Cognitive Constraint Modeling: A Formal Approach to Supporting Reasoning About Behavior

Andrew Howes (HowesA@cardiff.ac.uk)
School of Psychology, Cardiff University, Cardiff, Wales, UK CF10 3YG

Alonso Vera (avera@mail.arc.nasa.gov)
NASA Ames Research Center, MS 262-3 Moffet Field, CA 94035

Richard L. Lewis (rickl@umich.edu)
Department of Psychology, University of Michigan, Ann Arbor, MI 48109-1109

Michael McCurdy (mmccurdy@arc.nasa.gov)
NASA Ames Research Center, MS 262-3 Moffet Field, CA 94035

Abstract

Cognitive Constraint Modeling (CCM) is an approach to reasoning about behavior that (1) provides a framework for investigating the hypothesis that skilled behavior is the optimal solution to a constraint satisfaction problem defined by objective, environmental, knowledge, and architectural constraints, (2) derives predictions of behavior from formal specifications of theory, (3) supports reasoning using both dependency-based and cascade-based ontologies for expressing temporal relationships between processes. A software tool that demonstrates the potential advantages of CCM is described. The tool, called CORE, can be used to partially automate the generation of behavioral predictions given a specification of the constraints. We explore the application of CORE to dual-task data previously modeled with EPIC and ACT-R.

Introduction

When people acquire a skill they are able to adapt their behavior so as to incrementally improve the value of some utility function. With practice, the scope for improvement attenuates and performance asymptotes. It may asymptote at a level that is consistent with constraints imposed by the environment or perhaps at a level determined by the knowledge that is brought to the task. The bounds may instead be imposed by the human cognitive architecture. More plausibly, the asymptote may be determined by a combination of constraints, including the stochastic and temporal profiles of the task environment *and* the human cognitive, perceptual, and motor systems. The approach to the asymptote is bounded by a multiplicity of constraints (Simon, 1992).

There has of course been much work aimed at modeling skilled behavior and its acquisition (e.g. Anderson and Lebiere, 1998; Meyer and Kieras, 1997; Taatgen and Anderson, 2003). The purpose of the current paper is to provide an initial demonstration of how models of skilled behavior can be generated by the formal derivation of behavior descriptions from multiple constraints, and in particular, how this approach supports reasoning about asymptotic bounds on skilled behavior. The specific objectives of the paper are:

(1) To introduce the hypothesis that skilled behavior is the optimal solution to a constraint satisfaction problem defined by architecture, task environment, and knowledge constraints.

(2) To introduce a formal modeling approach, called CCM, that directly supports reasoning about the optimal bounds on skilled behavior. By using deductive inference and constraint satisfaction algorithms, CCM computes the necessary consequences of the constraints imposed by the task environment, by strategic knowledge, and by the cognitive architecture. These constraints may determine, for example, which cognitive and environmental processes can execute in parallel and which have sequential dependencies.

(3) To specify two ontologies which provide alternative information processing vocabularies for the cognitive and task theory, and the resulting descriptions of behavior. The first is a straightforward formalization of temporal dependencies, implicit in existing work based on CPM-GOMS. The second is a richer ontology that permits specifying sets of communicating information processes, where both the processes, inter-process communication channels and buffers are subject to resource constraints. This framework has much in common with McClelland's cascade model (McClelland, 1979). Both ontologies are formally defined by a set of declarative axioms that are part of the model specification.

