

Final Report on:

An architectural overlay: Modifying an architecture to help cognitive models understand and explain themselves
(N00014-02-1-0021)

Submitted to: Office of Naval Research
Attn: Harold Hawkins ONR Code: 342
800 North Quincy Street
Arlington, VA 22217-5660

Submitted by: Frank E. Ritter and Steven R. Haynes
College of Information Sciences and Technology
316G IST Building
University Park, PA 16802
Phone: 814-865-4453
Email: ritter@ist.psu.edu

Betzy A. Wilson
Manager, Research & Admin. Services
College of Information Sciences and Technology
334 IST Building
University Park, PA 16802
Phone: (814) 863-6801
Fax: (814) 865-7484
e-mail: bwilson@ist.psu.edu

20060302 165

Contractual Contact: Office of Sponsored Programs
110 Technology Center Building
University Park, PA 16802
(814) 865-1372

A Non-Profit Educational Institute

Table of Contents

1. Technical Summary of the Project	1
Herbal provides explanations	1
Herbal works now	2
Herbal's design is robust and reusable	2
Herbal appears easy to use.....	4
dTank makes it easy to test models	5
SegMan offers a way for models to interact with interfaces	6
Work with G2A	7
Work with Vista	7
Summary	7
2. Future Work with Herbal	8
Task 1: Extending Herbal.....	8
Task 2: Testing Herbal	9
Task 3: Applying Herbal	9
Task 4: Updating dTank.....	9
Task 5: Creating intelligent opponents with Herbal.....	10
Summary	10
3. Relevance to the Navy of These Results	10
Creating models more quickly, more accurately, more affordably to help with simulation and training.....	11
More understandable agents and opponents.....	11
Application to timely problem of anti-terrorism force protection.....	11
More intelligent opponents.....	11
An environment for testing modeling and behavior representations.....	12
Based on an extensively used architecture	12
Accelerators of these payoffs	12
4. References	14
Appendix : Outputs	17
Papers submitted	17
Papers and other outputs completed.....	18
Presentations.....	18

1. Technical Summary of the Project

Two challenges confront efforts to develop effective intelligent systems to work in complex domains: making them easier, faster and less expensive to develop, and making them easier to comprehend, use, and trust (Haynes, Councill, & Ritter, 2004). This applies to cognitive models as well (Ritter, 2004). We have created a high-level behavior representation language and integrated development environment (IDE) called Herbal (Cohen, Ritter, & Haynes, 2005) that answers these challenges. Herbal provides explanations, it works, has been used by 60 people as the basis of a course on modeling and as a tutorial at the BRIMS conference, and the data so far suggest a 3 to 7 x performance increase compared to developing Soar code directly. The next steps are to evaluate and extend Herbal, and to apply Herbal to a real-world problem.

This proposal makes intelligent systems more comprehensible to developers, users, and other stakeholders through the development and evaluation of the high-level behavior representation language, Herbal, and related tools. These systems are designed from the ground up based on user studies and explanation theory to account for the complexities of human-computer interaction with intelligent systems, both in development and in use. Two key design features of the approach are the ability to capture explicit design rationale from development activities, and then to use this design knowledge as the basis for explanations of the resulting system's structure, behaviors, and purpose.

Designing, implementing, and using intelligent systems can be difficult tasks considering that the background and expertise of developers can vary from novice to expert (Ritter et al., 2003). Part of the problem stems from the fact that cognitive architectures most often use low-level programming languages to model behavior. Both Soar (Laird, Newell, & Rosenbloom, 1987) and ACT-R (Anderson et al., 2004) for example, use production rules as their primary programming construct. A wider variety of users can be supported in the task of developing intelligent systems using a language that maps more directly to the user's domain. For example, if this high level language is built on top of Soar, and closely resembles the problem space theory used by the Soar architecture (Newell, 1990), it would make Soar easier to use, while maintaining the features of Soar that make it interesting, including learning, interaction, and reactivity.

Herbal provides explanations

Herbal provides an integrated development environment for users to create models in Soar. The resulting model is compiled into Soar productions. The IDE and an associate viewer for use at runtime (both shown in Figure 1) help to explain the structure and behavior of cognitive models.

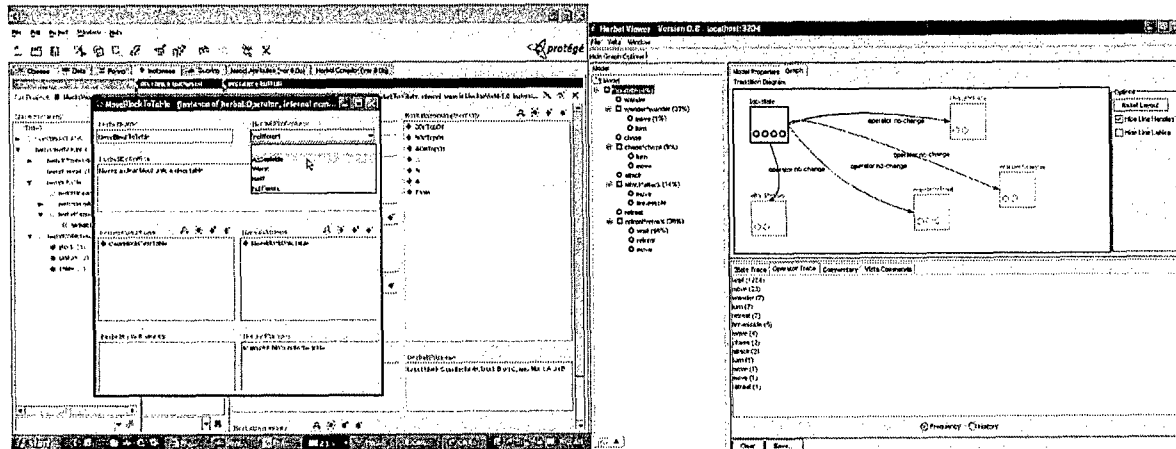


Figure 1. Herbal interface (left) and Herbal Viewer Interface (right). An operator instance pane is on the top of the IDE. The Herbal and model ontologies are visible in the far left pane of each window.

Herbal provides support for answering core model-related questions: *What* questions of models (identity of objects, definition, relation, and events), *How* questions (how do I use the model, how do I use the model's interface), *How does the model work*, *Where* questions (where did the event happen), *Functional* explanations (the purpose of the model), and *Pragmatic* explanations (why didn't something else happen?). The explanations are available in graphical and textual output.

