropriate instructional strategies (blue box) and the learner's effect on the training environment (green box – tailored psychomotor learning event). As in the LEM, domain-independent instructional strategies along with conditions in the tailored learning environment influence the selection of domain-dependent instructional tactics. Finally, instructional tactics are divided into two main actions taken by the adaptive tutor: (1) act on the learner through feedback and assessments or (2) act on the training environment to change its complexity to more closely match the learner's capabilities (Vygotsky, 1978). The action by the tutor on the learner or the environment usually results in some change to the learning, performance, or physical state of the learner.

Usability Evaluation and Interaction design Findings

The findings in this section are based on the examination of four psychomotor tasks involving different primitive skills: land navigation in live environments (Sottilare & LaViola, 2015), reconnaissance in interior spaces with augmented reality effects (LaViola, et al., 2015), adaptive marksmanship in virtual environments (Goldberg & Amburn, 2015), and hemorrhage control task with pressure sensors (Sottilare, Hackett, Pike & LaViola, 2016, in review). There are four primary findings from our evaluation of adaptive training for tasks in the psychomotor domain and all have some relationship to acquiring measures.

1. *Measures may be challenging to capture unobtrusively in real time*: an important step in the interaction design of adaptive tutors is to ensure that measures of success are known and methods are available to readily capture these measures to support learner state determination (e.g., learning and performance states) and instructional strategy selection.

2. *The complexity of psychomotor tasks is generally high when compared to cognitive tasks*: complexity is compounded by the degrees of uncertainty and variation in successful solutions among learners when performing physical tasks. ITSs must be able to recognize and account for this variation in classifying learner performance. While model tracing approaches may be simpler to implement, their ability to recognize "successful" performance may be very limited in psychomotor domains. Adaptive instruction of psychomotor tasks should include reinforcement learning methods to account for high degree of variance in successfully executing the task.

3. *The complexity of psychomotor tasks is generally high when conducted in live environments when compared to psychomotor tasks in virtual or mixed reality environments*: complexity in this case is specifically tied to the ability to unobtrusively measure physical movement in uninstrumented spaces when compared to measuring movements in an instrumented interior space.

4. *Space is king*: While other measures (e.g., agility, balance, coordination, speed, power, and reaction time) appear more often in the literature with respect to psychomotor tasks, space or the understanding of how a learner moves through space and their location in space is basic to successful measurement of psychomotor tasks.

Recommendations and Future Research

Based on our findings, we recommend the following capabilities be implemented within the GIFT architecture to support individual and team training in psychomotor domains:

- Provide gateway interfaces for sensors/sensor suites that support real-time sensing of learner movement (including hand, arm, and leg movement) at variable distances from 2 to 30 ft.
- Provide gateway interfaces for sensors/sensor suites that support real-time, clear sensing of multiple learners moving through a finite space (e.g., a density of up to 5 learners in a 100 ft^2 area).
- Provide methods to support remote sensing of objects (e.g., pressure sensors on bandages and tourniquets) in the training environment to provide measures related to learning and performance.
- Adapt the learner model to include standard attributes for primitive psychomotor skills (see Figure 21) with descriptors/metadata for most recent assessment and skill decay profiles.

Future research related to adaptive training in psychomotor task domains should focus on the following opportunities:

- Research to identify measures for physical and cognitive moderators (e.g., level of physical or mental fatigue, motivation, goal alignment and orientation, and emotional states)
- Research to identify measures of agility, balance, coordination, speed, power, and reaction time and development of classifiers to predict future psychomotor performance based on these primitive skills
- Research to identify assessment methods for determining psychomotor learning levels (perceptions to origination)
- Research to identify methods to optimize selection of instructional strategies and tactics during a psychomotor task

Conclusions

GIFT has been used to demonstrate adaptive training in multiple psychomotor task domains (e.g., marksmanship, land navigation, building reconnaissance, and medical triage/hemorrhage control) over the last 2 years. While GIFT is currently flexible enough to support training in the psychomotor domain, its capabilities to sense learner movements could be greatly enhanced through the application of existing sensor technologies (e.g., Microsoft Kinect and other state-of-the-art depth-sensing technologies). Research is needed to improve methods to acquire measures and classify learner psychomotor learning states. Significant effort should be dedicated to enhancing the GIFT learner model with respect to standard attributes and domain-unique attributes for the psychomotor domain. Potential candidates for standard learner model attributes include primitive skills that may be used to assess future success in psychomotor domains.

References

About Health. (2016, March 29). Retrieved from http://sportsmedicine.about.com/od/glossary/g/Agility_def.htm.

Anderson, L. W. & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of Educational Objectives: Complete edition, New York : Longman.

Britannica.com (2016). Psychomotor learning. Retrieved from http://www.britannica.com/topic/psychomotor-learning

Goldberg, B. & Amburn, C. The application of gift in a psychomotor domain of instruction: a marksmanship use case. In: Sottilare R, Sinatra A, editors. Proceedings of the 3rd Annual GIFT Users Symposium (GIFTSym3); 2015 Jun 17–18; Orlando, FL. Aberdeen Proving Ground (MD): Army Research Laboratory (US); c2015. ISBN 978-0-9893923-8-9.

Graesser, A.C. & D'Mello, S. (2012). Emotions During the Learning of Difficult Material. In: Brian H. Ross, Editor(s), Psychology of Learning and Motivation, Academic Press, Volume 57, Chapter 5, Pages 183–225, ISSN 0079-7421, ISBN 978012-3942937, 10.1016/B978-0-12-394293-7.00005-4.

Jung, C.G. (1971). Psychological Types. London: Routledge & Kegan Paul. (Collected Works of C.G. Jung, Vol. 6).

Krathwohl, D.R., Bloom, B.S. & Masia, B.B. (1964). Taxonomy of Educational Objectives: Handbook II: Affective Domain. New York: David McKay Co.

LaViola, J., Sottilare, R., Garrity, P., Williamson, B., Brooks, C. & Veazanchin, S. (2015). Using Augmented Reality to Tutor Military Tasks in the Wild. In Proceedings of the Interservice/Industry Training Simulation & Education Conference, Orlando, Florida, December 2015.

Merriam-Webster (2016, March 29). Power. Retrieved from http://www.merriam-webster.com/dictionary/power.

Simpson, E. (1972). The classification of educational objectives in the psychomotor domain: The psychomotor domain. Vol. 3. Washington, DC: Gryphon House.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. International Journal of Artificial Intelligence in Education, 12(1), 40–62.

Sottilare, R., Holden, H., Brawner, K. & Goldberg, B. (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training. In Proceedings of the Interservice/Industry Training Simulation & Education Conference, Orlando, FL, December 2011.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. &H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: US Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R. (2013). Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model - Research Outline. US Army Research Laboratory (ARL-SR-0284), December 2013.

Sottilare, R. (2015). Dimensions and Challenges in Domain Modeling for Adaptive Training. In R. Sottilare & A. Sinatra (Eds., 2015) 3nd Annual GIFT Users Symposium (GIFTSym3), Orlando, Florida, 17–18 June 2015. US Army Research Laboratory, Orlando, FL. ISBN: 978-0-9893923-8-9

Sottilare, R., Burke, S., Johnston, J., Sinatra, A., Salas, E. & Holden, H. (2015). Antecedents of Adaptive Collaborative Learning Environments. In Proceedings of the Interservice/Industry Training Simulation & Education Conference, Orlando, FL, December 2015.

Sottilare, R. & LaViola, J. (2015). Extending Intelligent Tutoring Beyond the Desktop to the Psychomotor Domain: A survey of smart glass technologies. In Proceedings of the Interservice/Industry Training Simulation & Education Conference, Orlando, FL, December 2015.

Sottilare, R., Hackett, M., Pike, W. & LaViola, J. (2016, *in review*). Adaptive Instruction for Medical Training in the Psychomotor Domain. In J. Cohn, D. Fitzhugh, and H. Freeman (Eds.) Special Issue: Modeling and Simulation Technologies to Enhance and Optimize the DoD's Medical Readiness and Response Capabilities of the Journal for Defense Modeling & Simulation (JDMS).

Vygotsky, L.S. (1978). Mind in Society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.

CHAPTER 17 – Domain Modeling in a Psychomotor World: A Marksmanship Use Case

Benjamin Goldberg and Charles Amburn
US Army Research Laboratory

Introduction

Recent technological advancements have extended computer-based training and education practices beyond the traditional desktop environment and into domains requiring physical interaction to conduct tasks. In this chapter, we examine a psychomotor use case as it relates to domain modeling within the Generalized Intelligent Framework for Tutoring (GIFT), and present considerations to address when developing any system of this nature. GIFT is a domain-independent architecture developed for the purpose of building adaptive training functions and intelligent tutoring systems (ITSs) across an array of tasks and a conceivably unlimited set of knowledge, skills, and abilities (KSAs). A current objective is to apply GIFT tools and methods in an Army-valued skill domain that incorporates psychomotor components of task execution and KSA development (Sottilare, Sinatra, Boyce & Graesser, 2015). The intention is to broaden ITS modeling techniques beyond the traditional cognitive dimensions of learning. In this instance, individuals are reliant upon knowledge and skills associated with psychomotor functions rather than cognitive application used to solve problems. From an adaptive training and ITS perspective, this new psychomotor paradigm requires a shift in how the domain space is modeled to associate this new type of input for informing assessment. Rather than modeling and monitoring steps toward solving a problem and identifying misconceptions and impasses along the way, the psychomotor use case focuses on behavior and its inherent influence on performance; specifically, what nuances of a task are dictated by variations in patterns of behavior and what strategies can be enacted to assist an individual in acquiring the ability to replicate a desired behavior across multiple trials. The end-state goal is to enhance training systems to support physical skill development through deliberate practice techniques (Ericsson, 2006, 2014) pedagogically managed by ITS assessment and feedback methods.

From an implementation standpoint, applying adaptive training to a psychomotor domain requires the same piece parts associated with developing any ITS. At a minimum, this includes (1) data types at a granular enough level to inform appropriate assessments, (2) established models of expert performance to inform performance state determinations, and (3) a pedagogical model that guides practice and accelerates skill acquisition through formative feedback and adaptive sequencing of practice events. These three components create an adaptive learning effect chain where raw data are used to inform assessments, and assessments are used to inform instructional tactics based on learning theory and cognitive psychology (Sottilare, 2015; Sottilare, Ragusa, Hoffman & Goldberg, 2013; see Figure 1). Each one of the mentioned piece parts is dependent on the domain being instructed and the technologies incorporated in its execution. To guide the remainder of this discussion, we focus in on the domain of marksmanship.

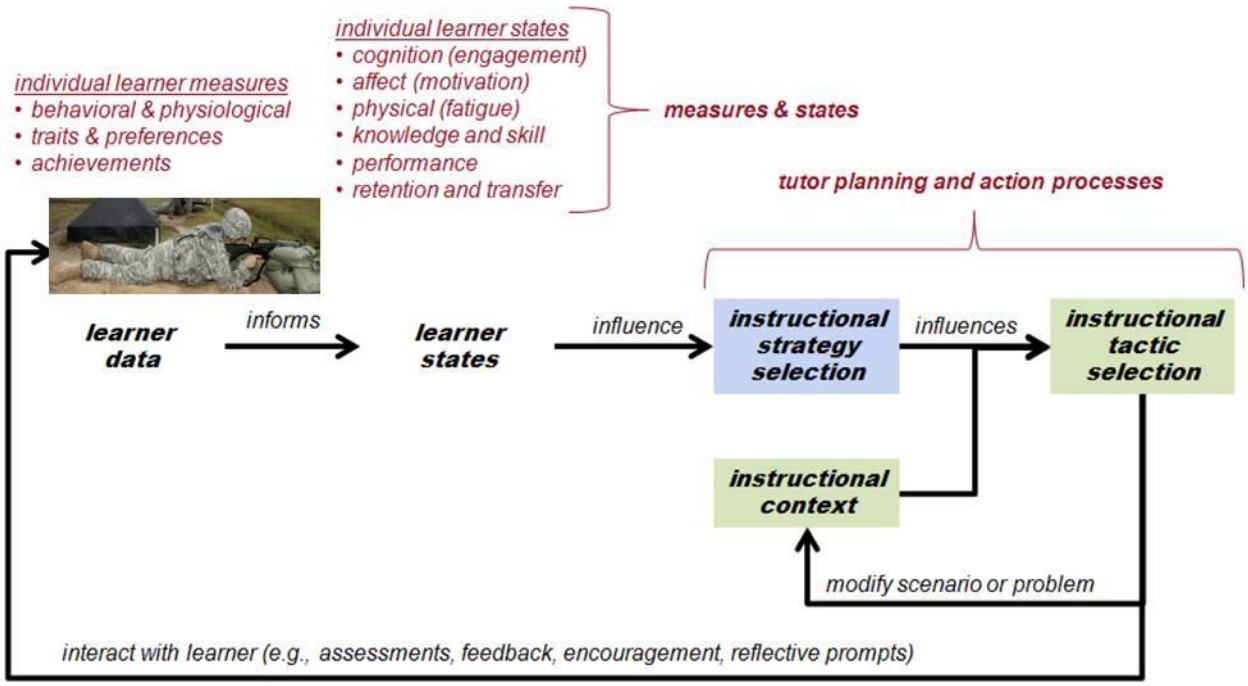


Figure 1. The learning effect model's chain of inference procedures within a closed-loop adaptive training system (Sottilare, 2015).

The Domain of Basic Rifle Marksmanship

For an initial use case, we identified marksmanship as an excellent candidate to steer development efforts. To scope the problem space, we selected the KSAs associated with instruction and training surrounding basic rifle marksmanship (BRM). This is due to the high volume of individuals who complete this form of training each year and the Army's large investment in marksmanship-dedicated simulators that provide initial exposure through interactive hands-on training that replicates live range type exercises. This approach to initial skill acquisition is attractive because it provides a safe (i.e., no live ammunition), cost-effective environment (i.e., no cost on ammunition) for initial contact with the basics of handling and firing a rifle.

Directing our attention to the skill components of the domain, rifle marksmanship is a complex psychomotor task demanding high physical and mental coordination. As with learning any complex skill, training starts with the basics. BRM involves the execution of fundamental procedures to consistently strike a target in a manner that can be replicated over multiple trials. These fundamentals are captured in the Army's Rifle Marksmanship Field Manual (FM 3-22.9), which serves as the foundation for BRM instruction. Acquiring proficiency in applying BRM fundamentals is essential before an individual can progress to more complex skill applications (e.g., hitting targets quickly across varying distances, hitting moving targets, etc.).

Current BRM Training Methods

The Engagement Skills Trainer (EST) is a simulated firing range designed as a cost-saving solution for deliberate practice of BRM fundamentals. It allows individuals to realistically replicate the procedures of operating a rifle for the first time in a controlled and safe environment. The EST was selected for our research because of its ability to collect behavioral data at a granular enough level to inform model repre-

sentations that perform assessments. To effectively monitor performance and diagnose error, the weapons used in the EST were instrumented with various sensor technologies (e.g., trigger pressure sensor, accelerometer, breathing strap, and aim trace laser). In its current state of practice, these data streams provide visual tools for instructors to observe trainee behavior leading up to and following the execution of a shot so as to better assess performance and diagnose deficiencies (Figure 2). This puts the responsibility of assessment and remediation directly on the instructor, resulting in inconsistent subjective opinions driving training practices.

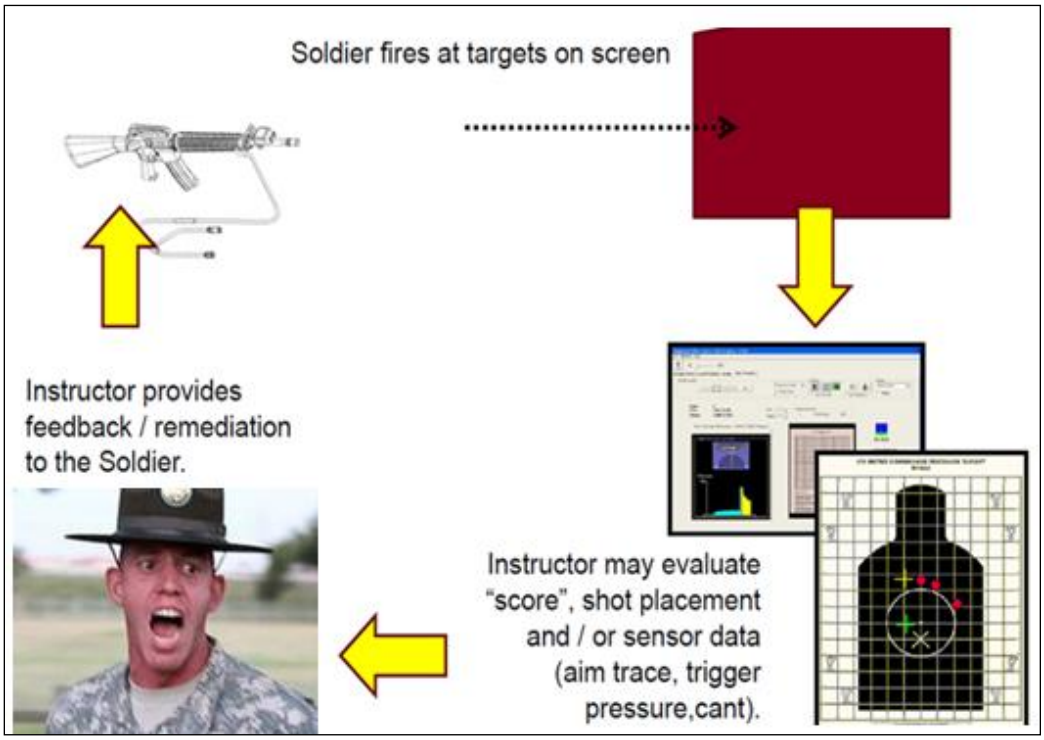


Figure 2. Current BRM training in the EST.

There are a couple recognized issues to this approach. First, prior research has shown a lack of consistency between BRM instructors in terms of error diagnosis (James & Dyer, 2011), and often, the data itself are overlooked due to difficulties in accessing and efficiently interpreting data outputs. Secondly, the throughput of trainees on the EST for BRM-related training is quite large, making it relatively impossible to provide personalized instruction for all. Currently, it is common to have 15 trainees at a time on an EST with only 1–2 instructors providing support.

Designing an Adaptive BRM Training Method

In an effort to enhance the EST to support tailored instruction across all users, work has started on integrating adaptive training technologies that enable real-time performance diagnosis for triggering objective-based guidance and remediation (Amburn, Goldberg & Brawner, 2014; Goldberg, Amburn, Brawner & Westphal, 2014). This process is intended to instill a repetitive behavior of proper fundamental techniques that become engrained in the trainee's future skill application (i.e., trainee exhibits autonomous application of fundamentals when executing a basic marksmanship task).

To facilitate this objective, we are investigating how GIFT can be applied to consume EST sensor data for the purpose of constructing data-driven assessment logic. The goal is to establish robust modeling techniques that classify sensor stream inputs against a set of designated behavioral objectives that align with BRM fundamentals. In this instance, GIFT would consume both performance-derived outcomes calculated within the EST and raw sensor data logged during task execution. Based on designated domain representations, GIFT would analyze inputs against a set of criteria to gauge performance and diagnose error if appropriate. GIFT could then apply pedagogical reasoning and direct feedback to the trainee through a communication interface; or data and feedback could be communicated directly to the instructor to support a data-informed human intervention (as seen in Figure 3).

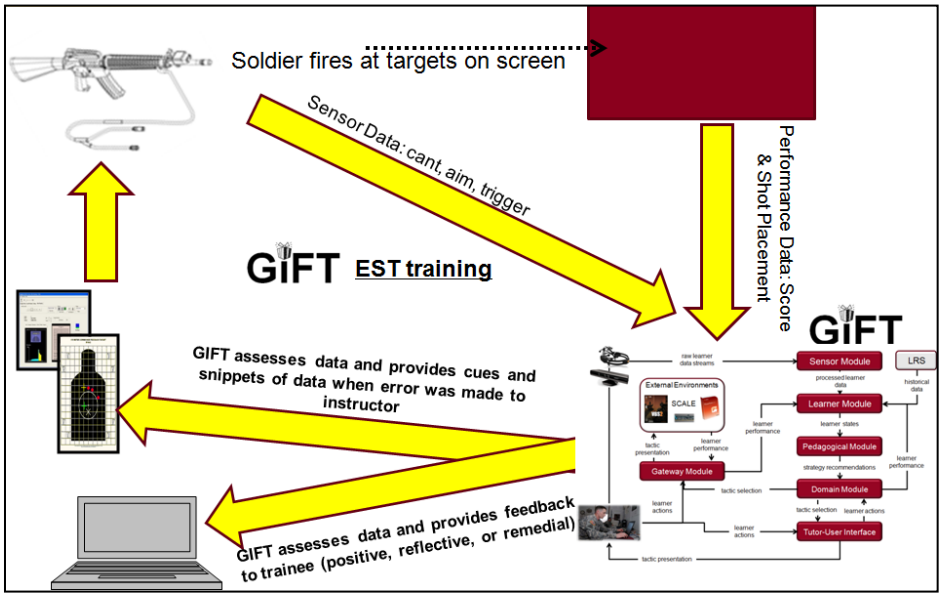


Figure 3. BRM training using the EST and GIFT.

In the remainder of this chapter, we present domain modeling approaches to support a BRM use case. We discuss dependencies and limitations of three domain modeling techniques. The techniques selected are influenced by pedagogical considerations that drive their implementation. We are specifically interested in investigating feedback techniques that work best when managing a psychomotor-based procedural task. The caveat is that each feedback technique is inherently linked to the type of assessment a domain model affords. We discuss this dependency in more detail as we highlight each hypothetical domain modeling application.

Domain Modeling a Psychomotor Task

The initial task in the development of an adaptive marksmanship capability is establishing domain models in GIFT that can act on real-time data and accurately inform learner state classifications. Understanding the components of GIFT and how domain information is configured to support underlying assessment logic is critical.