The paper has the following structure. We first introduce the background to our work on CCM and then describe a reasoning tool, called CORE (Constraint-based Optimal Reasoning Engine). We describe the application of CORE, using the temporal dependency axioms, to reasoning about a dual task experiment first reported by Schumacher, Lauber, Glass, Zubriggen, Gmeindl, Kieras, Meyer (1999) and subsequently modeled by Byrne and Anderson with ACT-R/PM (Byrne and Anderson, 2001). In doing so we show that CORE is flexible enough to support inference about the implications of both central and peripheral bottleneck theories of dual task performance. CORE requires 42 simple, universally quantified, declarative statements to

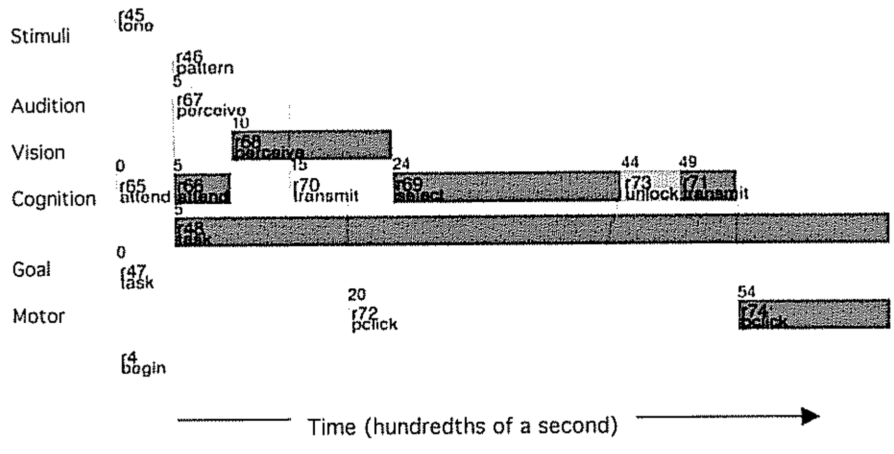

Figure 1: A CCM-d prediction of performance on Schumacher et al 's (1999) Experiment 3 task

specify the task, strategies, architecture, and axioms required to reason about the dual task We also report that the optimal schedule for Schumacher et al 's task suggests that participants may have been using a strategy not previously considered Lastly, we introduce, and describe the benefits of, the cascade-based axioms

Background

Predicting how long it will take people to perform a task is difficult, but important It is difficult because human performance depends on a multiplicity of complex interacting constraints that derive from the environment, from human psychology, and from knowledge that people bring to the task Skilled performance of a routine task usually involves the execution of a number of parallel but interdependent streams of activity: For example, one hand may move to a mouse; while the other finishes typing a word; and the eyes begin to fixate on a menu while the required menu label is retrieved from memory Each of these processes takes a few hundred milliseconds, but together they form behaviors that take many seconds. Importantly, the details of how processes are scheduled, of how they are ordered, and of the implications of their inter-dependencies, has significant consequences for the overall time requirement

There are a number of scientific and engineering tools that support the prediction of skilled performance time Many of these tools share a common intellectual origin in the Model Human Processor (MHP: Card, Moran & Newell, 1983) Card et al introduced the MHP as an engineering model in which human cognition was described as a set of communicating processors each of which had parameters (e g for cycle time) derived from human experimental psychology More recent engineering tools, particularly CPM-GOMS (Gray, John, Atwood, 1993), have also utilized the processor and process framework EPIC, a production system architecture that synthesizes more recent results in cognitive and perceptual psychology (Meyer and Kieras, 1997) was also influenced by the work of Card et al (1983) Most recently, ACT-R/PM (Byrne and Anderson, 1998) extended the ACT-R architecture with a set of EPIC-like perceptual-motor modules

The strength of these strands of work is evident in the range of experimental findings for which explanations can be offered; for the successful efforts at delivering scientifically validated tools to applied practitioners (e g Gray, John, Atwood, 1983); and for the rigor that is evident in the insistence that theory be expressed computationally (e g Byrne and Anderson, 2001; Meyer and Kieras, 1997) However, there are issues Below we have listed three that were significant in motivating the work reported in this paper

(1) It is difficult to inspect and modify architectural assumptions (Cooper and Shallice, 1995) Cognitive architectures embody architectural assumptions in underlying code, and are not easy to change This would not be a problem if the details of an architectural theory were stable and comprehensive enough to be applied to a wide range of tasks. But in the forseeable future the modeler will find it valuable to easily manipulate and add architectural assumptions that are still under debate in the field