Herbal works now

Herbal has been taught to over 60 people, 35 for a semester course on cognitive modeling at Penn State (IST 402 Emerging technologies: Models of human performance), and 30 paying tutees for 3 hours at BRIMS (Ritter, Morgan, Stevenson, & Cohen, 2005). This usage has improved Herbal, and suggests that Herbal has all the functionality Soar provides on its own, but is easier to use. A manual supports usability (Cohen & Ritter, 2004), and to improve its usability we are currently building a library of models.

Herbal's design is robust and reusable

Herbal uses an explicit design. User needs were gathered from a study on how experts and novices interact with Soar Tech's Situation Awareness Panel (Avraamides & Ritter, 2002). The study results were reanalyzed to find the kinds of questions Soar users ask (Councill, Haynes, & Ritter, 2003; Haynes, Ritter, Councill, & Cohen, submitted January 2005). These results are highlighted in Figure 2. The users' questions were used along with several explanation scenarios to generate a robust, reusable, responsibility-based design based on a compiler (Haynes et al., 2004). This explanation approach is based on the structure of the model as well as information gathered from the modeler as they create the model (design rationale) (Haynes, 2001).

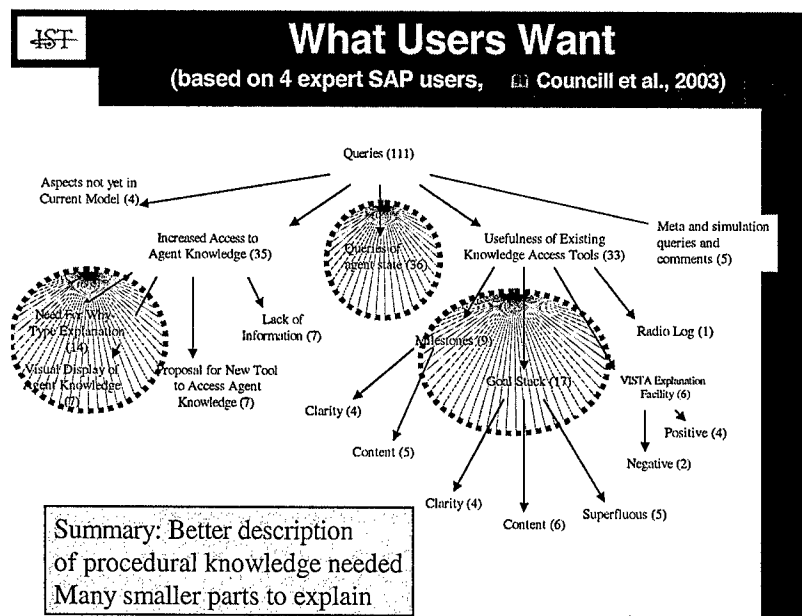


Figure 2. Summary of the usability study on what users want (Councill et al., 2003). The circles indicate major categories of questions.

Herbal's IDE is realized as an extension to popular ontology editor, Protégé, and an existing ontology language, RDF. The basic ontology, the objects models are created from, is shown in Figure 3. This use of standard tools makes it easier for new modelers to adopt the development environment, and means that cognitive models are represented in a portable data format. A model is compiled into Soar rules using an XSLT script. The explanations are provided graphically in the IDE as well as in a tracing tool. Many explanations are graphical, but we are also generating textual explanations concurrent with the model running.

These results (data and design) are reusable by all intelligent agents that want to create explanations—we have reported what needs to be done to support explanations (Haynes et al., submitted January 2005).

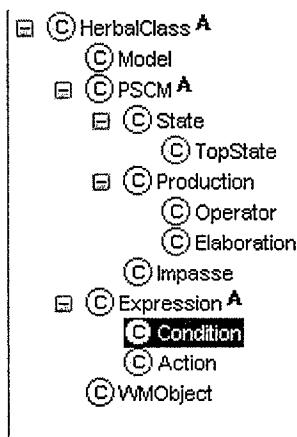


Figure 3. Herbal Ontology Class Hierarchy, taken from its interface.

Herbal appears easy to use

Herbal can be taught within a semester course. While we did not perform formal studies, anecdotally, students taking this course found Soar easier to use than previous classes have. When an undergraduate created a medium-sized dTank model, he was writing code (measured as productions) as quickly as a CMU graduate student using the TAQL Soar compiler, which itself was 3x normal Soar speed (Yost, 1993). This result is documented in a recent conference paper (Morgan, Cohen, Haynes, & Ritter, 2005). (The UG is now working at Soar Tech full-time). Figure 4 shows the time per rule (up to about 30 rules), and the cumulative average time per rule. By the end of the model the time per rule is approximately 3 minutes per rule. This is the time that Yost reported for himself and other CMU graduate students using TAQL.

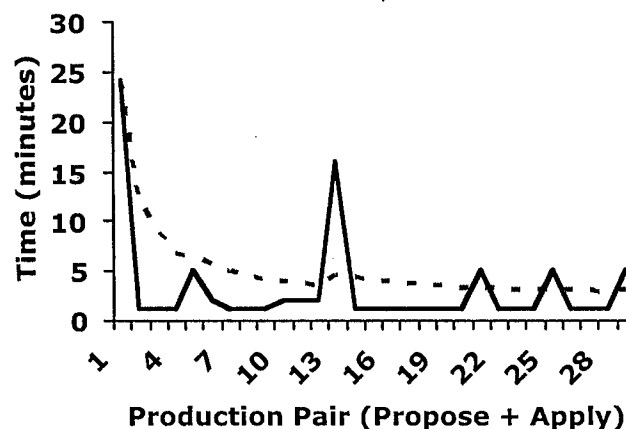


Figure 4. Time per rule (solid) and cumulate average time per rule (dashed).

Herbal has a code expansion ratio of approximately 8 to 1 on a simple system (blocks world). This number would, of course, change with different systems. The expansion ratio is fair because comments represent either explanation material currently used, or explanation material to be used. Brooks (of the *Mythical man-month* book, 1975) would argue that this will lead to a 8x speedup writing code because programmers he studied appear to program a set amount of code per day across languages, thus higher level languages are more productive. Here, however, some of the 8x is providing explanation, not just decreasing programming time.