Modeling to the GIFT Standard

Building assessment logic in GIFT is done within the domain module. It is here where conditions are written that designate performance state information that guides GIFT's Adaptive Tutoring Learning Effect Chain (ATLEC) (see Figure 1; Sottilare, Ragusa, Hoffman & Goldberg, 2013). To do so, first an author establishes a representation of task concepts and objectives against which a learner will be assessed. The schema for a domain model was designed to support domain-independency across all remaining modules within the GIFT architecture. It allows an author to establish an ontological representation of concepts and objectives that dictate the level of granularity associated with a particular task's assessment (Sottilare, Sinatra, Boyce & Graesser, 2015). This representation is encoded in what is called GIFT's domain knowledge file (DKF), where assessment practices are configured to support the ATLEC implantation.

In the context of a psychomotor use case, assessment practices are conceptually linked to three factors: performance, behavior, and physiology (Behneman et al., 2012; Chung et al., 2011). In BRM training, the critical variable that designates qualification standards is performance. In current settings, if trainees can consistently meet performance standards, then they are advanced to the next period of instruction. If a trainee does not meet standard, then coaching is required. Yet coaching cannot be based solely on performance outcomes. As such, psychomotor adaptive training requires complimentary assessments that aid in identifying what specific aspects of the task to deliberately focus on.

In this instance, coaching is based on moderators of performance. For marksmanship, these moderators are behavior patterns exhibited during task execution. This behavior is inferred from weapon sensors and physiological markers recorded during task execution. As such, assessments in GIFT link data types with the concepts they inform. As an example, the fundamental of trigger squeeze will be linked to the trigger sensor embedded directly on the weapon. With this association, a model will be developed to inform state representations that will guide pedagogical decisions. These state representations consist of four levels: (1) unknown, (2) below-expectation, (3) at-expectation, and (4) above-expectation.

While a domain model in GIFT can produce state assessments on performance moderators, it is important to define logic to resolve conflict between performance and behavior. As an example, if a basketball player exhibits an unconventional approach to shooting free throws as identified by some assessment logic (e.g., shooting underhand), yet makes a high percentage of shots, should the system intervene to correct the behavior? Or should the system recognize the superior performance level and allow the behavior to persist? Though some coaching might improve even the most superior of performers, it is often in the case of the low performer where immediate coaching is necessary.

Domain Modeling from an Assessment Standpoint

A current open question is what type of model is most suitable in a psychomotor-based training domain. To start, we must conceptually identify the modeling techniques that support assessment practices and the underlying assumptions based on their implementation. Each technique is confined to data made available from a training platform, with additional channels of information supported by commercial-off-the-shelf wearable sensing technologies. For the purpose of this discussion, we are limiting the data types for model development to include (1) performance outcomes as they relate to grouping and zeroing procedures produced by the training application, (2) trainee behaviors logged during the execution of a shot as collected from a suite of sensors embedded directly on the EST weapon (e.g., trigger pressure, cant angle of the rifle, aim trace, butt-stock pressure), (3) a physiological sensor collecting breathing patterns and heart rate data, and (4) a neuro-physiological sensor (e.g., electroencephalogram) used to monitor brain activity and cognitive load during practice exercises.

With a specified list of data fields, we present three high-level overviews of domain modeling within a psychomotor task environment and how those models are applied to guide coaching and pedagogical practice. The three approaches being investigated are (1) expert models of desired performance, (2) buggy-libraries that associate with common novice error, and (3) neuro-physiological approaches that focus on brain activity exhibited during skill development. Before we review each approach, there are a few assumptions that should be covered. First, GIFT needs the ability to consume and process sensor data in real time to support run-time instantiations of an assessment model. This is currently completed in GIFT's sensor module. Within the sensor module, raw data are filtered, transformed, and processed based on specified configurations. Identifying upfront how data will be used to inform assessment is important as that will dictate the methods to compute the variable required to properly assess behavior in relation to task fundamentals.

Expert-Informed Model Development

First, we focus on modeling moderators of performance and building mathematical representations of expert behavior. This representation is used as a way to recognize individuals whose behavior deviates from an established desired outcome. Behaviors of interest are task-dependent and should associate with a fundamental aspect of skill application. Identifying the behaviors to track is traditionally informed through a domain or task analysis. In terms of the BRM use case, the concepts and objectives selected were taken directly from FM3-22.9. We selected the four "fundamentals" of marksmanship taught in BRM to build the expert-informed models around, including (1) stable body position, (2) proper breath control, (3) steady trigger squeeze, and (4) sight alignment. These fundamentals are organized within GIFT's modeling schema standard as described above.

To this effect, we are defining concepts mapped in GIFT's DKF to these recognized fundamentals, and assigning data variables and combinations of variables as the input for guiding inference procedures. An assumption associated with this approach is that experts in the domain of marksmanship will exhibit common and consistent behaviors across all collected data fields. This method is primarily exploratory, as we seek to identify statistically reliable trends in behavior as they relate to expert performance of a BRM procedure. The output of this technique is descriptive expert models based on trends in behavior within specified time windows prior to and after the execution of a shot. Outcomes of initial experimentation with the Army Marksmanship Unit's Service Rifle Team are in support of this technique, with resulting models holding up during cross-fold validation procedures (Goldberg et al., 2014). As the subjects in this model development were the Army's best shots, the variance in performance was rather small. This resulted in the development of descriptive models that quantitatively represented behavior as a range of values within two standard deviations of a metric's absolute mean value. We have extended these techniques to account for a signals derivative over time, with an area under the curve (AUC) calculation being used as the basis of model formulation. In addition, we explored regression analysis techniques to identify the behaviors that account for the most variance in performance outcomes.

Yet, how effective are these models at informing feedback practices within an adaptive training environment? This is an important question to ask for any modeling approach applied across a psychomotor domain of instruction. What good is an assessment technique if it only supports shallow pedagogical interventions that have little impact on skill development and acquisition? An expert model of this nature can tell you what a trainee is doing that is "not" reflective of expert performance, but offers no insight on the specific error that individual is making. This limits the type of pedagogy a system like GIFT can provide. As such, the first round of experiments focus on generic remediation materials that map directly to a fundamental concept being violated (i.e., if a trainee is assessed poorly on trigger squeeze, then GIFT presents feedback linked directly to proper execution of that behavior). This approach also keeps models independent of one another in terms of machine learning techniques, as the performance state-space is only

that of expert. To extend the assessment space to account for specific error types, it is important to explore the efficacy of establishing buggy-library influenced models.

Novice-Informed Model Development (Buggy-Library)

In an effort to build more informative assessment models in BRM, an approach we will apply once a testbed is established involves the development of a buggy-library (i.e., model of misconceptions). In other words, we want to model common errors novices make when learning BRM procedures. This approach varies from expert models, as it takes behavior data and determines a specific type of error made among a bank of available choices. This bank is based on common impasses identified in James and Dyers' (2011) Rifle Marksmanship Diagnostic and Training Guide. In this document, the authors correlate common performance patterns with fundamental mistakes linked to task behavior. These recognized mistakes serve as available annotations to base supervised machine learning techniques on. An annotated data set will be collected based on the opinion of expert marksmen as they observe novices perform BRM procedures.

In essence, this becomes a machine learning problem. Expert annotators are instructed to observe and classify behavior based on available data streams. Common errors will include things like improper breathing, squeezing the trigger too quickly, poor body alignment, etc. This creates a rich state-space to drive machine learning methods to identify patterns in data that consistently designate this behavioral outcome. With an annotated data set, analyses can be conducted that look across the available dependent measures to identify consistent clusters that designate a reliable correlation between cause and outcome. This approach provides a much more prescriptive assessment of skill application and can inform a specific intervention designated to correct the identified misconception.

However, this method is prone to error as the annotation of novice mistakes is subjective by nature. Inter-rater reliability is key to the success of this method, as mutual agreement across experts will assist in creating data-driven models that are representative of expert opinion. A potential pitfall is the recognized inconsistency of instructors in BRM training schoolhouses. James and Dyer (2011) note that trainers of BRM vary in the tactics and procedures they teach along with their complete understanding of the behaviors involved in the task (Goldberg et al., 2014). Nonetheless, we will examine the feasibility of producing a buggy-library capability in GIFT to inform BRM assessment practices. This will be conducted during our first interaction with novice performers in an effort to validate the application of expert models in identifying non-expert behavior.

Physiology-Informed Model Development

The last domain modeling procedure we introduce is based on work surrounding brain monitoring techniques that involve the tracking of electroencephalogram (EEG) signals. Prior work in this area has yielded interesting findings, with a focus on techniques to promote cognitive readiness within individuals as they engage in activities to develop a new skill. In this instance, cognitive state representations, as determined by EEG signal classifiers, are used to determine an individual's readiness for building skill in memory that translates to future application. This application of neuro-feedback is believed to accelerate an individual's progression from novice to expert in a skill-based domain. Prior research has shown expert performers in the domains of marksmanship, golf, and archery to exhibit common EEG profiles as they mentally prepared prior to executing a shot (Berka, Behneman, Kintz, Johnson & Raphael, 2010). This profile was established as a comparative benchmark to assess novice EEG signals against. Berka et al. (2010) used this finding to create the Adaptive Peak Performance Trainer, which integrated a learner's psycho-physiological state into the system's automated feedback, with preliminary results showing individuals training on marksmanship to accelerate their learning trajectories when such information was made available.

Such systems are still being assessed within controlled laboratory settings. An example can be seen in a technology called Neurobridge, which applies neuroscience-based software to optimize how a trainee's brain functions in real time during the execution of a task procedure (Cubic, 2014). In the marksmanship example, Neurobridge is applied in an open learner model (OLM) approach that displays an individual's cognitive state, with the OLM serving as a mechanism to promote deep concentration through coaching interactions. As learners focus their attention, the system responds by adapting the OLM image to display a cue they are ready to train. This is a novel approach to extend the current state of the art, but currently there is a lack of empirical evidence to support its application.

This approach to instruction introduces elements of augmented cognition, as the EEG sensor is applied to inform individuals on their readiness to engage in an activity. What is potentially powerful about this approach to assessment is its ability to translate across tasks and domains. If a reliable profile of EEG trends can be established, learner modeling can be applied to keep individuals aware of their readiness state. However, there are concerns that must be addressed when considering this type of approach. The main question is, do interventions of this nature actually make a difference to warrant the application of expensive EEG headsets? While the costs of these devices have reduced over time, they are not yet at a selling point to support wide application. In addition, are these sensor technologies reliable enough to inform these types of cognitive states in an environment outside of a laboratory setting? There are many questions that must be answered, but this approach to domain modeling is interesting as it extends beyond basic procedural assessments.

Future Research and Conclusions

In this chapter, we presented considerations for extending GIFT modeling and pedagogical techniques to support a psychomotor training domain, in this instance, BRM. With an established workflow for building models and a testbed to run studies, future research will involve numerous components of marksmanship-related training efforts. This includes investigating the application of these approaches on new weapons not associated with BRM, as well as investigating the feasibility of training more advanced skills (e.g., hitting moving targets). In addition, future research will also examine guidance within a psychomotor use case that will investigate variations in timing, specificity, and modality of feedback. Another area of research will focus on human performance related questions that involve elements of deliberate practice and sequencing of training tasks to better improve skill acquisition and retention.

This work is among the first to address the need for automated, personalized, and intelligent psychomotor training. Methods for measurement, model creation, expert validation, and novice diagnosis in this type of domain are addressed, along with their implications of being implemented in a domain-independent framework (Sottilare, 2013). This chapter presents a way to measure items of interest that corresponds with skill and ability and how to use those items to create models that represent disparate assessment techniques for coaching purposes.

In terms of modeling psychomotor domains to guide adaptive instructional methods, this work lends itself to lessons learned that can inform recommendations for future practitioners and future developmental features native to GIFT's authoring and run-time environments. While the elements of a psychomotor task are often unique to the environment that task is executed within, there are common practices that can be established for assisting individuals in building data-driven domain models that inform automated coaching practices.

The initial step is establishing an ontological representation of the skills and fundamentals associated with a particular domain and linking those concepts to task features and behaviors that will drive assessment implementations. A recommendation for GIFT is to build an authoring interface that supports this upfront

top-down requirements analysis, where task concepts and fundamentals are linked to behaviors, and behaviors are linked to available data types that can be used for model development. In addition, research should be conducted to identify available sensor types that can be used for modeling psychomotor components, along with the conditions they are most suitable within. This can assist researchers and training developers in identifying the appropriate sensor technologies that provide the appropriate granularity of data to support the learning effect chain inference procedures.

With an established requirements document based on task analysis techniques, the next step is building mathematical models that can be used for real-time assessment practices. This capability requires GIFT to consume raw sensor data, filter the data for reducing the signal-to-noise ratio, transform the data into a variable/metric appropriate for analysis purposes (e.g., calculating an absolute derivative), and parse the metric into a specified time-window for running statistical techniques against (e.g., calculating a rolling average; conducting an area under the curve function, etc.). We recommend developing a tool to assist in this process, reusing methods often and where appropriate. It is also believed artificial intelligence (AI) methods can be explored to automate this process for the purpose of identifying consistent data trends across varying time segments that correspond with task execution behaviors. The goal would be investigating methods to automatically generate sensor-driven descriptive behavior models based on a preconfigured set of features and threshold parameters.

The final recommendation falls outside the scope of domain modeling, but its dependency can influence how assessment practices are configured. We recommend leveraging research examining human performance and skill development from a cognitive psychology and kinesiology perspective. The cognitive psychology suggestion is more from a pedagogical standpoint, but task scheduling should account for what is known about how an individual builds skill and develops muscle memory across blocked vs. interleaving practice trials (Birnbaum, Kornell, Bjork & Bjork, 2013). From a kinesiology viewpoint, understanding the relationships between physical motor movements, task difficulty, and skill acquisition should be accounted for in modeling the progression of tasks an individual should perform during a set of training sessions. The main argument here is that theory should not be ignored; it should help identify what matters in learning psychomotor skills and what components should be modeled to help coach along the way.

References

Amburn, C., Goldberg, B. & Brawner, K. (2014). *Steps Towards Adaptive Psychomotor Instruction.* Paper presented at the The Twenty-Seventh International Florida Artificial Intelligence Research Society (FLAIRS) Conference, Pensacola Beach, FL.

Behneman, A., Berka, C., Stevens, R., Vila, B., Tan, V., Galloway, T., Johnson, R. & Raphael, G. (2012). Neurotechnology to accelrate learning: during marksmanship training. *IEEE Pulse*, 3(*1*), 60–63.

Berka, C., Behneman, A., Kintz, N., Johnson, R. & Raphael, G. (2010). Accelerating training using interactive neuro-educational technologies: applications to archery, golf, and rifle marksmanship. *International journal of Sports and Society*, *1*(4), 87–104.

Birnbaum, M. S., Kornell, N., Bjork, E. L. & Bjork, R. A. (2013). Why interleaving enhances inductive learning: The roles of discrimination and retrieval. *Memory & Cognition*, *41*(3), 392–402.

Chung, G., Nagashima, S., Delacruz, G., Lee, J., Wainess, R. & Baker, E. (2011). Review of Rifle Marksmanship Training Research. Report 783. *National Center for Research for Research on Evaluation, Standards, and Student Testing (CRESST).*

Cubic. (2014). Cubic Showcases New Technology at I/ITSEC 2014. Retrived on 05 February 2016 from: http://www.cubic.com/News/Blog/Articles/ID/1147/Cubic-Showcases-New-Technology-at-IITSEC-2014.

Ericsson, K. A. (2006). The influence of experience and deliberate practice on the development of superior expert performance. *The Cambridge handbook of expertise and expert performance*, 683–703.

Ericsson, K. A. (2014). *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games*: Psychology Press.

Goldberg, B., Amburn, C., Brawner, K. & Westphal, M. (2014). *Developing Models of Expert Performance for Support in an Adaptive Marksmanship Trainer.* Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

James, D. R. & Dyer, J. L. (2011). Rifle Marksmanship Diagnostic and Training Guide: DTIC Document.

Sottilare, R., Sinatra, A., Boyce, M. & Graesser, A. (2015). Domain Modeling for Adaptive Training and Education in Support of the US Army Learning Model--Research Outline. Aberdeen Proving Ground (MD): Army Research Laboratory (US); Report No.: ARL-SR-0325.

Sottilare R. (2015). Challenges in moving adaptive training and education from state-of-art to state-of-practice. Presented at Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice Workshop at the 17th International Conference on Artificial Intelligence in Education Workshop; 2015 Jun; Madrid, Spain.

Sottilare, R. A., Ragusa, C., Hoffman, M. & Goldberg, B. (2013). *Characterizing an Adaptive Tutoring Learning Effect Chain for Individual and Team Tutoring.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Sottilare, R. (2013). Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model--Research Outline. Aberdeen Proving Ground (MD): Army Research Laboratory (US), Report No.: ARL-SR-0284.

CHAPTER 18 – Domain Modeling in AutoTutor

Zhiqiang Cai, Arthur Graesser, and Xiangen Hu
University of Memphis

Introduction

As a generalized framework for intelligent tutoring systems (ITSs), the Generalized Intelligent Framework for Tutoring (GIFT) integrates various types of ITS modules. Among all the modules, conversational ITS modules might be the most complex one, since it adds complex conversations to any type of ITS environment. Each type of module in ITS may have its specific considerations on domain modeling. This chapter presents our experience in domain modeling on "AutoTutor", which has been integrated as a conversation module in GIFT.

The domain model has long been considered as one of the major components of an ITS. For example, Elsom-Cook (1997) assumed that the structure of an ITS is around a "trinity" of components: "the knowledge about the domain being taught, about how to interact with the learner, and about the student." Murray (1999) described four models in authoring ITSs: the interface model, the domain model, the teaching model, and the student model. In the preface of the first book of this series, Sottilare, Graesser, Hu and Holden (2013) indicated that domain models in an ITS contain "the set of skills, knowledge, and strategies of the topic being tutored." They added that a domain model "normally contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit."

Obviously, the term "domain" in an ITS means a specific field or scope of knowledge, such as "algebra", "critical thinking", "psychology", etc. People who have a deep understanding of a domain are called "domain" experts. A domain model represents domain experts' ideas, skills, and the ways that they solve domain problems. In ITS development, domain experts are responsible for authoring domain knowledge. A good domain model provides a structure to minimize domain experts' authoring time and maximize the quality of the content.

Domain knowledge can be decomposed into "knowledge components" (KCs). Koedinger, Corbett, and Perfetti (2012) defined a KC as "an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks." This definition connects knowledge, learner, and tasks, which relates the domain model to the student model. For practical reasons, Koedinger et al. (2012) generalized the definition of KCs as "pieces of cognition or knowledge" that include "production rule", "schema", "misconception", "concept", "principle", "fact", or "skill".

A collection of KCs represents the knowledge of a domain. However, the domain knowledge cannot be taught only by directly presenting the KCs in a didactic manner as in a classroom lecture. A tutoring system usually requires more active application of KCs by giving students a set of problems. By solving the problems, students learn the domain knowledge at a deeper level. Moreover, we believe that a problem can be decomposed into a set of expectations (i.e., steps in a problem or answers to a question) and misconceptions (i.e., incorrect mental models, errors, bugs). KCs are mapped to expectations/misconceptions in problems. Students' knowledge about the KCs is then assessed through the performance on solving the problems (Figure 1).

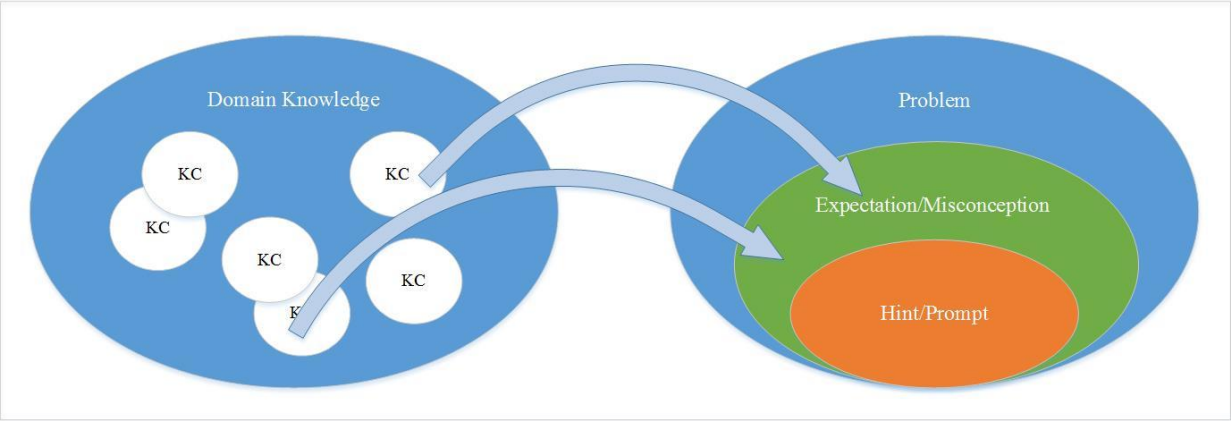


Figure 1. Mapping KCs to expectations and misconceptions.

AutoTutor (see the next section) is a problem-oriented tutoring system. Students learn by working on a solution to the problem. The system presents interactive learning content to students and helps students with conversations through animated agents. It is the highly focused intelligent conversation that makes AutoTutor different from many other ITS. In this chapter, we provide the latest AutoTutor domain model and the process of authoring AutoTutor domain knowledge.

AutoTutor Problems and Knowledge Components

An AutoTutor system usually contains a collection of problems or lessons that cover the content of a specific domain. For example, the computer literacy AutoTutor (Graesser, Wiemer-Hastings, Wiemer-Hastings, Kreuz & the Tutoring Research Group, 1999) contains 37 problems that cover the topics of hardware, operating system, and the Internet. Physics AutoTutor (Graesser, Lu, Jackson, Mitchell, Ventura, Olney & Louwerse, 2004; VanLehn et al., 2007) contains 11 practical problems that cover some of Newtonian laws in conceptual physics. Operation ARIES, an AutoTutor system that teaches critical scientific thinking, contains 21 training lessons, 21 case study problems, and 16 practice problems (Millis, Forsyth, Butler, Wallace, Graesser & Halpern, 2011; Halpern, Millis, Graesser, Butler, Forsyth & Cai, 2012; Cai, Forsyth, Germany, Graesser & Millis, 2012; Forsyth, Graesser, Pavlik, Cai, Butler, Halpern & Millis, 2013). CSAL AutoTutor, a system that helps adults in reading, contains over 30 lessons that cover a complete set of reading comprehension strategies (Graesser, Cai, Baer, Olney, Hu, Reed & Greenberg, in press).

