Developing a Human Error Modeling Architecture (HEMA)[1]

Michael E. Fotta
D.N. American
1000 Technology Drive
Fairmont, WV 26554
mike.fotta@dnamerican.com

Michael D. Byrne
Rice University
6100 Main St., MS-25
Houston TX 77005
byrne@rice.edu

Michael S. Luther
Booz Allen Hamilton
8283 Greensboro Dr
McLean, VA 22102
luther_michael@bah.com

Abstract

Although computational cognitive architectures have been applied to the study of human performance for decades no such architecture for modeling human errors exists. We have undertaken the development of a Human Error Modeling Architecture (HEMA), building on the ACT-R cognitive architecture. In developing HEMA we first set the context of what error types HEMA would handle and what overall cognitive performance process (a Framework for Human Performance) was being assumed. We then identified the cognitive functions which were failing and how they are failing when an error occurs. An analysis of these failures, in relation to the Framework, enabled us to specify a set of General Error Mechanisms. Comparison of these mechanisms to existing ACT-R mechanisms identified where ACT-R could be used, where modifications where necessary and where new mechanisms or modules would be need to develop HEMA. A conceptual design for HEMA was then proposed.

1 Introduction

Human error is continually cited as a cause in major disasters and minor mistakes. However, human error can often be traced to a system design which creates situations beyond a human operator's capabilities. Consider, for example, the Defense realm. Given the speed of weapons systems (e.g., supersonic aircraft, missiles) an operator must often filter, process and make decisions at a speed that does not allow for careful consideration of all the information. Human error at such times can lead to serious and even deadly consequences, such as "friendly-fire" incidents. Providing insight into the human error consequences resulting from a particular system design would enable designers to chose between alternative designs and modify a design to reduce error occurrence or enable recovery from human errors. Our research seeks to develop a Human Error Modeling Architecture (HEMA) that provides this insight by simulating errors that operators will experience as a result of a system design.

Efforts at modeling human error to provide predictive power are scarce. Their have, however, been a large number of taxonomic and descriptive efforts to explain human error behavior. Some of the most well known of these are the Generic Error Modeling System (GEMS) approach (Reason, 1991), the stages-of-action model (Norman, 1986) and the Cognitive Reliability and Error Analysis Method (CREAM) (Hollnagel, 1998). However, they are neither mechanistic, which limits their explanatory power, or predictive. Without predictive power these approaches cannot generally be used to determine which of two designs would generate fewer or less serious errors.

In more recent years there have been attempts to predict errors at a more mechanistic level. One example is the work of Byrne and Bovair (1997), which presented a computational account of a class of errors known as postcompletion errors (e.g., leaving a bankcard in an ATM or leaving the original on a photocopier). Another illustrative recent example comes from Anderson, Bothell, Lebiere, and Matessa (1998), which used ACT-R as a model and showed that it was possible to predict both the rate and content of the errors made in a task.

A fundamental problem in modeling human error is that it is the same human perceptual-cognitive-motor system producing all behavior, whether erroneous or not. Thus, to effectively model human error, it will be necessary to have a relatively complete model of the entire perceptual, cognitive, and motor systems. Computational cognitive architectures such as ACT-R, EPIC or SOAR provide such models (see Byrne, 2003a, for a review). Thus, a

[1] This research was funded under SBIR Phase I Contract: N00014-03-M-0357 for the Office of Naval Research.

cognitive architecture, properly modified, could serve as a basis for a predictive model of human error (Byrne, 2003b). Our research pursues this approach to develop a Human Error Modeling Architecture (HEMA).

We chose to use ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere, & Quin, 2004) as the basis for developing HEMA. ACT-R contains mechanisms which can produce "erroneous" behaviors even when the ostensibly "correct" pieces of declarative and procedural knowledge are present in the system. Furthermore, ACT-R has been extensively and successfully applied to model many domains of human performance, has wide acceptance as a computational cognitive architecture, and has been applied to human error modeling in some instances.