The results of the G2A compiler (GOMS to ACT-R) system is consistent with the Herbal results. G2A was also created as part of this project to explore high-level behavior compilers. G2A provides a 7.5x code expansion. It allowed a developer to create ACT-R models in 30 min. that took a good masters student 2 weeks (~ 60 hours) to write in ACT-R. The speed-up reported at a recent ONR workshop of up to 30x is thus conservative—the speedup is probably more like 100x. This speedup is documented in a journal article (St. Amant, Freed, & Ritter, 2005), based on a conference paper that won best applied paper at the ICCM conference (St. Amant & Ritter, 2004).

There is increasing interest in high level modeling languages, for example, Visual Soar, SoarDoc, ViSoar, CAST, and HLSR. In comparison to other projects in this area, John et al's work creates models, but not as fast as G2A, and not as generally as Herbal. The Pearson/Laird system does not provide as general a tool as Herbal. HLSR and the Pearson/Laird tool are not implemented to the same level as Herbal, and certainly not tested to the same degree as Herbal, in terms of general usability, programming speed, and code expansion. We were struck by how many things Herbal does now that others propose to do. Herbal generates the types of explanations users want, and a mapping from question type to answer was provided in the talk slides presented in June 2005.

Herbal has been praised publicly by several people. Chris Glur on the Soar mailing list in June 2005 noted "After the first few hours spent looking into SOAR, I wondered why one was forced to grovel down at low level. And I've been waiting decade(s) for what Herbal seems to describe." Brian Haugh of IDA noted at the BRIMS'05 tutorial that "...it deserves more funding." During and after the BRIMS tutorial we received 8 enquiries.

dTank makes it easy to test models

Part of Herbal's success is because we can create and test interesting Herbal models quickly using dTank (acs.ist.psu.edu/dTank). dTank is a Java-based game explicitly developed for testing explanation of dynamic, adversarial models (Morgan, Ritter, Stevenson, Schenck, & Cohen, 2005). It provides a simplified version of the large synthetic environments like ModSAF for testing ideas—It is a ModSAF-like sandbox.

dTank appears to be 100 to 1,000x faster to use (setup a model connection, add a model, run a model) than a full synthetic environment. These numbers are based on having junior and senior undergraduate students and graduate students at Penn State and Lock Haven use dTank within 5 minutes. dTank is not a replacement for systems like ModSAF, jSAF, and OTB. But when their size and complexity are not needed, dTank offers a usefully sized, ready-built alternative with some surface validity.

dTank allows multiple players and teams on multiple machines, and works where Java works. Recently it has been made more robust. The correspondence between agents and human vision has been improved; its speed has been improved; its interface has been made easier to use; and we have now four architectures use it (CAST, Soar, Jess, and Java). We have examined how to include additional architectures like ACT-R, JACK, and COJACK. Our preliminary work suggests that these should be straightforward.

dTank has been used by an Army MURI (Sun, Council, Fan, Ritter, & Yen, 2004) to compare teamwork in different architectures. It has been used at Lock Haven U. (www.lhup.edu/~mcohen/dTank/dTankJess.htm, Cohen, 2005) and at the Federal U. of Uberlandia (Brazil) to teach agent programming. At least two other universities and one government lab and one private contractor have made enquiries about using it. We have an informal agreement that a version will be included in the ACT-R distribution in the future.

SegMan offers a way for models to interact with interfaces

Models need to be able to interact with task environments. As a portion of our project Ritter had several students work with other resources to explore this problem. Sim-eyes and -hands are a way for models to interact directly with interfaces. We continued to work on apply SegMan (St. Amant & Riedl, 2001) to support this approach. SegMan is a tool developed by Robert St. Amant. It parses screen bitmaps and generates operating system function calls to allow models and agents to interact with interfaces.

Figure 5 shows some of the interfaces that we have used SegMan to interact with. In these cases the interfaces were not modified to support the interaction. Thus, the first telephone interface, while written in Tcl/Tk, did not have to be modified to be interacted with (St. Amant et al., 2005). The driving game was written in Java, and taken from the net. We used SegMan to predict driving performance, and we did this without modifying the interface (Ritter, Van Rooy, St. Amant, & Simpson, in press). The casino interface was, in fact, presented as a bitmap to the user through a browser, and was designed to not be usable by non-human agents. A graduate student in a weekend was able to create a model, install SegMan, and test a theory about gambling strategies. The cell phone is a bitmap taken from the Internet. We created a model to test how to optimize its design (St. Amant, Horton, & Ritter, accepted 2005). The robot interface was not instrumentable. A master's student used an ACT-R model to predict human performance on the interface (Kukreja, 2004).

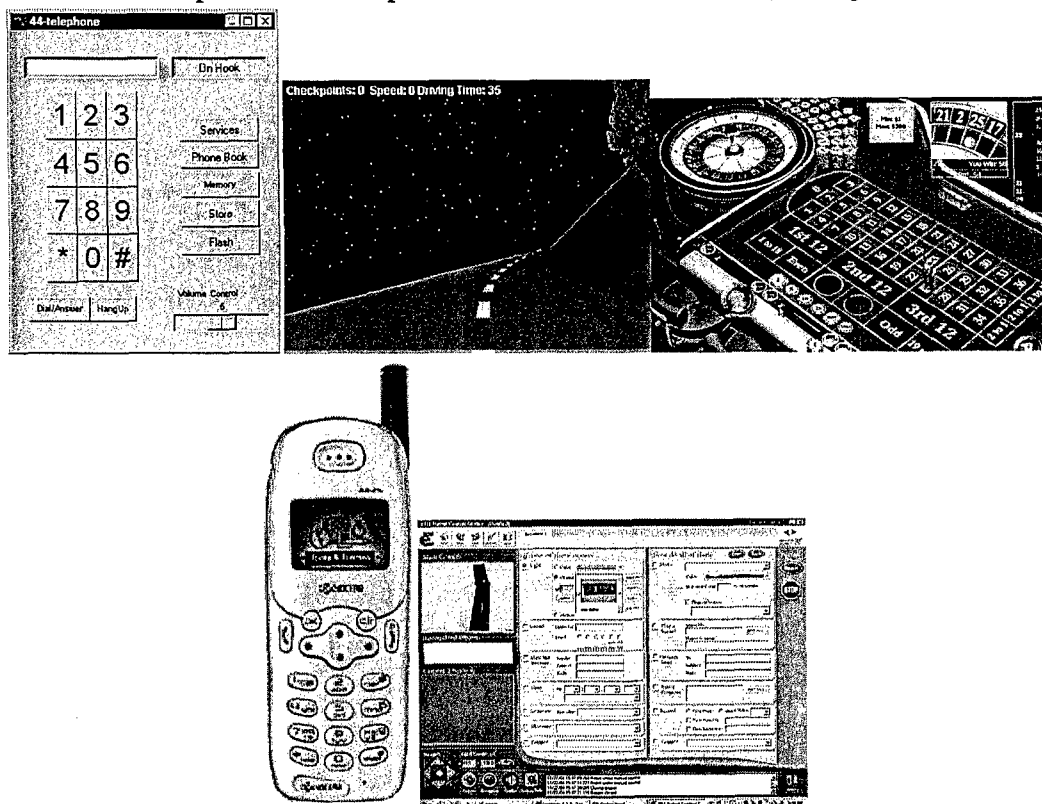


Figure 5. Interfaces that models interact with using SegMan.