(2) Model predictions can be a function of theoretically-irrelevant or implicit assumptions Current approaches force modeling at certain fixed levels of abstraction In general, computational cognitive architectures force computational completeness in order to incrementally simulate behavior But one consequence is that modelers must specify the details of procedural knowledge and the representations used in long term and short-term memory, which may not be intended as theoretical commitments

(3) It is difficult to predict the asymptotic bound on skilled behavior Though learning architectures such as ACT-R/PM could in principle automatically asymptote to the appropriate skilled behavior, the mechanism is an open research problem and puts a robust learning theory on the critical path to efficiently modeling skilled behavior

CORE: A tool to support reasoning about behavior

CORE takes as input a set of mathematically stated constraints on behavior and outputs a prediction In this respect it shares some similarities to the work of Duke and Duce (1999). One of the formats for the output, a CPM-GOMS-like Pert chart, is illustrated in Figure 1 The prediction in the Figure is for a dual task behavior studied by Schumacher et al (1999) and subsequently modeled in ACT-R/PM by Byrne and Anderson (2001) Each box represents a process Task 1 (light gray processes) is to respond to a tone (high or low) with a left-finger key press, and task 2 (dark gray) is to respond to a pattern with a right-finger key press In the Figure, time is represented on the horizontal access and each row represents a different resource or processor, perception at the top, through cognition and goal, to motor actions The task processes represent the temporal extent of the representation of each task on the goal.

Following Card et al (1983) constraints are described in terms of the temporal and resource properties of a distributed set of processors, each with its own processing capabilities. Each processor is defined in terms of a set of parameters and a defined set of processes Each process has parameterized limits (min, max) on its duration The duration of motor movements can be automatically determined by a calculation of Fitts's Law The duration of cognitive and perceptual processes may be directly determined from the empirical literature (e g estimates of the time required to switch attention), or by functions that, for example, model hypotheses about the behavior of retrieval mechanisms.

The relationships between the processes represented in Figure 1 are an attempt to reproduce the assumptions adopted by Byrne and Anderson (2001). The processing sequence for each stimulus is: attend to the stimulus, perceive the stimulus, select a response, transmit a command to the motor system. The duration of the select process is determined by ACT-R/PM's retrieval function

The lines between processes in Figure 1 represent temporal *dependencies*. A dependency is a type of constraint that specifies that one process must be scheduled after another has finished While the particular prediction illustrated in Figure 1 has been constrained by dependencies, cognitive constraint modeling is not limited to dependency-based representation of theories The constraints that specify the meaning of dependencies are the essence of a set of CCM axioms that we call CCM-d (CCM-dependency). CCM-d provides a formal specification of the CPM-GOMS modeling framework (Vera et al., 2004) (An alternative set of axioms, called CCM-c, for CCM-cascade, is described later in the current article)

Representing a theory

Constraints on behavior are specified to CORE in terms of relationships between events in the environment, tasks, and psychological processes

The semantics of the language for expressing statements is a subset of second-order predicate calculus An entity is represented as a set of elements where each element is either an ordered pair, or a triple where the first element is '++'. For example, the following reads, there exists a cognitive process called initclick that must be scheduled after process U_j

$$\exists P_i \ \{ (isa,process)\ (name.initclick)\ (resource,cognition)\ (++.after.U_j) \} \subseteq P_i \qquad (1)$$

Each pair consists of an *attribute* and a *value*. A set must only contain a single element with a particular attribute (e g there must only be, at most, one pair that matches the pattern (name,_)) Each triple consists of the symbol '++', an attribute, and a value For triples, there are no restrictions on the attribute or value Triples support the expression of sets in which an attribute can have multiple values The features in (1) are specified as a subset of P_i ($\subseteq$) Further features may complete the specification of this process

Sets that represent processes, must have a start attribute and a duration attribute This can be represented with the statement that all P_is that contain the pair (isa,process), must also contain a start time S_i and a duration D_i

$$\forall P_i \ \{ (isa,process) \} \subseteq P_i \rightarrow \{ (start.S_i).\ (duration,D_i) \} \subseteq P_i \qquad (2)$$