How do we know that a collection of problems/lessons covers the knowledge of a specific domain? How are the KCs overlapped across the problems? How are the KCs repeated in the problems? In order to answer these questions, a complete set of KCs about the domain needs to be identified. Identifying domain KCs is a task for domain experts. Domain experts need to be aware of the following three issues in creating a KC list:

(1) KCs need to be at a proper grain level. On one hand, coarse-level KCs will cause inaccurate assessment about students' learning. On the other hand, there could be too many fine-level KCs to be practically managed by tutoring systems.

(2) Identifying some relations between KCs could be useful. One important relation is "prerequisite". Prerequisite relation is useful for adaptive problem selection. Unfortunately, this relation is often debatable. Another important relation is degree of overlap. Consider two KCs, A and B, the over-

lap relation could be A contains B, B contains A, A and B have overlap and none of them contains the other, or A and B do not overlap. This relation helps assessing students' mastering of the knowledge. For example, if the performance of task shows that a student mastered the KC A, and A contains B, then we can infer that the student mastered B.

(3) KCs need to be complete. However, it might be impossible to make a complete and still manageable list of KCs. Domain experts may need a principled iterative process to make a good judgement on the completeness.

Once KCs are identified, the tutoring problems/tasks can then be designed around the KCs. In AutoTutor, a problem usually provides the learner about a half hour of interactions. Every problem/task should be associated with a subset of the KCs. The collection of the problems should cover all KCs enough times so that the student's learning curves (Koedinger et al., 2012) can be traced and the knowledge mastering can be accurately assessed.

The mapping from KC to problems (see Figure 1) can help with revising both problems and KCs. By systematically inspecting the KCs in each of the problems, missing KCs or relations may be found. Problems may be added, removed, and revised in order to achieve a complete and balanced coverage of the KCs. For example, Table 1 shows a part of the mapping from KCs to problems of our ElectronixTutor, which is under development in a project funded by Office of Naval Research (ONR). The labels of the row entries refer to the type of learning problem, such as deep reasoning (DR), knowledge check (KC), and particular question (Q). The labels of the column entries refer to the knowledge components. As can be seen in the cells, there are three problems that cover capacitor-behavior, but only one that covers capacitor-parameter. Therefore, an additional problem may be added to allow more coverage of capacitor-parameter.

Table 1. Mapping between problem and KCs.

	Capacitor-parameter	Capacitor-behavior	RC-Circuit-function	LC-Circuit-function	Inductor-behavior
DR-Q1		1	1	1	1
DR-Q2			1	1	
KC-Q1	1	1			
KC-Q2					1
KC-Q3		1			
KC-Q4			1	1	

AutoTutor Scenes and Conversations

Graesser (in press) often describes AutoTutor as a system that helps students learn by holding a conversation in natural language. Intelligent natural language conversation is the feature that makes AutoTutor different from many other tutoring systems. AutoTutor conversations are usually accompanied by an adaptive sequence of "scenes". An AutoTutor "scene" can be built with a set of simple HTML5 pages or a complex 3D virtual environment. Figure 2 shows three scenes in a CSAL lesson composed by three HTML5 pages. On the first page, computer agents give an introduction to the steps about how to use context clues to get the meaning of unknown words. The second page presents a text with a highlighted target word and three definitions for a student to choose. The tutor agent (left) asks the student agent (right) and the human student to choose the correct definition. Based on their answer, the tutor agent gives immediate feedback ("Great", "Correct", "No", etc.), followed by a hint or prompt if help is needed, and points out

the context clues in the text. The third page presents a new target word and gives the human student a new opportunity to learn the use of context clues.

Figure 2. CSAL AutoTutor scenes.

AutoTutor scenes based on HTML5 pages are simple. Yet, it can present rich types of interactions to students, such as multiple choice questions, fill-in-the-blank questions, rearranging sentences, and so on. Of course, the most important interaction from the standpoint of AutoTutor is the natural language conversation. AutoTutor scenes and conversations are adaptively presented. The next scene to be presented is usually determined by student's performance on the interactions of the past scenes. At each tutoring turn, a student's action, verbal input, and emotions are sent through the client interface to the AutoTutor Conversation Engine (ACE). ACE sends back a sequence of instructions to the client program to make changes to the scene, move to a selected next scene, and send speeches to computer agents.

There are many HTML5 e-learning authoring tools that can be used to create attractive and interactive HTML5 pages to present instruction content. However, creating adaptive interactions mixed with intelligent natural language conversation is challenging to domain experts. The reason is that "adaptivity" implies complex logic and needs expansive supporting resources. In order to make the interactions adaptive, the system needs to track student performance, which involves student modeling, data storage, and data analysis. In order to make intelligent natural language conversation, the system needs to have natural language processing (NLP) resources. Responding to mixed user inputs (actions, verbal inputs, emotions, etc.) needs complex production rules.

The latest AutoTutor authoring tool allows domain experts to create new instruction content by modifying template HTML5 pages without worrying about the complex adaptivity logic. An AutoTutor template page contains a set of predefined elements. Domain experts modify texts, images, videos, and agent speeches associated with the template page to make a customized content page. A "player" page is responsible for loading content pages and transferring messages among a loaded content page, animated agents and ACE.

While customizing a template page is simple, authoring an AutoTutor template is not. A complete AutoTutor template contains a collection of template pages, a "player" page, together with default texts, images, videos, and speeches. A template also has a set of production rules, cascading style sheets (CSS) files and JavaScript files. A good template may involve authors from different fields. A web page designer is responsible for the look and feel and the usability of the pages; a JavaScript programmer is responsible for interaction functions; and an AutoTutor rule designer is responsible for production rules. Consequently, the tasks of a domain expert are to focus on the content, whereas a community of other experts handles technical details that are beyond the proficiencies of most domain experts.

Template authors have the freedom to create any kind of template page. However, all template pages have required features:

(1) **Adaptive** - Pages are *adaptively* presented to learners, in the sense that the learner's performance on the presented pages determines which page is presented next. The learner's performance scores are carried over from page to page until the end of the problem.

(2) **Conversational** - Conversational agents are available on each page. When working on a page, a learner can get a better understanding of the presented content by having deep conversations with the conversational agents.

(3) **Interactive** – The pages are supposed to be interactive. That is, there should be interaction elements, such as clickable buttons, draggable objects, highlightable texts, selectable items, editable text areas, and so on.

(4) **Changeable** - An AutoTutor template page is designed for change. That is, the contents of the page are designed to be easily changed by domain experts who may not have any understanding of HTML/JavaScript/CSS. All changeable elements have attributes stored in JavaScript configuration files for easy manipulation.

Template pages also have a set of required JavaScript functions:

(1) Init() - Initialize elements and notify the player the page is ready.

(2) Lock() - Lock the interactive elements.

(3) Unlock() - Unlock the interactive elements.

(4) GetMessage(string messageName) - Prepared for the player to request a message.

(5) GetEvent(string action) - Prepared for the player to send an action to the page for execution, such as "GotoNextItem".

(6) SendEvent(string message) - Notify player with a message.

(7) GotoNextPage() - Replace the current page by another page.

(8) UpdateScore() - Update user's performance score.

(9) GetStepName() – Report the name of the current step on the page.

(10) GotoStep(string stepName) - Move to the step specified by the step name.

The above HTML5 page-based AutoTutor system is a way to deeply integrate conversation with multimedia content. This allows learners to read, watch, and do while they talk. We want to point out here that the above template-based knowledge model can be easily integrated into GIFT, especially the web version. The simplest way is to create an iFrame on any web page to hold an AutoTutor "player" page. The communication between GIFT modules and AutoTutor can be done through JavaScript messaging.

AutoTutor Conversation Scripts

With the help of the AutoTutor authoring tool and templates, domain experts can easily represent a problem by a set of interactive HTML5 pages. Domain experts may find it difficult to modify the default conversation scripts in a template without a good understanding of the AutoTutor conversation script elements. We give a brief description of the elements in this section. More information can be found in AutoTutor papers such as Cai, Feng, Baer, and Graesser (2014); Graesser (in press); and Nye, Graesser, and Hu (2014).

There are two types of conversation packs in AutoTutor. One is called "rigid pack" and the other is called "tutoring pack". A rigid pack contains a fixed sequence of conversations between computer agents. Rigid packs are usually used in the opening and closing of a conversation.

A tutoring pack has a complex structure because it contains the spoken contributions to adaptively respond to students' inputs. Domain authors start modifying a template tutoring pack from a "main question". The main question is the central part the conversation. Once the main question is answered by the student, the conversation goes to the closing rigid pack and ends. Therefore, the length of a conversation depends on the complexity in answering the main question. If the main question is very simple, then there is not much to talk about and the conversation will usually be short. When the main question requires a complex answer, the conversation may be long, and complex conversation scripts will be needed.

In AutoTutor, a complex answer to the main question is decomposed into a set of expectations. An expectation is a part of the answer to the main question, usually in the form of a single sentence that corresponds to a knowledge component. All expectations collectively answer the main question. During every step of the conversation, a student's input will be compared with each expectation. If an expectation is not matched, AutoTutor will provide the student a hint or a prompt to help the student construct an answer that matches the expectation. In AutoTutor, a hint is a question that requires an answer in the form of a sentence or clause. A prompt is a question that targets a single highly relevant word in the expectation. Domain experts need to author these hints and prompts.

An AutoTutor conversation script also contains answers to the main question, expectations, hints, and prompts. The answers are prepared for two purposes. One is to compare with students' inputs and the other is for computer agents to speak. The answers can be in different types. Two important types are "Good" and "Bad". If a question could be answered in different ways, multiple "Good" answers can be prepared. When a student's input matches a "Good" answer, the question is considered answered. A "Good" answer may be delivered by a computer agent if the student cannot answer the question. When a "Bad" answer is matched, AutoTutor will give a negative feedback and continue with a new hint or prompt to further help the student.

In recent years, AutoTutor systems often use *trialogs* (Cai et al., 2014; Cai, Graesser & Hu, 2015; Graesser, Li & Forsyth, 2014). A trialog involves two computer agents, usually a "tutor" agent and a "student" agent. The questions and answers can be prepared for both agents. Both the tutor agent and the student agent can ask a question to the human student. When authoring such questions, domain experts need to write the questions in the language of a tutor and a student respectively. "Good" answers can be prepared for both agents. However, "Bad" answers are only for student agent. "Bad" answers play an important role in trialogs. When a student's answer is bad, it is most helpful if the system can immediately give negative feedback, such as "No", or "That's not right". However, because the NLP module is never perfect, there is a chance that a correct answer from a student matches a bad answer. Obviously, giving a negative feedback to a correct answer is awful and harmful to learning. Trialogs can help with this. When a student's input matches a bad answer, AutoTutor may respond in the following way:

- The student agent says something like "I think I know the answer".
- The student agent says the matched bad answer.
- The tutor agent gives a negative feedback to the student agent.

Trialogs also allow the use of pedagogical strategies, such as vicarious learning, tutoring, and teachable agent. More about this can be found in Cai et al. (2014).

Conclusion

GIFT has various types of modules. Two important issues make conversational modules different from typical types of GIFT modules. First, the conversational scripts need to contain a family of users' possible responses as opposed to a singular response. Second, the conversation rules are complex, especially when users' actions and environmental states are included as rule conditions. Over the last two decades, we have allocated considerable efforts into developing AutoTutor script authoring tools (Cai et al., 2015; Hu, Cai, Han, Craig & Graesser, 2009; Susarla, Adcock, Van Eck, Moreno & Graesser, 2003). We believe that template-based authoring is the best approach to use. While we are continuously improving AutoTutor Script Authoring Tools, it is our hope that the domain model we provided here can be applied to other GIFT modules. We recommend the following for GIFT domain model authoring tools:

- Minimal technical background knowledge should be assumed on domain experts. A working template should be provided for domain experts to create content of similar style and functionality as in previous applications.
- Template authoring tools are needed for technical experts that embrace different skills, such as understanding language, global discourse patterns, pedagogy, sensing technologies, programming, art, and so on.
- Authoring conversations is hard. Conversation authoring tools should provide a mechanism for iterative authoring so that domain experts can review users' input and continuously make improvements to conversation scripts.

Acknowledgments

The research was supported by the National Science Foundation (SBR 9720314, REC 0106965, REC 0126265, ITR 0325428, REESE 0633918, ALT-0834847, DRK-12-0918409, 1108845), the Institute of Education Sciences (R305H050169, R305B070349, R305A080589, R305A080594, R305A090528, R305A100875, R305C120001), Army Research Lab (W911INF-12-2-0030), and the Office of Naval Research (N00014-00-1-0600, N00014-12-C-0643). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, Institute of Education Sciences (IES), or the Department of Defense (DOD).

References

Cai, Z., Feng, S., Baer, W. & Graesser, A. (2014). Instructional strategies in trialog-based intelligent tutoring systems. In R. Sottilare, A.C. Graesser, X. Hu & B. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Instructional Management* (pp. 225–235). Orlando, FL: US Army Research Laboratory.

Cai, Z., Forsyth, C. Germany, M. L, Graesser, A. & Millis, K. (2012). Accuracy of tracking student's natural language in OperationARIES!: A serious game for scientific methods. In S. A. Cerri and B. Clancey (Eds.) *Proceedings of the 11th International Conference on Intelligent Tutoring Systems* (*ITS 2012*) (pp. 629–630). Berlin: Springer-Verlag.

Cai, Z., Graesser, A. C. & Hu, X. (2015). ASAT: AutoTutor Script Authoring Tool. In R. Sottilare, A.C. Graesser, X. Hu & K. Brawner (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools & Expert Modeling Techniques* (pp. 199–210). Orlando, FL: US Army Research Laboratory.

Elsom-Cook, M. (1993). Student modeling in intelligent tutoring systems. *Artificial Intelligence Review 7*, 227-240.

Forsyth, C. M., Graesser, A. C., Pavlik, P., Cai, Z., Butler, H., Halpern, D. F. & Millis, K. (2013). OperationARIES! methods, mystery and mixed models: Discourse features predict affect in a serious game. *Journal of Educational Data Mining*, *5*, 147–189.

Graesser, A.C. (in press). Conversations with AutoTutor help students learn. *International Journal of Artificial Intelligence in Education*.

Graesser, A. C., Cai, Z., Baer, W., Olney, A. M., Hu, X., Reed, M. & Greenberg, D. (in press). Reading comprehension lessons in AutoTutor for the Center for the Study of Adult Literacy. In S. Crossley and D. McNamara (Eds.), *Adaptive educational technologies for literacy instruction*. New York: Routledge.

Graesser, A. C., Li, H. & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science, 23*, 374–380.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A. M. & Louwerse M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments & Computers, 36*, 180–193.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R. & the Tutoring Research Group (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive System Research, 1,* 35–51.

Halpern, D. F., Millis, K., Graesser, A. C., Butler, H., Forsyth, C. & Cai, Z. (2012). Operation ARA: A computerized learning game that teaches critical thinking and scientific reasoning. *Thinking Skills and Creativity*, *7*, 93–100.

Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T. & Graesser, A. C. (2009). AutoTutor Lite. In V. Dimitrova, R. Mizoguchi, B. Du Boulay & A. C. Graesser (Eds.), *Proceedings of 14th International Conference on Artificial Intelligence in Education. Building Learning Systems that Care: From Knowledge Representation to Affective Modelling* (p. 802). Amsterdam: IOS Press.

Koedinger, K. R., Corbett A. T. & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science 36*(5), 757–798.

Millis, K, Forsyth, C., Butler, H., Wallace, P., Graesser, A. & Halpern, D. (2011). Operation ARIES! A serious game for teaching scientific inquiry. In M. Ma, A. Oikonomou & J. Lakhmi (Eds.*) Serious games and edutainment applications* (pp.169–196). London, UK: Springer-Verlag.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education* 10(1), 98–129.

Nye, B.D., Graesser, A.C. & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education. 24*, 427–469.

Sottilare, R., Graesser, A.C., Hu, X. & Holden H. (2013). Preface. In R. Sottilare, A.C. Graesser, X. Hu & H. Holden (Eds). Design Recommendations for Intelligent Tutoring Systems: Learner Modeling (pp. ii-xiii). Orlando, FL: US Army Research Laboratory.

Susarla, S., Adcock, A., Van Eck, R., Moreno, K. & Graesser, A. C. (2003). Development and evaluation of a lesson authoring tool for AutoTutor. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo & K. Yacef (Eds.), *AIED2003 Supplemental Proceedings* (pp. 378–387). Sydney: University of Sydney School of Information Technologies.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A. & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, *31*(1), 3–62.

CHAPTER 19 – Modeling Mathematical Reasoning as Trained Perception-Action Procedures

Robert L. Goldstone[1], Erik Weitnauer[1], Erin R. Ottmar[2], Tyler Marghetis[1], and David H. Landy[1]
[1]Indiana University, [2] Worcester Polytechnic Institute

Introduction

For the last several years, our group has been involved in a project that can be construed in terms of exploring the relation of formal knowledge and perception. At first sight, formal cognition as seen in scientific and mathematical reasoning involves developing deep construals of phenomena that run counter to untutored perception. In fact, Quine (1977) considered a hallmark of advanced scientific thought to be that it no longer requires notions of overall perceptual similarity as the basis for its categories. The background rationale for this claim is that unanalyzed perceptual similarities may lead one astray. For example, marsupial wolves may closely resemble placental wolves, but they are evolutionarily rather distant cousins. In general, as a scientist or child (Carey, 2009) develops more complete, systematic knowledge about the reasons something has a property, then overall perceptual similarity becomes decreasingly relevant for generalizations.

While there is certainly justification for placing superficial perception and principled understanding in opposition, we have been exploring the converse strategy of trying to ground scientific and mathematical reasoning in perceptual processing (see also Kellman, Massey & Son, 2010). One reason for thinking that perception and formal thinking can be brought closer together is that we can train our perceptual processes to do the right thing, formally speaking. Perception and visual attention are highly educable processes. We can train our perception to process stimuli in an efficient manner for tasks that are important to us. An advantage of linking high-level processes to perception is that we can co-opt our neurologically large and phylogenetically early perceptual areas – areas that are the result of millions of years of evolutionary research and development. Finally, there are suggestive correlations across individuals between perceptual and conceptual abilities (Goldstone & Barsalou, 1998). For example, schizophrenics have difficulty inhibiting both inappropriate thoughts and irrelevant attributes. Autistics often suffer from overly selective attentional processes, including hypersensitivity to sensory stimulation and overly narrow language generalization.

Based on these considerations, we have developed a hypothesis we call Rigged Up Perception-Action Systems (RUPAS), which states that an important way to efficiently perform sophisticated cognitive tasks is to convert originally demanding, strategically controlled operations into learned, automatically executed perception and action processes. These tasks can be understood as on par with the "visual routines" proposed by Shimon Ullman (1984) to account for how people extract information from a visual scene using processes such as shifting attentional focus, indexing items, tracing boundaries, and spreading activation from a point to the boundary of an area. In this chapter, we apply this general theoretical approach of exploring ways in which our sophisticated, formal reasoning abilities are grounded in perception and action to the specific knowledge domain of algebra. Algebra is one the clearest case of widespread symbolic reasoning in all human cognition (Anderson, 2007). Showing that perceptual factors influence even algebraic reasoning provides prime facie support for the premise that perception-action grounding cannot be ignored for almost any cognitive task.

Related Research

To study the influence of perceptual grouping on mathematics, we gave undergraduate participants a task to judge whether an algebraic equality was necessarily true (Landy & Goldstone, 2007a). We were interested in whether perceptual and form-based groupings would be able to override participants' general knowledge of the order of precedence rules in algebra, according to which multiplications are executed before additions. We tested this by having perceptual grouping factors either consistent or inconsistent with order of precedence. For example, if shown the stimuli in the top row of Figure 1, participants would be asked to judge whether f + z * t + b is necessarily equal to t + b * f + z. In fact, this is not a valid equality. However, in the incongruent version of the physical spacing manipulation, the narrow spacing around the "+" signs might encourage participants to group the f and z together to form a "f + z" unit, as well as forming a "t + b" unit. If participants then match up these units on the left and right sides of the equation, they will find the same two units on the right side, leading them to respond "valid." As predicted, participants make 38% more errors on trials like this in which the formally determined order of operations is incongruent rather than congruent with the perceptual grouping suggested by the surround lines and circles. Other methods for manipulating perceptual groupings, like varying the connectedness of dots surrounding the mathematical expression (middle row) and proximity in alphabet (bottom row) also affect validity judgments. Participants continued to show large influences of grouping on equation verification even though they received trial-by-trial feedback. Feedback reduced, but did not eliminate the influence of these perceptual cues. This suggests that sensitivity to grouping is automatic or at least resistant to strategic, feedback-dependent control processes.

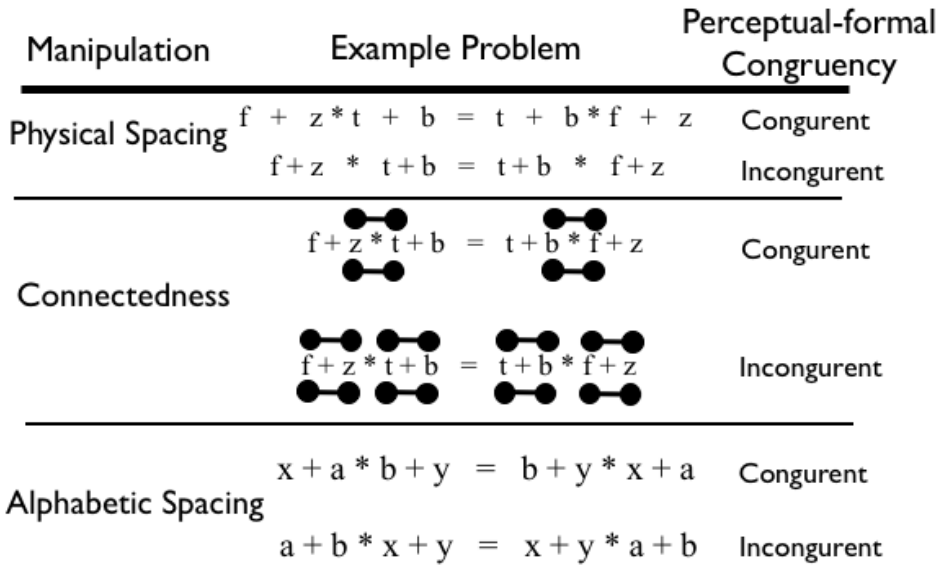


Figure 1. Samples from three experiments reported by Landy and Goldstone (2007b). Participants were asked to verify whether an equation is necessarily true. Grouping suggested by factors such as physical spacing, connectedness of contextual geometric forms, and proximity in the alphabet, could be either congruent or incongruent with the order of precedence of arithmetical operators (e.g., multiplications are calculated before additions). The physical manipulations shown bias participants to perceptually group the symbolic expressions. When formed perceptual groups are congruent with formal order of precedence then validity judgments are much more accurate than when they are incongruent.

