2

AL-TP-1992-0054

ARMSTRONG LABORATORY

A COGNITIVE ARCHITECTURE FOR HUMAN PERFORMANCE PROCESS MODEL RESEARCH

Michael J. Young

HUMAN RESOURCES DIRECTORATE
LOGISTICS RESEARCH DIVISION

NOVEMBER 1992

INTERIM TECHNICAL PAPER FOR PERIOD DECEMBER 1991 - JULY 1992

AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

This paper has been reviewed and is approved for publication.

MICHAEL J. YOUNG
Contract Monitor

BERTRAM W. CREAM, Chief
Logistics Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1992

3. REPORT TYPE AND DATES COVERED
Interim - December 1991 - July 1992

4. TITLE AND SUBTITLE
A Cognitive Architecture for Human Performance Process Model Research

5. FUNDING NUMBERS
C - F33615-91-D-0009
PE - 62205F
PR - 1710
TA - 00
WU - 60

6. AUTHOR(S)
Michael J. Young

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
Wright-Patterson AFB, OH 45433-6573

8. PERFORMING ORGANIZATION REPORT NUMBER
AL-TP-1992-0054

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Armstrong Laboratory
Human Resources Directorate
Logistics Research Division
Wright-Patterson AFB, OH 45433-6573

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AL-TP-1992-0054

11. SUPPLEMENTARY NOTES
Armstrong Laboratory Technical Monitor: Michael J. Young, AL/HRGA, (513) 255-8229

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Technical Paper proposes a new cognitive architecture for human performance process (HPP) model research. HPP models are engineering models of human performance. They represent the human information-processing system as a series of subsystems consisting of input and output, memory and processing subsystems. HPP models are used in engineering studies of new systems where the goal of the study is to predict human performance. This report proposes a new cognitive architecture for HPP model research. This new architecture is based upon the holon theory of mind previously proposed by Arthur Koestler. The holon theory of mind proposes that mind is both highly modular and hierarchically organized. The report provides an overview of HPP model research, describes the holon theory of mind, and discusses how this theory can form the basis of a new cognitive architecture for HPP model research.

14. SUBJECT TERMS
cognitive architectures
cognitive psychology
computational theory
human performance model
human performance process models
implementation architectures

15. NUMBER OF PAGES
41

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

This report proposes a new cognitive architecture for human performance process (HPP) model research. HPP models are engineering models of human performance. The Armstrong Laboratory, Logistics Research Division (AL/HRGA), is developing HPP models to support operability analysis studies of new and retrofit command and control systems. Operability analysis investigates allocation of function issues between man and machine, crew sizing and composition, and crew skill-level requirements to provide acquisition decision-makers accurate system performance and manpower life-cycle cost data for alternate system configurations. HPP models support operability analysis by serving as team players in system-modeling simulations.

This work supports the AL/HRGA advanced development Operability Assessment System for Integrated Simultaneous Engineering (OASIS; WU# 2940-03-02) and was accomplished by the author under Research, Development, Training and Evaluation Support work unit (1710-00-60). The author wishes to thank Marilyn Jager Adams and Kevin Corker for reading earlier drafts of the paper and making useful sugestions for improving the substance and readability of the text.

Introduction

This paper proposes a new cognitive architecture (Pylyshyn, 1991) to support the development of advanced process models of human performance. Human performance process (HPP) models attempt to emulate human behavior through simulation of specific human information processing attributes and processes (Card, Moran, & Newell 1983, 1986; Corker, 1991). HPP models represent the human information processing system as an engineering system consisting of input/output devices, processors, and memory storage subsystems. Information flow and transformation through the model are described in terms of key parameters denoting processing limitations of the subsystems. The goal of HPP model research is to develop models capable of predicting human performance.

Designing an HPP model requires architectural specification on two levels of abstraction: computational theory and implementation. The computational-theory (Marr, 1982) specification defines a psychological model capable of instantiating the behaviors of interest. It delineates information flow and transformation as the information proceeds through the information-processing system. The computational theory is the level at which the input/output devices, processors, and memory storage subsystems are specified in terms of their psychological functionality.

The implementation-level (Anderson, 1990) specification casts the psychological model in computational terms. Implementation-level specification delineates the choice of data structures and data-manipulating techniques (inferencing techniques, algorithms, etc.) to be employed to realize the computational-theory specification. The implementation level is not psychologically or physiologically real. It is an approximation (Anderson, 1990) employed to simulate a biological "computer" within a computer architecture.

For any given computational-theory specification there exist many equivalent implementation-level specifications. Implementation-level specifications may vary as to the data structures, inferencing techniques, and programming languages chosen to implement the computational-theory specification. However, each independent implementation specification still realizes the full functionality of the computational-theory specification.

Choice of an actual implementation-level specification is based on additional requirements that are best realized at the implementation level. For example, the Armstrong Laboratory (AL) employs HPP models as "team players" in real-time, human-in-the-loop simulations and to conduct cognitive science research by comparing the performance of alternative HPP models to human-in-the-loop operators. These applications require that the HPP model implementation have real-time simulation capability, and be designed to be highly modular.

In this paper, we introduce terminology which is similar in meaning to other well-known terms. In addition, we employ other commonly used terminology in somewhat new ways. Our choice of terminology is based on the goal of clearly distinguishing between model descriptions at the computational-theory level (i.e., the psychological model being proposed) and implementation level (i.e., how the psychological model will be simulated within a computer). Whenever possible, we use existing terminology with an added explanation to clarify our use of the term. However, there are instances for which we believe new terminology is appropriate, both to give credit to the intellectual antecedents of the ideas we are proposing and to clearly distinguish between computational-theory and implementation-level descriptions of the model being proposed.

In this paper, we first introduce the class of models we are interested in developing and discuss their applications. Next, we discuss the limitations of existing models and propose a new cognitive architecture for HPP model research. Finally, we conclude by discussing future research issues.

Human Performance Process Models

A state-of-the-art HPP model was developed by AL through a contract with BBN Systems and Technologies (Corker et al., 1991). This model consists of four subcomponent models: visual, auditory, cognitive, and psychomotor. The visual subcomponent models two types of visual processing: active gaze and monitoring.

Active gaze represents focused and directed movement to a target point; monitoring represents a scanning process. The parameters modeled for both active gaze and monitoring are field of view, velocity of motion, saccade, and fixation pause. The auditory subcomponent models the communication protocol employed by human operators, and the bandwidth and memory limits of human auditory processing capabilities. The psychomotor subcomponent models Fitts' formulation relating movement time, distance, and accuracy to provide a probability of error. The cognitive subcomponent model depicts cognitive activity via a procedural representation consisting of "If some condition exists Then execute some action" statements, and an inferencing engine which controls the application of the If/Then statements (which are normally called productions).

In addition to the subcomponent models, each operator model has an individually defined, updatable world representation which is a description of the world as the operator knows it. It contains rules for decisions, an awareness of external events as seen through the operator's perceptual processes (i.e., audition and vision subcomponent models), and a declarative description of the world as the operator knows it. The declarative, or factual, information is represented as frames. Declarative information includes "knowledge" concerning aircraft (types of aircraft and their capabilities), the operator's equipment (individual components of the equipment and how to operate it), and rules of engagement (knowledge about the operator's expected behavior).

