Lebiere, C., Archer, R., Warwick, W., & Schunk, D. (2005). Integrating Modeling and Simulation into a General-Purpose Tool. In Proceedings of the 11th International Conference on Human Computer Interaction. Las Vegas, NV.

Integrating Modeling and Simulation into a General-Purpose Tool

Christian Lebiere, Rick Archer, Walter Warwick & Dan Schunk

Micro Analysis and Design
4949 Pearl East Circle
Boulder, CO 80301
clebiere@maad.com, rarcher@maad.com, wwarwick@maad.com, dschunk@maad.com

Abstract

This paper describes a series of efforts to integrate a cognitive modeling architecture (ACT-R) with a task network simulation tool (IMPRINT). The goal is to combine the advantages of both approaches while minimizing their shortcomings. We describe a number of applications in which different ways of combining the two systems are attempted. The benefits of each combination scheme are described as well as their limitations. Preliminary indications suggest that while each approach might be best suited for some applications, in general the highest degree of integration is favored as providing the best leverage of the two modeling paradigms.

1. Introduction

Modeling and simulation systems are designed around a particular technical approach, and with a specific purpose in mind. However, once they have been designed and implemented and their users have bought into them and become experts at their use, their origins are often forgotten. The systems become a prism through which every problem can be viewed. As the saying goes, when all you have is a hammer, everything starts to look like a nail. This is particularly true for software systems with the power and flexibility of general programming languages such that it seems every problem can be represented and solved within that framework. However, just because a problem can be solved using a given tool doesn't mean that it is the right tool for the job. Systems are typically targeted, implicitly or explicitly, at a category of tasks for which their assumptions are well-suited, and for which they provide the right constraints and mechanisms to minimize effort and maximize reward (be it in terms of computational power or predictiveness). Often, that category of tasks is the one that is of primary interest to their designers, and a natural synergy develops where the tool is designed with those tasks in mind, and the availability of the tool naturally leads its user to focus on tasks for which it is well suited. However, once a substantial investment in terms of money, time and most importantly conceptual mind-space has been made, users naturally tend to want to use their favorite tool for problems for which they are not entirely well-suited. This doesn't imply that the system should be abandoned, but that it needs to be extended.

There are a number of ways to extend the power of a modeling or simulation tool. One approach (the "bag of tools") is to simply use a different tool when it is better suited for that purpose. One can add to the hammer the screwdriver and the wrench and have a pretty diverse tool set. However, as applications become more complex, tasks often present diverse characteristics that make different tools well-suited for different aspects of the task, but none well-suited for all of it. One needs some kind of combination of those tools. Another approach ("embrace and extend") is to enhance a given tool by adding new capabilities upon the existing base. While this approach has the advantage of conservatively limiting disruptions to incremental additions, it runs the risk of stretching the paradigm upon which the tool was built to areas beyond those for which it is a natural fit, leading to a gradual deformation in which use becomes more and more unnatural. The final approach ("integration") is to extend the capabilities of the original tool into a new domain by integrating it with another system that has proven to be well-suited for that domain. This approach has the advantage of relying upon proven technologies on both sides, but it carries the potential danger, if done wrong, of creating a strange and ugly hybrid. One needs to find the right way to merge and leverage the capabilities of the distinct tools, without sacrificing their power. We will focus on this last approach.

When integrating modeling and simulation tools, a number of problems arise, both technical and conceptual. At the technical level, the two systems must communicate with each other. They must exchange information about the problem that they are solving, and they must synchronize their operations with each other to ensure that they work in concert. The simulation community has long recognized those problems and has developed increasingly general and powerful techniques to solve them, e.g., the High-Level Architecture (HLA) (Kuhl et al., 1999). In the projects we describe, we have not used those systems directly because of the high computational overhead and complexity involved, but we have used the underlying techniques to address the problems for which they were developed. At the conceptual level, it is essential for the two systems being integrated to develop a natural match. To properly leverage the strengths of each system, their relative concepts must connect across the integration boundary rather than clash with each other. In the domain of intelligent systems, the need to integrate systems originating from fundamentally different paradigms (e.g. symbolic and connectionist systems) has been recognized as necessary to reproduce the full capabilities of human cognition. There are however a broad array of techniques and levels of integration, with widely different properties and results (Wermter & Sun, 2000).

In this paper we explore a number of different ways of integrating our target systems. First, we introduce those systems, the Adaptive Control of Thought – Rational (ACT-R) cognitive architecture (Anderson et al., 2004) and the IMPRINT (IMproved Performance Research INtegration Tool) task network modeling tool (Archer & Adkins, 1999). We then describe three different ways of integrating those tools. Each approach draws on a particular point of connection between matching concepts in the two systems. The approaches are cumulative, with each subsequent one building upon the previous one but going one step further in the integration of the two systems. For each approach, we lay out its motivation and underpinning principle, detail its technical implementation, briefly describe existing application(s) and discuss its advantages and drawbacks. Finally, we attempt to draw general lessons about system integration in modeling and simulation systems.