Relationships between the start times and durations of processes are represented with simple integer-arithmetic constraints The following represents the assumption that a motor process is a necessary consequence of an initialization process, that a motor process cannot occur before its initialization process, and that the maximum temporal gap between the two processes is 300ms This constraint must hold irrespective of the task

$$\begin{aligned}\forall P_j \ &\{ (isa,process)\ (name,initclick)\ (start,S_j)\ (duration,D_j) \} \subseteq P_j \\ \rightarrow\ &\exists P_i.\ \{ (isa,process)\ (name.click)\ (start,S_i) \} \subseteq P_i \\ &\wedge S_j + D_j \leq S_i \\ &\wedge S_i - (S_j + D_j) \leq 300 \qquad (3)\end{aligned}$$

Given a set of axioms, statements of this form can be used to represent theoretical assumptions about the task environment, about instruction taking, about the strategies that people deploy, and about the human cognitive architecture

Crucially, universally quantified constraints specified in a predicate calculus are not production rules The constraints may appear to possess a similar surface form to production rules but, in fact, the semantics are very different Most importantly, unlike production rules, these declarative statements of theory are statements of what *must* be true irrespective of context They are not elements of a procedure that generates the description. The constraint must hold for every circumstance where its antecedent is met. The generation of a model with these constraints is entirely monotonic and the order of expansion can be (and often is) different to the predicted order of behavior.

Generating a prediction

Given constraints on behavior, CORE can be used to generate a prediction. This is a two-phase process.

Phase 1. CORE derives the necessary implications of the theory. For example, given a P_i (as defined in statement 1) above and rule (2), CORE would derive that the initclick process must have a start and duration:

$$\exists P_j . \{ (isa, process)\ (name, initclick)\ (resource.cognition)\ (++.source, U_j)\ (start, S_j)\ (duration, D_j) \} \subseteq P_j (4)$$

Subsequently, with rule (3), CORE can derive that there must be a motor click process, with a start time and duration constrained by the given equations.

Arithmetic constraints on the start time, duration, and costs of a process are posted to a constraint store that is implemented in a Sicstus Prolog variant of Constraint Logic Programming for Finite Domains (CLP FD: Jaffar & Lassez, 1987). Much of the power that CORE provides is a direct consequence of CLP FD functionality (Vera et al., 2004). Importantly, the scheduling algorithms provided by CLP FD make it possible for an analyst using CORE to focus on the declarative specification of theory without worrying about the theory-irrelevant algorithms by which behavioral implications will be derived.

At the end of phase 1 the values of the start, duration, and other parameters, such as cost, are constrained by the posted equations, but their values are not yet uniquely determined.

Phase 2. Phase 2 involves making a prediction by finding a particular behavior that is consistent with the set of constraints, i.e. phase 2 must identify a consistent set of values for variables that were posted to the CLP FD constraint store (e.g. start time, duration, cost). This is achieved by calling a function that uses constraint satisfaction to achieve variable assignment. This function can be configured to use a range of scheduling algorithms. Two are particularly important for the purposes of reasoning about cognition: greedy scheduling and optimal scheduling.

Greedy scheduling. Scheduling proceeds with the tick of a clock. On each tick, a process is selected that can be scheduled immediately. The process is assigned the appropriate start time. Greedy scheduling can be used to model ACT-R/PM and EPIC. A greedy scheduling algorithm is not guaranteed to give an optimal schedule.

Optimal scheduling. Using CLP FD, a branch-and-bound algorithm can be used to generate a schedule with the greatest utility. We have used a utility function that is maximal when cost is minimized. Cost is defined as the sum of the total duration of the schedule and the durations of the buffers required to store information. As we illustrate below the ease with which CORE can be used to generate predictions of the optimal behavior, given the theoretical assumptions, is one of its key advantages.

