

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Information-Requirements Grammar: A Theory of the Structure of Competence for Interaction

Permalink

<https://escholarship.org/uc/item/9hn6k8bp>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 27(27)

ISSN

1069-7977

Authors

Howes, Andrew
Lewis, Richard L.
Richardson, Juliet
et al.

Publication Date

2005

Peer reviewed

Information-Requirements Grammar: A Theory of the Structure of Competence for Interaction

Andrew Howes (HowesA@Cardiff.ac.uk)

School of Psychology, Cardiff University, Cardiff, Wales, CF10 3AT, UK

Richard L. Lewis (rickl@umich.edu)

Department of Psychology, University of Michigan, Ann Arbor, MI 48109-1109

Alonso Vera (avera@mail.arc.nasa.gov)

Carnegie Mellon University and NASA Ames Research Center, MS 262-3 Moffet Field, CA 9403

Juliet Richardson (Juliet.Richardson@convergys.com)

Convergys, Westbrook Centre, Milton Road, Cambridge, UK

ABSTRACT

In this paper, we argue that existing languages for representing routine cognitive tasks (such as GOMS, UAN, and PDL) can fail either because they demand that task competence is described using serial position to determine temporal order (and they are therefore overly restrictive) or because they demand that partial orderings are specified with temporal dependencies and other logical relationships (and they are therefore under-constrained). We propose a novel task description language, called Information-Requirements Grammar (IRG), which is motivated by a theory of how higher-level task performance is constrained by the information requirements and resource demands of lower-level tasks. We demonstrate the use of IRG and show how it replaces serial ordering and temporal dependencies with resource-bound information cascades between architectural information processes.

Introduction

We propose a theory of competence for tasks that is called Information-Requirements Grammar (IRG). It is motivated by the assumption that constraints on tasks derive from their need for information. The assumption is not novel. In particular, Gray, John and Atwood (1993) emphasized the importance of information flow between processes in describing complex behaviors. What are novel are the implications that we draw from it. We argue that application of the theory to the modeling of task knowledge can solve two serious problems with established approaches, particularly GOMS (Card, Moran & Newell, 1983), but also UAN (Hartson & Gray, 1992), PDL (Freed, Matessa, Remington &, Vera, 2003) and similar scheduling-based languages.

Cognitive Modeling Approaches

Since Card, et al. (1983) there have been numerous advances in task knowledge modeling techniques. For our purposes, we broadly categorize these into two approaches

on the basis of how they determine temporal ordering: (1) extensions and uses of techniques such as GOMS (Card, et al., 1983) that use the serial position of operators in the task description to determine temporal order; and (2) techniques such as PDL and UAN that specify temporal dependencies between operators.

To illustrate these techniques we use a payment inquiry (PI) task. The task involves a customer contacting a call center to check whether his/her previous payment has been credited to an account (the task is similar to that studied in the seminal work by Gray, et al., 1993). To answer the inquiry the call center agent must find the customer's record in the system and then view the key details of the most recent payment. For this type of task, both categories of cognitive modeling approach have problems.

First, let us examine an example GOMS method rule for part of the PI task, namely entering the customer's postcode as part of the search criteria to find the customer's account:

Enter postcode →

- Step 1: request postcode,
- Step 2: listen for postcode,
- Step 3: select postcode field,
- Step 4: type postcode.

(1)

This method specifies that to achieve the 'enter postcode' goal, steps 1 to 4 must be conducted in order. The serial position of step 3 (between step 2 and 4) in the method description determines its temporal ordering in a behavior prediction. However, often the environment does not impose strict ordering constraints between components of the method. For example, with most devices step 3, select field, could in fact be the very first step in the "enter" method, it could also be executed in parallel with steps 1 and 2. The problem, as observed by Gray, et al. (1993), is that a GOMS method does not specify which components can be reordered; rather there is a universal assumption that temporal ordering will reflect serial position in the method.

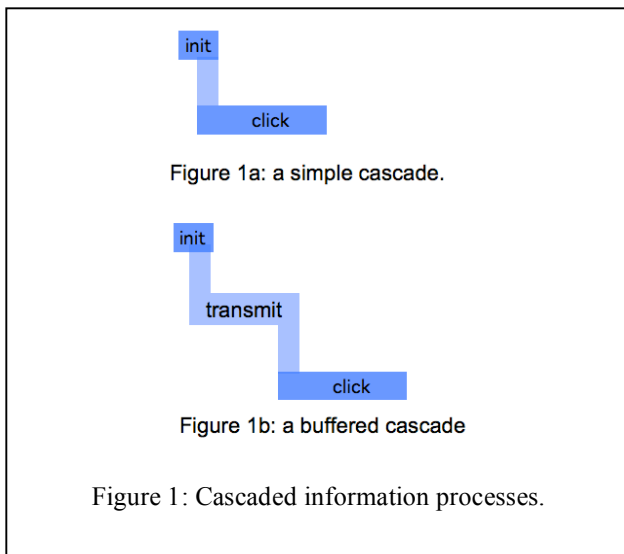


Figure 1: Cascaded information processes.