1.1 ACT-R

ACT-R is a unified architecture of cognition developed over the last 30 years at Carnegie Mellon University. At a fine-grained scale it has accounted for hundreds of phenomena from the cognitive psychology and human factors literature. The most recent version, ACT-R 5.0 (Anderson et al, in press), is a modular architecture composed of interacting modules for declarative memory, perceptual systems such as vision and audition modules, and motor systems such as manual and speech modules, all synchronized through a central production system (see Figure 1). This modular view of cognition is a reflection both of functional constraints and of recent advances in neuroscience concerning the localization of brain functions. ACT-R is also a hybrid system that combines a tractable symbolic level that enables the specification of complex cognitive functions with a subsymbolic level that tunes itself to the statistical structure of the environment to provide the graded characteristics of cognition such as adaptivity, robustness and stochasticity.

The central part of the architecture is the production module. A production can match the contents of any combination of buffers, including the goal buffer, which holds the current context and intentions, the retrieval buffer which holds the most recent chunk retrieved from declarative memory, the visual and auditory buffers that hold the current sensory information, and the manual and vocal buffers which hold the current state of the motor and speech module. The highest-rated matching production is selected to effect a change in one or more buffers, which in turn trigger an action in the corresponding module(s). This can be an external action (e.g., movement) or an internal action (e.g., requesting information from memory). Retrieval from memory is initiated by a production specifying a pattern for matching in declarative memory. Each chunk competes for retrieval, with the most active chunk being selected and returned in the retrieval buffer. The activation of a chunk is a function of its past frequency and recency of use, the degree to which it matches the requested pattern, plus stochastic noise. Those factors confer to memory retrievals, and behavior in general, desirable “soft” properties such as adaptivity to changing circumstances, generalization to similar situations, and variability (Anderson and Lebiere, 1998).

The current goal is a central concept in ACT-R, which as a result provides strong support for goal-directed behavior. However, the most recent version of the architecture is less goal-focused than its predecessors by allowing productions to match to any source of information, including the current goal, information retrieved from declarative memory, objects in the focus of attention of the perceptual modules and the state of the action modules. The content of many of those buffers, especially the perceptual buffers, might have changed not as a function of an internal request but as a result of an external event happening, perhaps unexpectedly, in the outside world. This emphasis on asynchronous pattern matching of a wide variety of information sources better enables ACT-R to operate and react efficiently in a dynamic fast-changing world through flexible goal-directed behavior which gives equal weight to internal and external sources of information.

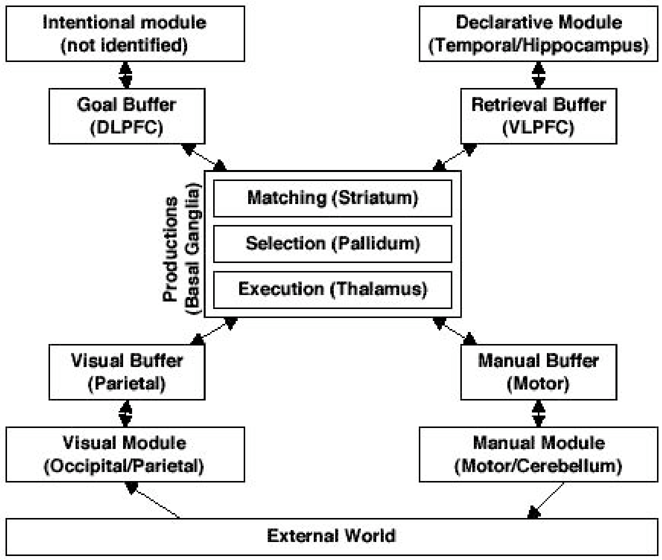

Figure 1: Modular View of the ACT-R Cognitive Architecture

There are three main distinctions in the ACT-R architecture. First is the procedural-declarative distinction that specifies two types of knowledge structures – chunks for representing declarative knowledge and productions for representing procedural knowledge. Second is the distinction between symbolic level, which contains the declarative and procedural knowledge, and sub-symbolic level of neural activation processes that determine the speed and success of access to chunks and productions. Third is a distinction between the performance processes by which the symbolic and sub-symbolic layers map onto behavior and the learning processes by which these layers change with experience.