Other research (Landy & Goldstone, 2010) indicates that people are heavily influenced by groupings based on perceptual properties when performing not only algebra but simple arithmetic as well. Despite being reminded of, and verbally subscribing to, standard order of precedence rules, our college student participants are much more likely to calculate an incorrect solution value of 25 for "2+3 * 5 = ?" than "2 + 3*5 = ?," presumably because the narrow spacing around the "+" in the former case biases people to calculate 2+3 before they multiply by 5.

People not only respond to the perceptual cues contained within symbolic representations, but they also add perceptual cues when they construct their own symbolic representations. Landy and Goldstone (2010) asked participants to write symbolic mathematical expressions for equations expressed in English such as "nine plus twelve equals nine plus three times four." Figure 2 shows an example of one participant's symbolic expression. From these expressions, we measured the physical space around the different operators. On average, the physical spacing was largest around "=", consistent with its role as the highest level structural grouping for the equation. The physical spacing was larger around the "+" than around the "X" in equations that had both of these operators. Our account of this result is that people produce notations that their own perceptual systems are well prepared to process. One noteworthy aspect of this empirical result is that people are creating perceptual cues that help them do the formally right thing even when this activity is not modeled for them by textbooks. Our corpus analysis reveals that most mathematics textbooks depict equal spacing around multiplications and additions. So, even though books do not use physical spacing to help students form useful perceptual groups in this instance, students still discover this practice. Furthermore, students who place wider physical spacing around lower, compared to higher, precedence operators also tend to produce the correct answer to math and logic problems (Landy & Goldstone, 2010). In this manner, we create notations that are aptly processed by our perceptual systems, and this is one of the reasons why perceptual systems should often be trusted rather than trumped.

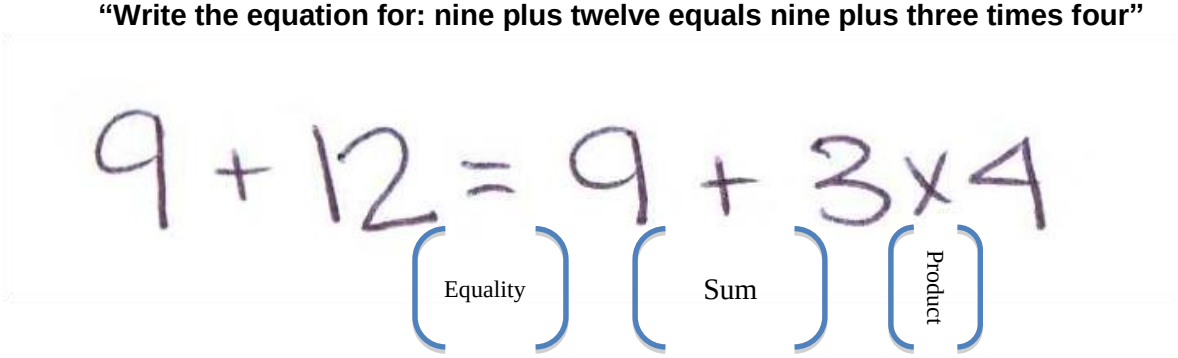


Figure 2. An example of a participants' drawn symbolic representation of an equation expressed in an English sentence, taken from Landy and Goldstone (2010). The physical spacing around the "=", "+", and "X" were measured and compared. Notice how in the drawn equation, the widest spacing is found around the "=" sign, followed by the "+" and then by the "X".

Rigged Up Perception Systems

The preceding examples illustrate ways in which we rely on perceptual processes to process symbolic notations. The lingering worry is that, as is the case with placental and marsupial wolves, or gold and pyrite (fool's gold), appearances may be misleading. If mathematicians use superficial perceptual cues to dictate how they will process mathematical notation, will they not often be led astray? One answer, described in the previous section, is that notations are not fixed and inflexible, but rather can be flexibly tuned to humans' perceptual systems because they are, after all, crafted by humans. This tuning occurs

within an individual's lifetime as with the case of physical spacing in students' written mathematical expressions, and also occurs on historic time scales. Much of the history of mathematical notation is one of changing over time to better fit human perceptual systems (Cajori, 1928). For example, the historic shift from representing "3 times the variable b plus 5" as "3×b+5," to later representing it as "3•b+5," and more recently as "3b+5," represents a consistent shift toward decreasing the spacing between operands that should be combined together earlier. Both individually and culturally speaking, we craft notations that let our perceptual systems better serve our cognitive needs.

As we tune our mathematical notations to fit our perceptual systems, we also tune our perceptual and attentional systems to fit math. People train their visual attention processes to give higher priority to notational operators that have higher precedence. The operator for multiplication, "×," attracts attention more so than does the notational symbol for the lower precedence addition operator, "+." People who know algebra show earlier and longer eye fixations to "×"s than "+"s in the context of math problems (Landy, Jones & Goldstone, 2008). Even when participants do not have to solve mathematical problems, their attention is automatically drawn toward the "×"s. When simply asked to determine what the center operator is for expressions like "4 × 3 + 5 × 2," participants' attention is diverted to the peripheral "×"s as indicated by their inaccurate responses compared to "4 + 3 + 5 + 2" trials (Goldstone, Landy & Son, 2010). The distracting influence of the peripheral operators is asymmetric as shown by the result that responding "×" to "4 + 3 × 5 + 2" is significantly easier than responding "+" in "4 × 3 + 5 × 2." That is, the operator for multiplication wins over the operator for addition in the competition for attention. This is not simply due to specific perceptual properties of "×" and "+" because similar asymmetries are found when participants are trained with novel operators with orders of precedence that are counterbalanced. The results suggest that a person's attention becomes automatically deployed to where it should be deployed to get them to act in accordance with the formal order of precedence in mathematics.

Rigged Up Action Systems

Mathematical symbol systems would not be very valuable if the only thing we could do with symbols was to perceive them. In fact, we transform symbols as well, to derive new implications and relations that lay dormant in their original form. One possibility is that symbolic transformations are executed internally using abstract representations. This account is tempting because notations like "2 × b = 14" seem to be straightforwardly translatable into hierarchical mental representations like "= (× (2, b), 14)." From this representation, propositional transformation rules like "= (× (a, b), c) ⇒ = (b, ÷ (c,a))" can be applied to solve for b. This kind of propositional transformational rule is powerful because of its generality and ability to operate on arbitrary inputs without any influence of their original spatial and perceptual properties (Newell & Simon, 1976).

However, as we have already seen, humans are indeed influenced by the spatial and perceptual properties of notations. Accordingly, the alternative account of symbolic transformation that we have pursued is to keep the symbolic form in its original spatial format, and apply spatial transformations within this world of notation. For the 2 × b = 14 problem, one candidate transformation is spatial transposition, in which the 2 is moved from the left side of the equality to the right side, where upon it is moved to the denominator of a 14/2 quotient. This spatial movement might be executed on paper or with number tiles if they are at the reasoner's disposal. More often, they are executed in the reasoner's mind. Although this transposition operation is highly intuitive, it is noteworthy that this kind of spatial transformation does not appear in most leading models of algebra (e.g., Anderson, 2007).

Displays like the one shown in Figure 3 were devised to measure if and when participants adopt a spatial transposition strategy for solving simple algebraic equations. Equations were superimposed on top of a vertically oriented grating that continuously moved to either the left or right. The movement of the grating

was either compatible or incompatible with the movement of numbers implicated by a transposition strategy. For the equation "4 * Y + 8 = 24" shown in Figure 3, a rightward motion of the grating would be compatible with transposition because, in order to isolate Y on the left side, the 4 and 8 must be moved to the right side. However, for the equation "24 = 4 * Y + 8," a rightward motion would be incompatible. Participants solved the equations more accurately when the grating motion was compatible with transposition.

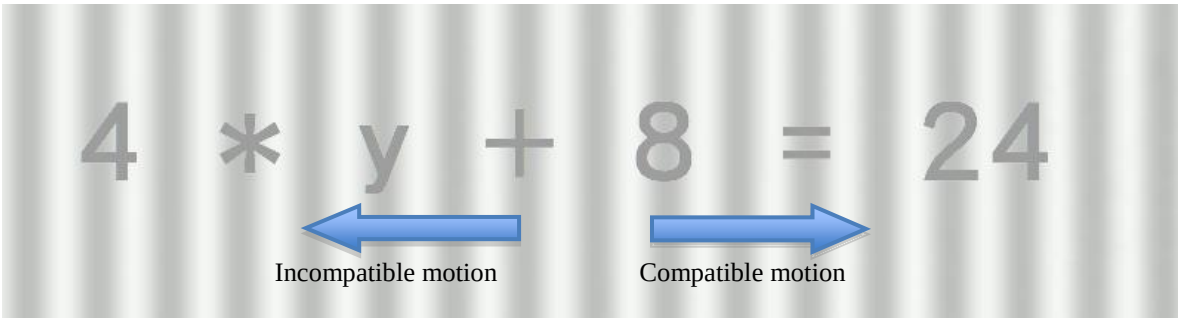


Figure 3. As participants solved for the variable in equations like the above, a vertically oriented grating continuously moved either to the left or to the right. Although irrelevant for the task, when the movement of the grating was compatible with the movements of the numbers required by spatial transposition, participants were more accurate.

The influence of background motion on algebraic solutions is consistent with a "visual routines" (Ullman, 1984) approach to mathematical cognition. According to this notion, people engage in dynamic, visual-spatial routines to perform perceptual computations. Of particular relevance to the perceptual leaning aspect of this transposition routine, we also found that participants who have taken advanced mathematics courses such as calculus are more affected by the compatibility of the background motion than students with less math experience. Accordingly, we conclude that the imagined motion strategy is a smart strategy that students come to adopt through experience with formal notations, rather than a strategy that students initially use while learning, and then abandon as their sophistication increases. Learned perceptual routines are not at odds with strong mathematical reasoning. They are often the means by which strong mathematical reasoning becomes possible. It is a smart strategy to take advantage of the scaffolding provided by space, using it as a canvas on which to project transforming motions.

Another result consistent with increasing use of space in notation with increasing mathematical sophistication is that older children rely more on physical spacing as a cue to perceptual organization than younger children. Braithwaite, Goldstone, van der Maas, and Landy (2016) analyzed a corpus of 65,856 8–12 year old Dutch children's solutions to simple math problems in which physical spacing was manipulated to be either congruent or incongruent with the formally defined order of operations. For example, the physical spacing in "2+7 × 5" is incongruent with the rule that multiplications are executed before additions, whereas "2 + 7×5" is congruent. Incorrect foils like 70, the answer that would be produced if the problem was incorrectly organized as (2+7) ×5, were much more common with the incongruent spacing. The fact that the difference in accuracy between incongruently and congruently spaced problems increased with age and math experience is not expected under the notion (e.g., Vygotsky, 1962) that mathematical development involves a shift from informal mechanisms to formal rules and axioms. Instead, the study shows that reliance on informal mechanisms can sometimes systematically increase with age.

Figure 4 shows other common actions related to mathematical reasoning. Each of them is a physical and spatial action that nonetheless can be made to align perfectly with formally valid operations. For example, if properly constrained, the operation of spatially swapping factors is formally sanctioned by the commutative property of multiplication. Likewise, the intuitive act of cancelling out the two 3's in the bottom

problem of Figure 4 can be formally sanctioned by a multiple step axiomatic derivation: (3×X)/(3×Y) ⇒ (3/3)×(X/Y) ⇒1×(X/Y) ⇒ X/Y. Future empirical work will be necessary to determine how often these actions are performed by mathematical reasoners and how effective they are. Our preliminary observations indicate that spatial actions like swapping A×B for B×A, splitting the a in a×(5+7) to form 5a + 7a, moving the 3 from the left side to the right side of the equality in X+3=8, simplifying 4×7 by projecting 28 on top of the original term, and canceling out the 3s in 3x/3y are commonplace and often times effectively deployed. While the notion that mathematical reasoning shifts toward abstraction as it develops is plausible, our initial observations of mathematicians "in the wild" suggest that they are at least as likely to employ these kinds of spatially concrete transformations as are less sophisticated reasoners. Sophisticated reasoners still use concrete actions – they just apply them more efficiently and felicitously.

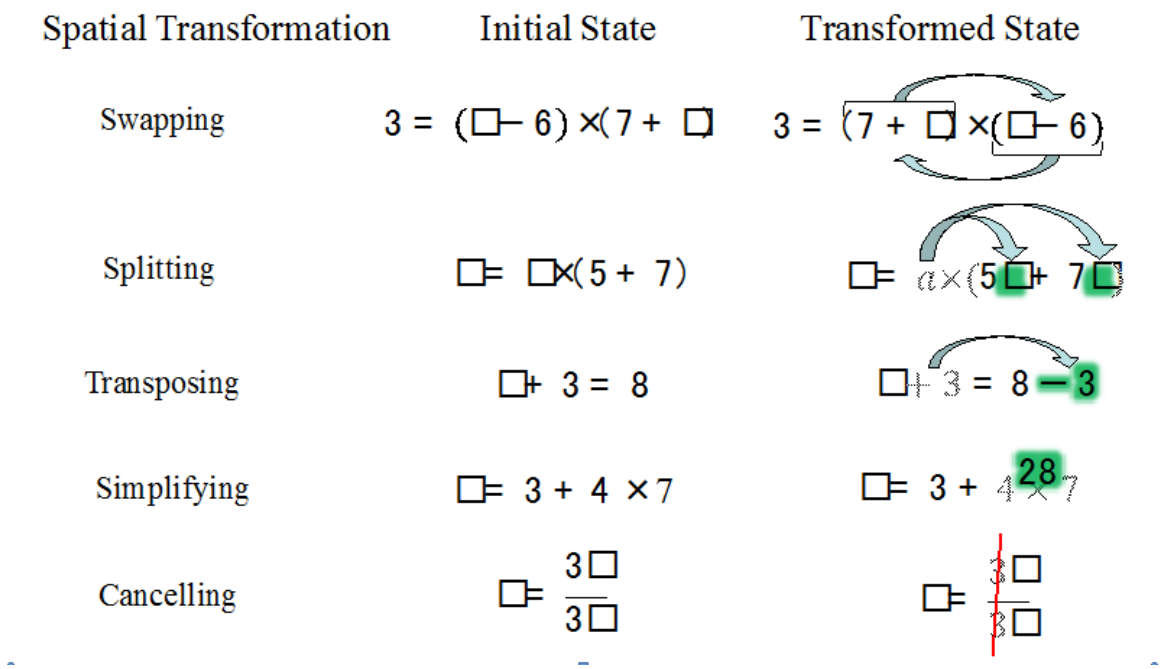


Figure 4. Examples of physical transformations within notational space. Operations like swapping factors, splitting a variable to make identical copies, transposing a term from one side of an equation to the other, simplifying by replacing one expression with another, and canceling factors in a numerator and denominator are commonly observed in mathematical reasoners, and are often employed in a cognitively efficient and valid fashion.

Discussion

On the basis of our laboratory investigations of rigged up perception and action routines for mathematical reasoning, we have implemented algebra tutoring software systems with a specific aim in mind – to help students rig up their perception and action systems for effectively processing algebraic notation and thinking mathematically. The currently most actively developed version of the system, named Graspable Math (http://graspablemath.com), allows students to interact in real time with math notation using perception-action processes. The system is a natural outgrowth of our empirical findings suggesting that people come to be proficient reasoners in science and mathematics not by ignoring perception, but by educating it (Goldstone, de Leeuw & Landy; 2015; Goldstone, Landy & Son, 2010; Landy, Allen & Zednik, 2014). Our intention is to construct a virtual sandbox for students to explore how algebra operates, and to devel-

op both intuitions and algorithms for performing mathematics (Ottmar, Landy, Goldstone & Weitnauer, 2015; Ottmar, Landy, Weitnauer & Goldstone, 2015).

One core design commitment of Graspable Math is that students must be able to intuitively see linkages between various components of mathematics. Figure 5 shows a screenshot from the system as a student works through the process of solving for two unknowns, x and y. The screenshot does not adequately show the real-time interactive experience and so we encourage readers to visit the project web page. The most immediate, intuitive linking is from one algebraic expression to the next via spatial transformations of the kind shown in Figure 4. Near the bottom right-hand corner of Figure 5, one can see that the user is picking up the -1 (shown in red), in the middle of the process, perhaps, of transforming the expression from y=-1+4x into y=4x - 1 or y+1=4x. The commutativity of addition is being effectively shown by the instantaneously reactive system; as the user moves the -1 to the right of 4x, the 4x moves over to give room to -1. If the -1 crosses the equal sign, it immediately transforms into a +1, letting users viscerally see the deep mathematical relation that if some X is equal to a function of some Y, then Y is also equal to the inverse of that function applied to X. Some teachers resist spatial transposition, viewing it to be an illegitimate algebraic transformation. They object, "You shouldn't teach students that they can just move the 2 of y-2=5 to the right side and changing its sign. Students should go through the axiomatically justified steps of adding 2 to both sides of the equation, yielding y-2+2=5+2, and then simplifying to y=5+2." To this objection, we respond that the teacher's preferred solution is *one* justifiable transformation pathway, but mathematics is rich enough to permit multiple axiomatizations of algebra, and the spatial transformations shown in Figure 4 provide an axiomatization that can also be shown to be formally valid. The traditional additive axiom of addition states: if two quantities are equal and an equal amount is added to each, they are still equal. The alternative, spatial axiomatization has three noteworthy advantages. First, it is a much more psychologically intuitive axiomatization because it has been designed to be efficiently processed by human perception-action systems. Second, it is more efficient, requiring two fewer transformations. Given that there is non-negligible "fail rate" (e.g., a student getting the wrong result, or giving up entirely) for each transformation, streamlining algebraic transformations is a valuable enterprise. Finally, the transposition operation makes intuitive the general mathematical pattern $X = F(X) \equiv F^{-1}(X) = Y$ that is almost completely hidden in the traditional, additive axiom of addition.

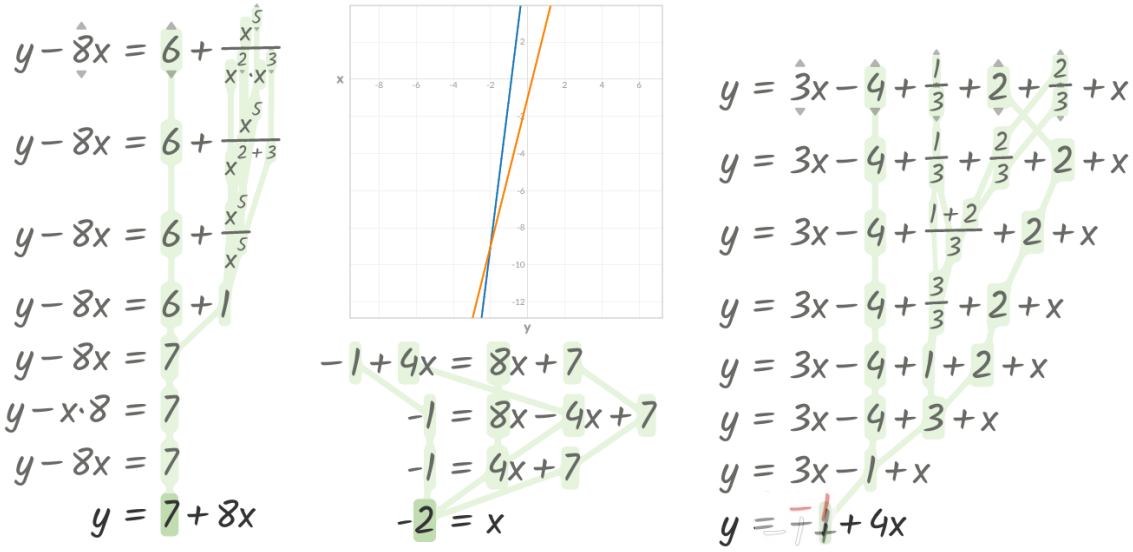


Figure 5. A screenshot from Graspable Math showing three kinds of linkages between representations. First, interactive animations, such as the moving red - 1 in the lower-right corner, provide intuitive, dynamic, and

spatial linkages between successive steps in an algebraic derivation. Second, the green pipelines show how symbolic elements are related to each other across several steps of a derivation. Third, the linear equations on the left and right sides are dynamically linked to the graph in the middle. Manipulating the graph's lines instantaneously affects the symbolic quantities in the equations, and vice versa.

Another kind of linkage that Graspable Math allows users to see and create is between algebraic notation and other representations. Figure 5 also shows a line graph corresponding to the two equations on each of its sides, where each equation is expressed as a line. The graph-algebra linkage is dynamic and interactive. As users scrub through different values for the constants in the algebraic notation, they see how those values affect the slope and intercept of the lines. Alternatively, if a user manipulates the slope or intercept of the line within the graph, the yoked symbolic numbers will automatically adjust. This kind of bidirectional linkage allows each representation to contribute to understanding where it shines most. Relations between lines are highly salient in the graph format. In particular, the intersection point between the lines is conspicuous and can provide an impetus for students to try to understand what is unique about that point. A fuller answer to that graph-inspired question is provided by the linked notation. In particular, by substituting what Y is equal to from one equation into the Y term for the other leads to the center equation shown below the graph in Figure 5. This center equation, expressed solely in terms of X because the Ys have been eliminated, can then be solved for X, which provides a symbolic tie-in to the point on the X-axis where the two lines intersect. An important part of our philosophy for Graspable Math is that neither the graph nor the algebraic notation is primary or privileged. Both can provide an improved understanding of the other, and when placed in correspondence, allow an understanding that transcends what either can provide on its own. For example, a student who first solves for X and Y algebraically may wonder what this solution looks like on the graph. Plotting the point corresponding to the solution of the two-equation system shown in Figure 5, {-2,-9}, then she augments the student's understanding of this solution to (literally) see it as the single point that lies on both of the lines corresponding to the two equations.