Reasoning about dual task performance

In Schumacher et al.'s (1999) experiment (Experiment 3) participants were required to respond to a tone and a visual pattern with key presses that depended on whether the tone was high or low and whether the pattern contained a particular feature. The tone and the pattern were presented with a small gap of between 50 and 1000ms (Stimulus Onset Asynchrony). Participants were asked to prioritize the tone task. The tone task response times were, on average, unaffected by SOA. In contrast, the mean pattern task response time, at a short SOA (50ms), was less than the sum of the tone task and pattern response times at long SOAs (> 500ms). This finding has been taken as evidence that some elements of tone and pattern task were performed in parallel at short SOAs. Byrne and Anderson were interested in modeling Schumacher's data using ACT-R/PM in order to demonstrate that cognitive parallelism is not required to explain these results. They argued that the results can be modeled with either the EPIC or ACT-R/PM assumptions and that Schumacher's data provides evidence for strategic deferment of the pattern task response.

Specification and Inference

We demonstrate that the theoretical assumptions of Byrne and Anderson (2001) and separately of Schumacher et al. (1999) can be precisely expressed as a small set of predicate calculus constraints, and that CORE can be used to support reasoning about their behavioral consequences.

We used 42 universally quantified constraints to capture the theoretical assumptions underlying the architecture and strategies deployed in Byrne and Anderson's model (see www.cf.ac.uk/psych/howesa/ccm). Together with the CCM-d axioms, these constraints are a mathematically-complete specification of the theory underlying Figure 1. They capture constraints on the task environment, the strategy, the architecture, and in addition the axioms of CCM-d (It would in fact be possible to use many fewer constraints but we attempted to write them in a way that was general enough to enable reuse.) We selected parameters to fit the performance time and ran the model with a greedy scheduling algorithm to check its performance. It produced the same pattern of results as reported by Schumacher and modeled by Byrne and Anderson (2001).

One of the constraints that is particularly important for the predictions made by Anderson and Byrne's model states that the duration of retrieval is sensitive to whether the retrieval request is issued when the tone task overlaps in time with the pattern task. This relies on an implementation of the ACT-R retrieval time function, *retrieval_time(B,S,T)*, where B is the base level activation, S is the source activation, and T is the returned retrieval duration. The source activation is lower per task when there are multiple concurrent tasks.

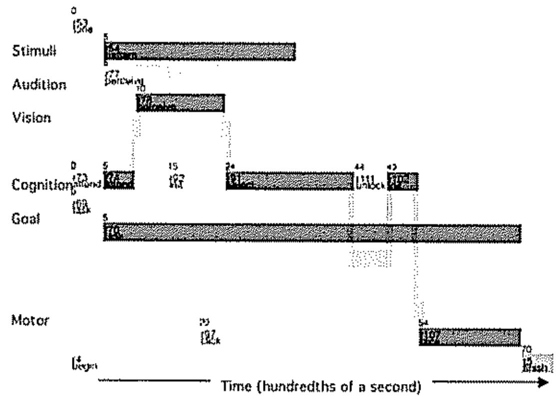

Figure 2: A CCM-c prediction of performance on Schumacher et al.'s (1999) experiment 3 task.

Subsequently, we modified the specification in order to remove the ACT-R/PM assumption that there is a central bottleneck on human cognition and permit EPIC-like concurrent cognitive processing. Only a handful of changes were required to make this alteration, demonstrating the claim that CORE facilitates reasoning about the consequence of different architectural assumptions. First, the tone task and pattern task cognitive processes were assigned to separate resource streams, and second, an unlock process was introduced and its duration adjusted to fit the data.

The fact that ACT-R and EPIC resource assumptions can be captured with such similar sets of constraints is unsurprising given their shared intellectual history (though it would be interesting to compare the Lisp code). In general, the space of theories that can be represented with CCM is determined by the requirement that theories are described in terms of processes, processors, the relationships between them, and by the axioms (CCM-d or CCM-c).