Human cognition can be characterized as having two principal components: 1) the knowledge and procedures codified through specific training within the domain, and 2) the natural cognitive abilities that manifest themselves in tasks as diverse as memory, reasoning, planning and learning. The fundamental advantage of an integrated architecture like ACT-R is that it provides a framework for modeling basic human cognition and integrating it with specific symbolic domain knowledge of the type specified by domain experts (e.g., rules specifying what to do in a given condition, a type of knowledge particularly well-suited for representation as production rules). However, performance described by symbolic knowledge is mediated by parameters at the sub-symbolic level that determine the availability and applicability of symbolic knowledge. Those parameters underlie ACT-R's theory of memory, providing effects such as decay, priming and strengthening and make cognition adaptive, stochastic and approximate, capable of generalization to new situations and robustness in the face of uncertainty. They also can account for the limitations of human performance, such as latencies to perform tasks and errors that can originate from a number of sources. Finally, they provide a basis for representing individual differences such as those in working memory capacity, attentional focus, motivation and psychomotor speed as well as the impact of external behavior moderators such as fatigue through continuous variations of those subsymbolic architectural parameters that affect performance in complex tasks (e.g. Lovett, Reder and Lebiere, 1999; Taatgen, 2001).

Similar computations are at work in other modules, such as the perceptual-motor modules. Especially important are the parameters controlling the time course of processing as one attempts to execute a complex action, or as one shifts visual attention to encode a new stimuli (Byrne and Anderson, 2001). ACT-R can predict not only direct quantitative measures of performance such as latency and probability of errors, but from the same mechanistic basis can also arise more global, indirect measures of performance such as cognitive workload. While ACT-R has traditionally shied away from such meta-awareness measures and concentrated on matching directly measurable data such as external actions, response times and eye movements, it is by no means incapable of doing so. Lebiere

(2001) proposed a measure of cognitive workload in ACT-R grounded in the concept of unit task (Card, Moran and Newell, 1983) and validated against empirical data collected using the NASA-TLX workload assessment method.

1.2 IMPRINT

IMPRINT (Archer and Adkins 1999) was developed for Army Research Laboratory Human Research Effectiveness Directorate (ARL-HRED) by Micro Analysis and Design to conduct human performance analyses very early in the acquisition of a proposed weapon system. It consists of a set of automated aids to assist researchers in conducting human performance analyses. IMPRINT assists a user in estimating the likely performance of a new system by facilitating the construction of flow models that describe the scenario, the environment, and the mission that must be accomplished (see Figure 2).