The HPP model works in the following way. Information enters the world representation through the perceptual modalities. The cognitive subcomponent model continually tries to match condition clauses with the data in the world representation. If a match is triggered, the execution clause generates activities. These activities are then executed through the appropriate subcomponent model. The execution of activities can also change data in the world representation, often resulting in the generation of additional activities.

The cognitive subcomponent model is an elaborate attempt to incorporate knowledge-based modeling techniques into an HPP model. Knowledge-based models depict the problem-solving processes of experts. The experts' knowledge is represented in symbol structures, along with rules for manipulating the knowledge. Knowledge is often stored as heuristics--"rules-of-thumb" that individuals employ in making decisions. The knowledge-based model employed in the HPP model is an expert system (Hayes-Roth et al., 1983). It is unique in that the control strategy guiding the inferencing engine is based, in part, on a psychological model of how human memory operates.

Potential Human Performance Process Model Applications

HPP models are being developed to permit psychological principles and data to influence the design of new systems. HPP models have several potential applications. One application is in system modeling test-beds where they can be used in engineering studies to evaluate design options.

System modeling is the explicit modeling of the functional operation of the system, the environment in which the system operates, and the human operators of the system. System modeling allows a designer/analyst to play out specific scenarios in simulation, capture operational measures, and thus track operability impacts resulting from specific system design decisions.

A system-modeling test-bed typically consists of scenario generation tools, system prototyping tools, human-in-the-loop simulation tools, and data collection and analysis tools. An analyst employs the test-bed to create a soft prototype of the system under consideration. (A soft prototype is a physical mockup or simulation mockup of the system in which the functional operation of the system is simulated in software.) The analyst defines representative scenarios under which the system will be employed. The system design is then analyzed using human-in-the-loop simulation technology.

HPP models integrated into the system-modeling test-bed serve as team members in a multicrew environment, operating individual crew stations. The HPP models interact with the human in the loop through voice generation and recognition systems, as well as the system interface. In this capacity, HPP models reduce experimental variability by providing behavioral replication (on the part of the team members) across simulation trials and decrease study costs by reducing the number of operational personnel that need to participate in the design studies.

A second application is employing HPP models in cognitive science research to test psychological theories and computational strategies. This approach does not eliminate the black-box problem where alternative models, each of which replicates the human behavior under study, cannot be shown to be a veridical representation of the way actual humans process information. However, this approach does allow one to compare alternative micromodels within the context of a complete simulation of human information processing. The comparison of micromodels can be in terms of the psychological theory or computer implementation. This application is made possible by the capability to compare the models to human performance when both are integrated into a system-modeling test-bed.

HPP models are typically coded in a modular fashion to readily support substitution of one module representing one theory or computational technique for another, thus allowing side-by-side comparison of theories and techniques. In addition, the ability to substitute an actual human-in-the-loop operator for a modeled operator during identical simulation trials allows differing aggregate HPP models to be compared with human performance. This additional ability to perform side-by-side comparisons of operators and models creates a unique research vehicle for human performance research. Combined, the ability to perform side-by-side comparisons of models to models and models to actual operators is a new, powerful research technique for cognitive science.

AL is in the process of developing a new system-modeling test-bed for application to real-world problems and cognitive science research. This system-modeling test-bed goes by the lengthy name of Operability Assessment System for Integrated Simultaneous Engineering (OASIS). The objective of OASIS is to develop a set of user-friendly tools to support designers in building soft prototypes of competing multicrew workstation designs. Tools under development include tools to define user requirements, create soft prototypes of consoles, and develop appropriate scenarios to test operability issues.

One aspect of OASIS will be a documented interface standard for HPP models. This interface standard should allow diverse model developers--including us--to link their models to the OASIS system for testing and evaluation, and to participate in operability evaluation studies.

Limitations of Existing Human Performance Process Models

Compared to humans, extant models have very limited capabilities. Several facets of HPP models need continued research and development. These facets are divided into behavioral and structural issues.

Behavioral limitations arise from the inability of current models to predict human performance. Ideally, HPP models should be able to depict human performance limitations across a wide variety of tasks. Development and validation of such models would allow HPP models to be incorporated into computer-aided design workstations where they could animate (provide the cortex for) anthropometric man models.

Behaviors to be modeled include: performance limitations due to high task demands, skill-level differences between personnel, concurrent task performance, complex decision-making, strategic behavior, context and task prioritization, and procedural flexibility. The ability to model these behaviors will allow designers to investigate allocation of function concerns (between man and machine), crew sizing and composition issues, and man-machine interface issues as component factors in a "fly-off" between alternative system designs.

Structural limitations stem from a lack of sophistication in depicting the information-processing system in contemporary HPP models. Lack of structural sophistication is the primary constraint preventing the development of models embodying complex behaviors. There are three dimensions in which additional structural sophistication is needed: top-down and bottom-up data processing, sequential and parallel information processing, and knowledge representation strategies.

Human cognition requires both bottom-up and top-down data-processing strategies. Bottom-up processing involves a series of processing steps for which the output of one step serves as the input for the next. During bottom-up processing, information is transformed from small perceptual units to larger aggregate chunks. Also, in bottom-up processing the outcome of a lower processing step never affects the next higher step in the processing chain. Bottom-up processing is also called data-driven processing because the data "drives" the processing of information.

In contrast, top-down processing uses situational context and general world knowledge to guide the processing and interpretation of information. The perceiver employs conceptual structures (schemas) and world knowledge to filter incoming information. Contextual factors are used to make appropriate interpretations. Top-down processing is also called conceptually driven processing because conceptual structures help determine perception.

Humans employ both top-down and bottom-up processes; these interact in perceiving stimuli, focusing attention, and solving problems. Top-down processing, for example, focuses the bottom-up processing on stimuli categories expected in the incoming data. Combined top-down and bottom-up data processing has been shown, experimentally, to play a key role in perception, both visual and auditory.

Extant HPP models ordinarily employ only one data-processing strategy, usually the bottom-up (data-driven) approach employed in production (knowledge-based) systems. Models based on a production-system approach suffer limitations. Production systems are data-driven and, hence, are not well-suited for modeling intentional behavior. An analyst employing an HPP model built around a production system usually must manipulate the scenario in order to elicit the desired behavior in the operator model. It would be preferable to merely program the desired intentions into the cognitive model.

Top-down processes normally direct bottom-up processing. This guidance effectively limits the amount of search required by the bottom-up processes in decoding stimuli, thus greatly enhancing pattern recognition capability. Contemporary production-system-based models do not have an efficient means of incorporating top-down knowledge. Hence, there is no good mechanism to speed up the data-driven process by focusing the pattern recognition search. This inability to incorporate top-down processing limits the size of the production system that can be built and incorporated into a real-time test-bed.