Skill and optimal constraint satisfaction

In order to explore the hypothesis that skilled behavior is the optimal solution to a constraint satisfaction problem, we switched from greedy scheduling to optimal scheduling (a parameter change in the input specification). CORE generated a novel prediction. In the behavior for a task with a 50ms SOA and a simple pattern, rather than choose to schedule the pattern selection process so that it was concurrent with the tone task, CORE chose a schedule in which selection (i.e. retrieval) is deferred until after the tone task has finished. The benefit is that by deferring selection, the overall time requirement is slightly reduced. This is because even though the selection process starts later (after the tone click in Figure 1) it has a much shorter duration (only 30ms compared to the 250ms). The resulting difference in the overall time cost of the schedules is marginal but the qualitative difference in the strategies is dramatic. The analysis exposes a necessary consequence of ACT-R's retrieval function and the assumption that people adapt strategies to reduce time cost.

The example illustrates the way in which CORE can facilitate the exposure of a logically required implication of a set of theoretical assumptions. Byrne and Anderson's ACT-R model does not make this prediction because it does not optimize over the total cost of the behavior. Optimal scheduling exposes the possibility that participants strategically defer retrieval so as not to incur the costs of concurrent processing.

Our analysis also raises a question about a fundamental assumption embedded in the ACT-R retrieval function: That retrieval time is not dynamically adjusted with changes in source activation occurring during retrieval. I.e. the sensitivity of the retrieval time function to source activation is limited to the value of the source activation at the time of the retrieval request. An alternative assumption would be that retrieval could take advantage of increases in source activation that occur after a retrieval request is made but before a chunk is delivered. With this alternative assumption, deferred retrieval in the Schumacher task would carry no advantage. Which assumption provides a better model of human retrieval is an empirical question that is not answered by Schumacher's data.

Cognition as cascading information processing

CORE is flexible enough to accept theory specifications expressed relative to a range of different sets of axioms. For the work reported above we used axioms that were based on the notion of a dependency (CCM-d). However, there are intrinsic limitations of dependency-based axioms (Vera et al., 2004) and we have therefore been working on the specification of a set of axioms that is based in part on the idea of a cascade as a mechanism for representing overlapping, communicating processes. Our formalization builds on McClelland's (1978) original cascade assumptions to explicitly include the declaration of resource-limited communication channels and buffers between processes.

Figure 2 illustrates a prediction derived by CORE using CCM-c axioms. The start times and durations of the processes are the same as in Figure 1. The difference between the figures is that the relationships between processes are expressed in terms of cascades. These capture the resource requirements and temporal constraints on the inter-process communication channels. The cascades are represented in Figure 1 as the lightest gray bars that run between the processes. One advantage of cascades is that they prevent cognitively implausible process orderings that would be legal using CCM-d. For example, the process ordering init(x), init(y), click(y), click(x) is legal in CCM-d but cognitively implausible because it assumes no cost to buffering information between the cognitive intention and the motor action.

Discussion

We have introduced a framework and a tool for making inferences about the implications of formally specified theories of skilled behavior. The tool uses a constraint logic-programming environment to support the inference of the asymptotic bound on skilled behavior given a specification of the constraints on the task environment, on perception, on cognition, and on action

Our investigations are at an early stage. We have so far explored the potential of dependency and cascade axioms on only a handful of tasks. In addition to the dual task described in the current paper we have also used CORE to generate predictions for a range of applied tasks including a call-center accounts advise task, and a laboratory version of an Automated Teller Machine.

Our aim in conducting this work was not to recast ACT-R/PM and EPIC in a formal language. The aim was to provide a tool that could assist in the prediction of the asymptotic bound on skilled behavior given constraints on, not only, objective and environment but also on strategies and architecture. We concur with Simon (1992) that an analysis of the optimal adaptation given all of these sources of constraint provides a more accurate estimation of behavior. While the extent to which we can achieve our aim is yet to be determined, we have presented arguments for the scientific merit of deductive inference in exploring the asymptotic bound on skilled behavior. We have shown that by deriving the optimal schedule of behavior for these constraints, logically implied but previously unexplored predictions of behavior can be exposed. The fact that a novel prediction was generated for a task that has been the subject of a number of published studes illustrates that the benefits of cognitive constraint modeling go beyond redescription of existing theory.