A third kind of linkage, depicted in green in Figure 5, is between the different steps of a derivation. Graspable Math uses interactive animations to show the spatial transformations that connect adjacent steps in a derivation, but often times it is useful for mathematical reasoners to see the overall correspondences between elements of a long derivation. In converting the equation of a line from the point-slope form of $y-7=3(x-2)$ to slope-intercept form of $y=3x+1$, it is illuminating to see why and how the intercept depends on both the point and the slope of the original form, but the slope does not. Graphically speaking, if a line needs to hit a particular point {2,7}, then as the slope of the line becomes increasingly shallow, it will have to hit the Y-intercept at increasingly high points. Algebraically speaking, the intercept is seen to depend on the y-coordinate of the point {2,7} in a directly proportional way, but is seen to depend on the x-coordinate in a multiplicative manner with the slope. Together, juxtaposition of the algebraic and graphical representations of linear systems takes advantage of the superior spatial relation highlighting of the graph representation and the superior highlighting of quantity dependencies of the algebraic representation.

We have engaged in a considerable amount of testing of Graspable Math in classroom and informal learning contexts. Students are generally highly enthusiastic and engaged when interacting with the system. Achieving these high levels of engagement is, in itself, a major achievement in educational design given that most students find algebra to be one of the least liked topics in all of their K-12 school experience! Moreover, on tests of transfer of mathematical reasoning to new problems presented in a paper-and-pencil format, Graspable Math has been shown to have educational benefits compared to standard practices for teaching the same content. When deploying Graspable Math in educational contexts, instructor-coaches should be aware of the potential for students to acquire "mal rules" – procedural transformations that are not formally valid (Sleeman, 1984). For example, because cancelling out terms is so aesthetically and kinesthetically agreeable for many students, there is some tendency to overgeneralize from those situa-

tions where spatial cancelling is valid (e.g., crossing out the 3's in 3x/3y) to those where it is not (e.g., (3+x)/(3+y)). One might draw the conclusion from these infelicitous perception-action generalizations that these seductively intuitive but dangerous perception-action routines should be avoided. Relying on formal and explicit rules may be safer (Kirshner & Awtry, 2004). However, we have had success by instead training perception-action routines to become more nuanced in their triggering conditions and deployment. One way to train this subtlety is by placing the forms 3x/3y and (3+x)/(3+y) side by side, showing that the cancelling transformation can apply to one but not the other, and then showing what the allowed spatial transformations of the latter form is (3/(3+y)) + (x/(3+y)). There is substantial pedagogical mileage to be gained by adopting a growth mindset rather than a fixed mindset (Dweck, 2012) toward our own perceptual abilities.

Recommendations and Future Research

The most immediate recommendation stemming from our research for the Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare et al., 2013) is to develop additional support for incorporating interactive, spatial simulations, and cultivating training within these rich environments. We have been impressed by the extent to which training, even for skills normally considered "intellectual," is most effectively enabled by educating perception and action routines. From our perspective, the GIFT initiative has excelled at incorporating explicit discourse and self-explanation into computer-based training systems. This activity could be effectively combined with our laboratory's emphasis on implicit perception-action training. Rather than treating explicit understandings and implicit perception-action routines as parallel and independent, the two routes to proficient skilled performance are mutually supportive and understanding their interplay in human learning is an exciting area for improving computer-based tutoring. GIFT already has modules that are well positioned to track learners' detailed actions. By leveraging these modules, and developing new capacities for mouse tracking, gesture recognition, open-ended procedurally generated simulations, and dynamic systems, a general platform could be built for studying how skill develops from synergies between implicit and explicit learning.

When thinking about a content domain, there is a tendency to focus on the facts, explicit strategies, and formal models that underlie the domain. While these are certainly important components, it is easy to neglect the more implicit "feels" that an expert develops for a domain (Goodwin, 1994). These feels are often underappreciated precisely because they are implicit and hard to put into words. Even for the apparently abstract and formal content domain of algebra, proficient practitioners develop feels that have a surprisingly large impact on what they are able to accomplish. These feels develop at both input and output ends of information processing. On the input end, mathematicians develop strong dispositions to perceptually organize and reorganize mathematical objects in ways that help them see patterns that are important to them. On the output end, mathematicians develop action routines that help them transform math in revealing ways. Moreover, the input and output sides are inextricably linked because what we perceive influences the actions we generate, and our actions transform the world to make fruitful perceptions more likely (Landy & Goldstone, 2007).

Given this perspective, our foremost recommendation going forward is to consider ways to incorporate perception-action procedures into models of domain knowledge. For education purposes, it is beneficial for coaches, trainers, teachers, and students to think about ways of adapting how to see and how to act. Although this kind of knowledge can be considered to be procedural, it is quite different from traditional procedural knowledge that is modeled after following recipes, rules, or algorithms. In algebra, it is one thing to explicitly know that the distributive property of multiplication over addition can be used to transform 3(X+Y) into 3X+3Y and another thing to reliably enact the spatial action routine that splits the 3 into the proper number of terms within the parenthesis and constructs the formally valid written expression. Perception and action procedures will likely involve different effective training techniques than

knowledge that involves either declarative knowledge or recipe following (Koedinger, Booth & Klahr, 2013). Compared to more explicit knowledge, embodied and grounded perception-action routines seem to benefit from prolonged and spaced practice, tight agent-to-environment coupling, scaffolded training support, and training that emphasizes active construction over passive study of solved problems.

Although explicit knowledge and perception-action procedures should be distinguished for purposes of optimizing their acquisition and use, it would be a mistake to treat the two kinds of knowledge as operating independently, in parallel. In fact, our second recommendation is to understand understanding itself as the interplay between explicit knowledge and implicit perception-action training. It may be tempting to try tackling the tasks of training explicit and implicit knowledge separately. Much of modern neuroimaging encourages treating different brain regions as modules whose activity levels can be separately assessed. This leads naturally to an approach toward training that stresses the importance of increasing or decreasing the activity of particular modules, much as one would practice curls in weightlifting to strengthen a particular muscle group such as the biceps. As an alternative vision, we think that successful training often involves the coordination of modules rather than their independent strengthening (Schwartz & Goldstone, 2016). Proficient mathematicians strategically think about ways of training their perception-action systems over time to do the Right Thing, formally speaking. They also think creativity and laboriously about ways to create coordinations between algebraic, geometric, topological, spatial, definitional, and model-based understandings of a situation. Developing learning contexts that allow our explicit and implicit understandings to mutually inform one another has great potential payoff for promoting learning outcomes that are efficient, robust, and broadly applicable.

References

Anderson, J. R. (2007). *How can the human mind exist in the physical world?* Oxford, England: Oxford University Press.

Braithwaite, D. W., Goldstone, R. L., van der Maas, H. L . J. & Landy, D. H. (2016). Informal mechanisms in mathematical cognitive development: The case of arithmetic. *Cognition, 149*, 40–55.

Cajori, F. (1928). *A history of mathematical notations*. Open Court Publishing Company: La Salle, Illinois.

Carey, S. (2009). *The Origin of Concepts.* New York: Oxford University Press.

Dweck, C. S. (2012). *Mindset: How You Can Fulfil Your Potential*. New York: Constable & Robinson Limited.

Goldstone, R. L. & Barsalou, L. (1998). Reuniting perception and conception. *Cognition, 65*, 231–262.

Goldstone, R. L., de Leeuw, J. R. & Landy, D. H. (2015). Fitting Perception in and to Cognition. *Cognition, 135*, 24–29.

Goldstone, R. L., Landy, D. H. & Son, J. Y. (2010). The education of perception. *Topics in Cognitive Science, 2*, 265–284.

Goodwin, C. (1994). "Professional Vision." American Anthropologist 96(3): 606–633.

Kellman, P. J., Massey, C. M & Son, J. (2010). Perceptual learning modules in mathematics: Enhancing students' pattern recognition, structure extraction, and fluency. *Topics in Cognitive Science, 2*, 285–305.

Kirshner, D. & Awtry, T. (2004). Visual salience of algebraic transformations. *Journal for Research in Mathematics Education, 35* (4), 224–257.

Koedinger, K., Booth, J. & Klahr, D. (2013). Instructional complexity and the science to constrain it. *Science, 342*, 935–937.

Landy, D., Allen, C. & Zednik, C. (2014). A perceptual account of symbolic reasoning. *Frontiers in Psychology, 5*, 275. doi: 10.3389/fpsyg.2014.00275

Landy, D. & Goldstone, R. L. (2007a). How abstract is symbolic thought? *Journal of Experimental Psychology: Learning, Memory & Cognition, 33*, 720–733.

Landy, D. & Goldstone, R. L. (2007b). Formal notations are diagrams: Evidence from a production task. *Memory & Cognition, 35*, 2033–2040.

Landy, D. H., Jones, M. N. & Goldstone, R. L. (2008). How the appearance of an operator affects its formal precedence. *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society,* (pp. 2109–2114). Washington, D.C.: Cognitive Science Society.

Landy, D. H. & Goldstone, R. L. (2010). Proximity and precedence in arithmetic. *Quarterly Journal of Experimental Psychology, 63*, 1953–1968.

Newell, A. & Simon, H.A. (1976). Computer science as empirical enquiry: Symbols and search. *Communications of the ACM 19*(3), 113–126.

Ottmar, E.R., Landy, D., Goldstone, R. & Weitnauer, E. (2015). Getting From Here to There!: Testing the Effectiveness of an Interactive Mathematics Intervention Embedding Perceptual Learning. *Proceedings of the 37th Annual Conference of the Cognitive Science Society*. Pasadena, California: Cognitive Science Society.

Ottmar, E.R., Landy, D., Weitnauer, E. & Goldstone, R. (2015). Graspable Mathematics: Using Perceptual Learning Technology to Discover Algebraic Notation. in Maria Meletiou-Mavrotheris, Katerina Mavrou, and Efi Paparistodemou, eds., *Integrating Touch-Enabled and Mobile Devices into Contemporary Mathematics Education.* Hershey, PA: IGI Global.

Schwartz, D. L & Goldstone, R. L. (2016). Learning as coordination: Cognitive psychology and education. In L. Corno & E. M. Anderman (Eds.) *Handbook of Educational Psychology, 3rd edition.* New York: Routledge (pp. 61–75).

Sleeman, D. (1984). An attempt to understand students' understanding of basic algebra, *Cognitive Science*, *6*, 387–412.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.K. (2013). *The Generalized Intelligent Framework for Tutoring (GIFT).* Technical report.

Quine, W. V. (1977). Natural kinds. In S. P. Schwartz, ed., *Naming, necessity, and natural kinds*. Ithaca, NY: Cornell University Press.

Ullman, S. (1984). Visual routines. *Cognition*, *18*, 97–159.

Vygotsky, L. S. (1962). *Thought and language*. Cambridge, MA: MIT Press.

CHAPTER 20 – Sketch Understanding for Education

Kenneth D. Forbus
Northwestern University

Introduction

Spatial learning is a fundamental problem for education. Spatial learning involves two things. The first is learning how to represent and reason about space itself. The second is how spatial representations and reasoning can be used in learning other domains. Space is particularly important for science, technology, engineering, and mathematics (STEM) domains, since many scientific and engineering fields have spatial components (e.g., mechanics, chemistry, geoscience, and so on). Modeling these domains, as well as many others, requires modeling their spatial aspects. This requires formal representations of shape and space, drawing on cognitive science research on human visual and spatial representations and reasoning. It also requires the ability for domain modelers, learners, and intelligent tutoring systems (ITSs) to communicate about spatial concepts. People use a variety of means to communicate spatial concepts with each other, such as gesture, physical models, and sketching. Sketching is a sweet spot for ITSs because it is a powerful tool for spatial learning and it is a closer match to the affordances of today's computer technologies used to field ITSs, particularly as digital pens become more common (e.g., Tablet PCs, iPad Pro).

Why is sketching a powerful tool for spatial learning? There are three reasons. First, sketching is a natural way that people communicate spatial ideas. Graphs, maps, configurations, and sequences of events are all naturally depicted via sketches. Sketching between people is interactive, involving both drawing and language. This enables feedback to be provided immediately, by contrast with learners grappling with a diagram that complements a text they are reading. Second, sketching is a powerful way for people to work through ideas on their own. Creating a sketch, like writing, forces one to be more specific about an idea, in order to decide what to include, how to depict things, and how they are related. Third and finally, sketches can be used to work through abstract ideas in addition to concretely spatial topics. Concept maps, Venn diagrams, schematics, and object class diagrams are all examples of visual languages aimed at helping people grapple with abstract concepts spatially. Thus incorporating sketching widely in ITSs could potentially have revolutionary impact.

Despite this potential, very few ITSs have used sketching to date. Recent advances in artificial intelligence (AI) can change this. This chapter discusses recent progress in sketch understanding and how it can be used in education. It outlines the idea of open-domain sketch understanding, illustrating how models of human-like visual and spatial representations can be used across a range of domains. It also discusses recognition-based systems, which provide complementary services for specific domains. Both approaches are illustrated via field-tested examples. The consequences of incorporating sketches for the design of ITSs are discussed, closing with recommendations and directions for future research.

Related Research

This section discusses the two main approaches to sketch understanding, the open-domain approach (Forbus, Usher, Lovett, Lockwood & Wetzel, 2011) and the recognition-based approach (Stahovich, Davis & Shrobe,1998; Valentine et al. 2012). Ultimately, these approaches are complementary, as discussed later.

Open-Domain Sketch Understanding

Human vision is very powerful, and relies on both vast processing capabilities and rich, hierarchical representations (Palmer, 1999). Cognitive science is still far from a complete model of human visual and spatial representations and reasoning. Working with sketches instead of vision radically simplifies the problem (e.g., edge-finding, stereo vision, and lightness computations become moot). However, many hard problems remain (e.g., segmentation). Even for the subset of visual problems relevant for sketching, none of them have been completely solved. However, there has been enough progress that workable solutions for many educational situations exist. By workable, I mean that what the software sees is close enough to what people see that a tutoring system's visual understanding can be used in generating helpful feedback. This is the idea of open-domain sketch understanding (i.e., constructing a general high-level visual system for digital ink that can be used across a wide range of domains and applications).

Open-domain sketch understanding systems have been used to explore battlespace reasoning (Forbus, Usher & Chapman, 2003; Rasch, Kott & Forbus, 2002), teaching AI systems via multimodal instruction (Chang & Forbus, 2015; Hinrichs & Forbus, 2013; Lockwood & Forbus, 2009;) and solving physics problems expressed via diagrams (Chang, Wetzel & Forbus,2014). I focus on CogSketch (Forbus et al., 2011) here because it has seen the most use in education research and because it is being made freely available for research and applications.

Like other open-domain sketch understanding systems, CogSketch relies on a combination of quantitative and qualitative representations and processing. Its input consists of digital ink, which is encoded as polylines, each of which is an ordered list of points. Each polyline includes, in addition to the x,y coordinates and timestamps for each of its points, other visual properties of the ink (e.g., stroke thickness, color, and whether it is solid or dashed). Ink is initially grouped into glyphs (i.e., visual entities considered as a unit) as part of the drawing process. The CogSketch interface enables a user to say when they are finished with a glyph, or just draw and do segmentation later, depending on individual preference. The interface also provides a means for users to label what they intend their glyph to mean, using concepts drawn from CogSketch's knowledge base[1]. This explicit labeling is very important, since it ensures that the learner's ideas are accurately conveyed to the system even when they are incorrect. For example, an AI system that understands the layers of the Earth might conjecture that the student's innermost circle represents the Earth's core, whereas in the student's mind, they intend it to be Earth's mantle. CogSketch takes the difference between visual and spatial to lie in the interpretation of the sketch, with visual concerning properties computed with respect to the coordinate system of the sketch, and spatial concerning properties computed with respect to the world depicted. This part of a user's intent is also communicated explicitly. That is, users can specify the genre of a sketch (i.e., map, physical, abstract) and its pose (i.e., viewed from the top or side).

Qualitative representations are particularly important for spatial learning because they provide a bridge between perception and cognition. That is, qualitative representations carve up space and shapes into discrete symbolic elements, typically grounded in metric representations, which can then be reasoned about. CogSketch computes qualitative spatial relationships both between glyphs and within glyphs. Between glyphs, it computes qualitative topological relationships (e.g., disconnected, touching, inside) based on Region Connection Calculus (Cohn, 1996), positional relationships (e.g., above, leftOf), coarse shape properties (e.g., major axis, roundness), and relative sizes with respect to the frame (e.g., large, small, medium). It can also group sets of glyphs with similar properties, using gestalt principles. Within glyphs, the representations it can construct include edge-level representations, where ink is split into junctions at

[1] CogSketch uses contents from the OpenCyc knowledge base, which includes over 59,000 concepts and over 16,000 relations, all connected via 1.3 million facts.

visually salient points, and edge cycle representations, where connected sets of edges that decompose a glyph into 2D regions are constructed. These representations can be used to construct hypotheses about surfaces in three dimensions (Lovett, Dehghani & Forbus., 2008; Lovett & Forbus, 2013). The ability to shift among multiple levels can be important for tutoring. For instance, in providing feedback about engineering drawing exercises, CogSketch starts with glyph level descriptions, and based on how they match, it drills down as needed to finer-grained edge-level representations to provide more specific feedback.

All of the relationships computed by CogSketch are represented explicitly, as predicate calculus statements. This provides two benefits. First, it enables a tight coupling between visual and conceptual information. For instance, when a coaching system is providing feedback about possible motions of a mechanical system, it is capable of analyzing ink representing the two objects to introducing a new edge to explicitly represent their surface contact, and it can provide reasons for its conclusions, to enable students to understand its reasoning (Figure 1). Second, it enables the use of analogical matching to compare and contrast sketches. CogSketch uses SME, the Structure-Mapping Engine (Falkenhainer, Forbus & Gentner,1989; Forbus, Ferguson, Lovett & Gentner, in press), a computational model of Gentner's (1983) structure-mapping theory of analogy and similarity. There is psychological evidence that human visual similarity can be captured via the laws of structure-mapping (Sagi, Gentner & Lovett, 2012). Moreover, CogSketch's visual representations have been used to model multiple visual problem-solving tasks, including geometric analogies (Lovett, Tomai, Forbus & Usher, 2009), cross-cultural data on a visual oddity task (Lovett & Forbus, 2011), and Ravens' Progressive Matrices (Lovett, Forbus & Usher, 2010). These tasks provide evidence that CogSketch's visual processing and analogical matching processes are useful for comparing and contrasting sketches in human-like ways[2].

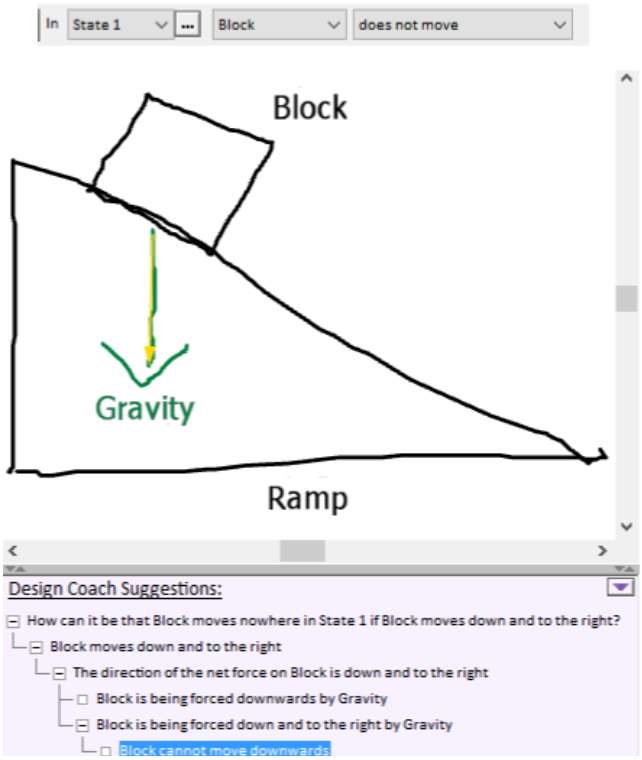


Figure 1. An example of feedback provided by the CogSketch Design Coach.

[2] For example, on the Ravens' test, a widely used test of human intelligence, the CogSketch model's performance places it in the 75th percentile, which is better than most adult Americans (Lovett et al., 2010).

CogSketch provides a rich visual language for communicating via sketching. So far I have focused on glyphs that represent specific shapes or objects, also known as entity glyphs. CogSketch supports two other kinds of glyphs as well. Relation glyphs describe binary relationships between other glyphs. Relation glyphs are drawn as arrows and their conceptual interpretation is provided by the user selecting among a set of available relationships (In the knowledge base CogSketch uses, there are just over 11,000 such relationships, but for any particular exercise, only a small subset of relevant relationships are exposed, to help learners stay focused). CogSketch guesses what other glyphs the relationship applies to, based on proximity of other glyphs to the ends of the arrow that the user draws. If its guess about the intended meaning is incorrect, users can change this via a simple drag and drop interface. Annotation glyphs provide links to specific kinds of conceptual information. Examples include numerical values with units (e.g., flow rates in a sketch of the Carbon Cycle are to be specified in petatons), directions of applied force or motion, and points of mechanical constraint. The type of annotation glyph being drawn is selected from a menu, and there are one or more attachment points that can be dragged to indicate what the annotation applies to. These representations are compositional when appropriate (e.g., relation glyphs can be used to link one relationship to another) to express causality or implication. This vocabulary is sufficient to express concept maps, as well as physical situations and many kinds of diagrams.