However, while ACT-R contains some error modeling mechanisms, it is unlikely to have all the components necessary for comprehensive error modeling. In order to develop HEMA we need to specify the extent to which ACT-R currently provides these mechanisms. Before accomplishing this we obviously had to define the error related mechanisms needed to develop HEMA. In order to define these mechanisms it is necessary to first identify the cognitive functions which can fail and how they can fail when an error occurs. Identification of these failures has to be set in the context of what errors are occurring and what overall cognitive performance process is being followed. Thus, the first two activities undertaken in developing HEMA were to define the cognitive processes that are involved as an operator performs a complex task and define the error types HEMA should handle.

This paper reports on the results of these tasks beginning with the last two and culminating in the development of the Human Error Modeling Architecture.

2 Define the Human Performance Process

In order to define the cognitive processes involved in human error it was necessary to develop a process model of how an operator performs in an environment likely to produce a variety of human errors. This framework, while a synthesis of readings of a variety of papers and texts, and discussions between the authors, owes much to the texts by Anderson (2000), Reason (1990), and Hollnagel (1998), and papers by Endsley (1999) and Leiden, et. al. (2001).

Figure 1 shows the top level diagram for a Framework of Human Performance (FHP) which describes the processes an operator goes through in performing a complex task. In order to reduce the complexity of this figure and supporting process figures the cognitive processes of attention, perception and memory have been combined in an APM Component and used wherever attention is likely to be applied to enable perception.

The process starts with the arrow on the left of the Attention-Perception-Memory (APM) Component. This represents a conscious intent to attend to the situation at hand, e.g., an airline cockpit display, a Combat Information Center. The operator makes a general assessment of the situation via the Understand Problem decision. Current environmental information from the APM and the operator's knowledge is used to quickly make this decision. If there is enough information the process flows to Set Intention(s). If not, the operator will seek further information using the APM and also directly from Memory. There is obviously a time constraint, but this is not shown here. As this process runs the operator is constructing an internal representation of the situation, the Perceived Situation, which is stored in Memory and retrieved and updated as the rest of the processes function.

In order to Set Intention(s) the operator retrieves information on similar past situations from Memory and compares this to the Perceived Situation. If a match is made to a past situation then the intention for this past situation is used. If a direct match is not made then the best match to either a previous situation or rules from a number of similar situations are used to infer an intention. The intention is stored in Memory.

The operator must then *Form Plan(s)* for each intention. Plans are formed on the basis of the extent to which a plan exist in schemas or other knowledge for the current intention. Reason's (1990) scheme of skill-based, rule-based and knowledge-based processing is used to define the type of plan here. The plan is stored in Memory.