New HPP models that incorporate both top-down and bottom-up processing need to be developed. The development of such models will advance the modeling of complex decision-making and concurrent tasking. In both processes, the operator's intentions and expectations strongly influence data processing and behavioral outcomes.

Another limiting structural factor of extant HPP models is that they only employ one type of information-processing, the form normally associated with conscious-reasoning processes. Humans, however, appear to have two distinct information-processing systems (Rasmussen, 1986). One system is a slow, sequential, limited-capacity processor operating within human awareness. The main functions of this processor include handling unique situations, engaging in symbolic reasoning, and performing rational deductions. The second system is a distributed, parallel-processing, high-capacity processor operating outside human awareness. This processor decodes information captured by the senses and executes psychomotor activities.

Existing models normally only model the sequential limited-capacity processing system. The rationale for this approach is that, since the distributed parallel system is massively parallel, it most likely does not play a role in defining human performance limitations. This justification appears shortsighted for (at least) two reasons.

First, the parallel system plays a vital role in information acquisition. For example, the human eye makes several movements each second, pausing momentarily to fixate on separate points; such movements are called saccades. These saccades keep our internal representation of the visual environment from fading and detect changes in the environment. Interwoven with these saccades are automatic eye movements which pick up data related to the operator's current task (e.g., the automatic scanning of displays). These eye movements are controlled (directed) by a theoretical construct, called the internal model, that apparently is also part of the parallel system. Any HPP model that incorporates visual scanning must account for these movements in order to accurately predict human performance. Finally, interwoven with this automatic monitoring of the environment are deliberate, conscious intentions to look at things. HPP models need to accurately depict both information-processing systems to model visual monitoring and active gazing.

Second, failure to model the parallel system appears to engender the general frame problem: how to efficiently represent a complex, changing world (Janlert, 1986). A key issue is how to update the internal representation of changes in the external world. Most artificial intelligence systems attempt to calculate what changes in the external world will occur based on the actions taken, rather than actually recording changes as they occur. Analogously, these systems are similar to a blind man attempting to predict changes in the external world based upon his actions. As pointed out by VanLehn and Ball (1991), an alternate approach is to have a system which records changes as they occur. The parallel information processing appears to fulfill this function.

Effective modeling of both the serial and parallel information-processing systems will further the development of models capable of representing skill-level differences. Novices and experts differ in their use of these two information-processing systems. For example, the scanning patterns of experts are clearly different from novices; experts see and assimilate different information.

A second example, independent of vision, is that experts and novices distribute task performance functions differently between the two systems. Experts perform some tasks without conscious awareness (i.e., they have automatized the task), whereas novices must engage conscious awareness to perform the same task. Modeling both systems appears crucial to modeling skill-level differences.

The final structural limitation is that humans appear to represent and employ knowledge in different formats, including differing levels of abstraction. In contrast, most contemporary HPP models employ only one, or occasionally two, knowledge representation schemes. A key issue here is the continuing controversy within psychology as to how the external world is represented in memory and whether one or multiple representation schemes are employed in the process. For discussion purposes, we will briefly consider three alternate schemes: propositionally based representations, analogical representations, and procedural representations.

Propositionally based schemes represent knowledge as a collection of symbols (Rumelhart & Norman, 1983) that are "formal" statements expressing relationships about the represented environment. Relationships are expressed in attribute-value pairs (e.g., attribute: color, value: blue). The set of relationships have been hypothesized to have many forms: networks (semantic nets), frames (frame systems, scripts), or logical formula. Propositionally based systems are often proposed to depict "meaning-of" relationships in long-term semantic memory.

Analogical representation maps characteristics of the external world directly into the mind. For example, spatial properties would be directly mapped onto the mental representation. The key aspect of an analogical representation is that, as the external object changes, the internal image undergoes the same change. Much of the research suggesting that humans employ analogical images comes from studies of mental rotation (Kosslyn, 1980). In these studies, subjects are asked to determine as quickly as possible if two images are the same or different. On some trials the images are the same, though one image has been rotated. In other trials the images are different. Experiments have shown that a subject's response time is directly related to the rotational offset for trials in which the images are the same. It appears that the subjects are mentally rotating the images to determine if they are the same or different. Analogical representations have been proposed for geographical and size information. In addition, analogical codes have been proposed for the internal model that drives visual scanning (Wickens, 1992).

Procedural schemes represent knowledge as procedures for employing knowledge. Procedural knowledge is the "how to" knowledge employed in skilled behavior. It is employed to guide actions like speech or motor movements. Within the human mind, procedural knowledge tends to be inaccessible to inspection (Rumelhart & Norman, 1983). The most common machine implementation of procedural knowledge is the production system (employed in the HPP model described above).

Contemporary models typically incorporate one or two of the above knowledge-representation schemes. It is common to use semantic nets (or a frame system) to model semantic memory and a production system to model procedural knowledge. Very few contemporary models include analogical knowledge.

A second knowledge-representation problem for existing HPP models is their inability to represent and employ knowledge at differing levels of abstraction. Rasmussen (1986) has investigated cognitive task formulation and execution, and found that individuals engaged in complex decision-making employ fluid task formulations. These formulations vary, from consideration of physical forms and physical functions through consideration of generalized function and abstract function to consideration of functional purpose. Individuals shift their problem conceptualization between levels while engaged in decision-making. Extant HPP models lack this sophistication; they have "one-dimensional" thought processes.

New HPP models that incorporate multiple forms of knowledge representation need to be developed. In particular, these models should support analogical representation to improve their capability to reason about spatial problems. In addition, HPP models need to support knowledge inferencing at multiple levels of abstraction. These two capacities will enable new models to depict the processes supporting complex decision-making.

Architectures and Frameworks

The previous section defined the behavioral and structural limitations that the next generation of HPP models must overcome. Our approach to overcoming these limitations is to apply a computational theory to the development of a new cognitive architecture that will support a broad-based research agenda on HPP models. The concept of architecture, however, has been used by different authors for different purposes. Before introducing our proposal for a new cognitive architecture, we must present our definition of the term. We begin by reviewing three authors' use of the concept.

Newell (1990) defines an architecture as a fixed structure that realizes a symbol system. An architecture is a system description at the register-transfer level (computer-processor level). Architectures are fixed structures that support system performance and provide boundaries which separate structure from content. This implies that architectural structure is not synonymous with hardware. Fixed mechanisms are often cast in software. Newell (1990) gives several examples of software mechanisms (e.g., dynamic memory allocation, data structures, etc.) that are architectural in nature.

For Newell, the key issue in defining an architecture is its "fixedness." The architecture is fixed, whereas the "content" of behavior changes. While Newell notes that architectures may change, the rate of architectural change is very slow when compared to the time scale of system behavior.

Anderson (1987) defines the term architecture to be the interface between the algorithm[1] and implementation levels. For Anderson, the architecture specifies the components through which the algorithms are implemented. These components appear to correspond to data structures and inferencing techniques which implement the functionality of the algorithm level at the implementation level.

Pylyshyn (1991) defines the term cognitive architecture to be the level of abstraction at which the states being processed receive a cognitive interpretation. He points out that an algorithm cannot be specified without making assumptions about the architecture that will process the algorithm. Pylyshyn believes the cognitive architecture may include large-scale organizational structure such as modularity of subsystems, as has been proposed by Chomsky (1980) and Fodor (1983).