Many domains involve the depiction of multiple states of affairs, either showing how a situation evolves over time or alternatives to be contrasted. To support this, CogSketch sketches can consist of multiple subsketches. Each subsketch has its own local coordinate system and interpretation, although persistent objects can be constructed by copy/pasting glyphs from one subsketch to another. To get an overview of the sketch, the metalayer treats every subsketch as a higher-order glyph. Like other types of glyphs, these glyphs can be related via arrows and annotated. Thus the subsketches can be used to form comic graphs, a generalization of comic strips because they allow branches and joins, unlike the traditional linear structure of comic strips. Because this visual representation enables multiple states to be displayed at once, it can facilitate comparison by reducing working memory load. This is an important contrast to animation, which although popular for dynamic displays, can lead to issues of students not noticing important information and not being able to compare subsequent states due to memory load. Psychological evidence (Tversky, Morrison & Betrancourt, 2002) suggests that sequences of pictures (i.e., a sequence of qualitative states) are often better than animation. It is certainly easier for a student to produce such a sequence of drawings and be given feedback on them, than to force students to provide all of the additional information needed to make an animation.

Examples of Open-Domain Sketch Understanding for Education

CogSketch was designed as a platform for creating new kinds of sketch-based educational software. Two types of such software have been developed to date: Sketch Worksheets and Design Coach. Each is summarized.

Sketch Worksheets (Yin, Forbus, Usher, Sageman & Jee, 2010) helps students learn about situations and systems whose characterization depends to some degree on spatial layouts. Figure 2 shows a simple example of a greenhouse effect worksheet. Sketch Worksheets can be authored by domain experts, using a special interface in CogSketch. That is, authors construct a solution sketch as a subsketch within a sketch representing the worksheet. They choose what concepts, relationships, and annotations are relevant to the worksheet from the underlying knowledge base, or (with more effort) they can extend the knowledge base with their own contributions. CogSketch automatically elaborates what they draw via visual processing. The author marks what feedback should be given if particular relationships (visual or conceptual) are not satisfied. For example in Figure 2, the student has forgotten to add a radiation flow from the atmosphere back to the planet. The worksheet author provided advice, in natural language, to be presented if this occurred, so the lack of that flow causes the tutor to present this advice. This information can also be organized into rubrics (e.g., the Radiation Arrows entry on the feedback pane), and a numerical score can be

assigned to each such difference. When a student tackles a Sketch Worksheet, they are drawing on a subsketch that is distinct from the author's solution, but their ink and conceptual labels are processed in the same way. Feedback is provided on demand, based on differences between the solution sketch and a student's sketch, as computed by a subject-matter expert (SME). To avoid overloading students, feedback is filtered based on rubrics (i.e., more basic issues are presented before more subtle ones). Moreover, the integrated visual and conceptual analogy is used to highlight the aspects of the sketch involved in each piece of feedback, providing context for students to understand how the feedback applies to their sketch.

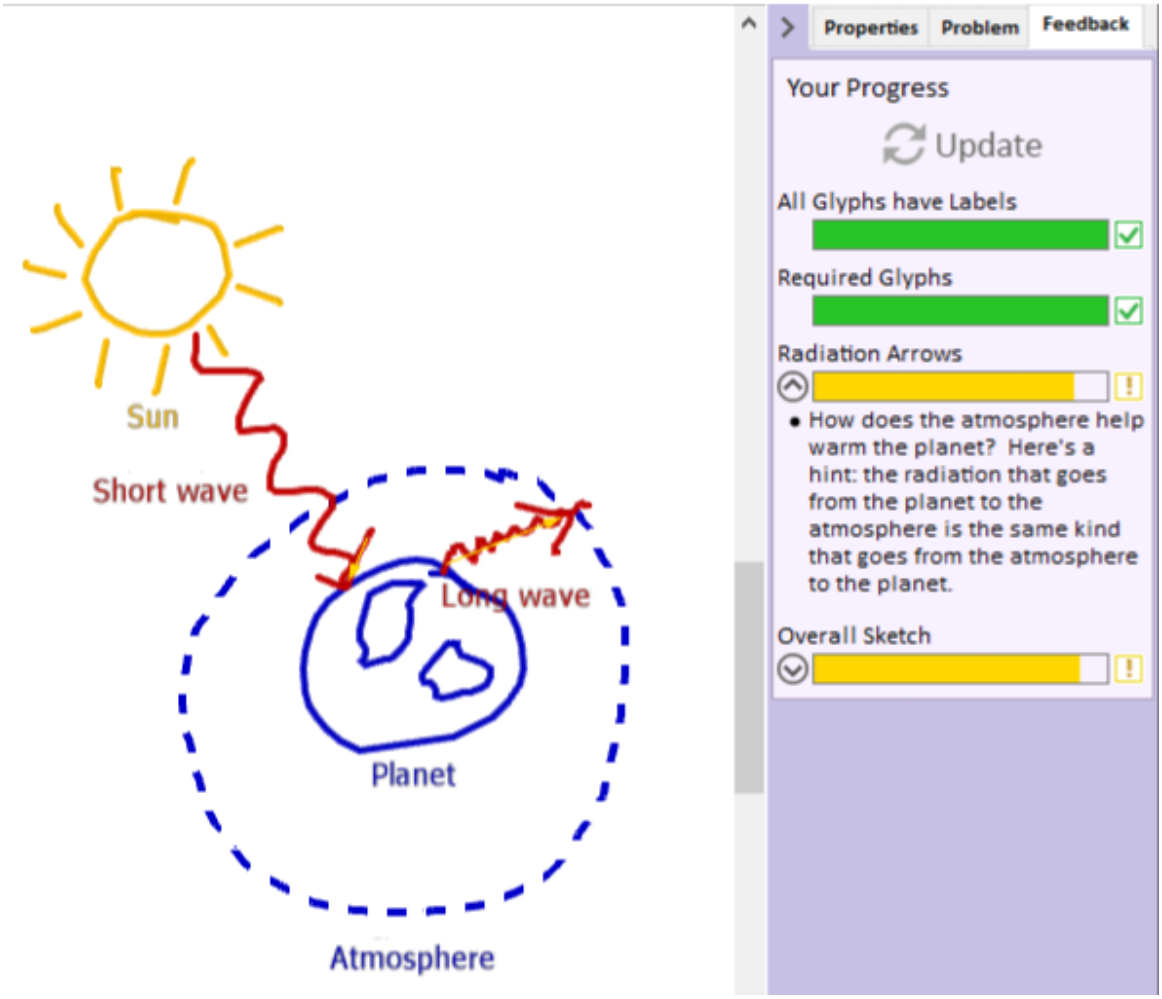


Figure 2. Feedback from Sketch Worksheets, generated via analogy with an instructor's sketch.

Classroom experimentation motivated several different extensions to this model. First, misconception sketches can be provided when there are known problems that students have on an assignment. When a misconception sketch matches the student's sketch, the advice it provides is given instead of advice based on differences with a solution sketch. Second, subsketches can have a background image against which sketching is done. This is important for many types of problems, such as marking up photographs and interpreting graphs. Since CogSketch does not process bitmaps, instructors use entity glyphs to mark up photographs in their solution(s). Such glyphs can be used as quantitative ink constraints, so that a student's glyph must be a quantitatively accurate tracing, in location, orientation, and scale, to the instructor's glyph, up to some tolerance. Finally, quantitative matching of numerical value annotations is carried out, again with appropriate tolerances and unit conversions.

There are additional analytic tools to help instructors and researchers mine sketches for information about student performance. CogSketch automatically keeps a complete temporal record of every user interface (UI) operation, and can reconstruct exactly what a student's sketch looked like each time they requested feedback and what feedback they were given. A gradebook provides a simple means for instructors to grade worksheets for classes, including an opt-in scheme that transmits their student data, anonymized onsite for privacy, to Northwestern University as a contribution to an ongoing corpus of sketches to facilitate future research.

Sketch worksheets have been used with a variety of student ages and domains. For example, fifth grade students who were learning about the human circulatory system that used three Sketch Worksheets showed significant gains on two out of three pre/post-tests (Miller, Cromley & Newcombe, 2013). Most classroom experiments have concerned college geoscience, with pilot experiments conducted at Northwestern and Carleton. The most extensive experiment was using sketch worksheets across an entire semester of an introductory geoscience class at University of Wisconsin-Madison. Importantly, the worksheets used in this experiment were developed by a geoscience graduate student and faculty member, not by AI or computer science researchers. Although most geoscience instructors think that sketching is important, few assign sketches as often as they think that they should, due to the onerous burden of grading them. In the Madison experiment, parallel pencil and paper worksheets were also developed, and additional investment was made to grade those as well, providing a comparison of Sketch Worksheets against an idealized situation. Student pre/post-test scores for Sketch Worksheets were just as good as for the pencil and paper worksheets, demonstrating that this technology can lead to the same outcomes as good human tutoring, although far more efficiently (Garnier et al., 2014)

Sketch Worksheets were designed to be domain-general and not involve any reasoning beyond CogSketch's default capabilities. The Design Coach illustrates a very different type of system, aimed at a specific domain and involving domain-specific reasoning and feedback. The goal of the Design Coach was to help introductory engineering design students become more confident in explaining themselves via sketching. Instructors at Northwestern and elsewhere view this as a serious issue, what can be thought of as sketch anxiety, analogous to math anxiety, which is a form of stereotype threat that can significantly impact women and underrepresented minority students (Beilock, Gunderson, Ramirez & Levine, 2010). The idea was to enable students to practice using sketching to explain designs in a socially safe setting (i.e., CogSketch). The Design Coach provides feedback when the student's explanation is inconsistent or incomplete. As a simplistic example, suppose a student drew ramp and block in Figure 1 and said that it was not going to move (top of Figure 1). The bottom of Figure 1 shows the Design Coach's feedback, based on qualitative mechanics reasoning, involving visually identifying a region of surface contact between the block and the ramp, reasoning about the net forces on the block (the student did not specify friction), and the mechanical constraint imposed by the orientation of the ramp.

This is, of course, a very simple example: Most designs include multiple states, either representing different modes or an unfolding behavior, so students draw comic graphs to describe multiple states, using arrows on the metalayer to describe how they relate to each other as part of their explanation. This provides new opportunities for feedback (e.g., when the sequence of behaviors that a student draws is inconsistent with what they say about the design). Non-sketch input is provided via a simple menu system that lets students construct what looks to them like sentences, but lead to predicate calculus representations internally that CogSketch can reason about, as in the top of Figure 1. In a pilot experiment, just one Design Coach exercise during a quarter was sufficient to significantly reduce sketch anxiety, compared to a control group who did not use it (Wetzel & Forbus, 2015).

Recognition-Based Sketch Understanding

Some domains involve using vocabularies of visual symbols to express situations, plans, and designs. Examples include schematics in digital and analog electronics, unified modeling language (UML) diagrams, and military battle plans. While many equate sketch understanding with sketch recognition, that is a fundamental mistake, for two reasons. First, in human to human sketching, people talk while sketching, indicating what their glyphs represent and what relationships between them are important versus unimportant. In other words, recognition is a catalyst, not a requirement, for sketching. Second, recognition works well when there are visual symbols, whose meaning is fundamentally distinct from their spatial properties. Much of what people draw when sketching in STEM domains are not visual symbols. The outline of a geological fault or marker bed, for example, is a function of the processes that gave rise to the formation of those geologic structures, not a human convention for depicting a component in an artifact. The mapping between conceptual entities and visual properties in general is many to many: A circle could represent a planet, an orbit, a cross section of a vein or pipe, or a bubble, for example. Nevertheless, for domains where visual symbols can be used, or defined, recognition can improve the fluency of sketching.

It is important to realize that sketch recognition is an extremely hard problem. To understand why sketch recognition is difficult, it is useful to consider an edge case: handwriting recognition. Today's handwriting recognizers are amazingly good, especially if they are provided with stroke timing information in addition to the visual depiction of the digital ink. The symbol vocabulary is fairly large, in English, 52 characters (upper and lower case) plus punctuation. However, the relationships between characters are strictly linear, rather than being arrayed in multiple configurations (ignoring crossword puzzles, other word games, and artistic layouts). The sketch recognition problem is harder because in general visual symbols can have multi-dimensional relationships. Moreover, in handwriting recognition there are strong constraints due to language, including between character relationships and higher-order relationships involving common sequences of words. Such relationships do exist in other domains where sketch recognition is useful. For example, in an analog circuit diagram it is rare to see a transistor without at least one resistor connected to it, and there are visual conventions on geometry (e.g., electrical grounds tend to be drawn on the bottom of a circuit) that can facilitate recognition. However, the amount of training data available for most domains is much smaller than today's machine learning techniques tend to require.

Despite these difficulties, impressive progress has been made on using sketch recognition systems in education. This progress has been based on combining existing technologies and extending them as needed to support particular uses. For example, one problem with any recognition technology is that, today, the user is being trained as much as the system. This drawback can be turned into an advantage, when the topic of instruction is how to draw visual symbols, as in teaching Kanji (Taele & Hammond, 2009). The strokes in Kanji are drawn in a specific order so that, when someone is writing quickly, the distortions introduced will be recognizable across different people. Similarly, when teaching music annotation (Taele, Barreto & Hammond, 2015), the point is to learn how to use visual symbols to express meaning in another domain.

The MECHANIX system (Valentine et al., 2012) helps students with statics problems. Here recognition is divided into two distinct problems, recognizing the lines that constitute the struts in a mechanical structure, versus writing the equations that describe the structure. MECHANIX has now been successfully fielded in classes at Texas A&M. Another system for statics problems, Newton's Pen 2 (Lee, Jordan, Stahovich & Herold, 2012), was developed and successfully deployed for courses at University of California Riverside.

An interesting case where sketch recognition has been used is in pentop computing systems, where a pen is used with special paper to produce diagrams in ink that are human-readable, but also lead to a digital trace that can be used for assessment and feedback. Kirchoff's Pen (de Silva et al., 2007), for example, provided feedback to help students learn aspects of electrical circuit analysis. Given the limited amount of

processing on such pens, these systems required that students learn a particular order in which strokes need to be made to enable recognition, and thus have fallen out of favor compared to using tablet PCs.

An important problem in recognition-based systems is automatic segmentation. That is, what pieces of ink should be considered together? This is typically solved by using spatial and temporal proximity as a heuristic for grouping (Cohen et al., 2015). Recognition and segmentation can interact (e.g., recognizing a set of strokes as characters denoting a resistor's value versus strokes describing the resistor itself). Information from multiple modalities can help with both segmentation and recognition. For example, in Sketch-Thru-Plan (Cohen et al., 2015), a multimodal sketch recognition system for enabling commanders to express military battle plans, accuracy went from 73% to 96% when both speech and ink recognition were used for mutual disambiguation, versus ink recognition alone.

These examples illustrate that sketch recognition, when used carefully, can provide value for educational software systems. However, the drawback is that recognition errors can distract students, focusing them more on the interface than on the topic they are trying to learn. An interesting comparison can be made with speech recognition. Speech recognition has seen a considerable amount of attention by the research community over the past 40 years, and with that amount of time and a considerable investment of resources, only now is starting to become really useful on an everyday basis. Some, but not all, of the bottlenecks have been due to lack of raw computational power, to be sure. But sketch recognition has had far less attention and investment than speech recognition, while being an arguably more difficult problem, so it should not be surprising that it has farther to go before it will be similarly useful.

Since the focus of this book is on domain modeling, and the role of sketch recognition is improving interface fluency rather than adding new domain modeling capabilities, it is not considered further here. However, as recognition technologies improve, they could be integrated into open-domain sketching systems to suggest conceptual labels for student ink. Such suggestions could make interaction more fluent when recognition is successful, but if it fails, students could easily correct it via the usual conceptual labeling interface. It may also be the case that the richer representations provided by CogSketch's visual processing could substantially enhance recognition, but this is an empirical question.

Discussion

Progress in sketch understanding offers the opportunity to create domain models that include visual and spatial information. The use of qualitative spatial representations, automatically computed from digital ink, provides a means for people and software to communicate concerning spatial topics that has not been available before. This new capability provides three advantages for domain modeling:

(1) Domain experts can express their visual and spatial knowledge via sketching. As research with Sketch Worksheets shows, this can lead to materials authored entirely by domain experts, without AI practitioners in the content generation loop. This is crucial for scalability.

(2) Students can express their understanding of spatial situations via sketching. Moreover, there is evidence that the order in which students draw elements of diagrams, and what they do and do not include, can provide valuable information for assessment, even during simple copying tasks (Jee et al., 2014).

(3) Visual information can be combined with conceptual representations to support domain-specific reasoning, as illustrated the Design Coach research. Research to date suggests that the same set of visual relationships should suffice for a wide range of STEM domains. Thus having a broad range

of visual and spatial representations already developed and reusable should drastically cut the cost of developing new domain theories that involve visual processing and spatial knowledge.

This suggests that sketches should become a common content medium for ITS. Analogical processing over formally represented sketches provides a form of case-based reasoning, which can be used to rapidly author new exercises in domains and provide examples for other kinds of case-based tutors. It also suggests that, when spatial content is involved in a domain, it is worth incorporating sketching interfaces in ITS. With the rise of pen computing, as illustrated by Microsoft's Surface computers and the iPad Pro, the hardware to support sketching well is rapidly becoming commonplace. However, even when using a mouse and doing explicit segmentation and conceptual labeling, sketching interfaces are easy for students as young as fifth grade to use.

Recommendations and Future Research

Here are two recommendations for the Generalized Intelligent Framework for Tutoring (GIFT) specifically, and two recommendations for further research. Regarding GIFT, I suggest the following:

(1) Integrate a sketching interface and sketch understanding into GIFT. Since CogSketch has already been developed as part of a decade-long National Science Foundation (NSF)-funded effort and is publically available, it is a good place to start. This will require mostly engineering effort. A distributed version is already under development, with lightweight HTML5 interfaces for ink capture that could be integrated within the GIFT interface framework, with the heavier visual, conceptual, and analogical processing happening via a CogSketch instance running on a server or cloud service. The simplest version of this is incorporating the current Sketch Worksheet model as lessons within other ITSs and curricula, providing interconnections between the internal representations of rubrics with representations used by other tutoring systems. For example, CogSketch tracks timing information at the level of individual ink strokes and glyphs. It can provide information about when the student asked for feedback, what feedback they were given, and what the students were seeing at the time. This information can be analyzed in many ways: Spatial Intelligence and Learning Center (SILC) psychologists and learning scientists have used it to analyze student learning (e.g., Jee, Gentner, Uttal, D, Sageman,, Forbus,, Manduca, C., Ormand, C., Shipley, T. & Tikoff, 2014), and there is potential for finding patterns of misconceptions by using analogical generalization over a corpus of sketches (Chang & Forbus, 2014).

(2) Develop a Sketching Academy, by analogy with Khan Academy, where learners can go to learn spatially heavy domains, such as physics, mechanics, geoscience, and biology. GIFT would provide the overall interface and recommendation services (e.g., based on how a student has done so far, provide suggestions as to what problem to try next).

Here are two recommendations regarding future research:

(1) Explore the use of a combination of image processing and sketch understanding to automatically (or semi-automatically) encode diagrams and other image-based training materials for domain model construction. Unlike many of today's computer vision techniques, which rely on pixel-based processing, it has long been known that mammalian vision includes edge-finding as one of its basic operations. An intriguing possibility is to use off-the-shelf edge-finders to produce digital ink, which might then be processed via CogSketch into material that could be integrated into a domain theory. This involves tackling several difficult research questions, such as automatic segmentation of what can be very rich visual stimuli and doing context-driven recognition (e.g., diagrams and text refer to each other).

(2) Explore multimodal science learners both as an aid to domain modelers and as a step toward true Socratic tutors. In Socratic dialogues, either party can bring up new examples to illustrate a concept or sharpen a distinction. This requires combining both sketch understanding and natural language understanding for robust multimodal interactions. It also requires extremely broad and deep background knowledge, so that the software Socratic tutor can make sense of student-provided examples. Therefore, as a way to bootstrap such systems, I suggest first investigating how to build multimodal science learners, that is, systems which can be taught science topics via multimodal interactions and via learning by reading (including diagrams). This intermediate kind of system provides a setting for exploring better techniques for multimodal interaction and accumulating domain knowledge at scale. The accumulated knowledge and improved communication abilities will provide the basis for the next step, of incorporating additional pedagogical models (built perhaps in part by the system reflecting on its own experiences as a student).

To summarize, sketch understanding, especially open-domain sketch understanding, shows promise as a new medium for domain modeling and interaction in ITS. The naturalness of sketching, both for domain modelers creating spatially laced content (as, for example, much of STEM is) and for students, suggests that sketching could potentially have a revolutionary impact. The time is right for larger-scale experiments, to understand both how to scale up sketch understanding for domain modeling and interacting with students, and how to best integrate it with other forms of educational software.

Acknowledgements

Maria Chang, Madeline Usher, Matt McLure, and Tom Hinrichs provided valuable feedback. CogSketch research is supported by the SILC, an NSF-funded Science of Learning Center (SBE-1041707).

References

Beilock, S. L., Gunderson, E. A., Ramirez, G. & Levine, S. C. (2010) Female teachers' math anxiety affects girls' math achievement. *Proceedings of the National Academy of Sciences*, 107, 1860–1863

Chang, M.D. & Forbus, K.D. (2014) Using analogy to cluster hand-drawn sketches for sketch-based educational software. *AI Magazine*, 35(1), 76–84.

Chang, M.D. & Forbus, K.D. (2015). Toward Interpretation Strategies for Multimodal Instructional Analogies. *Proceedings of the 28th International Workshop on Qualitative Reasoning (QR2015).* Minneapolis, MN.

Chang, M. D., Wetzel, J. W. & Forbus, K.D. (2014). Spatial Reasoning in Comparative Analyses of Physics Diagrams. C. Freksa et al. (Eds.), *Spatial Cognition 2014*, LNAI 8684, pp. 268–282.

Cohen, P., Kaiser, E., Buchanan, M., Lind, S., Corrigan, M. & Wesson, R. (2015) Sketch-Thru-Plan: A Multimodal Interface for Command and Control. *Communications of the ACM*, 54(4), pp. 56–65.