This approach lets models be applied to more interfaces, and while not perfect yet, it appears to do so faster than previous approaches of instrumenting interfaces. We estimate that it can interact with new cell phone interfaces, for example, 20 times faster than instrumenting them.

Results from the G2A supports the role of compilers in modeling

We also worked on a compiler for ACT-R. this project was also done with Robert St. Amant. This compiler, GOMS to ACT-R (G2A), was primarily done by St. Amant for other purposes. Our paper was awarded the Siegel-Wolf Award for best applied modeling paper at the 2004 International Conference on Cognitive Modeling (St. Amant & Ritter, 2004), and was subsequently published as a journal article (St. Amant et al., 2005).

G2A appears to allow users to generate code about 20x faster than plain ACT-R. This measure is based on two people creating models to use cell phones (including some of the ones shown in Figure 5). It took about 2 weeks for a masters student who knew ACT-R to make an ACT-R model, and about two hours using G2A. (Thus, the ratio is something like 40 hours to 2 hours, but related changes will remain much faster.) G2A code expands at about a factor of 7.5 to 1, which suggests that 20x might be a somewhat high prediction.

Work with Vista illustrates the role of displays

We put some effort into exploring how to use Vista to help create displays for Herbal. Figure 6 shows a display we created to show sequential behavior (Daughtry & Ritter, 2005). It has been used to explain models in Herbal in reports (Morgan, Ritter et al., 2005), for example, to illustrate strategies. Figure 6 shows how a tank in dTank basically waits until something is visible. Then some cognition occurs in preparation for attack, then another precept indicates that it is time to attack.

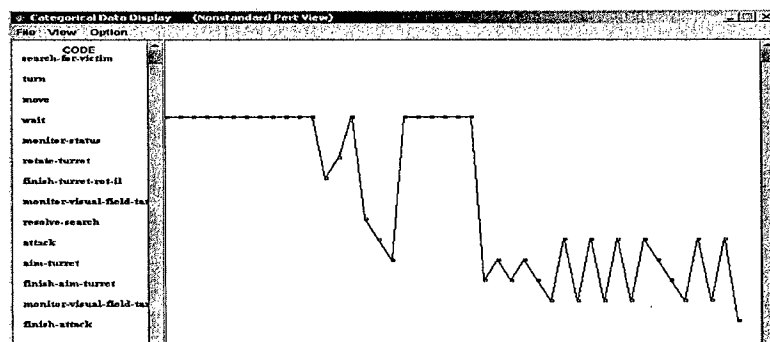


Figure 6. Vista display of a Herbal agent playing dTank.

Summary

The project that developed Herbal and dTank has created a reusable design to create high-level behavior representation languages that can provide explanations at the model level. The design is based on user studies and surveys of the literature. The resulting models are organized graphically in an ontology editor, and should be easier to comprehend and reuse because of this standard, portable representation (but this reusability needs to be tested further).

The resulting system, Herbal, provides now nearly all the explanation types that users request, but the usability of these explanations still need to be improved.

Herbal has been initially tested with over 60 users. These tests have led to some successes as noted in its use by undergraduate students, graduate students, and colleagues (Cohen, 2005; Cohen, Ritter, & Haynes, 2005; Morgan, Cohen et al., 2005). Herbal will need to be tested with more advanced users creating larger systems.

Herbal's design appears to be quite sound, and it should now be applied at this point by modifying Herbal to compile into additional base languages besides Soar, such as JESS and ACT-R. It should also be applied by creating models that use the metacognitive facilities Herbal can include in a model to create an intelligent opponent in dTank.

dTank works, but it needs to be made even easier to use as people find more uses for it, and want to run larger simulations to test models, and to run these tests more routinely.

These open tasks are presented in expanded form in the next section.

2. Future Work with Herbal

Here we describe the new tasks suggested by the technical basis above.

Task 1: Extending Herbal

Herbal needs to be extended. Herbal has been used to create modest sized systems ranging from 10 to 100 rules. The basic technology, Protégé, has been used to create ontologies with up to 100,000 objects, so we believe that Herbal can scale up, but this remains an open, empirical question. One has to imagine, however, that 400 operators in Herbal would be far easier to write and manipulate than 1,200 unstructured productions, but this improved usability remains to be demonstrated for larger models. Smaller models appear to be at least 3x faster to write. We are currently writing some models to test how Herbal scales, but this remains an area to test and extend Herbal, and resources are needed to demonstrate a scaling.

There are also several areas where our current users have pointed out where Herbal can be expanded and improved. These include the ability to do some automatic model checking and the ability to make the details easier to enter and write. We also will use the design rational in Herbal to provide concurrent audio output from the model. This output will provide an additional explanation modality

Programming in Herbal language involves instantiating objects using these ontological classes. The process of programming a model is reduced to instantiating objects from a set of classes, instead of coding structure implicitly in and across a large set of heterogeneous Soar productions. When objects are instantiated, the ontology includes additional attributes that allow the developer to describe the intent of the object. This descriptive information is automatically embedded in the generated Soar code as comments when the model is compiled, and will be used in future version of Herbal to generate concurrent and post-task explanations. The explanation display (the Herbal Viewer) also includes a listing of the

model operators, and tools to explore and manipulate them, and their substructures in a uniform way.

The Herbal Viewer is a tool designed to provide a portal into an intelligent agent developed using the Herbal IDE. The Viewer currently includes basic explanation capabilities that will be significantly expanded as part of the work proposed here. In particular, we propose to use results from the various analytic and empirical evaluation studies discussed below as part of a feedback loop driving further development of the Herbal tools.