Pylyshyn developed the concept of cognitive impenetrability to delineate operations that derive from the implementation (or functional, in his terms) architecture as opposed to the cognitive architecture. Basically, operations at the implementation level are not affected by the goals and beliefs of the entity, whereas

---

[1] Anderson's algorithm level defines an explicit psychological model (not an architecture) in terms of mental procedures and knowledge. See Anderson (1990) for a discussion of terminology.

those at the cognitive architecture can be. Pylyshyn believes that discovering the cognitive architecture of the mind is a central concern of cognitive science.

We believe that Newell's conception of architectures is too limited; it only exists at the register-transfer level. We also believe that architectures exist at implementation and computational-theory levels. Architectural specification at the implementation level corresponds to Anderson's conceptualization of defining the components (data structures and inferencing techniques) through which the computational-theory specification is implemented. Architectural specification at the computational-theory level corresponds to defining what Pylyshyn calls the cognitive architecture. It is a conceptualization of organization structure which defines the relationships between the semi-independent modules. (Henceforth we will use computational-theory architecture and cognitive architecture synonymously.)

Architectural specification at the computational-theory and implementation levels fixates the structure of an intelligent agent in the same manner as a register-transfer level architecture does: *by demarcating a class of models that can be built at a specific level of abstraction.* Similarly, these additional levels of architecture support system performance by providing a structure from which the behavioral content can emerge. Finally, knowledge and symbol-level architectures are also "fixed" in that if they change, they change very slowly.

A second term we need to introduce and define is *framework*. A framework is a set of architectures, each at a different level of abstraction, which supports instantiation of a multilevel model specification. It is similar to a test-bed. Typically, a framework consists of a computational theory, implementation, and computer architectures. As an example, a framework might consist of a collection of tools hosted on a serial computer (computer-level or register-transfer-level specification) which support building backward-chaining, knowledge-based systems (implementation-level specification). These tools could be employed to build HPP models (computational-theory-level specification).

In this example, the framework consists of three architectures: computational-theory, implementation-level, and computer architecture.

Computational-theory architecture limits the models that can be built to the class of data-driven, pattern-matching models employing one working memory. Implementation-level architecture limits the class of models that can be built to those that employ backward-chaining, knowledge-based systems. Computer-level architecture further limits this class to backwards-chaining, knowledge-based systems running on serial computers.

We now introduce a new computational-theory-level architecture (cognitive architecture) called the holon concept. A computational theory specifies the morphological architectural possibilities of an intelligent system. It specifies the tenets that define the possible relationships between knowledge sources. We first introduce the intellectual antecedents of the proposed cognitive architecture, then describe the architecture in general terms.

The Holon Concept

The holon concept was developed to explain the hierarchical behavior of organizations (Koestler, 1967; 1976). A holon can be any structural or functional subsystem in a biological, social, or cognitive hierarchy which manifests rule-governed behavior. Figure 1 depicts a biological holon hierarchy. In this example, the body is an organism made up of subsystems such as the circulatory system, digestive system, and nervous system. These system, in turn, are made up of organs, which are made up of tissues, which are made up of cells.

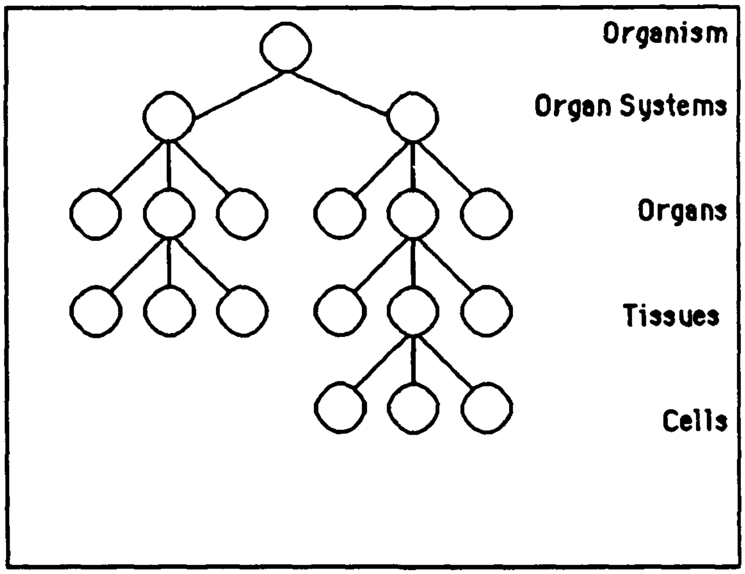

Figure 1
Biological Holon Hierarchy

Holons have three defining characteristics: a) they behave as quasi-autonomous wholes, b) the structure of their hierarchy affects information processing, and c) their behavior is governed by fixed rules and flexible strategies.

Each holon behaves as a quasi-autonomous whole. For example, each holon depicted in Figure 1 (organs, tissues, cells) is capable of functioning in vitro as a quasi-independent whole. Further, each holon displays its own timing basis for behavioral patterns of activity. This expression of autonomy is referred to by Koestler (1976) as the self-assertive tendency. The self-assertive tendency expresses itself in cognitive holons as instinctive rituals (the fixed-action patterns of animals) and habit patterns (gestures, handwriting, speech patterns). In social holons, the self-assertive tendency appears as social norms, moral imperatives, and value systems.

Holons are linked together in hierarchies called holarchies (Koestler, 1976). The number of levels in a holarchy is referred to as its depth; the number of holons on a given level is the span. Levels within a holarchy demarcate processing points at which the abstraction level of the information changes. Typically, information comes in at one level of abstraction and, through processing within the holon, is used to develop a new information product--usually at a different level of abstraction.

Branching lines in a holarchy represent communication channels. Holons on different levels but within the same branch structure can communicate only through channels. However, holons in different branch structures have the potential to communicate directly with other holons. In addition, holons can effect changes in the general environment. This ability to modify the environment is a form of indirect communication.

There are different classes of holarchies. The two most common are input holarchies and output holarchies. Input holarchies convert complex input patterns into coded signals (Koestler, 1976). Each level of an input holarchy acts as a filter or classifier, identifying the input signal as some higher-order construct. The abstracted knowledge of that construct is then passed up the holarchy. Figure 2 is an example of a visual detector input holarchy.