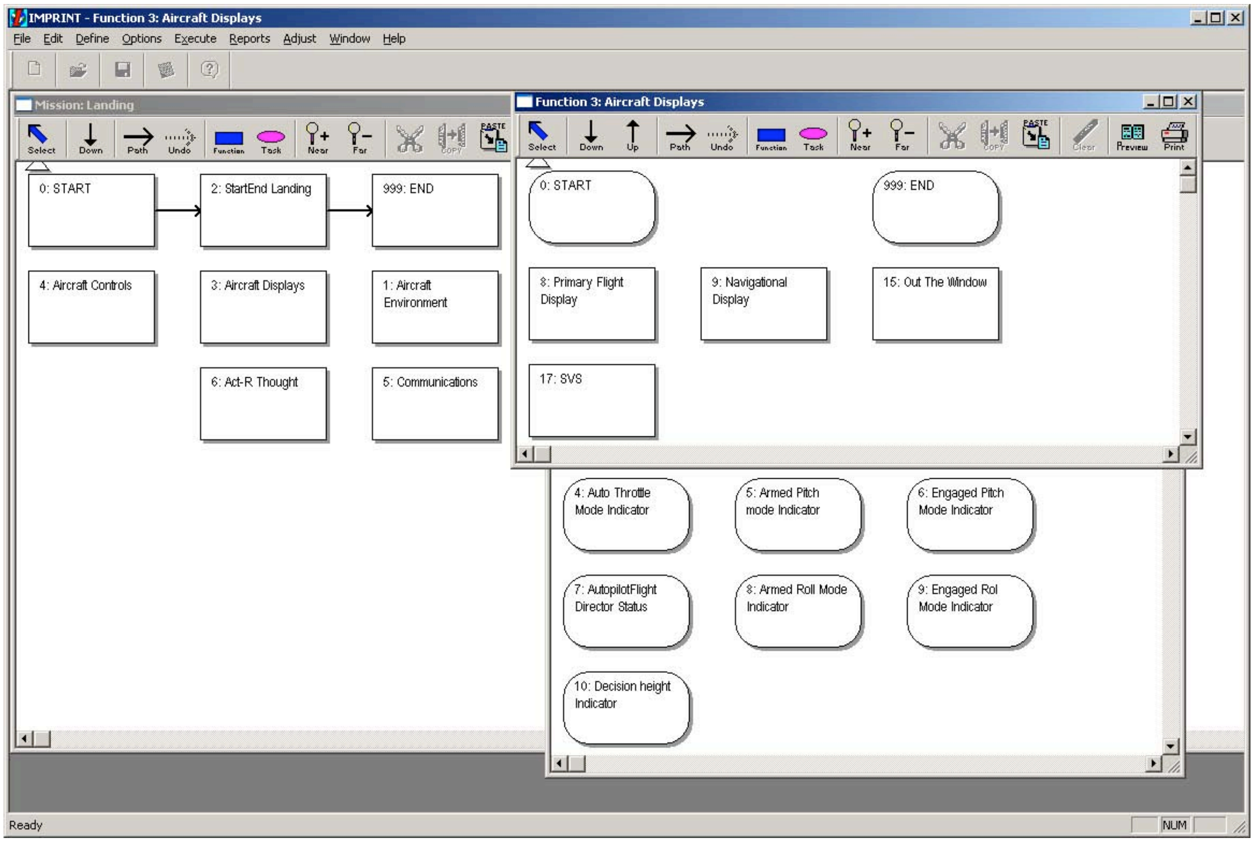

Figure 2: IMPRINT Task Network View

Since it is typically easier to describe the mission by breaking it into smaller “sub” functions than trying to describe the mission as a whole, users build these models by breaking down the mission into a network of functions. Each of the functions is then further broken down into a network consisting of other functions and tasks. Then, a user estimates the time is will take to perform each task and the likelihood that it will be performed accurately. Finally, by executing a simulation model of the mission multiple times, you can study the range of results that occur. A description of the variability of each element can be obtained for further analysis. Additionally, at the completion of the simulation, IMPRINT can compare the minimum acceptable mission performance time and accuracy to the predicted performance. This will determine whether the mission met its performance requirements. Additionally, IMPRINT incorporates the results of the research in cognitive workload that has been represented as computer algorithms (e.g., McCracken and Aldrich, 1994). Given a description of the tasks and equipment with which humans are engaged, these algorithms support assessment of when workload-related performance problems are likely to occur, and often include identification of the quantitative impact of those problems on overall system performance.

IMPRINT has been used successfully to predict human performance in complex and dynamic operational environments. It has been shown to be easy to use, and fairly quick to apply. It does not include an embedded model of cognitive or psychological processes. Rather, it relies on the modeler to specify and implement these constructs. IMPRINT was implemented in C on the Windows platform. It includes a graphical user interface for model authoring, a library of existing weapon systems, expandable function libraries, data collection and display modules, and built-in optimization and animation tools for simulation development. These capabilities provide an easy-to-use interface for the development of simulation models that can be used to study and assess human processes.

1.3 Strengths and Weaknesses

ACT-R and IMPRINT can be viewed as representative examples of two different approaches to human performance modeling (Laughery, Lebiere & Archer, in press). ACT-R, and cognitive architectures in general, is an instance of first-principle models. They operate by representing explicitly the goals, knowledge and processes of human cognition. Performing complex tasks is then modeled by combining elementary actions and knowledge elements (the "atomic components of thought") into increasingly complex and usually goal-directed processing. This is a fundamentally bottom-up approach to modeling human performance. IMPRINT, and task network modeling in general, is an instance of the reductionist approach to modeling. These models use human-system task sequences as the primary organizing structure, with individual models of human behavior for each task or task element connected to this task sequencing structure. The process of modeling human behavior involves taking the larger aspects of human-system behavior and then successively reducing them to increasingly smaller elements of behavior until a level of decomposition is reached at which reasonable estimates of human performance for the task elements can be made and represented in the network as parameters of the lower-level tasks. This is a fundamentally top-down approach to modeling system and human performance.

These two approaches have complementary advantages and disadvantages. ACT-R models are distinguished by their high-fidelity to human behavior across a rich array of behavioral measures, including latency, probability and type of errors, eye and motor movements, and even functional Magnetic Resonance Imaging (fMRI) indicators of localized brain activity. Because of the generality of its architecture, ACT-R can perform any task that human cognition can accomplish. Moreover, because it breaks down performance on complex tasks in terms of basic steps of cognition whose components and mechanisms have been validated in laboratory experiments, it brings significant predictive power from the constraints that those processes and mechanisms put upon its performance. However, with the high-fidelity come a number of requirements. Building ACT-R models of complex tasks can be very time-intensive. It also requires knowledge of specific user strategies and representations that might not be available in many situations. Running ACT-R models can be computationally demanding in complex simulations. And finally, high-fidelity might not always be an essential requirement of the model, or it might only matter in some parts of a task, or it might be involved in a trade-off with other desirable qualities such as robustness.

IMPRINT models, in large part because they follow a top-down methodology and are not committed to any particular analysis level, have opposite pros and cons. Model development is usually tractable for even very complex tasks, because the decomposition level can be adjusted upward as complexity increases and low-level details become less important. For the same reasons, they also tend to be very efficient to run and can be run many times in Monte Carlo simulations. Because task network models are not committed to any particular mechanisms, they can be used to represent the system as well as the operators interacting with it, making for a very efficient, integrated simulation. The main drawback of those models is that precisely because they are not committed to any principled mechanisms or processes, the basic tasks at the level at which the decomposition approach stops need to be parameterized. Values indicating the time taken to perform the task, the probability of errors and the resulting workload need to be specified. Sometimes those data are available from human performance results, but often it needs to be estimated, or it becomes a free parameter of the model.