Cohn, A. (1996) Calculi for qualitative spatial reasoning. In J. Calmet, J. A. Campbell & J. Pfalzgraph (Eds.), *Artificial Intelligence and Symbolic Mathematical Computation*, LNCS 1138 (pp. 124–143). New York: Springer Verlag.

de Silva, R., Bischel, D., Lee, W., Peterson, E., Calfee, R. & Stahovich, T. (2007). Kirchhoff's Pen: a pen-based circuit analysis tutor. *Proceedings of the 4th Eurographics Workshop on Sketch-Based interfaces and Modeling (SBIM 2007)*, pp. 75–82

Falkenhainer, B., Forbus, K. D. & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1–63.

Forbus, K., Ferguson, R., Lovett, A. & Gentner, D. (in press). Extending SME to Handle Large-Scale Cognitive Modeling. *Cognitive Science*.

Forbus, K., Usher, J., Lovett, A., Lockwood, K. & Wetzel, J. (2011). CogSketch: Sketch understanding for Cognitive Science Research and for Education. *Topics in Cognitive Science*. 3(4), pp. 648–666.

Forbus, K., Usher, J. & Chapman, V. (2003). Sketching for Military Courses of Action Diagrams. *Proceedings of IUI'03*, January, 2003. Miami, FL.

Garnier, B., Ormand, C., Chang, M., Matlen, B., Tikoff, B., Shipley, T. & Forbus, K. (2014). Testing CogSketch geoscience worksheets as an effective spatial learning tool in introductory geoscience courses: *Geological Society of America annual meeting* (Vancouver, BC).

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155–170.

Hinrichs, T. & Forbus, K. (2014). X Goes First: Teaching a Simple Game through Multimodal Interaction. *Advances in Cognitive Systems*, 3:31–46.

Jee, B. D., Gentner, D., Uttal, D. H., Sageman, B., Forbus, K., Manduca, C. A., Ormand, C. J., Shipley, T. F. & Tikoff, B. (2014). Drawing on Experience: How Domain Knowledge Is Reflected in Sketches of Scientific Structures and Processes. *Research in Science Education*, 44(6), 859–883.

Lee, C., Jordan, J., Stahovich, T. & Herold, J. (2012). Newton's Pen II: An Intelligent Sketch-based Tutoring System and its Sketch Processing Techniques. *Proceedings of the Eurographics Sketch-Based Interfaces and Modeling Workshop*.

Lockwood, K. & Forbus, K. (2009). Multimodal knowledge capture from text and diagrams. *Proceedings of KCAP-2009.*

Lovett, A. & Forbus, K. (2011). Cultural commonalities and differences in spatial problem solving: A computational analysis. *Cognition, 121*, 281–287.

Lovett, A. & Forbus, K. (2013). Modeling spatial ability in mental rotation and paper-folding. *Proceedings of the 35th Annual Conference of the Cognitive Science Society*. Berlin, Germany, 930–935.

Lovett, A., Dehghani, M. & Forbus, K. (2008). Building and comparing qualitative descriptions of three-dimensional design sketches. *Proceedings of the 22nd International Qualitative Reasoning Workshop.* Boulder, CO.

Lovett, A., Tomai, E., Forbus, K. & Usher, J. (2009). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science. 33*(7), 1192–1231.

Lovett, A., Forbus, K. & Usher, J. (2010). A structure-mapping model of Raven's Progressive Matrices. *Proceedings of CogSci-10*.

Miller, B., Cromley, J. & Newcombe, N. (2013). Exploration of CogSketch as an Instructional Tool in Middle School Science. http://bit.ly/XmgZzE.

Palmer, S. (1999). *Vision Science: Photons to Phenomenology*. MIT Press.

Rasch, R., Kott, Al & Forbus, K. (2002). AI on the Battlefield: An experimental exploration. *Proceedings of the 14th Innovative Applications of Artificial Intelligence Conference*, July, Edmonton, Canada.

Sagi, E., Gentner, D. & Lovett, A. (2012). What difference reveals about similarity. *Cognitive Science*, *36*(6), 1019–1050.

Stahovich, T., Davis, R. & Shrobe, H. (1998). Generating multiple new designs from a sketch. *Artificial Intelligence*, *104*, pp. 211–264.

Taele, P. & Hammond, T. (2009). Hashigo: A Next-Generation Sketch Interactive System for Japanese Kanji. *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference (IAAI).* 153–158.

Taele, P., Barreto, L. & Hammond, T. (2015). Maestoso: An Intelligent Educational Sketching Tool for Learning Music Theory. *Proceedings of the Twenty-Seventh Innovative Applications of Artificial Intelligence Conference (IAAI).*

Tversky, B., Morrison, J. & Betrancourt, M. (2002). Animation: can it facilitate? *Int. J. Human-Computer Studies*, *57*, 247–262.

Valentine, S., Vides, F., Lucchese, G., Turner, D., Kim, H., Li, W., Linsey, J. & Hammond, T. (2012). Mechanix: A Sketch-Based Tutoring System for Statics Courses. *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference (IAAI)*. pp. 2253–2260.

Wetzel, J. & Forbus, K. (2015). Increasing Student Confidence in Engineering Sketching via a Software Coach. In Hammond, T., Valentine, S., Adler, A. & Payton, M. (Eds.) *The Impact of Pen and Touch Technology on Education*. pp. 107–118, Springer.

Yin, P., Forbus, K., Usher, J., Sageman, B. & Jee, B. (2010). Sketch Worksheets: A Sketch-based Educational Software System. *Proceedings of the 22nd Annual conference on Innovative Applications of Artificial Intelligence*.

BIOGRAPHIES

Editors

Dr. Arthur Graesser is a professor in the Department of Psychology and the Institute of Intelligent Systems (IIS) at the University of Memphis (UofM), as well as an Honorary Research Fellow at University of Oxford. He received his PhD in psychology from the University of California at San Diego. His primary research interests are in cognitive science, discourse processing, and the learning sciences. More specific interests include knowledge representation, question asking and answering, tutoring, text comprehension, inference generation, conversation, reading, education, memory, emotions, artificial intelligence (AI), computational linguistics, and human-computer interaction (HCI). He served as editor of *Discourse Processes* (1996–2005) and is the current editor of the *Journal of Educational Psychology* (2009-2014). His service in professional societies includes president of the Empirical Studies of Literature, Art, and Media (1989–1992), the Society for Text and Discourse (2007–2010), the International Society for Artificial Intelligence in Education (2007–2009), and the Federation of Associations in the Behavioral and Brain Sciences Foundation (2012–2013). In addition to publishing over 600 articles in journals, books, and conference proceedings, he has written 3 books and co-edited 16 books. He and his colleagues have designed, developed, and tested software in learning, language, and discourse technologies, including AutoTutor, AutoTutor-Lite, AutoMentor, ElectronixTutor, MetaTutor, GuruTutor, DeepTutor, HURA Advisor, SEEK Web Tutor, Personal Assistant for Lifelong Learning (PAL3), Operation ARIES!, iSTART, Writing-Pal, Point & Query, Question Understanding Aid (QUAID), QUEST & Coh-Metrix.

Dr. Xiangen Hu is a professor in the Department of Psychology and Department of Electrical and Computer Engineering at UofM and senior researcher at IIS, and a visiting professor at Central China Normal University (CCNU). He received his MS in applied mathematics from Huazhong University of Science and Technology, MA in social sciences, and PhD in cognitive sciences from the University of California, Irvine. He is the Director of Advanced Distributed Learning (ADL) Center for Intelligent Tutoring Systems (ITSs) Research & Development and a senior researcher in the Chinese Ministry of Education's Key Laboratory of Adolescent Cyberpsychology and Behavior. His primary research areas include mathematical psychology, research design and statistics, and cognitive psychology. More specific research interests include general processing tree (GPT) models, categorical data analysis, knowledge representation, computerized tutoring, and advanced distributed learning. He receives funding for the above research from the US National Science Foundation (NSF), US Institute for Education Sciences (IES), ADL of the US Department of Defense (DOD), US Army Medical Research Acquisition Activity, US Army Research Laboratories (ARL), US Office of Naval Research (ONR), UofM, and CCNU.

Dr. Benjamin D. Nye is the Director of Learning Sciences at University of Southern California Institute for Creative Technologies (USC-ICT). His major research interest is to identify best practices in advanced learning technology, particularly for frontiers such as distributed learning technologies (e.g., cloud-based, device-agnostic) and socially situated learning (e.g., face-to-face mobile use). Research interests include modular ITS designs, modeling social learning and memes, cognitive agents, and educational tools for the developing world and low-resource/low-income contexts. He received his PhD in systems engineering from the University of Pennsylvania in 2011. In his recent work as a research professor at UofM, he led work on the shareable knowledge objects (SKO) framework integrating ITS services such as AutoTutor for the ONR ITS Grand Challenge, helped data mine effort a corpus of 250k human-to-human online tutoring dialogs (part of the ADL PAL initiative), collaborated on ONR's PAL3 tutoring architecture for supporting life-long learning, and is an advisor and book editor for the ARL Generalized Intelligent Framework for Tutoring (GIFT) expert workshop panel. His research tries to remove barriers development and adoption of ITSs, so that they can reach larger numbers of learners, which has traditionally been a major roadblock for these highly effective interventions. He also believes that the future of learning science depends on large, sustainable platforms with many users, where efficient sampling techniques can be used to drive new designs for experiments. Finally, he is interested in making the process of science more efficient, such as by advanced metadata and analysis for scholarly publications.

Dr. Andrew Olney presently serves as associate professor in both the IIS and Department of Psychology and as director of the IIS at UofM. He received a BA in linguistics with cognitive science from University College London in 1998, an MS in evolutionary and adaptive systems from the University of Sussex in 2001, and a PhD in computer science from the UofM in 2006. His primary research interests are in natural language interfaces. Specific interests include vector space models, dialogue systems, unsupervised grammar induction, robotics, and ITSs.

Dr. Anne M. Sinatra is an adaptive training scientist in ARL's Human Research and Engineering Directorate, Advanced Training and Simulation Division (ATSD). She works on the GIFT project and is the lead for the Team Modeling for Adaptive Training and Education research vector. Her research interests are focused on cognitive and human factors psychology. She has specific interest in how information relating to the self and about those that one is familiar with can aid in memory, recall, and tutoring. Her dissertation research evaluated the impact of using degraded speech and a familiar story on attention/recall in a dichotic listening task. Her work has been published in the *Journal of Interaction Studies* and in proceedings including the Human Computer Interaction International (HCII) Conference and Human Factors and Ergonomics Society (HFES) Conference. She has a combination of over 30 publications and conference papers. Prior to becoming an ARL scientist, she was an ARL post-doctoral fellow and graduate research associate with University of Central Florida's (UCF) Applied Cognition and Technology (ACAT) Lab, and taught a variety of undergraduate psychology courses. She received her PhD and MA in applied experimental and human factors psychology, as well as her BS in psychology from UCF.

Dr. Robert A. Sottilare leads adaptive training research within ARL where the focus of his research is automated authoring, automated instructional management, the extension of tutors to psychomotor task domains, and evaluation tools and methods for ITSs. His work is widely published and includes articles in the *Cognitive Technology Journal*, the *Educational Technology Journal*, and the *Journal for Defense Modeling & Simulation*. He is a co-creator of GIFT, an open-source tutoring architecture, and is the chief editor for the *Design Recommendations for Intelligent Tutoring Systems* book series. He is a visiting scientist and lecturer at the US Military Academy and a graduate faculty scholar at UCF. He received his doctorate in modeling and simulation (M&S) from UCF with a focus in intelligent systems. In 2012, he was honored as the inaugural recipient of US Army Research, Development and Engineering Command's (RDECOM) Modeling & Simulation Lifetime Achievement Award, and in 2015, he was honored with the National Training and Simulation Association (NTSA) Governor's Award for Modeling & Simulation Lifetime Achievement.

Authors

Mr. Charles R. Amburn is the senior instructional systems specialist for the Advanced Modeling & Simulation Branch (AMSB) of ARL-HRED in Orlando, FL. After obtaining both a film degree and a Master's in instructional systems design from UCF, he began his DOD civilian career in the Advanced Instructional Systems Branch at the Naval Air Warfare Center Training Systems Division. There, he worked on special projects for the Navy and Marine Corps for 10 years before becoming the Lead Instructional Designer for the Army's Engagement Skills Trainer program at the Program Executive Office of Simulation, Training and Instrumentation (PEOSTRI), Orlando, FL. At PEOSTRI, he was responsible for several innovations in the way immersive training scenarios were created, experienced, and assessed; and this drive to push the boundaries of what's possible in simulations and training is what led him to ARL's ATSD. Since joining ARL in 2011, his award-winning research projects have spanned various domains including augmented reality, terrain visualization, and adaptive training systems.

Dr. Tiffany Barnes is an associate professor of computer science at North Carolina State University, who leads research in using data to personalize learning experiences, in creating games for education, exercise and energy, and in broadening participation. She received an NSF Faculty Early Career Development (CAREER) Award for her novel work in using data to add intelligence to science, technology, engineering, and mathematics (STEM) learning environments, and an NSF Improving Undergraduate STEM Education (IUSE) award to combine data-driven hints with data-driven pedagogical choices for learning in logic, probability, and programming. She co-leads the NSF Students and Technology in Academia, Research, and Service (STARS) Computing Corps that engages college students and faculty in tiered mentoring and leadership in outreach, research and service, and the NSF Beauty and Joy of Computing (BJC)-STARS project to develop faculty and teacher leaders to scale professional development for the new Computer Science Principles course. She serves on executive boards for the Association for Computing Machinery (ACM) Special Interest Group on Computer Science Education (SIGCSE), the International Educational Data Mining Society, the Artificial Intelligence in Education Society, and the IEEE Special Technical Community on Broadening Participation (STCBP). She has been on the organizing committees for several conferences, including SIGCSE, Educational Data Mining, and the Foundations of Digital Games. In 2015, she founded the IEEE STCBP's annual conference on Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT).

Dr. Benjamin Bell is a principal and founder of Aqru Research and Technology, where he leads Aqru's programs in AI for simulation, training, human-machine interaction, and work support environments. His research has addressed the use of simulation for training and education across a spectrum of applications, including K-12, higher education, military, and national security training. He has held academic faculty positions, chief executive positions in industry, and serves in leadership roles for several international conferences. He is an assistant adjunct professor at Embry Riddle, holds a PhD from Northwestern University and is a graduate of the University of Pennsylvania.

Dr. Winston "Wink" Bennett is the division technical advisor for the Warfighter Readiness Research Division of the 711th Human Performance Wing Air Force Research Laboratory. Through his more than 25 years of service in the Air Force research community, he has achieved international recognition as a leader in education, training, and performance measurement research. He has led the development of numerous research products that have since become part of the operational military community and, as such, significantly improved the mission effectiveness of US and coalition personnel. He pioneered training, education, and measurement technologies, as well as transitioned research results to the operational military and scientific and commercial communities. His leadership in developing methods permitting the systematic identification of training and education requirements – and the design and evaluation of technology and tools for addressing those requirements – is a hallmark of his scientific prowess and the basis

of many of his contributions. His efforts have produced some of the most groundbreaking training technology and research findings in the field and serve as a foundation for other researchers and practitioners to follow. Among the many examples of his project leadership success is the Mission-Essential Competency methodology, a new way to define and assess training requirements and capabilities for military and civilian occupations. He is a Fellow of Division 14 (The Society for Industrial and Organizational Psychology [SIOP]) and 19 (The Society for Military Psychology) of the American Psychological Association and he is an Air Force Research Laboratory Fellow. His work has not only contributed immeasurably to the body of scientific knowledge, but has transformed the approaches used by the US military and many coalition nations to establish (train), evaluate, and maintain their combat and mission support readiness.

Dr. Elizabeth "Beth" Biddle. is a Boeing associate technical fellow with 12 years' service with The Boeing Company. She currently provides technical leadership in the Boeing Research & Technology (BR&T) Advanced Learning organization to support the development of advanced training technologies. Prior to joining Boeing, she had over five years' experience in leading human performance and training research and development activities for academic, government, and small business organizations. She was awarded the Modeling & Simulation Award in Training by the Defense Modeling & Simulation Office (DMSO) in 2001 and nominated as a Charter Member to receive the Certified Modeling & Simulation Professional (CMSP) certification in 2002. She is currently the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Program Chair and was previously Secretary for the Simulation Interoperability Standards Organization (SISO) Conference Committee. She holds a PhD in industrial engineering and management systems from UCF, a MS in counseling and human development from Troy State University and a BA in psychology from Florida State University.

Dr. Michael W. Boyce is a third-year postdoctoral research associate at ARL-HRED, under the mentorship of Dr. Sottilare. His current focus is on investigating the use of adaptive training systems to support the instruction of military tactics and understanding how to design for ill-defined domains. His postdoctoral research project integrates GIFT and the Augmented Reality Sandtable (ARES). The goal of the project is to provide experimental data on measuring learner performance, as well as physiological and experiential data for cadets at the US Military Academy at West Point. He received his doctorate in applied/experimental human factors psychology from UCF in 2014.

Dr. Keith Brawner is a researcher for the Learning in Intelligent Tutoring Environments (LITE) Lab within ARL-HRED. He has 10 years of experience within US Army and Navy acquisition, development, and research agencies. He holds a Master's and PhD degree in computer engineering with a focus on intelligent systems and machine learning from UCF. His current research is in machine learning, active and semi-supervised learning, ITS architectures, and cognitive architectures. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs toward next-generation training.

Dr. Bert Bredeweg is an associate professor at the Informatics Institute within the University of Amsterdam (The Netherlands), leading the qualitative reasoning (QR) group. His research focus is the development of tools and expertise that support the acquisition of conceptual understanding of dynamic systems through conceptual modeling and simulation. Topics of interest include knowledge capture, QR, learning by modeling, cognitive diagnose, and HCI. He has supervised over 70 MSc and PhD students. He is an active member of the ecological informatics, QR, and educational technology communities. He is a regular senior reviewer and board member for the associated conferences and journals. He was part of the Computing Community Consortium (CCC)/NSF (USA) roadmap development (B. Woolf (Ed.), 2010), and consultant for SRI International (HALO project, 2010/2011), both USA, and acted four times as invited special issue editor for leading journals (most recently, guest editor for *IEEE Transactions on*

Learning Technologies – special issue: 6(3), 2013, pp. 194–257 – together with Dr. B.M. McLaren and Dr. G. Biswas). He acquired and coordinated the DynaLearn project (EU FP7).

Dr. Peter Brusilovsky has been working in the area of adaptive systems and e-learning for many years. Since 1993 he has participated in the development of several adaptive web-based educational systems including ELM-ART, a winner of 1998 European Academic Software Award. He was involved in developing practical e-learning courses and systems as a Director of Computer Managed Instruction at Carnegie Technology Education, one of the first e-learning companies in the United States. Currently, he continues his research on adaptive e-learning as a professor of information science and intelligent systems at the University of Pittsburgh. He has published numerous research papers and several books adaptive systems and e-learning. He is the editor-in-chief of *IEEE Transactions on Learning Technologies* and a board member of several other journals. He is also the immediate past president of User Modeling, Inc., a professional association of user modeling researchers.

Mr. Zhiqiang Cai is a research assistant professor with IIS at UofM. He has a MS in mathematics received in 1985 from Huazhong University of Science and Technology, P. R. China. After 15 years of teaching mathematics in colleges, he has worked in the field of natural language processing and intelligent systems. He is the chief software designer and developer of Coh-Metrix, OperationAries, CSAL AutoTutor, and many other text analysis tools and conversational tutoring systems. He has coauthored over 70 publications**.**

Dr. William J. Clancey is a senior research scientist at the Florida Institute for Human and Machine Cognition. His research relates cognitive and social science in the study of work practices and the design of agent systems. He has developed AI applications for medicine, finance, education, robotics, and spaceflight systems. He received a PhD in computer science at Stanford University and BA in mathematical sciences at Rice University. He was a founding member of the Institute for Research on Learning (1987–1997) and Chief Scientist of Human-Centered Computing at NASA Ames Research Center (1998–2013). He is a Fellow of the Association for Psychological Science, Association for Advancement of AI, National Academy of Inventors, and the American College of Medical Informatics. His seven books include *Working on Mars: Voyages of Scientific Discovery with the Mars Exploration Rovers* (recipient of AIAA 2014 Gardner-Lasser Aerospace History Literature Award). He has presented invited lectures in over 20 countries.

Mr. Brandt Dargue is an associate technical fellow at BR&T performing research into current and future training technologies including simulations, automated performance assessment, adaptive scenarios, intelligent tutoring, virtual environments, augmented/virtual reality, mobile platforms, gaming concepts, gaming technologies, interactive electronic technical manuals (IETMs), and electronic performance support systems (EPSSs). Employed at Boeing for 27 years, he has several patents in training technologies, has chaired or participated in several international standards development and study groups, has published/presented at numerous conferences, is a judge in the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Serious Games Showcase and Challenge, and was the program manager and principal investigator for research contracts such as The Enemy of Reason serious game and training effectiveness studies.

Dr. Michel Desmarais is professor at the Computer and Software Engineering Department of École Polytechnique de Montreal since 2002. His field of expertise is in the domains of HCI, e-learning, and AI. He has 15 years of experience in software project management. He was principal researcher of the HCI and computerized learning environments groups at the Computer Research Institute of Montreal between 1990 and 1998, where he directed a research program in HCI and computer assisted learning, and was involved in a number of research projects in close collaboration with private corporations. From 1998 to 2002, he was director of the web services department in a private company (mvm.com) and leader of a

number of R&D web-based software projects. He is the editor of the *Journal of Educational Data Mining* and has authored over 100 scientific publications. His research interests include student and user modeling, user-centered software engineering, HCI, probabilistic modeling, and recommender interfaces.

Dr. Michael Eagle is a postdoctoral fellow at Carnegie Mellon University's (CMU) Human-Computer Interaction Institute. His research focuses on deriving understanding from complex interaction data from intelligent tutors and video games. He has worked in data science at Blizzard Entertainment and Warner Bros. Interactive Entertainment (Turbine Inc.) He received a NSF Graduate Research Fellowship Program (GRFP) Honorable Mention award, a Graduate Assistance in Areas of National Need (GAANN) fellow, and Freeman-Awards for Study in Asia (ASIA) recipient. He was also the principal investigator on a NSF East Asia and Pacific Summer Institutes (EAPSI) grant, in which he traveled to Japan and collaborated with Japanese researchers also working in the educational data mining field. He graduated from North Carolina State University in December 2015 under the direction of Dr. Barnes.