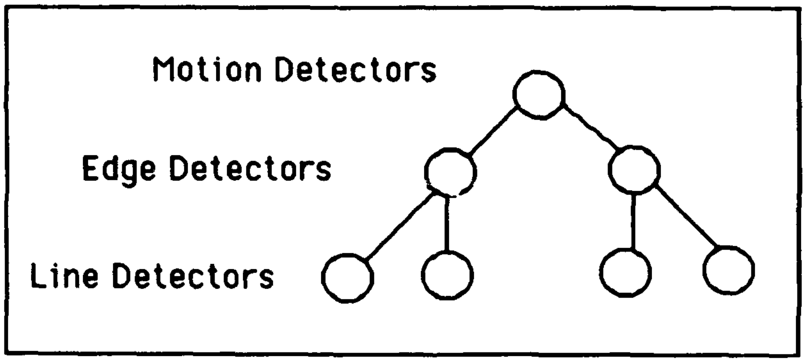

Figure 2
Visual Input Holarchy

Output holarchies work inversely: whereas an input holarchy abstracts information, an output holarchy elaborates (or embellishes) information. For example, a simple signal, called the trigger releaser, can cause the output holarchy to perform complex actions. Once the top-level holon is activated, each successive lower level of an output holarchy furthers defines and specifies the action to be taken.

Specific branches in a holarchy either analyze or further define (depending on whether it is an input or output branch) a specific dimension of the signal. For example, Koestler (1976) proposes that an auditory input holarchy for analyzing music would have separate branches for analyzing the melody and timbre of incoming stimuli.

Holons are governed by fixed rules (collectively called the holon canon) which are executed through flexible strategies. For a biological holon, the canon determines physical structure and activity patterns. For a cognitive holon, the canon represents rules of performance, such as the rules of enunciation, grammar, and syntax which govern speech production. For a social holon, the canon represents social norms.

The canon is applied, through flexible strategies, based upon the environment of the holon. This highlights a key point: each holon has its own world representation and control structure that determines the applicability of its canon. The following examples help illustrate this point.

Koestler (1976) uses the web-making activities of a spider to illustrate the difference between canon and strategies. A spider's inherited canon requires the radial threads of the web to always bisect the laterals at equal angles, thus forming a polygon. However, the spider will hang its web from three or more points of attachment depending upon its environment. The spider chooses its strategy based on the existing environment. Many instinctive animal behaviors, such as nest-building and beehive construction, share this dual characteristic of being based upon a fixed canon but executed through a flexible strategy which takes into account the external environment.

A second example is chess. The rules (canon) for playing chess are relatively simple. There are six different pieces which have their own rules of movement. There are also a few additional rules which govern things like taking turns (moving a piece), castling, and the end of the game. However, applying the rules is very difficult. The appropriate move in any situation depends on the positions of the other pieces. The player must have an internal representation of the game and a control strategy to apply the rules.

The requirement for an internal representation differentiates the holon from other similar concepts such as software macros. Macros are compiled bits of computer code used to perform frequently executed tasks like displaying windows on a computer screen, or responding to user input. Macros are handed (passed in computer terminology) parameters from the main program which tell the macro where the window should be drawn or where the user input activity occurred on the screen. In this example, the compiled code which permits activation of a specific function is analogous to the fixed canon, whereas the passing of parameters which provides flexibility in where the specific function is activated is analogous to the activation of a flexible strategy. Holons differ from macros in that holons maintain their own internal representations of the external environment.

Another important point is that the environment for individual holons within the same structure can be quite different. The "environment" for a holon may be external, internal, or both. Holons represent only that portion of the environment relevant to them. Koestler (1976) uses the example of driving a car to illustrate this point. The environment for a holon controlling the foot is tactile pressure of the foot on the accelerator pedal. The tactile feedback is used by the holon to keep speed steady. The environment for a holon controlling eye movement is much larger. The eyes need to take in the whole visual scene. Visual information is used to pilot the car, plan routes, and react to contingencies. An eye-movement holon must internally represent the whole external visual environment.

Holons at different levels of the holarchy represent the environment at differing levels of abstraction. As discussed above, abstraction is a function of how "high" the holons are in the holarchy. Each successive level of an input holarchy is continually abstracting incoming information. In the above example, the internal representation employed by the holon controlling the foot would probably be simple, perhaps a feedback loop. In contrast, the representation employed by the holon controlling the eye would most likely be highly abstract. The difference in the representations is due to the difference in the functional distance of the holons from external sensors. The foot-controlling holon is closer to the tactile sensor than the eye-controlling holon is to the visual sensors in terms of the number of intervening holons. Hence, the information in the visual holon is more abstracted, having been processed a greater number of times.

A Human Perfromance Process Model as a Holarchy

In this section, we describe how the holon concept can be used to define a computational-theory-level architecture for HPP model research. This description is a rough sketch, provided to help illustrate the holon and holarchy concepts. In the following sections, we discuss how an HPP model as a holarchy can overcome the limitations of existing models and address the morphological implications of modeling human-information processing as a holarchy.

Figure 3 depicts an HPP model as a holarchy. The apex of the holarchy is the central processing unit (CPU), Holon D. Information flows into the CPU from the visual and auditory recognition centers, Holons B and I, respectively. Intentions to move physically flow from the CPU to the motor control center, Holon E. Operation of the model is discussed in greater detail below.

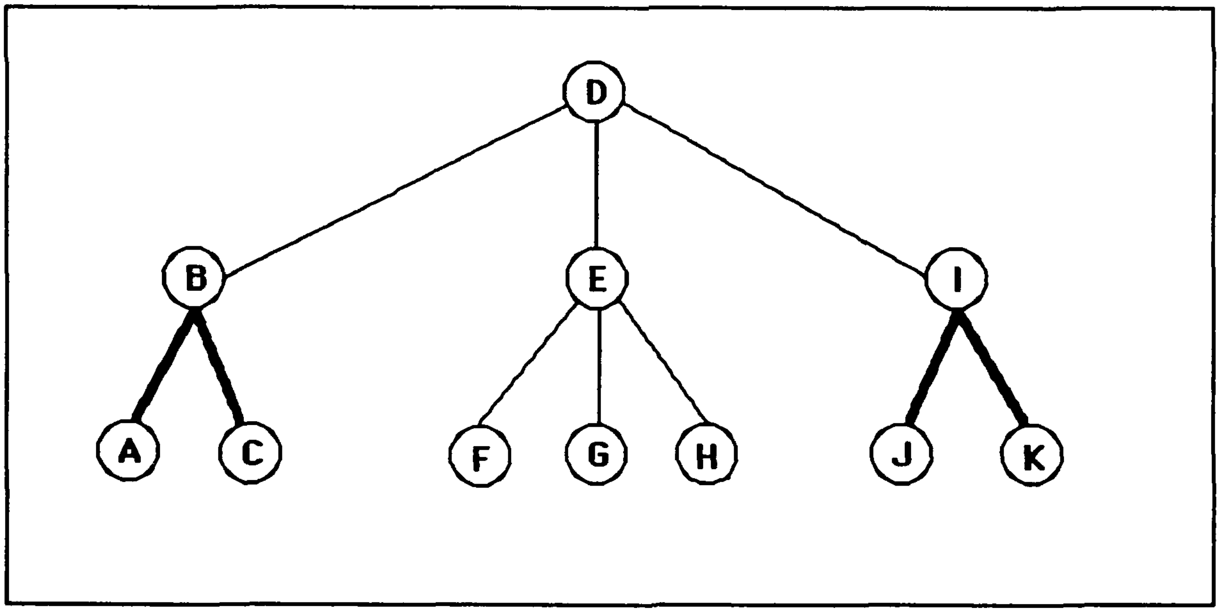

Figure 3
A Human Performance Process Model as a Holarchy

Binocular visual information flows into the system from Holons A and C, which represent scene-decoding centers. These holons each process sensory information from one eye. Information is integrated into a visual scene in Holon B. The channels from Holons A and C to Holon B are in bold to show that the level of model detail can be varied to accommodate different operability study requirements. Varying the level of detail is referred to as varying the resolution. For example, in most operability studies, binocular visual processing need not be modeled. Visual objects can be directly instantiated into the working memory of the visual-recognition holon. However, if a requirement arises to determine whether a signal can be seen, the depth of the holarchy can be extended, thus extending the resolution of the model, to incorporate additional visual analysis holons.