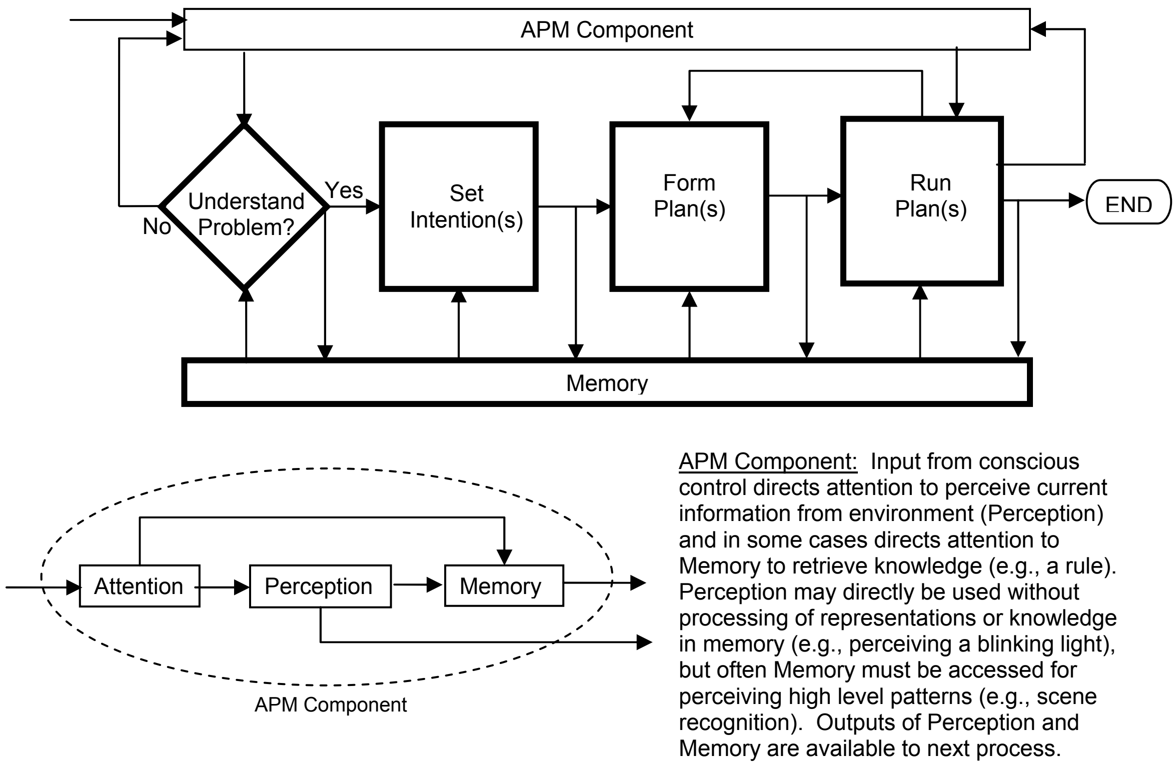

Figure 1. Top level process diagram for a Framework for Human Performance.

Finally, the operator must Run Plan(s). This consists of performing each action in the plan and evaluating the result in relation to the plan and current environmental information, from the APM. The extent of evaluation and modification of plans varies from almost none for skill-based plans to extreme for knowledge-based plans. In the simplest case the next action in the plan will be run or the plan will end. However, results may also indicate that either the intention or plan is no longer appropriate and need modified or dropped. The feedback loops indicate this.

Each process within Figure 1 has been further detailed in its own process description to provide a complete explanation of this framework. This enabled us to use this framework as a basis for identifying the cognitive functions which are involved in producing the error types (see Section 4). Unfortunately space precludes showing this detail here.

3 Define a Sample of Human Error Types

Although our review of the human error literature found many taxonomies and papers on individual errors we came to the conclusion that three approaches formed a broad sample representing the gamut of human error cognitive processing. Each of these approaches are well-described and comprehensive theories of human error and include fairly broad taxonomies. We propose that by identifying mechanisms that cover error types from these taxonomies we form a solid and broad basis for a human error modeling architecture. The three approaches are the Generic Error Modeling System (GEMS) of Reason (1990), Situation Awareness (SA) as put forth by Endsley (1999), and the Cognitive Reliability and Error Analysis Method (CREAM) of Hollnagel (1998). Table 1 shows the error types used in our research from each approach. The taxonomy is shown under the Source column, a high level for the error types in that taxonomy are shown in the Category column, and the Error Types for the Category appear in the last column. Detailed explanations of each error type can be found in the appropriate reference.

Reason's GEMS approach was chosen for a number of reasons: 1) this text is the dominant, most comprehensive descriptive text on human error, 2) it describes a range of performance including skill-based, rule-based and

Table 1. Error types used in study.