These research activities will result in a number of products for dissemination including the Herbal language specification, the Herbal language implementation as an open-source Protégé plug-in, and the Herbal Viewer as an open-source Java application. In addition, we have published papers on a number of topics related to the design and development component of the project. These present a reusable design and the lessons learned. We have prepared a paper on design patterns for explanation facilities in intelligent systems, explanation filtering and tailoring techniques, and explanation visualization.

Task 2: Testing Herbal

Herbal further testing and augmentation. We will be testing Herbal in two ways. We will run a more formal user study on how fast users can create models in Herbal. Second, we will modify Herbal to compile into another AI language, such as Sandia's Java Expert System Shell (JESS, Friedman-Hill, 2003).

Task 3: Applying Herbal

Herbal needs to be applied to new problems and new domains. This will provide further lessons for the language, but it will also harvest and apply the results so far. We will integrate Herbal agents into a large-scale system designed to enhance defensive anti-terrorism planning and resource allocation. We will develop a Herbal agent in the USMC anti-terrorism resource allocation decision model (Haynes, Kannampallil, Larson, & Garg, 2005) as an example use of Herbal in systems applicable to military, government, and industry contexts.

Task 4: Updating dTank

Herbal's associated sandbox needs to be upgraded. We have developed and used the dTank microworld (Morgan, Ritter et al., 2005) to illustrate Herbal's explanation capabilities. It is from 100x to 1,000x easier to use than JSAF and related large simulations, so it is useful for testing models. It is so easy to hook models up to it that undergraduates have used it routinely in class.

dTank has also been used by Java, Soar, Soar/Slip, and CAST agents (Ritter, Sun, Schenck, Gross, & Stevenson, in preparation). Two MoD projects and an DMSO project both plan to use it, and it has been used by an Army MURI (Sun et al., 2004). ACT-R has agreed, in principle, to include a version in their next ACT-R release. It has the promise of being a successful and widely used piece of software to support work in synthetic environments.

Task 5: Creating intelligent opponents with Herbal

Herbal can be used to create more intelligent opponents. Herbal provides a declarative representation of the knowledge in the model. We can create a model that reasons about its behavior and modifies its behavior and learns using meta-cognitive methods because this representation knowledge is available. This will provide a better model of learning, and also a more intelligent opponent. It will also stretch and demonstrate our explanation capabilities, as this will produce a model that has something to explain, and it will be a model that people can learn from.

The exploratory version of this model will be created in dTank. This approach makes the process more tractable, and allows much more rapid exploration.

Summary

Our multidisciplinary research program has helped introduce a new concept to cognitive architectures, that of a high-level language and compiler. The compiler is based on a theory and empirically derived design. The resulting compiler, Herbal, has been tested in a variety of ways. Feedback from its use and from an ONR program review workshop has led to suggestions for extensions and further tests.

Our design supports applying the results quite widely by putting them into existing pieces of software with large user communities. Our core team of investigators has shown that they are willing and able to combine their separate research programs to create a new approach to cognitive modeling, one that applies results from software engineering and empirical tests of software to the area of cognitive modeling.

Our team now understands how to design, create, test, and distribute a high level language for modeling. Our team has worked extensively together to create Herbal and dTank, and we are able to rapidly include changes to the system.

This project hastens the transition of basic research findings to practical application. The use of a high-level compiler for cognitive architectures can be and has been used immediately by others, and as we propose here, could also be applied to ACT-R models in synthetic environments. The tasks proposed here will provide several applications.

3. Relevance to the Navy of These Results

We note payoffs for the Navy here, as well as several accelerators of these payoffs. At an abstract level, the relevance of the research and technology development proposed here relates to any situation involving design of complex systems and subsequent exploration, explanation about these designs including their intention, the rationale behind their form, and the constraints that dictate what is possible in today's engineering, economic, and socio-political context. These results thus, with further development will apply to a wide range of systems.

Creating models more quickly, more accurately, more affordably to help with simulation and training

Models created in Herbal can be used in place of humans in simulation and training. Herbal appears to make models 3x to 7x faster to create. Making models easier to create will not just save time; ease of use should also lead to creating more complete and more accurate colleagues and opponents to support training activities. Making models easier to use for design and for synthetic environments is important for the Navy.

More understandable agents and opponents

Models created in Herbal or in tools that use Herbal's design will include more explanatory information. This will assist in their debugging, in their verification and validation, and in their use in synthetic environments. It also means that users and other project stakeholders are provided with a much richer depth of knowledge about the tools they use and the domains in which they work. Technology designs embed and inscribe significant knowledge above and beyond what is apparent to a user of a given technology, especially software. Among the foundations of the work proposed here is exposing this knowledge so that users are able to draw on the expertise marshaled in support of technology design.

Application to timely problem of anti-terrorism force protection

Anti-terrorism planning is currently among the most important and high profile domains in systems research. Since the events of 9/11 and subsequent analysis of the lessons learned from those events, it is broadly recognized that ensuring Force Protection and Homeland Security requires not only better access to information, but more effective use of the information that is already available to planners. Enormous quantities of information are generated by various forms of human, signals, electronic, and image intelligence gathering mechanisms, but human and group cognitive limitations severely constrain whether and how this information is comprehended.

Anti-terrorism planning involves identifying and prioritizing assets to be protected, determining the relative utility of different mitigations that can be applied to protect prioritized assets, and then selecting the portfolio of mitigation projects that provides the overall greatest net benefit given resource constraints. Such planning is rarely performed in a vacuum, it typically involves coordination among a number of key roles. Anti-terrorism planning is new to the job description for many of those currently working in this area. This makes the usability of software tools designed to support anti-terrorism planning activities a key requirement for these systems.

The purpose of the research proposed here is to better understand the usability of anti-terrorism/force protection planning systems and to contribute to their usability through the addition of intelligent agents to augment human cognition.

More intelligent opponents

The creation of an opponent that can learn through reflection using a domain ontology will provide more intelligent opponents. These opponents will be useful in training. They will

also be useful in analyses. An intelligent opponent that learns can test systems, like in the Rampart ATRP system (Haynes et al., 2005), to a higher level and more systematically.