One could view these two approaches as fundamentally opposite and incompatible. There are parallels to the symbolic-connectionist debate in the representation of human cognition. Proponents of the connectionist approach argue that representing the neural processes captures the true nature of brain processes and that symbolic approaches are merely descriptive. Proponents of the symbolic approach argue that they have captured the most productive level of abstraction to describe mental processes and that neural networks are too concerned with implementation details. The two approaches in isolation have significant limitations but also complementary strengths and the growing field of hybrid cognitive systems has recognized that the most productive approach is to try to combine them. We have adopted the same philosophy in our approach. With a focus on the modeling of complex, real-world tasks, we wanted to preserve the tractability and efficiency of task network modeling, but also to augment it in crucial areas with the constraints and predictiveness of high-fidelity cognitive models.

2. Task Level Integration

2.1 Approach

The first integration principle that we adopted is based upon the central organizing concepts of each approach, leading to a natural and powerful synergy. IMPRINT is focused on the concept of task, how high-level functions break down into smaller-scale tasks and the logic by which those tasks follow each other to accomplish those functions. The power of this top-down approach effectively stops at the final layer of task decomposition, at which point parameters like task times and execution conditions describing those tasks have to be provided by the user. Conversely, ACT-R is targeted at the "atomic" level of thought, the individual cognitive, perceptual and motor acts that take place at the sub-second level. Those individual cognitive operations are organized together in order to meet goals, which provides structure and direction to the flow of cognition.

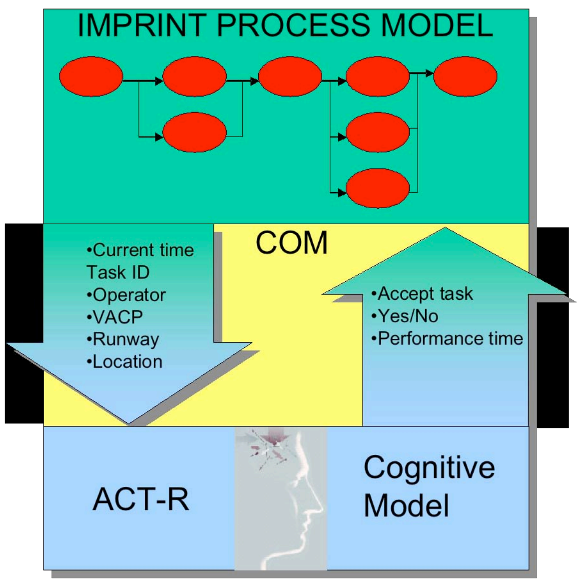

Figure 3: IMPRINT-ACT-R Communication

The ACT-R theory specifies in detail the performance and learning that takes place at each cycle within a specific goal, but has comparatively little to say about the selection of those goals. Since goals in ACT-R closely correspond to tasks in IMPRINT, that weakness matches perfectly IMPRINT's strength, which is to specify efficiently how complex tasks are accomplished from the coordination of simpler ones. Conversely, since IMPRINT requires the characteristics of each task to be specified as part of the model, ACT-R can be used to generate those detailed characteristics in a constrained, first-principled manner without requiring extensive data collection. In summary, the top-down and bottom-up approaches are reconciled because they meet at the same level, where one ends and the other starts.

The structure of a hybrid IMPRINT/ACT-R model is described in Figure 3. The IMPRINT model specifies the network of tasks and includes the definition of how higher-order functions are decomposed into tasks and the logic by which these tasks are composed together. For certain tasks where higher behavioral fidelity is required, IMPRINT sends to ACT-R over a Component Object Model (COM) link the state of variables providing a detailed description of that task. ACT-R then creates a goal corresponding to that task, with the components of the goal set to the description of the task. The ACT-R model for that goal is then run, producing detailed cognitive predictions including latency of the run, whether an error occurred, etc. Those results are then passed back over the same COM link to IMPRINT, which uses them as parameters of the task to advance the task network model.

2.2 Application

Lebiere et al (2002) described a practical application of the IMPRINT and ACT-R integration to the modeling of a complex and dynamic task. Researchers with the National Aeronautics and Space Administration (NASA) were interested in developing models of pilot navigation while taxiing from a runway to a gate. Research on pilot surface operations had shown that pilots can commit numerous errors during taxi procedures (Hooey & Foyle, 2001). NASA was hoping to reduce the number and range of pilot error during surface operations by using information displays that would improve the pilots' overall situation awareness. The IMPRINT model handled the higher level, task oriented parts of the taxiing and landing operations while ACT-R handled the more cognitive and decision making parts of the task.

The IMPRINT task network model consists of the tasks that the Captain and the First Officer perform from the time that the airplane approaches the airport until the airplane either commits a taxi navigation error or arrives at the correct terminal gate without committing an error. During the runway roll out segment, the taxi route and gate information is communicated to the crew by the control tower. This communication task shares that information with the ACT-R model through the COM link by calling an ACT-R goal to encode the information. As the aircraft approaches each potential runway turn-off, information about whether runway signage has been noticed or communicated is also passed to the ACT-R model. In turn, ACT-R passes information back to IMPRINT about whether to make a turn and in which direction to turn.