Source	Category	Error Types		
GEMS	Skill-based: Inattention	Double-capture slips Reduced intentionality Interference errors	Omissions following interruptions Perceptual confusions	
	Skill-based: Overattention	Omissions Repetitions	Reversals	
	Rule-based: Misapplication of Good Rules	First exceptions Informational overload General rules	Countersigns and non-signs Rule strength	Rigidity Redundancy
	Rule-based: Application of Bad Rules	Lack of Encoding Inaccurate encoding	Protection by specific rules Inelegant rules	Wrong rules Inadvisable rules
	Knowledge-based (KB)	Selectivity Out of sight out of mind Overconfidence	Workspace limitations Confirmation Bias Biased Reviewing	Illusory correlation Causality
	KB: Problems With Complexity	Delayed feed-back Causal series vs. Nets	Thematic vagabonding Processes in time	Encysting
CREAM	Observation (O)	Observation missed		
	O: False Observation	False reaction	False recognition	
	O: Wrong Identification	Mistaken cue	Partial identification	Incorrect identification
	Interpretation (I)	Delayed interpretation	Incorrect prediction	
	I: Faulty Diagnosis	Wrong diagnosis	Incomplete diagnosis	
	I: Wrong Reasoning	Induction error Wrong Priorities	Deduction error	
	I: Decision Error	Decision paralysis	Wrong Decision	Partial Decision
	Planning (P): Inadequate Plan	Incomplete plan	Wrong Plan	
	P: Priority Error	Wrong goal selected		
	Temporary, Person (TP)	Delayed response	Performance variability	Inattention
	TP: Memory Failure	Forgotten	Incorrect recall	Incomplete recall
	TP: Fear	Random actions	Freeze	
	TP: Distraction	Task suspended	Task not completed	Goal forgotten
Situation Awareness	Level 1: Failure to correctly perceive information	Data not available Misperception of data	Data discrimination/detection Failure to monitor or observe data	Memory loss
	Level 2: Failure to correctly integrate or comprehend information	Poor mental model	Use of incorrect mental model Over-reliance on default values	Other
	Level 3: Failure to project future actions or state of the system	Poor mental model	Over-projection of current trends	Other
	General	Habitual schema	Failure to maintain multiple goals	

knowledge-based, and 3) the development and use of plans is a cornerstone of the GEMS approach. Although other authors discuss the use of plans, none go into the detailed description of Reason. Any effort to model human error must account for the development, running, evaluation of and modification of plans.

CREAM is a comprehensive methodology for Human Reliability Analysis (HRA), an extensive area of research to ensure the reliability of complex systems such as aircraft, nuclear reactors, weapons systems, etc. At the highest level CREAM categorizes error types as occurring because of Man (People), Technology or Organizational factors. For our analysis we have used only the People related factors where the cognitive issues and error mechanisms are most likely to occur. The People grouping included one category, Permanent Person Related errors, which we did not include as it went to levels of physical impairments (e.g., deafness, bad eyesight) and individual processing styles (e.g., simultaneous scanning, successive scanning) that is beyond the scope of the first version of HEMA.

Endsley's situation awareness (1999) approach was chosen as: 1) its core tenant is an understanding and projection of an understanding of an entire situation, and 2) the error types form a reasonable taxonomy of perceptual and high-level cognitive processes. The errors which occur when a situation it is not understood is critical to a complete error modeling approach. The perceptual error types are not well covered by Reason, but provide a cross check to the CREAM observational errors. Finally, any error model should be able to account for errors related to complex cognitive processes (e.g., over-projection of current trends) as well as discretized cognitive processing failures.

4 Define Cognitive Functions Involved in Human Error

Our next step was to identify the cause of each error type in terms of a failure within the FHP. We then identified the cognitive function or functions which were involved in this failure. Table 2 shows how this was done for each error type within the Situation Awareness Level 1 category.

Table 2. Identifying the cognitive functions involved in Situation Awareness (SA) Level 1 errors.

Error Type	Location and Cause in FHP	Cognitive Function(s)
Data not available	Not a cognitive process failure.	None
Data hard to discriminate or detect	Occurs in Understand Problem. Could be either a perceptual failure or environment is outside or just at boundaries of perceptual limits.	Perception
Failure to monitor or observe data	Occurs in Understand Problem. Attention failure with perception involved.	Attention: Perception
Misperception of data	Occurs in Understand Problem. If due to influence of prior expectations this is a misperception of the Perceived Situation due to misapplication of declarative knowledge in Memory. If due to distraction this is an Attention failure in APM.	Perception: Memory or Attention.
Memory loss	Occurs in Understand Problem or Set Intentions. No longer in Working Memory, LTM or can not be accessed. Could be that task was shed in Set Intentions process if workload is high.	Memory: Memory loss

An "analysis table" with the type of information as shown in Table 2 was constructed for all error types. Our reasoning was that we would be able to identify commonalities within the cognitive functions by sorting on this column in the table. We hypothesized that these commonalities would enable us to derive a constrained list of General Error Mechanisms. Note that in a few cases (as shown for the "Data not available" error in Table 2) we decided that there was not a plausible cognitive explanation for the error type.