Intentions are formed in the CPU, Holon D. The intention to look at something (e.g., a specific display) originates in the CPU and is sent to the psychomotor control unit, Holon E. The psychomotor control unit internally represents the position of the body in three-dimensional space. When the psychomotor control unit receives the intention-to-look message, it calculates which parts of the body need to be moved to look at the display and sends out appropriate messages. For this example, assume that both the head and eyes need to move, Holons F and G, respectively. When the head- and eye-movement control centers receive their messages, they determine which muscle groups need to be activated, then send out messages to the muscle control centers (not shown). Once the eyes focus on the display, information enters the model through Holons A and C, the binocular scene-decoding-center holons.

So far we have discussed the model as a two-dimensional representation. However, the holarchy concept becomes very powerful when models are created in N-dimensional space. Figure 4 depicts the inside of Holon D, the CPU. The CPU holon is actually a holarchy in itself, which exists in a dimension tangential to the dimension in which the other holons exist.

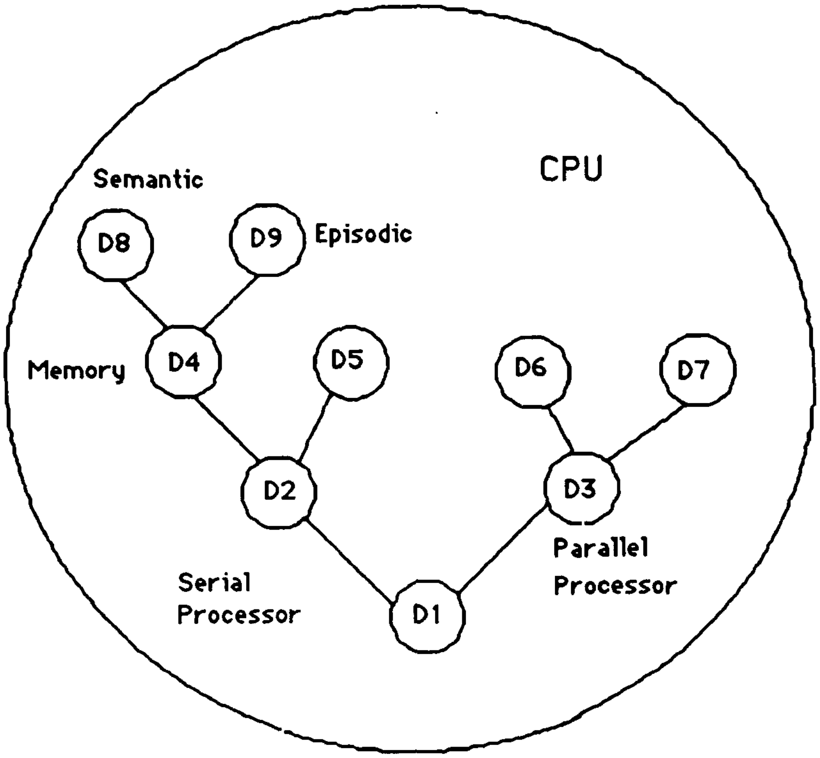

Figure 4
A Holarchy within a Holarchy

Use of N-dimensional space is another method for varying the complexity and resolution of the model. Individual holons can be expanded to whatever level of detail is required to model specific human performance attributes. In this example, the CPU becomes a holarchy consisting of serial- and parallel-processing subholarchies.

Holarchies and the Limitations of Existing Models

We believe the holon/holarchy formalism is an excellent way to depict an HPP model at the computational-theory level. The holon/holarchy formalism, in particular, seems sufficiently powerful to overcome the structural limitations of existing models.

The holarchy concept supports the development of models employing multiple data streams and processing centers. The hierarchical structure of the holarchy can easily be employed to create models that simultaneously implement top-down and bottom-up data-processing streams, and both serial and parallel information-processing centers, all within one model.

The modularity of the holon architecture supports the development of models that employ multiple forms of knowledge and knowledge representation. Because holons are encapsulated, semi-autonomous wholes, individual holons can represent and employ knowledge in different ways to include representing knowledge at differing levels of abstraction.

Further, the inherent versatility of the holon/holarchy formalism can better represent human behavior than other modeling approaches. For example, holons have three types of communication options. They can communicate via channels within a branch of the holarchy, directly (point-to-point) with other holons, or by effecting environmental change.

Communication options could, potentially, be employed to model skill-level differences. An expert would immediately and automatically react to a stimuli by sending a message from a recognition holon to an output holon, whereas a novice would be required to send the output of recognition higher up the holarchy for further processing before generating an output action. (This will, of course, require the development [or perhaps modification] of a learning theory that explains skill-level differences in terms of differences in the speed of pattern recognition in the external environment.)

Holons possess the capability to generate actions that potentially allow a wide range of behaviors to be manifested. Their generative capability results from the expression of the canon being situationally dependent on the environment. Since a holon can potentially encounter an infinite number of different environmental situations, virtually an infinite number of behaviors can be manifested. The capacity for generative actions should support expanding the behavioral repertoire of HPP models beyond procedural tasks into complex decision-making.

Finally, the modeling of performance limitations due to high task demands and concurrent tasking can potentially be accomplished through a careful combination of module (holon) topology specification (at the computational-theory level) and choice of parameters to be modeled by specific holons. Many of the existing explanations or theories of performance limitations due to high task demands do not, in our opinion, consider the topology of the information-processing system in sufficient detail.

For example, the multiple-resources theory (Wickens, 1980; 1990) proposes three structural dimensions of the human-information-processing system: processing stages, codes, and modalities. The processing-stages dimension defines two separate resources: perceptual-cognition (input processes) and response processes (output processes). The second dimension contrasts spatial and analog codes involved in information processing. The third dimension contrasts perceptual modalities (visual versus auditory input). The first two dimensions are associated with different resources in Wickens' model. The third dimension is not associated with resources; rather it affects resource utilization through interactions with the other two dimensions (Wickens, 1990). To the degree that tasks are similar across dimensions, they will use the same resources and, hence, interfere with one another.

Wickens' multiple-resource model has had only limited success in explicating performance limitations due to high task demands. We believe the predictive power of the multiple-resource theory could be enhanced by integrating the theory with a holon-based HPP model. Improved prediction could be achieved by lowering the level of granularity at which resources are utilized. Currently, resources are utilized or assigned to structural dimensions; the potential exists with a holon-based HPP model to assign resources to individual holons within a processing dimension, thus potentially improving the predictive capacity of the multiple-resources theory.