In the spirit of concentrating on the areas where cognitive accuracy is most critical, the ACT-R model focused on the task of memorizing and recalling the list of taxiways to follow after landing. This task is similar to the cognitive psychology task of list learning, for which an ACT-R model had already been developed (Anderson et al., 1998). We adapted that model to the task at hand while preserving its fundamental representation and parameters, thus eliminating degrees of freedom and inheriting that model’s empirical validation. The taxiing model could reproduce the full range of errors observed in human pilots. Using this integrated modeling and simulation tool allowed us to be able to represent a complex, dynamic task in efficient yet principled fashion and by using the strengths of each architecture, the modeling process was enhanced and streamlined.

3. Federated Integration

3.1 Approach

The previous approach based on the correspondence between IMPRINT tasks and ACT-R goals has a number of advantages. It allows each tool to still display its most powerful attributes while lessening its limitations. It provides a natural integration based on central concepts of each paradigm. And it allows a modular commitment to integration in which the degree of fidelity can be adjusted as needed. A significant shortcoming is that the structure of the behavior, and in particular the organization of decisions needed to accomplish complex tasks, is left to the task network model, with the cognitive architecture only providing fidelity in terms of how each low-level task is accomplished (latency, errors, workload). While this is fine when the task structure is relatively fixed, such as when it is strongly constrained by the nature of the task and formal procedures (as in the example above), it falls short when the decisions of how to decompose complex tasks in terms of simpler tasks and when to execute those reflects complex cognitive process such as adaptivity that are best represented in the cognitive model itself.

One solution would be to make those decisions themselves be tasks in the task network, and thus preserve the task-level integration paradigm described previously. However, this would become cumbersome if the decisions themselves are based not upon a well-understood task structure but upon unpredictable events such as interrupts and other asynchronous external information. To tailor the integration of ACT-R and IMPRINT to those kinds of dynamic environments, which are becoming an increasingly important domain of application, we situated the integration at a more basic level than the task or goal. That level is the concept of event. Task network models such as IMPRINT are based upon discrete event simulation tools like MicroSaint. As recently as ACT-R 4.0 (Anderson & Lebiere, 1998), ACT-R was centered on the concept of production cycle, but recognizing the need for more interruptible and less goal-centered behavior to operate successfully in dynamic environments, the architecture of ACT-R 5.0 is based on independent modules communicating with each other and with the external world through asynchronous events. Thus events are central concepts to both IMPRINT and ACT-R, and become a central concept in their integration as well.

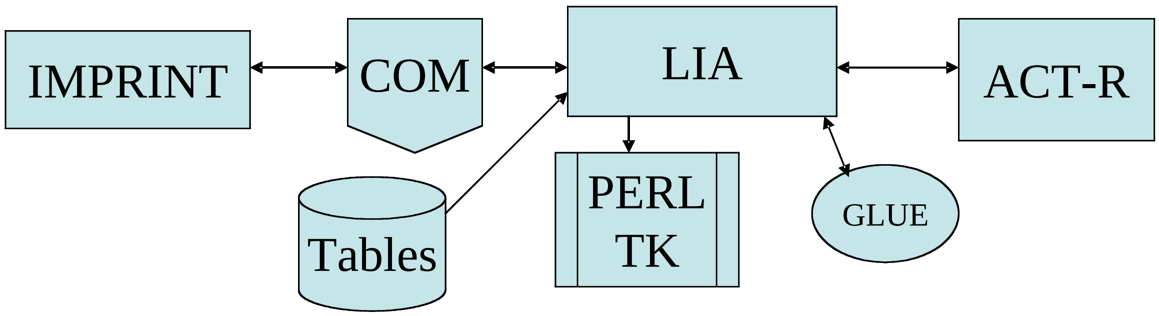

Figure 4: IMPRINT-ACT-R Federation

On this approach, IMPRINT and ACT-R relate on a level more like equal federates in an integrated simulation rather than the client-server relationship of task-level integration. Following the methodology of integrated simulation frameworks such as HLA, we developed a communication hub called Link IMPRINT-ACT-R (LIA) that provides a re-usable framework for federating IMPRINT and ACT-R models (see Figure 4). LIA facilitates data exchanges between IMPRINT and ACT-R and keeps them synchronized. LIA uses an External Model Call (EMC) protocol over a COM link for communication. Data exchanges are regulated by tables that specify the correspondence between variables and their types in the two models. A Perl/TK front-end can be used to display the state of the system. The entire communication system is re-usable across models except for a few lines of domain-specific "glue code," cutting down integration time from weeks or even months to just days.