5 Define a set of General Error Mechanisms

Sorting the analysis table on the Cognitive Functions grouped error types which had the same or similar failures. With commonalities in error types now identified and considering the specificity of the error (e.g., if a memory loss then where was it occurring in FHP and/or what other details specified when and how it occurred) we identified the General Error Mechanisms which would have to exist in order to account for all of these error types in our sample.

Table 3 shows a portion of the sorted analysis table. This table shows that by grouping error types by the Cognitive Function Memory with a specific explanation Memory Loss we identified five error types which could be accounted for by the same mechanism - a Decay mechanism. Similar occurrences were found throughout the complete sorted table for all error types, i.e., error types fell into groups by similar Cognitive Functions and explanations. In fact we found it necessary to hypothesize the existence of only 15 error mechanisms, although two did have subsets.

Table 3. Example of identifying a General Error Mechanism for error types with common cognitive functions.

Error Type	FHP Explanation	Cognitive Function	General Mechanism
SA: Memory loss	Loss in Working Memory, LTM, or task was shed in Set Intentions.	Memory: Memory loss	**Decay -**or may be that task was not entered but shed.
SA: Multiple goals	Loss of intention from in Working Memory or a task was shed in Set Intentions.	Memory: Working Memory Loss: Intention	**Decay** (of intention)-or intention was never entered due to task shed.
CREAM: TP-Goal forgotten	Loss of intention from Working Memory (may cause repetition of steps)	Memory: Working Memory Loss: Intention	**Decay** (of intention) -or intention was never entered but shed.
GEMS: Inattention Reduced intentionality	Reduction of strength of intention, or forgetting intention, in Working Memory.	Memory: Working Memory Loss: Intention	**Decay** (of intention) – reduction in strength
CREAM: TP-Loss of orientation	Loss of plan or part of plan from Working Memory.	Memory: Working Memory Loss: Plan	**Decay** (of plan or part of plan)

The hypothesized general mechanisms needed to simulate errors are:

- *Plan developer*: Develops plans given a situation. Can develop incomplete or inappropriate plan.
- *Compare Actions:* Compares expected action to action taken. Can fail due to monitor failure or bias.
- *Monitor*: Performs comparison at certain times to achieve an evaluation, but which can fail to monitor.
- *Attention (for perceptual information):* Allocates attention to perception, but can fail to do so.
- *Bias mechanism*: A bias (strength) which would tend to yield positive comparisons in Compare Action.
- *Rule match*: Matches current information to stored rules. Can fail to retrieve correct rules or apply correct action side of rule. Specific subsets include: General Rules, Rule Bias and Strength of Rule
- *Schema match*: Matches information from perception to entire schema. Can fail to correctly match.
- *Time constraint mechanism:* Places time constraint on various activities, e.g., choosing a rule.
- *Decay:* Reduction of strength of information (e.g., with chunks, rules, intention, plan).
- *Poor Learning (encoding):* Stores incorrect rules, but need to be (somewhat) logically related to learning in previous similar situations. Specific subsets include: Rule conditions not encoded or incorrectly encoded, Rule action incorrect or inefficient, Reduction in rule strength, and Reduction in strength of event schema.
- *Retrieval mechanism*: Retrieves information, but can fail to correctly retrieve.
- *Plan Controller*: Runs plans, but can fail in various ways, e.g., by failing to continue running plans.
- *Perceptual Mechanism:* Inputs perceptual information. Can fail to perceive some information.
- *Association Developer*: Develops associations from memory, but can fail, e.g., by developing narrow association net when deeper one should be developed.
- *Motor Mechanism:* Performs motor execution, but can fail to perform necessary action.

Note that all of these mechanisms are not directly error causing mechanisms. Many are functions that will have to be simulated (e.g., the Plan Developer) to account for the cognitive processes which lead to the error types within the FHP framework. Some mechanisms on the other hand are directly related to errors (e.g., the Bias Mechanism).