One potential concern is that if we were to write a set of constraints to capture the range of behaviors exhibited by, for example ACT-R, we would generate a set that was as large and formidable as ACT-R's Lisp code. Our response is twofold. First, we note that CCM is not a simple subset of ACT-R, it includes functionality, particularly optimization, that is not present in simulation architectures. Second, we point out, again, that our aim is not to recast ACT-R or EPIC in a formal language. More particularly, our aim, at present, is not to build a simulation architecture, rather it is to provide a tool for supporting reasoning about psychological theory. Much of the complexity of the ACT-R and EPIC implementations may be related to the simulation-based framework in which they are cast.

Our current work is aimed at further developing the generality of CORE. Most importantly, we need to take full advantage of the constraint satisfaction engine, CLP FD, that is used for the calculation of arithmetic parameters. In the present implementation of CORE, this engine is not used to reason about the symbolic inter-process constraints. We also need to work on using constraint satisfaction techniques that support reasoning about statistical distributions rather than just integer values.

In conclusion, we have introduced a constraint-based framework for reasoning about human behavior and argued for the utility of a specific tool called CORE. Our investigation suggests that partially automatic algorithms can be used to generate predictions of optimal human behavior from concise, theory-relevant, and readily modifiable, specifications of psychological theory.

Acknowledgments

This work was supported by ONR Grant number N00014041P2002. Bonnie John, Mike Byrne, and Duncan Brumby made a number of valuable comments on this work.

References

Anderson, J.R. (1990). *Rational Analysis*. LEA.

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

Byrne, M.D. & Anderson, J.R. (1998). Perception and action. In J. R. Anderson and C. Lebiere (Eds.) *The Atomic Components of Thought* (pp. 167-200). Mahwah, NJ: Erlbaum.

Byrne, M.D. & Anderson, J.R. (2001). Serial modules in parallel: The psychological refractory period and perfect time sharing. *Psychological Review, 108*, 4, 847-869.

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

Cooper, R. & Shallice, T. (1995). Soar and the case for unified theories of cognition. *Cognition, 55*(2), 115-149.

Duke, D.J. and Duce, D.A. (1999). The formalization of a cognitive architecture and its application to reasoning about human computer interaction. *Formal Aspects of Computing, 11*, 665-689.

Gray, W.D., John, B.E. and Atwood, M. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction, 8*, (3), 237-309.

Jaffar, J., & Lassez, J.L. (1987). Constraint logic programming. In *Proceedings of the ACM Symposium on Principles of Programming Languages*, ACM Press.

McClelland, J.L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86*, 287-330.

Meyer, D.E. & Kieras, D.E. (1997). A computational theory of human multiple-task performance: The EPIC information-processing architecture and strategic response deferment model. *Psychological Review, 104*, 1-65.

Schumacher, E.H., Lauber, E.J., Glass, J.M., Zurbriggen, E.L., Gmeindl, L., Kieras, D.E., & Meyer, D.E. (1999). Concurrent response-selection processes in dual-task performance: evidence for adaptive executive control of task scheduling. *Journal of Experimental Psychology: Human Perception and Performance*. *25*, 791-814.

Simon, H.A. (1992). What is an "explanation" of behavior? *Psychological Science, 3*, 150-161.

Taatgen, N.A. & Lee, F.J. (2003). Production Compilation: A simple mechanism to model Complex Skill Acquisition. *Human Factors, 45*(1), 61-76.

Vera, A., Howes, A., McCurdy, M., Lewis, R.L. (2004). A constraint satisfaction approach to predicting skilled interactive cognition. In *Procedings of the ACM Conference on Human Factors in Computing Systems CHI'04*, Vienna, April 24-29, 2004.