3.2 Application to NASA Approach and Landing

Best et al. (2004) applied the "federated" modeling methodology described above to model commercial aircraft approach and landing scenarios and the tasks the pilots must perform. The federated model was used to compare pilot procedures using current technologies with procedures using augmented displays such as a synthetic vision system (SVS). The federated model represents an aircraft and its environment as well as the pilots operating the aircraft. For the simulation of the aircraft, the IMPRINT federate represents the autopilot as well as the physics of the aircraft. These aspects include the aircraft's location in time and space, its deceleration, descent rate, and all physical changes in the aircraft including its landing gear, flap settings and air brakes. The IMPRINT federate also includes the controls and displays of the aircraft including all autopilot functions, the mode control panels, the primary flight display, the navigational display, and an out-the-window view. Finally, the IMPRINT federate handles all communication between the aircraft and air traffic control. With these controls and displays, IMPRINT is able to simulate how a plane will react in its environment when these controls and displays are manipulated. The ACT-R federate was used to model pilot and was composed of a set of goals, together with the procedural and declarative knowledge necessary to solve those goals. The top-level goal is essentially a monitoring loop that repeatedly sets subgoals to check the settings of the various controls. Each of these subgoals typically requires acquiring the value of one or more environmental variables (e.g., speed, altitude, etc.) by reading the instruments or looking out the window, then making a decision as to whether to change the controls. The critical feature of the ACT-R federate is that it learns to make decisions regarding which control to monitor and which source (traditional control panel or SVS display) to get the information from based on its experience. This precluded a hardwired task network representation of the control loop and required the delegation of that function to the cognitive architecture. The ACT-R federate displayed the proper sensitivity to the frequency at which controls needed to be manipulated as well as the relative efficiency of the various sources of information. Using LIA, the IMPRINT model constantly provided updates as to the state of the aircraft and the controls, while the ACT-R model communicated its actions upon the controls. A graphical representation of the simulation could have been hooked into LIA to display the current state of the simulation.

4. Full Integration

4.1 Approach

The principal shortcoming of the previous approach is that, despite significant progress in automating the integration using LIA, it is still a relatively lengthy, duplicative and error-prone process. Two separate models have to be generated, often by two separate modelers since few modelers are trained in both task network and cognitive modeling, and then reconciled during the integration phase, which slows down the modeling process and has the potential to introduce conceptual mismatches that are very difficult to track down. It would be desirable for both productivity and conceptual clarity to develop both models within an integrated framework. The two approaches developed previously, i.e., task-level integration and event-level integration, provide the groundwork for a fully integrated model. But for full integration, the models must also be unified within a conceptual framework that relies on the same concept. That final unification consists in applying the task network paradigm all the way down to the level of atomic cognitive steps. At first glance, task networks and production systems (which constitute the symbolic basis of ACT-R) appear to have little in common. Production systems consist of condition-action rules that test the current state of the system, find the best-matching rule, fire its actions, then repeat the process in a very loose control structure that appears to have nothing in common with rigid task networks. But it turns out that just as ACT-R has evolved to become based more on events than on the production cycle, it has also evolved more of a control structure. State slots in the goal and other buffer indicators often serve the purpose of enforcing a definite control structure upon the workings of the production system. That structure, however, while quite explicit in the head of the modeler is often implicit in the conditions and actions of the production rules, leading to difficult and unproductive engineering of the rule set to conform to the internal conception of the modeler. The final unification principle between IMPRINT and ACT-R is to apply the same task network decomposition methodology to represent sequential control structure at the level of individual cognitive actions.