6 Compare the General Error Mechanisms to ACT-R mechanisms

Given these necessary error mechanisms we then performed a comparison to ACT-R mechanisms as shown in Table 4. In other words we wanted to identify to what extent ACT-R could account for the General Error Mechanisms. Some of the General Error Mechanisms map straightforwardly to extant ACT-R mechanisms, some will require an extension of existing ACT-R mechanisms and some require new development in HEMA. Table 4 shows both the comparison to ACT-R and the proposed implementation in HEMA.

Table 4. Comparison of General Error Mechanisms to ACT-R

General Error Mechanism	Comparison to ACT-R	Implementation in HEMA
Plan developer	Could use productions and chunks in ACT-R but difficult. AI planning field provides better approaches.	Needs a new mechanism in HEMA - Plan Developer.
Plan Controller **Compare Actions** **Monitor** **Bias mechanism**	Combine into one mechanism for controlling and monitoring plans. Could revise and modify previous ACT-R goal management system, but revision more cumbersome than developing a new mechanism.	Needs a new mechanism in HEMA - Plan Controller which includes Compare Action, Monitor, and Bias Mechanism functions.
Rule Match	Mechanism exists in ACT-R.	Use ACT-R mechanism.
Schema Match	Schemas represented in ACT-R by large hierarchical chunks, but this works poorly and matching is slow.	Modify ACT-R by associating large chunks to develop specific schema structures.
Time Constraint Mechanism	A timing mechanism must be included as ACT-R has no time sense.	Extend ACT-R's Scheduler to include time sense.
Decay	Mechanism exists in ACT-R.	Use ACT-R mechanism.
Attention	Currently too constrained in ACT-R. Include functionality to vary level of attention, spatially or temporally application, etc.	Extend ACT-R mechanism. Start with current attention, but implement as separate module.
Poor Learning (encoding)	This "mechanism" is basically placing poor information into a knowledge representation such as ACT-R's.	Place incorrect information in modified ACT-R knowledge structure, i.e., add schema structure.
Retrieval	Mechanism exists in ACT-R.	Use ACT-R mechanism.
Perceptual Mechanism	Current ACT-R's Perception Module, in general, always correctly perceives the environment.	Modify ACT-R's Perception Module to allow errors.
Association Developer	ACT-R has associations between chunks.	Use ACT-R mechanism and develop poor associations before simulating performance.
Motor Mechanism	ACT-R's Motor Module, in general, always correctly performs the correct action.	Modify ACT-R's Motor Module to allow errors.

7 Design the Human Error Modeling Architecture

Once we identified which General Error Mechanisms ACT-R could handle and where extensions and new mechanisms where needed we designed a conceptual architecture for HEMA. Besides including the ACT-R mechanisms as identified in our analysis this design must also include components to handle all the General Error Mechanisms as described in Table 4 above. Furthermore, HEMA must also handle the processes of the FHP and the error types from which the General Error Mechanisms and hence the HEMA design derive.

Figure 2 presents a conceptual design for HEMA, as a UML component diagram, which meets these criteria. All mechanisms are either specifically shown or can be mapped to existing or modified ACT-R mechanisms (e.g., Rule Match, Decay). Much of HEMA can be accounted for with ACT-R as shown in the shaded area. Note that while generally modules only access buffers, we have included some module to module access via greyed lines to indicate new interactions within HEMA.