Methods, such as ACT-SIMPLE (Salvucci & Lee, 2003), GOMSL (Kieras, 1999) and G2A (St. Amant & Ritter, 2004), that rely on decomposing task descriptions into sequences of operators all suffer from this same problem of over-constraining the temporal order. They have limited or no scope for parallelization of operators.

In contrast, CPM-GOMS was developed specifically to model the information flow between tasks and the parallelization of activities (Gray, et al., 1993). Using CPM-GOMS some aspects of a theory of information flow could be imposed with temporal dependencies. However, the specification of the relationships between processes in terms of dependencies leads to under-constraint. Important aspects of the theory on which CPM-GOMS was based remained implicit. In particular,

1. The maximum duration of the gap between two processes (e.g., working memory bounds) cannot be specified.
2. It is difficult to specify constraints on whether a process can be scheduled between two other processes.

Vera, et al. (2004), describe how these problems can result in undesired processing schedules, with operators occurring in orders which are inconsistent with the theory, potentially leading to under-prediction of the task performance time. Similar approaches, such as UAN and ConcurTaskTrees (Paterno, Mancini, & Meniconi, 1997), suffer from similar problems. They under-constrain the temporal order. While constraints can be added to a particular model by the theorist, there is not a theory embedded within the structure of the language of the constraints imposed by the human cognitive architecture.

Information-Requirements Grammar

IRG is a grammar notation for representing hypotheses concerning competence for tasks. It is based on the following theoretical commitments:

- The execution of tasks is constrained by their *information and control requirements*.

- The performance of the component processes of a task and the transmission of information between them require resources and are subject to resource constraints.

The expansion of an IRG maps a task description into a set of processes and a set of constraints on inter-process information flow. Together, the processes and information flow constraints form a model of the processing involved when a person performs the task.

Before introducing the structure of IRG rules, we first describe what we mean by information flow constraints.

Information and Control Constraints

Tasks receive, transform, and transmit information. The availability of information constrains how and when a task can be conducted. For example, a call center agent cannot enter a value (such as the customer's address) into the system until he/she hears that address from the customer. In this case, typing the value is constrained by when it is heard. As a result, it is the flow of information that determines the temporal order of tasks. We refer to a task's need for information as an information constraint.

In addition to information constraints, tasks can be subject to control constraints. For example, a user may choose to delay typing into a field on the screen until he/she has observed that the cursor is in that field. In this case, typing the value is constrained by observation of the cursor state. Control constraints can be thought of as a special case of information constraints: a signal that a process has completed, or perhaps started, is required before another can proceed.

Competence in this view is knowledge of information and control constraints. This is a somewhat restricted view of competence, as it does not concern the content of the information transmitted. The point is that what it does include is just what is needed to figure out how to schedule processes given limited resources.

Resources and Resource Constraints

The transmission of information between processes imposes resource demands. Information can be transmitted from one process to another only if there is a physical substrate with which to carry and perhaps store that information. We assume that information transmission must be either between simultaneously instantiated processes or mediated by a buffer. In both cases we say that information is *cascaded* (a concept introduced by McClelland (1979) to explain speed-accuracy tradeoffs).

Our version of cascade theory commits to the following assumptions: Processes must overlap in time so as to transfer information. A process is executed by a processor (also known as a resource; a candidate set was proposed by Card, et al., 1983). A process has a minimum duration before it starts transmitting (incomplete) information and a duration by which time it is transmitting complete

effect ACTION after E1 and E2 : CTRL EFFECT → init ACTION after E1 and E2 transmit INIT in mb do ACTION on TRANSMIT hold MOTOR in mb display ACTION on MOTOR	- INIT, - TRANSMIT, - MOTOR, - CTRL, - EFFECT.
check ATTENDED to FIELD : RESULT → verify ATTENDED is FIELD hold VERIFIED in wm 1	- VERIFIED, - RESULT.
move_eyes_to FIELD : FIX → init eyes do shift_eyes on INIT hold SHIFT in vision	- INIT, - SHIFT, - FIX.
say STRING : MOTOR RESPONSE → init say transmit INIT in mb say1 STRING on TRANSMIT verbal response on MOTOR	- INIT, - TRANSMIT, - MOTOR, - RESPONSE.
hear RESPONSE : RESULT → attend auditory transmit ATTEND in ab perceive auditory RESPONSE on TRANSMIT hold PERCEIVED in ab	- ATTEND, - TRANSMIT, - PERCEIVED, - RESULT.
see EVENT at PLACE : RESULT → attend visual PLACE transmit ATTEND in vb perceive visual EVENT on TRANSMIT hold PERCEIVED in vb	- ATTEND, - TRANSMIT, - PERCEIVED, - RESULT.
see TARGET at fixation FX : RESULT → attend visual TARGET transmit ATTEND in vb perceive visual FX TARGET on TRANSMIT hold PERCEIVED in vb	- ATTEND, - TRANSMIT, - PERCEIVED, - RESULT.