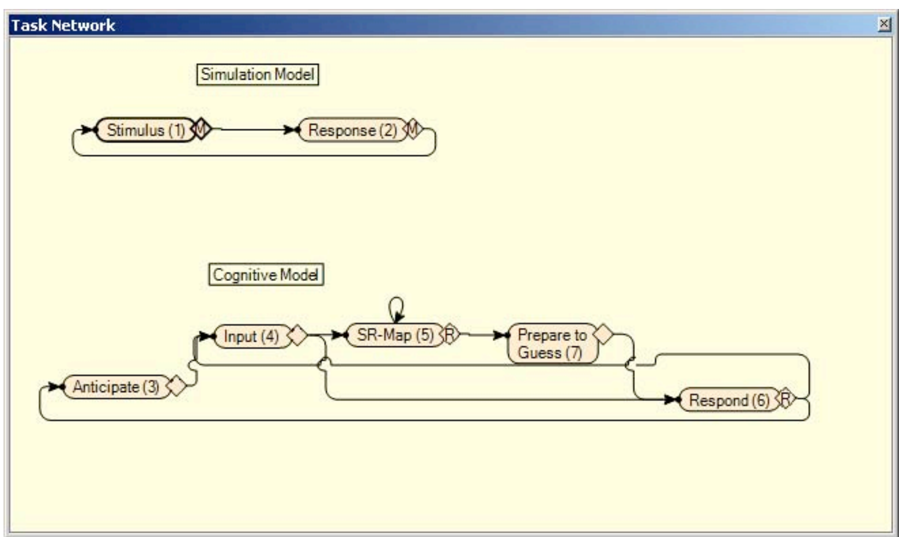

Figure 5: Fully Integrated IMPRINT-ACT-R Model of Sequence Learning

4.2 Application to Sequence Learning

Figure 5 presents a graphical representation of what a fully integrated ACT-R-IMPRINT model would look like (see Lebiere et al (2004) for details of the proposed approach). The task is sequence learning, a cognitive psychology paradigm in which subjects implicitly learn the locations of a series of events on the screen. The very simple two-task network at the top of the screen is sufficient to generate any sequence of events. It should be noted that this would traditionally be represented as carefully engineered Lisp code interacting with the ACT-R model, a process requiring both expert programming skills and careful integration by hand. The simulation network exchanges events with a network representing the cognitive model (bottom of figure). That network essentially represents each production as a task, with the logic behind the crafting of production rules made explicit in the chaining of the production rules. This representation has the advantage of corresponding directly to the modeler's mental model, with the additional benefit of preventing many errors by automatically supporting the basic control constructs. For instance, each memory retrieval will automatically invoke a test that ensures no condition (e.g., failed retrieval) goes unsupported, a common source of problems that leads ACT-R models to get stuck and spin forever in a behavioral black hole. It should be emphasized that despite its appearance as a task network, the cognitive model is still represented underneath by the very same ACT-R model, thereby preserving the established validity of the cognitive architecture.

5. Conclusion

This paper describes attempts at integrating the IMPRINT task network modeling tool and the ACT-R cognitive architecture. Principles underlying that integration including task-goal equivalency, a common event-based framework, and an extension of the task network graphical representation to cover production systems. Each integration effort can be viewed as gradually increasing the degree of integration, garnering additional advantages while preserving existing strengths. A natural implication for the integration of modeling and simulation systems, especially in the cognitive domain, is that the deepest integration is also the most powerful.

References

Anderson, J. R., & Lebiere, C. (1998). The Atomic Components of Thought. Mahwah, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Lebiere C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38,* 341-380.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . (2004). An integrated theory of the mind. Psychological Review 111, (4). 1036-1060.

Archer, S. G. & Adkins, R. (1999). Improved Performance Research Integration Tool (IMPRINT) Analysis Guide. Army Research Laboratory Technical Report, Aberdeen Proving Ground, MD.

Best, B., Lebiere, C., Schunk, D., Johnson, I. & Archer, R. (2004). Validating a Cognitive Model of Approach based on the ACT-R Architecture. Micro Analysis and Design Technical Report.

Byrne, M. D., & Anderson, J. R. (2001). Serial modules in parallel: The psychological refractory period and perfect time-sharing. Psychological Review, 108, 847-869.

Card, S.K., Moran, T.P., Newell, A. (1983). The psychology of human computer interaction. Hillsdale, NJ: Lawrence Erlbaum.

Hooey, B. L & Foyle, D. C. (2001). A post-hoc analysis of navigation errors during surface operations: Identification of contributing factors and mitigating strategies. In *Proceedings of the 11th Symposium on Aviation Psychology*. Columbus, Ohio: Ohio State University.

Kuhl, F., Weatherly, R., & Dahmann J. (1999). *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. NY, NY: Prentice Hall.

Laughery, K. R., Lebiere, C. & Archer, R. (in press). Modeling human performance in complex systems. In Salvendy, G. (Ed.) *2005 Handbook of Human Factors*.

Lebiere, C. (2001). A theory-based model of cognitive workload and its applications. In Proceedings of the 2001 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC). Arlington, VA: NDIA.

Lebiere, C., Biefeld, E., Archer, R., Archer, S., Allender, L., & Kelley, T. (2002). IMPRINT/ACT-R: Integration of a task network modeling architecture with a cognitive architecture and its application to human error modeling. In Proceedings of the Advanced Technologies Simulation Conference. San Diego, CA.

Lebiere, C., Schunk, D., Archer, R. & Warwick, W. (2004). Human Behavior Architecture Interface for Integrated Cognitive and Task Performance Model Development. Micro Analysis and Design Technical Report.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In Miyake, A. & Shah, P. (Eds.) Models of Working Memory: Mechanisms of Active Maintenance and Executive Control. New York: Cambridge University Press.

McCracken, J.H. and Aldrich, T.B. (1994). Analysis of Selected LHX Mission Functions: Implications for Operator Workload and System Automation Goals. Technical Note ASI 479-024-84(B) by Anacapa Sciences, June 1984.

Taatgen, N. A. (2001). A model of individual differences in learning Air Traffic Control. In Proceedings of the Fourth International Conference on Cognitive Modeling, pp. 211-216. Mahwah, NJ: Lawrence Erlbaum Associates.

Wermter, S., & Sun, R. (2000). *Hybrid Neural Systems*. Springer-Verlag, Heidelberg.