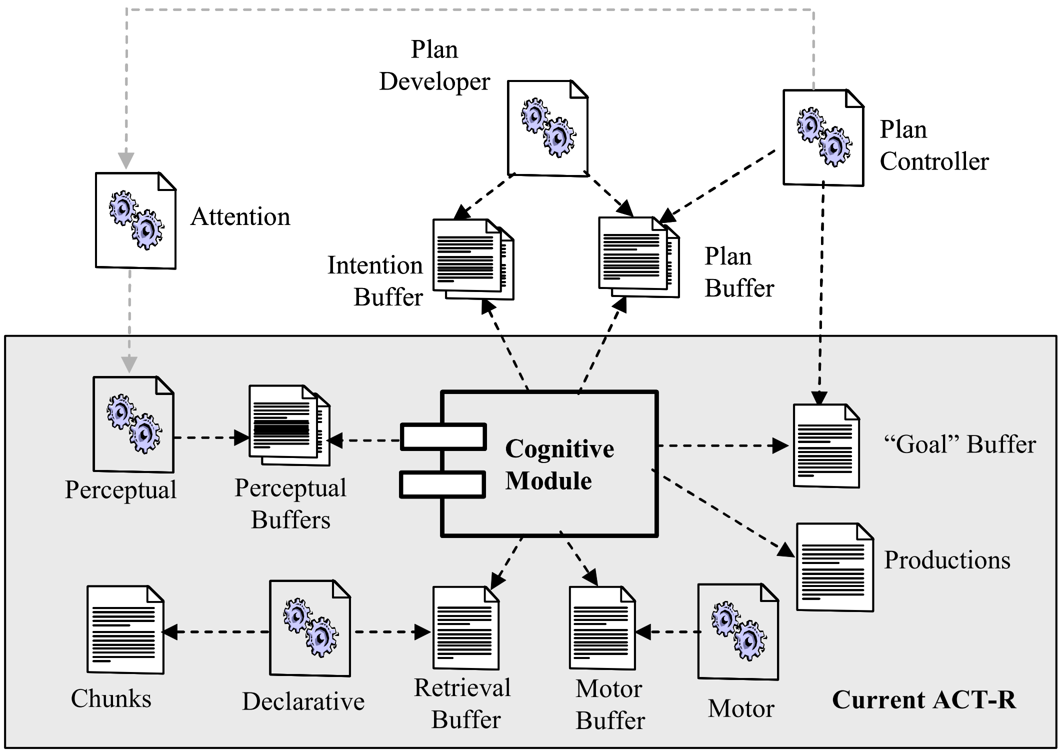

Figure 2. Design for a Human Error Modeling Architecture

The following discussion briefly shows how this component diagram could serve as a design for implementing the FHP (Figure 1):

- *Understand Problem:* The Attention Module directs the Perceptual Module to get information from the environment. The Cognitive Module builds a perceived situation which is placed in the Retrieval Buffer. The Cognitive Module can direct Attention to gather more perceptual information if the operator has time and needs more information.
- *Set Intentions:* Using Productions, Declarative Memory, Chunks and Retrieval Buffer contents the Cognitive Module sets an intention and places this in the Intention Buffer. It is assumed that the Cognitive Module can also estimate the effort needed (from information stored in Chunks) and remove or weigh intentions.
- *Form Plans:* The Plan Developer runs and uses the Cognitive Module to run Productions and Declarative Memory to access Chunks in order to form plans. For skill-based performance a well-rehearsed plan may already be available in Declarative Memory for the current situation. However, for rule or knowledge based performance the Plan Developer will have to run the Cognitive Module repeatedly and form a novel plan. The Plan is placed in the Plan Buffer.
- *Run Plans:* Plan Controller runs plans by retrieving a plan from Plan Buffer and first running the next action in the plan. This action could be: 1) requesting more information from the environment via Attention, Perceptual and Cognition Module to update the Perceived Situation, 2) accessing the Cognitive Module to derive further information from Declarative Memory, or 3) performing a motor activity via Motor Module. If the current action needs an evaluation (and we are assuming information relative to this

need must be placed in the plan) then the Monitor function of Plan Controller requests information on the environment via Attention and Perception to update the Perceived Situation. Plan Controller then activates the Compare Plan mechanism to perform an evaluation to determine current plan status. The Plan Buffer and if necessary the Intention Buffer are updated.

8 Error Types in HEMA

As pointed out above HEMA should be able to explain how to model the error types shown in Table 1. In order to do this we first performed another analysis and sorting of the error types in terms of which error types could be implemented with similar mechanisms in HEMA. This led us to a derivation of a HEMA error taxonomy, which at present has four major categories - Perception, Plan Development, Plan Control and Memory. Each major category has sub-categories of error types and in some cases even a secondary sub-category.

To give some example of the modeling within this error taxonomy we show a few error types in Table 5. The first column in Table 5 gives the name of a HEMA error type, the second column provides a description of how that error could occur using a system implementing the HEMA design, and the third column shows which original error types the HEMA error type can be traced to. Our design includes similarly detailed tables for each of the four major HEMA error categories.