Dr. Kenneth D. Forbus is the Walter P. Murphy Professor of Computer Science and Professor of Education at Northwestern University. He received his degrees from the Massachusetts Institute of Technology (MIT) (PhD in 1984). His research interests include QR, analogical reasoning and learning, spatial reasoning, sketch understanding, natural language understanding, cognitive architecture, reasoning system design, intelligent educational software, and the use of AI in interactive entertainment. He is a Fellow of the Association for the Advancement of Artificial Intelligence, the Cognitive Science Society, and the Association for Computing Machinery. He has received the Humboldt Award and has served as chair of the Cognitive Science Society.

Ms. Elena L. Glassman is an electrical engineering and computer sciences PhD candidate at MIT's Computer Science and Artificial Intelligence Lab, where she specializes in HCI. She uses program analysis, machine learning, and crowdsourcing techniques to create systems that help teach programming and hardware design to thousands of students at once. She has also taught and served in leadership positions for MIT Middle East Entrepreneurs of Tomorrow (MEET), which teaches gifted Israelis and Palestinians computer science and teamwork in Jerusalem. She earned her MIT electrical engineering and computer sciences BS and MEng degrees in 2008 and 2010, respectively, and expects to receive her PhD in 2016. She was awarded both NSF and National Defense Science and Engineering Graduate (NDSEG) fellowships and MIT's Amar Bose Teaching Fellowship. In addition to doing research at MIT, she has been a visiting researcher at Stanford and a summer research intern at both Google and Microsoft Research.

Dr. Benjamin Goldberg is a member of the LITE Lab at ARL-HRED in Orlando, FL. He has been conducting research in the M&S community for the past eight years with a focus on adaptive learning in simulation-based environments and how to leverage AI tools and methods to create personalized learning experiences. Currently, he is the LITE Lab's lead scientist on instructional management research within adaptive training environments and is a co-creator of GIFT. He is a PhD graduate from UCF in the program of M&S. His work has been published across several well-known conferences, with recent contributions to the Human Factors and Ergonomics Society (HFES), Artificial Intelligence in Education and Intelligent Tutoring Systems proceedings. He has also recently contributed to the *Journal Computers in Human Behavior* and *Journal of Cognitive Technology*.

Dr. Ilya Goldin is a Director of Data Science at 2U, Inc., where his role is to use learning sciences and data science to advance 2U technology for students and faculty. He has engaged in research on a variety of topics in adaptive and personalized learning, including mastery modeling, domain modeling and help-seeking in tutoring systems; personal epistemology in learning science; and peer assessment in open-ended learning activities. Prior to 2U, he was a research scientist in the Center for Digital Data, Analytics and Adaptive Learning at Pearson, and an IES Post-doctoral Training Program in Interdisciplinary Education (PostPIER) postdoctoral fellow at CMU. He holds a PhD in intelligent systems from the University

of Pittsburgh. He serves as an Edmund W. Gordon Fellow, funded by MacArthur Foundation and Educational Testing Service.

Dr. Robert Goldstone is Chancellor's professor in the Psychological and Brain Sciences Department and Cognitive Science Program at Indiana University, where he has been a faculty member since 1991. He received a BA from Oberlin College in 1986 in cognitive science, a Master's from University of Illinois in 1989, and a PhD in psychology from University of Michigan in 1991. His research interests include concept learning and representation, perceptual learning, educational applications of cognitive science, decision making, collective behavior, and computational modeling of human cognition. His interests in education focus on learning and transfer in mathematics and science, computational models of learning, and the design of innovative learning technologies. He was awarded two American Psychological Association (APA) Young Investigator awards in 1995 for articles appearing in the *Journal of Experimental Psychology*, the 1996 Chase Memorial Award for Outstanding Young Researcher in Cognitive Science, a 1997 James McKeen Cattell Sabbatical Award, the 2000 APA Distinguished Scientific Award for Early Career Contribution to Psychology in the area of Cognition and Human Learning, and a 2004 Troland research award from the National Academy of Sciences. He was the executive editor of *Cognitive Science* from 2001–2005, associate editor of *Psychonomic Bulletin & Review* from 1998–2000, and associate editor of *Cognitive Psychology* and *Topics in Cognitive Science* from 2007–2013. He was elected as a fellow of the Society of Experimental Psychologists in 2004, a fellow of the Association for Psychological Science in 2007, and a fellow of the Cognitive Science Society in 2006. From 2006 to 2011 he was the director of the Indiana University Cognitive Science Program.

Dr. Gregory A. Goodwin is a senior research scientist at ARL-HRED's ATSD in Orlando, FL. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He holds a PhD in psychology from Binghamton University and an MA in psychology from Wake Forest University.

Dr. Juho Kim [http://juhokim.com/] is a visiting assistant professor of computer science and a Brown Fellow at Stanford University. He will be an assistant professor in the School of Computing at the Korea Advanced Institute of Science and Technology (KAIST) starting fall 2016. His research interests lie in HCI, learning at scale, video interfaces, and crowdsourcing. He builds interactive systems powered by large-scale data from users, in which users' natural and incentivized activities dynamically improve content, interaction, and experience. He earned his PhD from MIT, MS from Stanford University, and BS from Seoul National University. He is a recipient of six paper awards from CHI and HCOMP, and the Samsung Fellowship.

Dr. David Landy is an assistant professor in the psychological and brain sciences at Indiana University in Bloomington, IN. He also earned his PhD in there in 2007 in computer science and cognitive science. In between, he was a post-doctoral research scientist at the University of Illinois at Urbana-Champaign and an assistant professor at the University of Richmond. His work on perceptual aspects of notational reasoning earned the 2007 David Marr prize from the Cognitive Science Society, and a new investigator award from the APA in 2008. His work on interactive and dynamic algebras has been funded by the Institute for Education Sciences.

Dr. Walter S. Lasecki is an assistant professor of computer science and engineering at the University of Michigan, Ann Arbor, where he directs the Crowds+Machines (CROMA) Lab. He and his students create interactive intelligent systems that are robust enough to be used in real-world settings by combining both human and machine intelligence to exceed the capabilities of either. These systems let people be more productive, and improve access to the world for people with disabilities. He received his PhD and MS from the University of Rochester in 2015 and a BS in computer science and mathematics from Virginia

Tech in 2010. He has previously held visiting research positions at CMU, Stanford, Microsoft Research, and Google[x].

Dr. Joseph J. LaViola Jr. is the Charles N. Millican Faculty Fellow and associate professor in the Department of Electrical Engineering and Computer Science and directs the Interactive Systems and User Experience Research Cluster of Excellence at UCF. He is the director of the M&S graduate program and is also an adjunct associate research professor in the Computer Science Department at Brown University. His primary research interests include pen-based interactive computing, 3D spatial interfaces for video games, human-robot interaction, multimodal interaction in virtual environments, and user interface evaluation. His work has appeared in journals such as *ACM TOCHI*, *IEEE PAMI*, *Presence*, and *IEEE Computer Graphics & Applications*, and he has presented research at conferences including ACM CHI, ACM IUI, IEEE Virtual Reality, and ACM SIGGRAPH. He has also coauthored *3D User Interfaces: Theory and Practice*, the first comprehensive book on 3D user interfaces. In 2009, he won an NSF Career Award to conduct research on mathematical sketching. He received a ScM in computer science in 2000, a ScM in applied mathematics in 2001, and a PhD in computer science in 2005 from Brown University.

Dr. Douglas B. Lenat is one of the world's leading computer scientists, and is the founder of the Cyc project and president of Cycorp. He Lenat has been a Professor of Computer Science at CMU and Stanford University and has received numerous honors including awarded the biannual International Joint Conference on Artificial Intelligence (IJCAI) Computers and Thought Award, which the highest honor in AI; named first Fellow of the Association for the Advancement of Artificial Intelligence, and Fellow of the American Academy for the Advancement of Science. He is a prolific author, whose hundreds of publications include *Knowledge Based Systems in Artificial Intelligence* (1982, McGraw-Hill), *Building Expert Systems* (1983, Addison-Wesley), *Knowledge Representation* (1988, Addison-Wesley), and *Building Large Knowledge Based Systems* (1989, Addison-Wesley). His 1976 Stanford thesis earned him the biannual IJCAI Computers and Thought Award in 1977. He received his PhD in computer science from Stanford University and his BA and MS in mathematics from the University of Pennsylvania. He is an editor of the *J. Automated Reasoning*, *J. Learning Sciences*, and *J. Applied Ontology*. He is a founder and Advisory Board member of TTI Vanguard, and is the only individual to have served on the Scientific Advisory Boards of both Microsoft and Apple.

Dr. Tyler Marghetis is a postdoctoral research scientist in the Department of Psychological and Brain Sciences at Indiana University. He is interested in how local, situated activity both reflects and reproduces larger sociocultural structures. He has particular interests in cultural practices associated with the domains of space, time, and mathematics. He has studied collaboration among mathematical experts; investigated children's emerging understanding of abstract time concepts; and conducted cross-cultural research on the conceptualization of space and number in indigenous communities in Mexico and Papua New Guinea. Running throughout this work is an attention to the web of mutual influences between bodies, technology, language, and reasoning. Before becoming a cognitive scientist, he competed internationally on the Canadian national wrestling team; this instilled an appreciation for the ways in which body and mind become regimented by habitual practice. He completed his PhD in cognitive science at the University of California, San Diego, and holds a Master's in mathematics education.

Mr. Behrooz Mostafavi is a PhD candidate at North Carolina State University under the direction of Dr. Barnes. His research focuses on improving individualized instruction in tutoring systems using data-driven methods and the mining of educational data. Applying and studying these methods to a tutor for propositional logic are the topics of his dissertation thesis. He will graduate with his doctoral degree in 2016.

Dr. Erin Ottmar is an assistant professor of psychology and learning sciences at Worcester Polytechnic Institute. Her research aims to develop and evaluate classroom interventions that improve mathematics

teaching and learning. She focuses on the intersections of educational, cognitive, and developmental psychology and is interested in understanding how cognitive and non-cognitive pathways combine to produce learning and growth in mathematics. She received her PhD in educational psychology in 2011 from the University of Virginia.

Dr. Philip I. Pavlik is an assistant professor and Director of the Optimal Learning Lab. The lab's mission is to describe models of learning so that these models can be used by instructional software to sequence and schedule practice. He completed his dissertation research with John Anderson in CMU's Psychology Department and has worked with Ken Koedinger in CMU's Human Computer Interaction Institute. He is current working on multiple existing grants and has applied for funding from both Department of Education (DOED) and NSF.

Dr. Anna N. Rafferty is an assistant professor at Carleton College. Her research focuses on using machine learning and computational cognitive science to build more effective personalized educational technologies. She is particularly interested in how probabilistic models of cognition can be leveraged to provide more effective feedback to learners and to adapt the experience of learners within an educational technology. She has also collaborated with groups focused on science education, such as the Web-based Inquiry Science Environment (WISE) and WestEd, to develop analytics and personalized feedback for chemistry activities used in the classroom. She received her doctorate in computer science from the University of California, Berkeley, and a Master's degree in symbolic systems from Stanford University.

Dr. Damon Regan is the external research and development lead at the ADL Initiative. He received his PhD in instructional technology and his BS in computer science from UCF. He received an MBA from the Crummer School of Business at Rollins College. At the ADL Initiative, he provides oversight of contracted research. Prior to his current role, he has led research projects focused on creating, sharing, and reusing learning objects, contributed to e-learning specifications promoting interoperability, and operated simulations supporting US Army training exercises.

Dr. Steven Ritter, founder and chief scientist at Carnegie Learning, has been developing and evaluating educational systems for over 20 years. He earned his PhD in cognitive psychology at CMU in 1992 and was instrumental in the development and evaluation of Cognitive Tutors for mathematics. Through leadership of Carnegie Learning's research department, he has led many improvements to the use of adaptive learning systems and math education in real-world settings. He is the author of numerous papers on the design, architecture, and evaluation of ITSs. He is lead author of an evaluation judged by the DOED's What Works Clearinghouse as fully meeting their standards and is lead author of a "Best Paper" at the International Conference on Educational Data Mining.

Dr. Erik Weitnauer researches and develops tools that support mathematical reasoning and learning at Indiana University Bloomington. He is the lead developer of Graspable Math, a dynamic algebra notation system. He earned his Master's and PhD from Bielefeld University, Germany.

Dr. Joseph Jay Williams is a research fellow at Harvard University's Office of the Vice Provost for Advances in Learning. He is also affiliated with the Intelligent Interactive Systems Group in Harvard Computer Science, and the ASSISTments K12 educational platform at Worcester Polytechnic Institute. His research designs adaptive systems for online content, by integrating research in psychology and education, human-computer interaction, and statistical machine learning. To make any static website become intelligently adaptive, he uses powerful systems for randomized A/B experiments. Examples range from adaptive explanations for how to solve math problems, to self-personalizing emails that change people's behavior. These systems continually crowdsource new "A" and "B" designs from psychological scientists, using randomized comparisons to evaluate how helpful these alternative designs are, for people with different profiles. Algorithms from statistical machine learning use this data for real-time enhancement

and personalization, by changing which designs are presented to future users. He completed a postdoc in 2014 at Stanford University's Graduate School of Education. He received his PhD in 2013 in from UC Berkeley's Psychology Department, where his research investigated why explaining "why?" helps learning, and used Bayesian statistics and machine learning to model human cognition. He received a BS from University of Toronto in cognitive science, AI, and mathematics, and is originally from Trinidad and Tobago.

Mr. Peng Xu is a PhD student in educational data mining in École Polytechnique de Montréal. His research interest is using machine learning and data mining techniques for educational data problems. He earned his BS in mathematics from Sichuan University, China.

adaptive training, 10, 97, 98, 150, 153, 163, 188, 190, 191, 192, 193, 195, 197, 199, 200
adaptive Training, 102, 191
adaptive tutoring, 5, 10
after action review, 154, 189
alternating least squares factorization, 43
Amburn, 98, 102, 163, 192, 195, 197, 239
architecture, 3, 9
artificial intelligence, 63, 69, 100, 128, 203, 225, 237
ASSISTments, 117, 130, 132
authoring tools, 6, 7, 8, 19, 34, 113, 115, 164, 208, 211
AutoTutor, 155, 163, 205, 206, 207, 208, 209, 210, 211
Barnes, 41, 94, 139, 141, 239
Bayesian knowledge tracing, 179
Behrooz Mostafavi, 244
Bell, 69, 239
Bennett, 69, 147, 239
Biddle, 94, 147, 240
Bloom's revised taxonomy, 30
Boyce, 195, 240
Brawner, 93, 102, 115, 185, 197, 240
Bredeweg, 58, 64, 66, 240
Brusilovsky, 163, 165, 176, 241
Cai, 163, 205, 206, 210, 211, 241
case-based reasoning, 29, 233
Clancey, 69, 71, 72, 241
cognitive diagnosis model, 41
cognitive task analysis, 149
Cognitive Tutor Authoring Tools, 45
CogSketch, 164, 226, 227, 228, 230, 232, 233, 234
computer-adaptive instruction, 165
crowdsourcing, 93, 94, 95, 100, 127, 128, 130, 132, 134
CyclePad, 64, 66
Dargue, 94, 147, 241
decision tree, 44, 45, 49
Desmarais, 39, 41, 43, 45, 94, 241
domain
  expertise, 21, 24
  ill-defined, 21, 22, 24, 29, 34
  learning, 15
  marksmanship, 195
  pedagogy, 21, 22, 24, 32
  psychomotor, 164, 193
  social, 187
  team, 187
  well-defined, 22, 153
domain knowledge, 8, 21, 24, 25, 30, 32, 33, 57, 60, 88, 98, 124, 141, 147, 149, 168, 170, 171, 205, 206, 221, 234
domain knowledge file, 155, 161, 199, 200
domain model, 3, 6, 15, 17, 39, 59, 60, 69, 70, 73, 74, 75, 77, 86, 87, 116, 123, 161, 168, 179
  characteristics and requirements, 73
  marksmanship, 164
domain module, 3, 6, 102, 103, 199
domain theory, 61
Eagle, 94, 139, 140, 141, 242
Engagement Skills Trainer, 196, 197, 239
expert model, 8
expertise, 9, 11, 15, 20, 21, 22, 23, 24, 33, 71, 88, 97, 128, 147, 148, 149, 150, 152, 154
  archetypes, 27
  domain, 21, 24, 124
  ill-defined, 21
  vs pedagogy, 21
  well-defined, 21
Forbus, 61, 64, 65, 164, 225, 226, 227, 230, 233, 242
Generalized Intelligent Framework for Tutoring, 4
GIFT, 3, 5, 6, 7, 8, 9, 10
Glassman, 94, 128, 135, 242
Goldberg, 98, 102, 104, 115, 163, 185, 192, 195, 197, 201, 242
Goldin, 94, 115, 119, 121, 242
Goldstone, 163, 213, 214, 215, 216, 217, 218, 221, 222, 243
Goodwin, 93, 98, 99, 221, 243
Graesser, 91, 93, 115, 163, 186, 195, 205, 206, 207, 210, 211, 237
Hu, 13, 115, 163, 205, 206, 210, 211, 237
ill-defined domain, 15, 17, 19, 20, 21, 23, 24, 26, 27, 29, 30, 31, 33, 34, 153
instructional systems design, 155
intelligent tutoring system, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 16, 17, 19, 21, 22, 23, 24, 26, 27, 28, 29, 30, 32, 33, 34, 39, 49, 50, 53, 54, 55, 69, 70, 71, 73, 74, 75, 81, 86, 87, 88, 94, 98, 107, 111, 112, 113, 127, 135, 138, 142, 143, 149, 150, 151, 153, 154, 155, 156, 161, 162,魴163, 164, 167, 170, 172, 176, 185, 186, 189, 190, 192, 195, 205, 206, 225, 233, 234
Kim, 94, 127, 128, 243
knowledge component, 16, 17, 27, 39, 94, 115, 116, 123, 141, 148, 149, 168, 179, 205, 206, 207, 210
Landy, 163, 214, 215, 216, 217, 218, 221, 243
Lasecki, 94, 127, 128, 243
LaViola, 163, 185, 192, 244
learner model, 3, 5, 8, 10
learner module, 3, 6
learning domain, 15
Lenat, 49, 100, 244
machine learning, 93, 94, 99, 127, 128, 129, 131, 134, 200, 201, 231
Marghetis, 163, 244
Markov Decision Process, 22
marksmanship, 97, 98, 99, 101, 102, 103, 104, 163, 189, 193, 195, 196
  adaptive, 163, 192
  Basic Rifle Marksmanship, 98, 99, 102, 196
  domain, 195

model, 102
tasks, 163
mental model, 25, 94, 95, 147, 148, 149, 151, 153, 154, 155
incorrect, 205
shared, 187
MOOClet, 134
Mostafavi, 94, 138, 139, 141, 142
Nye, 26, 238
Olney, 91, 93, 115, 206, 238
ontology, 17, 49, 50, 51, 54, 60, 61
component, 61
process, 61
Ottmar, 163, 218, 244
Pavlik, 94, 115, 117, 119, 206, 245
pedagogical model, 3, 9, 124
pedagogical module, 3, 6
pedagogy, 7, 15, 19, 21, 32, 33, 34
adaptive, 30
applied, 19
approaches, 15
domain, 22, 24, 32
strategies, 34
vs expertise, 21
well-defined, 28, 33
qualitative model, 57
Qualitative Process Theory, 61
qualitative representation, 57, 60
Rafferty, 94, 127, 245
Regan, 93, 98, 245
reinforcement learning, 129
Ritter, 27, 94, 115, 121, 245
Sinatra, 93, 107, 108, 109, 111, 112, 113, 159, 161, 195, 238
Sottilare, 102, 108, 109, 111, 112, 113, 115, 159, 161, 163, 185, 192, 195, 196, 202, 205, 221, 238
SQL-Tutor, 168, 172
structured query language, 29, 81, 83, 101
tasks
complex, 196
ill-defined, 26, 27, 34
marksmanship, 104, 163
psychomotor, 163, 193, 196
well-defined, 26, 27, 163
team cognition, 187
team learning, 34, 187
tutor model, 3, 4
user interface, 3, 7, 116, 124, 149, 150, 230
Weitnauer, 163, 218, 245
well-defined domain, 22
well-defined pedagogy, 33
Williams, 61, 94, 127, 128, 129, 130, 131, 132, 133, 245
Xu, 39, 41, 45, 94, 117, 118, 246

Design Recommendations for Intelligent Tutoring Systems

Volume 4
Domain Modeling

Design Recommendations for Intelligent Tutoring Systems (ITSs) explores the impact of intelligent tutoring system design on education and training. Specifically, this volume examines "Domain Modeling". The "Design Recommendations book series examines tools and methods to reduce the time and skill required to develop Intelligent Tutoring Systems with the goal of improving the Generalized Intelligent Framework for Tutoring (GIFT). GIFT is a modular, service-oriented architecture developed to capture simplified authoring techniques, promote reuse and standardization of ITSs along with automated instructional techniques and effectiveness evaluation capabilities for adaptive tutoring tools and methods.

About the Editors:

- ***Dr. Robert Sottilare*** *leads adaptive training research at the Army Research Laboratory and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*
- ***Dr. Arthur Graesser*** *is a professor in the Department of Psychology and the Institute of Intelligent Systems at the University of Memphis and is a Senior Research Fellow in the Department of Education at the University of Oxford.*
- ***Dr. Xiangen Hu*** *is a professor in the Department of Psychology at The University of Memphis and visiting professor at Central China Normal University.*
- ***Dr. Andrew Olney*** *is an associate professor and Director of the Institute for Intelligent Systems at the University of Memphis.*
- ***Dr. Benjamin Nye*** *is the Director for Learning Science Research at the Institute for Creative Technologies at the University of Southern California.*
- ***Dr. Anne Sinatra*** *is an adaptive training scientist at the U.S. Army Research Laboratory.*

A Volume in the Adaptive Tutoring Series

9 780989 392396