The Rampart-Attacker opponent may also be useful as a tool to explore how cognitive models can act as repositories of knowledge in domains such as terrorist psychology and tactics. One advantage of cognitive modeling architectures such as Soar is that models can be easily evolved in response to new research findings and psychological study results. In other words, as our research progresses we expect to make the Rampart-Attacker model increasingly intelligent and representative of the state-of-knowledge in terrorist psychology. The enhanced explanation capability of the Herbal development environment makes this knowledge more transparent to developers and users of agents created within the environment thereby increasing the utility of these agents as learning devices.

An environment for testing modeling and behavior representations

One reason it is expensive to develop models, architectures, and algorithms is because it is expensive to tie a model to task so that the model can 'do its stuff'. The cost of tying models to environments has been noted a few times, but it is an under-stated problem, although it has been noted in a few places (Pew & Mavor, 1998; Ritter, Baxter, Jones, & Young, 2000; Ritter et al., 2003). Many projects seem to be spending 1/4 of their time or more tying demonstration models to large environments when a simple test environment would help answer many development questions. dTank offers a useful environment for testing the effects of errors and their implementation in an architecture, the effects of moderators on performance, and so on. It appears to offer approximately a 1,000x speed up for this task.

Based on an extensively used architecture

Soar is an important cognitive architecture. It has been widely used in synthetic environments (e.g., Jones et al., 1999), and is often used for comparison with other architectures (e.g., Gluck & Pew, in press; Pew & Mavor, 1998; Ritter, Sun, Schenck, Gross, & Stevenson, submitted; Shakir, 2002). It is thus doubly useful to make Soar models easier to understand and to create. Soar models have shown that they are useful in synthetic environments.

Accelerators of these payoffs

There are several accelerators of these payoffs. Technology transfer will be hastened by our close ties with three other groups: (a) We have been sharing this work with Colin Sheppard, Dstl (formerly, Defense Evaluation and Research Agency, UK) as the UK has been pursuing a similar research program to make models easier to create. They have been exploring the use of the JACK agent architecture to create a more cognitive version called COJACK (Norling & Ritter, 2004). In addition of this connection as an outlet, some of the ideas of JACK, of explicit representation of plans, has given us ideas for developing Herbal.

(b) Ritter is part of the major cognitive modeling communities, Soar and ACT-R, and he and the ACS Lab maintain the Frequently Asked Questions List for both Soar and ACT-R. He will disseminate these results through presentations at the Soar Workshop, the ACT-R

workshop, at the International Cognitive Modeling Conference, and at the Cognitive Science Conference.

(c) Haynes has worked closely with both the United States Marine Corps, through the Marine Corps Research University, and the Defense Threat Reduction Agency (DTRA) on system approaches to managing the terrorist threat. If funded, the work proposed here will be informed by these efforts and will in turn inform ongoing work with these Department of Defense organizations.

In addition to the traditional outputs of technical reports, conference papers, and journal articles, we take advantage of the Internet to provide additional supporting documentation. For these tools, we have created web sites in order to encourage reuse of the theories and models. These web sites have included the Herbal system, its manuals, and dTank and its documentation and supporting materials. They are available as acs.ist.psu.edu/Herbal and acs.ist.psu.edu/dTank. We have created movies illustrating model behavior before, and they have been useful for explaining the model when the model is not available or is difficult to install. As well as supporting classes and local users, these websites have lead to at least one user picking up dTank and using it from a distance.

4. References

- Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036-1060.
- Avraamides, M., & Ritter, F. E. (2002). Using multidisciplinary expert evaluations to test and improve cognitive model interfaces. In *Proceedings of the 11th Computer Generated Forces Conference*, 553-562, 502-CGF-002. Orlando, FL: U. of Central Florida.
- Brooks, F. P. (1975). *The mythical man-month: Essays on software engineering*. Reading, MA: Addison-Wesley Pub. Co.
- Cohen, M. A. (2005). Teaching agent programming using custom environments and Jess. *AISB Quarterly*, 120(Spring), 4.
www.lhup.edu/mcohen/Publications/AISBQSpring2005.pdf.
- Cohen, M. A., & Ritter, F. E. (2004). Herbal tutorial (Tech. Report No. 2004-2): Applied Cognitive Science Lab, School of Information Sciences and Technology, Penn State.
acs.ist.psu.edu/acs-lab/reports/cohenR04.pdf.
- Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2005). Herbal: A high-level language and development environment for developing cognitive models in Soar. In *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*, 133-140. 105-BRIMS-043. Orlando, FL: U. of Central Florida.
- Councill, I. G., Haynes, S. R., & Ritter, F. E. (2003). Explaining Soar: Analysis of existing tools and user information requirements. In *Proceedings of the Fifth International Conference on Cognitive Modeling*, 63-68. Bamberg, Germany: Universitäts-Verlag Bamberg.
- Daughtry, J. M., & Ritter, F. E. (2005). Supporting the active AI and cognitive modeling developer: CaDaDis version 2.1. *AI and the Simulation of Behaviour Quarterly*, 119, 4.
- Friedman-Hill, E. (2003). *JESS in action*. Greenwich, CT: Manning Publications.
- Gluck, K. A., & Pew, R. W. (Eds.). (in press). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Lawrence Erlbaum.
- Haynes, S. R. (2001). *Explanations in information systems: A design rationale approach*. Unpublished Ph.D. dissertation, Departments of Information Systems and Social Psychology, London School of Economics.
- Haynes, S. R., Councill, I. G., & Ritter, F. E. (2004). Responsibility-driven explanation engineering for cognitive models. In *AAAI Workshop on intelligent agent architectures: Combining the strengths of software engineering and cognitive systems*, 46-52. Menlo Park, CA: AAAI Press.
- Haynes, S. R., Kannampallil, T. G., Larson, L. L., & Garg, N. (2005). Optimizing anti-terrorism resource allocation. *Journal of the American Society for Information Science and Technology*, 56(3), 299-309.
- Haynes, S. R., Ritter, F. E., Councill, I. G., & Cohen, M. A. (submitted January 2005). Explaining intelligent agents. *Journal of Autonomous Agents and Multi-Agent Systems*.
- Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, 20(1), 27-41.
- Kukreja, U. (2004). *Towards model-based evaluations of human-robot interfaces*. Unpublished MS thesis, Department of Computer Science and Engineering, The Pennsylvania State University.

- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1-64.
- Morgan, G. P., Cohen, M. A., Haynes, S. R., & Ritter, F. E. (2005). Increasing efficiency of the development of user models. In *Proceedings of the IEEE System Information and Engineering Design Symposium*.
- Morgan, G. P., Ritter, F. E., Stevenson, W. E., Schenck, I. N., & Cohen, M. A. (2005). dTank: An environment for architectural comparisons of competitive agents. In *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*, 133-140. 105-BRIMS-044. Orlando, FL: U. of Central Florida.
- Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Norling, E., & Ritter, F. E. (2004). A parameter set to support psychologically plausible variability in agent-based human modelling. In *The Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS04)*, 758-765. New York, NY: ACM.
- Pew, R. W., & Mavor, A. S. (Eds.). (1998). *Modeling human and organizational behavior: Application to military simulations*. Washington, DC: National Academy Press. books.nap.edu/catalog/6173.html.
- Ritter, F. E. (2004). Choosing and getting started with a cognitive architecture to test and use human-machine interfaces. *MMI-Interaktiv-Journal*, 7, 17-37. useworld.net/mmij/musimms.
- Ritter, F. E., Baxter, G. D., Jones, G., & Young, R. M. (2000). Supporting cognitive models as users. *ACM Transactions on Computer-Human Interaction*, 7(2), 141-173.
- Ritter, F. E., Morgan, G. P., Stevenson, W. E., & Cohen, M. A. (2005). A tutorial on Herbal: A high-level language and development environment based on Protégé for developing cognitive models in Soar. In *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*. Orlando, FL: U. of Central Florida.
- Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R., Gobet, F., & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review*. Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center (HSIAC), formerly known as the Crew System Ergonomics Information Analysis Center (CSERIAC). iac.dtic.mil/hsiac/S-docs/SOAR-Jun03.pdf.
- Ritter, F. E., Sun, S., Schenck, I. N., Gross, J. B., & Stevenson, W. E. (in preparation). Extending the concept of control in cognitive architectures using a competitive environment. In W. Gray (Ed.), *Integrated models of cognitive systems*. New York, NY: Oxford University Press.
- Ritter, F. E., Sun, S., Schenck, I. N., Gross, J. B., & Stevenson, W. E. (submitted). Exploring the implications of including errors in cognitive architectures with a competitive environment. In W. Gray (Ed.), *Integrated models of cognitive systems*. New York, NY: Oxford University Press.
- Ritter, F. E., Van Rooy, D., St. Amant, R., & Simpson, K. (in press). Providing user models direct access to interfaces: An exploratory study of a simple interface with implications for HRI and HCI. *IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans*.

- Shakir, A. (2002). Assessment of models of human decision-making for air combat analysis. [Draft paper, to be put on the web with permission, at <http://acs.ist.psu.edu/papers/shakir02.pdf>].
- St. Amant, R., Freed, A. R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research*, 6(1), 71-88.
- St. Amant, R., Horton, T. E., & Ritter, F. E. (accepted 2005). Model-based evaluation of expert cell phone menu interaction. *Transactions on CHI*.
- St. Amant, R., & Riedl, M. O. (2001). A perception/action substrate for cognitive modeling in HCI. *International Journal of Human-Computer Studies*, 55(1), 15-39.
- St. Amant, R., & Ritter, F. E. (2004). Automated GOMS to ACT-R model generation. In *Proceedings of the International Conference on Cognitive Modeling*, 26-31. Mahwah, NJ: Lawrence Erlbaum.
- Sun, S., Councill, I. G., Fan, X., Ritter, F. E., & Yen, J. (2004). Comparing teamwork modeling in an empirical approach. In *Proceedings of the Sixth International Conference on Cognitive Modeling*, 388-389. Mahwah, NJ: Lawrence Erlbaum.
- Yost, G. R. (1993). Acquiring knowledge in Soar. *IEEE Expert*, 8(3), 26-34.

Appendix : Outputs

Papers submitted or in preparation

- Haynes, S. R., Ritter, F. E., Councill, I. G., & Cohen, M. A. (submitted January 2005, currently being revised). Explaining intelligent agents.
- Haynes, S. R., Cohen, M. A. & Ritter (submitted May 2005, currently being revised). Designing patterns for explaining intelligent agents.

Journal articles published or in press

- Ritter, F. E. (2004). Choosing and getting started with a cognitive architecture to test and use human-machine interfaces. *MMI-Interaktiv-Journal's* special issue on Modeling and Simulation in HMS. 7. 17-37. [in English, abstract in German]
useworld.net/mmij/musimms
- Ritter, F. E., Van Rooy, D., St. Amant, R., & Simpson, K. (in press). Using a simulated user to explore human-robot interfaces. *IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans*.
- St. Amant, R., Freed, A. R., & Ritter, F. E. (2005). Specifying ACT-R models of user interaction with a GOMS language. *Cognitive Systems Research*, 6(1), 71-88.
- St. Amant, R., Horton, T. E., & Ritter, F. E. (accepted 2005). Model-based evaluation of expert cell phone menu interaction. *Transactions on CHI*.

Refereed conference publications

- Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2005). Herbal: A high-level language and development environment for developing cognitive models in Soar. In *BRIMS'05*. 177-182. 05-BRIMS-044. Orlando, FL: U. of Central Florida.
- Councill, I. G., Haynes, S. R., & Ritter, F. E. (2003). Explaining Soar: Analysis of existing tools and user information requirements. In *Proceedings of the Fifth International Conference on Cognitive Modeling*. 63-68. Bamberg, Germany: Universitats-Verlag Bamberg.
- Haynes, S. R., Councill, I. G., & Ritter, F. E. (2004). Responsibility-driven explanation engineering for cognitive models. In *AAAI Workshop on Intelligent Agent Architectures: Combining the Strengths of Software Engineering and Cognitive Systems*.
- Morgan, G. P., Haynes, S., Ritter, F. E., & Cohen, M. (2005). Increasing efficiency of the development of user models. In *Proceedings of the 2005 Systems and Information Engineering Design Symposium*. Ellen J. Bass, (ed). IEEE and Department of Systems and Information Engineering, University of Virginia: Charlottesville, VA.
- Morgan, G. P., Ritter, F. E., Stevenson, W. E., Schenck, I. N., & Cohen, M. A. (2005). dTank: An environment for architectural comparisons of competitive agents. In *BRIMS'05*. 133-140. 05-BRIMS-043. Orlando, FL: U. of Central Florida.
- Ritter, F. E., Cohen, M. A., Morgan, G. P., & Stevenson, W. (2005). Herbal: A high-level language and development environment for developing cognitive models in Soar. 05-BRIMS-041. A tutorial presented at and in the proceedings of *BRIMS '05*. xxix-xxxi. 05-BRIMS-041.