Architectural Issues

Computational-theory architectural specification makes explicit one's paradigmatic assumptions, particularly pretheorectical ideas, about the nature of the mind. It reflects a commitment to a specific paradigm of research. Commitment to the holon architecture implies a commitment to believing the capabilities of the mind are both hierarchically and modularly organized, communicate through message passing, and employ knowledge at differing levels of abstraction.

In recent years, there has been an active discussion in the research literature as to whether the mind is modular and, if so, the division of specific faculties to different modules. Chomsky (1980), Fodor (1983, 1986), and Minsky (1986) have all proposed models that postulate some form of modularity. These theories differ greatly in terms of their idealization (Kosslyn, 1984); that is, some are more (or less) specific than others in discussing given details about how modularity is realized.

Modular theories of mind, in general, propose a lack of continuity between subsystems of the mind. For example, Fodor (1983, 1986) postulates that there is a distinct subsystem for perception and information within this subsystem is encapsulated. Information encapsulation means that computational processes within the subsystem do not have access to all the information an organism possesses. In addition, Minsky (1986) has proposed a hierarchical modular system for which information at different stages of information processing is at different levels of abstraction.

Architectures define classes of models that can be instantiated at a given system level. For an architecture at the computational-theory level, the classes of models that can be instantiated correspond to different theoretical perspectives concerning the functionality of the mind. For modular theories of the mind, the classes represent different ways to divide the capabilities of the mind among different modules. Specific theories, for example, differ on whether there is a separate module for speech only, or for perception only, or whether speech, perception, and cognition all have separate modules. Further, theories differ on whether the individual modules (e.g., a speech module) are also modular (i.e., is the speech module made up of submodules) and/or whether the submodules are hierarchically organized. Finally, different theories make different claims as to the functionality of separate modules in terms of processing capability, data structures, and communication capability.

Computational-theory-level architectures--like the holon architecture--are instantiated through implementation-level architectures. Implementation-level architectures are realized as *computer languages* (VanLehn & Ball, 1991). Implementation-level architectures are programming languages plus some additional tools that allow a model developer to build specific models. The combined capability of the programming language and tools define the class of models that can be realized.

Specific tools needed to implement the holon architecture include object-oriented programming tools (for message passing, data encapsulation, etc.), and editors to load and structure knowledge. In addition, the programming language must either support asynchronous parallel execution of computer processes or support the simulation of such processes.

Just as a specific computational-theory architecture can be implemented by diverse implementation-level architectures, a specific implementation-level architecture can support diverse computational-theory-level architectures. We believe that one implementation-level architecture can be developed that will support research on most current modular theories of the mind, including the holon architecture.

The capacity for an implementation-level architecture to support multiple computational-theory architectures is achieved through *"implementation via convention."* Implementation via convention is the practice of implementing a programming technique through convention, as opposed to implementing the technique explicitly in the programming language. A classical example of implementation via convention is the "goto" command. Good programming guidance is to never use a goto statement. However, most high-level languages do support a goto command.

For the holon architecture, the most likely constructs to be implemented via convention are message-passing constructs. Currently, three message types are required to implement the psychological functionality (defined by the computational-theory specification) of the holon architecture at the implementation level: hierarchical, broadcast, and point-to-point.

Hierarchical messages (paths) are used to define the branches of a holarchy structure and the part/whole relationship between holons. Each holon has one message path to its parent and one to each of its children. Because holons have additional communication options, two additional message types are also required.

Broadcast messages are sent from one holon to all other holons and are normally be used to denote changes in the internal environment of the holarchy.

Point-to-point messages are sent from one holon to another. These messages differ from hierarchical messages in that the sending and receiving holons are in different branches of a holarchy.

Implementing hierarchical messages via convention as a special case of point-to-point message passing will allow one implementation-level architecture to support computational-theory architectures that differ as to whether they postulate hierarchical modules in addition to modularity. For models that postulate hierarchical arrangements between modules, the model developer will define parent/child relationships by creating a special acquaintance list that defines the parent/child relationships. This will negate the need to have a special hierarchical message method or construct.

In addition, the capability to create N-dimensional models can be implemented by convention. A holarchy that exists in a dimension tangential to another is only allowed to communicate through the one holon that connects the two dimensions. No point-to-point or broadcast message is allowed, even though a "special" case of point-to-point message passing does the communicating. This functionality can also be implemented through a special acquaintance list and programming convention, as opposed to the development of a special communication method.

Why is the ability to use one implementation-level architecture to implement multiple computational-theory-level architectures important? Modular theories of the mind are relatively new and untried. Developing an implementation-level architecture to support exploration of the capacity of various computational-theory architectures to build predictive HPP models will require a fair amount of resource investment. Developing an implementation-level architecture that supports research on multiple theories of the mind increases the potential research opportunities that can be achieved with the architecture, while ameliorating the risk that a specific architecture will not prove sufficient for the development of HPP models.

Future Research

Continued research and development is needed at both the computational-theory and implementation levels to realize an HPP model based on the holon formalism. At the computational-theory level, defining a cognitive architecture is not the same as defining an HPP model. The cognitive architecture defines the class (or range) of models that can be built. To design a specific HPP model, the model developer must employ the tenets of the computational theory to delineate a specific model[2] . For architectures based on the principle of modularity, this involves specifying the topology of the modules representing subcomponents of the information-processing system. In general, module topology delineation requires that the number of modules and their functionality (or purpose), internal processes (methods), and available communication paths be specified.

Architectural tenets for a holon architecture class model further require that individual branches of the holarchy represent different modalities (vision, audition, psychomotor, etc.). Subbranches within a branch (modality) must represent different dimensions of the stimulus being analyzed (or, in the case of psychomotor modality, different limbs or major muscle groups that are being activated). Furthermore, for each holon (module), the model developer must specify, in addition to its purpose (e.g., pattern recognition, categorization, planning, etc.), what aspects of the "world" the holon maintains in its working memory, any additional data it uses to accomplish its purpose, its acquaintances (to include parent, children, and "friends"), the types of messages it can send and receive (broadcast, point-to-point, and hierarchical), and methods for processing messages.

---

[2] Delineating a specific model corresponds to creating an algorithm-level specification (Anderson, 1987; 1990), which is an explicit formulation of a psychological theory in computational terms.

In the next phase of this effort, we will define a specific HPP model based on the holon architecture. The key issue is defining the appropriate number of modules and determining their functionality. To accomplish this, we are planning a multiple-track approach.

First, we will develop a multitask scenario representative of the behavior we want the HPP model to perform. This scenario will be represented as a multilevel goal hierarchy (Adams, Tenney, & Pew, 1991). Analysis of the scenario will provide a first cut of the psychological functionality the HPP model must embody.

We will selectively review the experimental psychological literature concerned with the decomposition of mental function. Identifying and characterizing basic psychological processes is of central concern. There is vast experimental literature addressing short- and long-term memory, episodic and semantic memory, procedural and declarative memory, lexical- and rule-based routes to reading, and lexical and syntactic components of number processing (just to name a few) which postulates a variety of processes and distinct psychological functions.