The final step in the conceptual design was the development of sample UML sequence diagrams for each category. These diagrams describe in detail the sequence of actions which would occur in HEMA when a particular error occurs. Sequence diagrams, and supporting information, for each error type will be developed for use in directing the implementation of HEMA.

Table 5. Example of HEMA Error Types and Mapping to original error types.

HEMA Error	Description	Derived From
Plan Development: I. Plan Incorrect: A.Wrong Plan Set	Plan Developer forms wrong plan. The Cognitive module retrieves intention from Intention Buffer then accesses Declarative Memory but retrieves an incorrect schema or rule(s) for the plan. The Plan Developer places this incorrect plan in the Plan Buffer.	CREAM P: Inadequate Plan: Wrong Plan
Plan Control IV. Evaluate Interpretation - Time Delay	Plan Controller runs, but Compare Action runs before Goal Buffer is updated with new information about effect of action. This could be modeled by introducing a delay in the Cognitive Module performing schema/rule matching after action is performed	GEMS KB: Problems with Complexity: Delayed feed-back
Perception: II. Perceptual Attention Failures: C. Wrong Features	Attention directs Perceptual Module to incorrect features in environment (saliency dominates over logic).	GEMS KB: Selectivity
Memory: VI. Memory Loss: B. Decay	This is currently accounted for in ACT-R by activation being reduced on Chunks in declarative knowledge so these Chunks are not retrieved.	CREAM TP: Memory Failure: Forgotten and SA - Level 1: Memory loss

9 Summary

In order to develop a Human Error Modeling Architecture it was first necessary to first develop a complete process flow of human performance - a Framework of Human Performance (FHP). The development of the FHP, along with the identification of a broad sample of human error types enabled the description of cognitive function failures in the context of the FHP. Identifying the commonalities in these failures led to the proposition of fifteen General Error Mechanisms that could account for the error types sampled. Comparison of these mechanisms to the ACT-R architecture demonstrated that ACT-R could account for many of these mechanisms and serve as a basis for HEMA. The proposed HEMA design includes a core ACT-R with extensions (e.g., schema, extended Perceptual Module) and additional modules.

Our final goal is to provide the results of HEMA to system designers for use in assessing the error incidences likely to occur given a proposed system design. To reach this goal HEMA will be the core component on a larger system Human Error Model for Error Tolerant Systems (HEMETS). HEMETS must include capabilities to interface HEMA with system design simulations and provide system designers with some form of "error prediction report" for each system design. Further research will consider the best ways to achieve these goals while implementing the Human Error Modeling Architecture.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(1036-1060).

Anderson, J.R., Bothell, D., Lebiere, C., and Matessa M. (1998). An Integrated Theory of List Memory. *Journal of Memory and Language*, 38, 341-380.

Byrne, M.D. (2003a). Cognitive Architecture. In J.A. Jacko and A. Sears (Eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 97 - 117). Mahwah: Lawrence Erlbaum Associates.

Byrne, M. D. (2003). A mechanism-based framework for predicting routine procedural errors. In R. Alterman & D. Kirsh (Eds.), *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Byrne, M.D. and Bovair, S. (1997). A working memory model of a procedural error. *Cognitive Science*, 21, 31-61.

Endsley, M.R. (1999). Situation Awareness and Human Error: Designing to Support Human Performance. *Proceedings of the High Consequence Systems Surety Conference*, Albuquerque.

Hollnagel, E. (1998). Cognitive Reliability and Error Analysis Method. Oxford, UK: Elsevier.

Leiden, K. K., Laughery, R., Keller, J., French, J., Warwick,W. and Wood, S.D. (2001). A Review of Human Performance Models for the Prediction of Human Error. NASA System-Wide Accident Prevention Program. Moffet Field: NASA.

Norman, D.A. (1986). Cognitive Engineering. In D.A. Norman, S.W. Draper (Eds.) *User Centered System Design: New Perspectives on Computer Interaction* (pp. 31 - 61). Hillsdale: Lawrence Erlbaum Associates.

Reason, J.T. (1990). Human Error. New York: Cambridge University Press.