- Shah, K., Rajyaguru, S., St. Amant, R., & Ritter, F. E. (2003). Connecting a cognitive model to dynamic gaming environments: Architectural and image processing issues. In *Proceedings of the Fifth International Conference on Cognitive Modeling*. 189-194. Bamberg, Germany: Universitäts-Verlag Bamberg.
- St. Amant, R., Riedel, M. O., Ritter, F. E., & Reifers, A. (2005). Image processing in cognitive models with SegMan. In *Proceedings of HCI International '05*. (Invited.) Volume 4 - Theories Models and Processes in HCI. Paper # 1869.
- St. Amant, R., Horton, T. E., & Ritter, F. E. (2004). Model-based evaluation of cell phone menu interaction. In *Proceedings of the CHI'04 Conference on Human Factors in Computer Systems*. 343-350. New York, NY: ACM.
- Sun, S., Councill, I., Fan, X., Ritter, F. E., & Yen, J. (2004). Comparing teamwork modeling in an empirical approach. In *Proceedings of the Sixth International Conference on Cognitive Modeling*. 388-389. Mahwah, NJ: Lawrence Erlbaum.

Other outputs completed

- Cohen, M. A., Ritter, F. E., Haynes, S. (2004). An introduction to Herbal. In *Proceedings of the Soar XXIV Workshop*. 75-77. The Soar Group, University of Michigan.
- Cohen, M. A. & Ritter, F. E. (2004). Herbal Tutorial. (Tech. Report No. 2004-2). Applied Cognitive Science Lab, School of Information Sciences and Technology, Penn State. acs.ist.psu.edu/reports/cohenR04.pdf.
- Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2004). An introduction to Herbal. In the *Proceedings of the XXIV Soar Workshop*, University of Michigan.
- Cohen, M. A. (2003). *Developing Intelligent Systems: Soar, Herbal, and dTank*. Presented at the Lock Haven University Association of Computing Machinery Symposium, Lock Haven University.
- Cohen, M. A., & Ritter, F. E. (2004). *The Herbal Tutorial (Tech. Report No. ACS 2004-2)*. Applied Cognitive Science Lab, School of Information Sciences and Technology, the Pennsylvania State University.
- Councill, I. G., Morgan, G. P., & Ritter, F. E. (2004). dTank: A competitive environment for distributed agents (Tech. Report No. 2004-1). Applied Cognitive Science Lab, School of Information Sciences and Technology, Penn State. acs.ist.psu.edu/acs-lab/reports/councillMR04.pdf.
- Morgan, G. P., Ritter, F. E., Schenck, I., & Cohen, M. A. (2005). dTank: A neutral environment for teaching and testing agent architectures. In *Proceedings of the Soar XXV Workshop*. 66-68. The Soar Group, University of Michigan.
- Ritter, F. E., Cohen, M. A., Haynes, S., Morgan, G. P. (2005). Herbal and the Herbal Viewer. In *Proceedings of the Soar XXV Workshop*. 75-77. The Soar Group, University of Michigan.

Presentations

Presentations without associated papers where this work was reported and support acknowledged.

- Cohen, M. A. (2003). *Developing Intelligent Systems: Soar, Herbal, and dTank*. Presented at the Lock Haven University Association of Computing Machinery Symposium, Lock Haven University.
- Ritter, F. E. (2006). Some new components for an automatic user for testing human-computer interfaces. Computer Science Department Colloquium Series, Tufts University.
- Ritter, F. E. (2005). Steps towards an automatic user for testing human-computer interfaces. Psychology, TU Chemnitz, departmental seminar.
- Ritter, F. E., Sun, S., Schenck, I., Cohen, M., Gross, J. B., & Stevenson, W. E. (March 2005). Control in Soar (and related digressions). Presented at the *AFOSR Integrated Models of Cognitive Systems Workshop*.
- Ritter, F. E., St. Amant, R., Reifers, A., Morgan, G., Schenck, I. N., & Cohen, M. (2005). Steps towards a CAD/CAM system for Human-Computer Interfaces. School of Computer Science and Information Systems, Pace University.
- Kukreja, U. & Ritter, F. E. (January, 2005). Towards improving human-robot interaction. Psychology Cognitive Brown Bag series, Penn State.
- Ritter, F. E. (September, 2004). Steps towards a CAD/CAM system for Human-Computer Interfaces. Department of Computer Science, Tufts University.
- Also presented at BBN Labs.
- Ritter, F. E. (2004). Projects modeling architectural changes. Human Research and Engineering Directorate, Army Research Lab, Aberdeen Proving Ground.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 22-01-2006		2. REPORT TYPE Final		3. DATES COVERED (From - To) November 2001 to 9/30/05	
4. TITLE AND SUBTITLE An architectural overlay: Modifying an architecture to help cognitive models understand and explain themselves.				5a. CONTRACT NUMBER N00014-02-1-0021	
				5b. GRANT NUMBER N00014-02-1-0021	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
6. AUTHOR(S) Ritter, Frank E. Haynes, Steven R.				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) College of Information Sciences and Technology The Pennsylvania State University				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research Attn: Harold Hawkins ONR Code: 342 800 North Quincy Street Arlington, VA 22217-5660				10. SPONSOR/MONITOR'S ACRONYM(S) ONR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited					
DISTRIBUTION STATEMENT A Approved for Public Release Distribution Unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Two challenges confront efforts to develop effective intelligent systems to work in complex domains: making them easier, faster and less expensive to develop, and making them easier to comprehend, use, and trust (Haynes, Councill, & Ritter, 2004). This applies to cognitive models as well (Ritter, 2004). We have created a high-level behavior representation language and integrated development environment (IDE) called Herbal (Cohen, Ritter, & Haynes, 2005) that answers these challenges. Herbal provides explanations, it works, has been used by 60 people as the basis of a course on modeling and as a tutorial at the BRIMS conference, and the data so far suggest a 3 to 7 x performance increase compared to developing Soar code directly. The next steps are to evaluate and extend Herbal, and to apply Herbal to a real-world problem.					
15. SUBJECT TERMS Cognitive models, affordable cognitive models, high level behavior representation, synthetic environments, explanation.					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU		21
Unclassified	Unclassified	Unclassified			
19a. NAME OF RESPONSIBLE PERSON Frank E. Ritter					19b. TELEPHONE NUMBER (Include area code) +1 (814) 865-4453