Next, we will selectively review the cognitive neuropsychological literature to review proposals for isolable systems that underlie human performance. The results from these two literature reviews will be synthesized to create an initial proposal of the modular functionality of the human mind.

Finally, we will collect examples of what Newell has called the "regularities of behavior" (Newell, 1990). These regularities are well-documented (i.e., well-supported by experimental data) behavioral phenomena. Example behaviors include item recognition, typing skill, and skill acquisition. These well-documented behaviors will be used as constraints for which the initial model must account.

Our goal is to create a model that accounts for psychological phenomena in an "emergent" manner. That is, regularities of behavior and other psychology phenomena, such as memory and attention, must result from the cognitive architecture and the embodied knowledge.

Several issues at the implementational level must be resolved. The ideal implementation architecture for the holon computational-theory architecture would be an asynchronous, parallel-processing computer environment. Individual holons could be independent computer processes within this computer environment. The sum of holon processes would constitute the HPP model. However, there are some unsolved problems.

The implementation architecture will require a very sophisticated operating system. There are (at least) two key issues to consider here. First, it is unlikely that the number of holons (i.e., independent computer processes) will match the number of computer processors. Hence, the operating system must have some means of scheduling the use of computer processors. Second, it is likely that there will be some very challenging problems associated with coordinating the timing of message passing between holons. If the combined group of holons consitutes a model, the activity of the holons must be coordinated through the operating system. This control problem could be severely exacerbated if one holon, which provides input to an integrating holon at a "higher" functional level, takes significantly longer to process information than the other input-providing holons. In this case, the "integrating" holon would have to store the other input (from the faster holons) while it waited for the "slower" holon.

We are taking a two-prong approach to resolve these problems. The first method is to thoroughly investigate the creation of an implementation architecture on a parallel-processing computer system. We are initiating a requirements study to further define the issues and to attempt to determine the amount of effort required to create an asynchronous parallel-processing computer environment for HPP model research. Based on the outcome of this study, we will either proceed immediately in attempting to develop this architecture or wait until the technology catches up with our designs.

The second method is to attempt to realize the holon-implementation architecture on a serial computer. In this approach, the implementation architecture would be realized within a discrete-event-simulation environment. At every "tick of the clock," all events for all holons would be processed.

However, there are two problems associated with this approach. First, at the start of a "tick" it is unknown which holons are going to send messages to which other holons, all of which will require processing and might modify ongoing holon activity. A way is needed to "scope out" processing and, hence, scheduling at the beginning of each tick. An earlier AL-sponsored effort (Corker et al., 1991) developed some of the requisite technology. However, additional work is needed to support an architecture of this complexity. Second, there will very likely be a severe penalty in performance for simulating a parallel process on a serial computer. Whether a serial computer system can support real-time simulation of a holon hierarchy must be determined.

Finally, there is a problem that affects both the computational theory and implementation perspectives of the model: examinability. The development of models, in general, based on the concept of modularity, is relatively new and untried. A significant amount of "trial and error" will be required to create a functional model. At the computational-theory level, the trial and error effort will focus primarily on determining the topology of the model; at the implementation level it will focus primarily on determining computational effectual processes. A key research issue is the development of methods that will allow the developers to examine the workings of the model as design parameters on both relevant levels of abstraction are perturbed.

REFERENCES

Adams, M. J., Tenney, Y. J., & Pew, R. W. (1991). *Strategic Workload and the Cognitive Management of Advanced Multi-Task Systems.* Dayton, OH: Wright-Patterson Air Force Base.

Anderson, J. R. (1987). Methodologies for Studying Human Knowledge. *Behavioral and Brain Sciences*, 10, 467-505.

Anderson, J. R. (1990). *The Adaptive Character of Thought.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Card, S. K., Moran, T. P., & Newell, A. (1986). The Model Human Processor: An Engineeering Model of Human Performance. In K. Boff, L. Kaufman, & J. Tomas (eds.), *Handbook of Perception and Human Performance*, Vol. II. New York, NY: John Wiley & Sons.

Chomsky, N. (1980). Rules and Representations. *Behavioral and Brain Sciences,* 3, 1-61.

Corker, K. M. (1991). Model Development and Implementation with the AIRT C2 Evaluation Methodology. Unpublished working paper.

Corker, K. M., Crammer, N. L., Henry, E. H., & Smoot, D. E. (1991). *Methodology for Evaluation of Automation Impacts on Tactical Command and Control (C2) Systems* (AFHRL-TR-90-91). Wright-Patterson AFB, OH: Air Force Human Resources Laboratory, Logistics Research Division.

Fodor, J. A. (1983). *The Modularity of Mind.* Cambridge, MA: MIT Press/Bradford Books.

Fodor, J. A. (1986). Precis of The Modularity of Mind. *Behavioral and Brain Sciences*, 8, 1-42.

Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. (1983). *Building Expert Systems.* Reading, MA: Addison-Wesley.

Janlert, L. (1986). Modeling Change-The Frame Problem. In Z. W. Pylyshyn (ed.), *The Robot's Dilemma.* Norwood, NJ: Ablex.

Koestler, A. (1967). *The Ghost in the Machine.* London: Cambridge University Press.

Koestler, A. (1976). *Janus: A Summing Up.* New York, NY: Random House.

Kosslyn, S. M. (1980). *Image and Mind.* Cambridge, MA: Harvard University Press.

Kosslyn, S. M. (1984). Mental Representation. In J. R. Anderson & S. M. Kosslyn (eds.), *Tutorials in Learning and Memory.* San Fransico, CA: Freeman.

Marr, D. (1982). *Vision.* San Francisco, CA: Freeman.

Minsky, M. (1986). *The Society of Mind.* New York, NY: Simon and Schuster.

Newell, A. (1990). *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Pylyshyn, Z. W. (1991). The Role of Cognitive Architecture in Theories of Cognition. In K. VanLehn (ed.), *Architectures for Intelligence.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Rasmussen, J. (1986). *Information Processing and Human Computer Interaction.* New York, NY: North-Holland.

Rumelhart, D. E., & Norman, D. A. (1983). *Representation in Memory.* Center for Human Information Processing Report (CHIP-116). San Diego, CA: University of California, San Diego.

VanLehn, K., & Ball, W. (1991). Goal Reconstruction: How Teton Blends Situated Action and Planned Action. In K. VanLehn (ed.), *Architectures for Intelligence.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Wickens, C. D. (1980). Processing Resources in Attention. In R. Nickerson (ed.), *Attention and Performance VIII.* Hillsdale, NJ: Lawerence Erlbaum Associates.

Wickens, C. D. (1990). *Processing Resources and Attention.* Aviation Research Laboratory Report (ARL-90-4). Urbana-Champaign, IL: University of Illinois.

Wickens, C. D. (1992). *Engineering Psychology and Human Performance.* New York, NY: HarperCollins.

☆U.S.Government Printing Office: 1993 — 750-061/60238