Figure 2: IRG specification of mappings between Abstract Functional Operators and architectural processes.

information. It also has a maximum duration, by which time it is no longer executing and its results are not available. Some function relates the accuracy of information produced to the duration since the process started (Howes, Vera, Lewis & McCurdy, 2004).

The constraint imposed on process start times and durations by a cascade that transfers information from process i to process j , where i and j have start times S_i, S_j and end times E_i, E_j , can be defined as a pair of inequalities:

$$S_i < S_j < E_i \quad (2)$$

These constraints assert the need for overlap between i and j but without representing the speed-accuracy tradeoff functions. (The tradeoff functions are beyond the scope of the current paper.) Figure 1 illustrates two models that are consistent with this constraint. In Figure 1a, a cognitive init must overlap in time with the motor process that it causes. The period of time during which information flows from the init to the click is illustrated with a faded connector, representing the cascade, between them. In Figure 1b, the flow of information between the init and the click is mediated by a buffer, but the relationships between the init

and the buffer and the buffer and the click are each consistent with the cascade assumption.

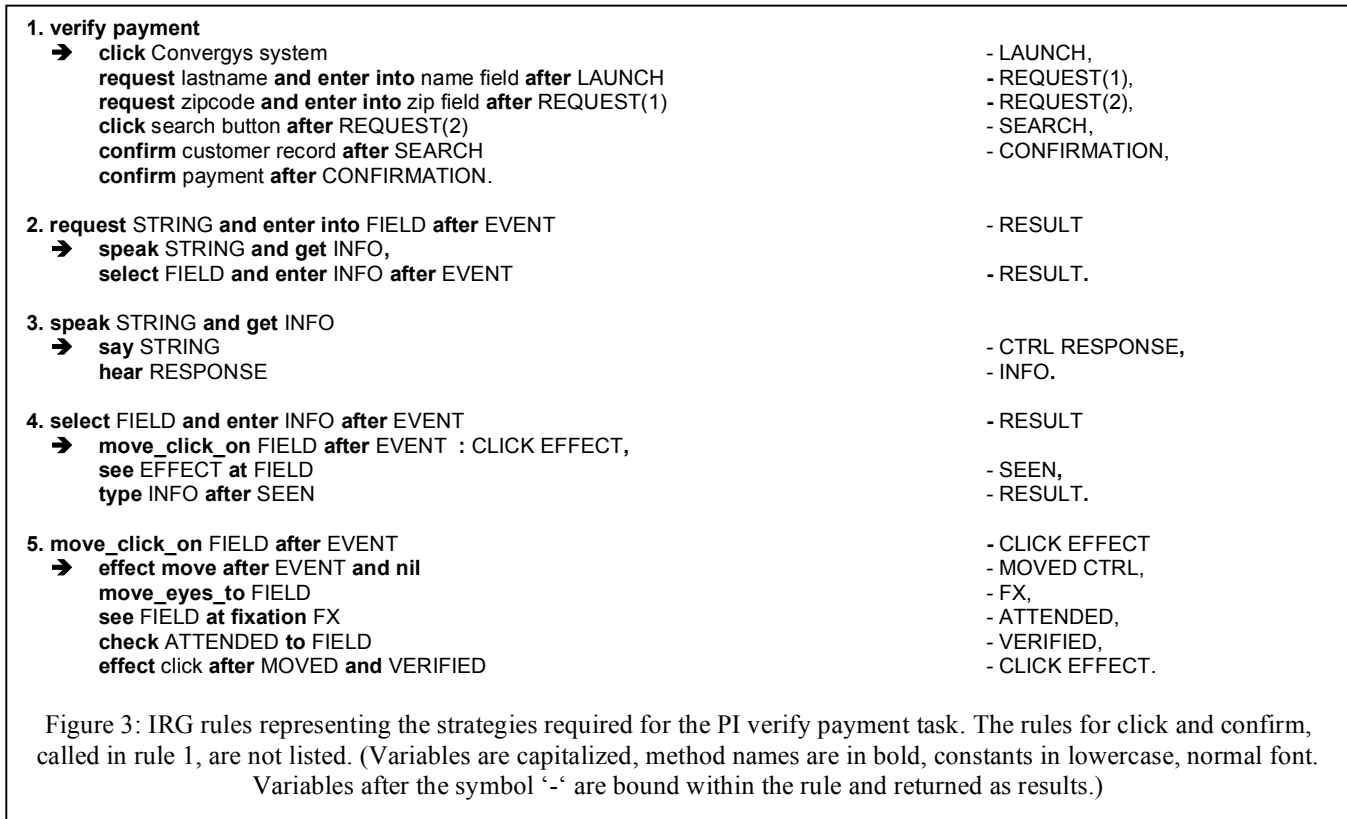
Importantly, the relationship defined by a cascade is the *only* temporal constraint between processes permitted in IRG.

IRG Rules for Primitive Architectural Operators

We refer to the buffer in Figure 1b as a transmit process. Together the three processes (init, transmit, and click) form a simple example of what we call Architectural Process Cascades (APCs). APCs model the fixed and immutable functionality provided by the cognitive architecture. APCs can be defined with IRG rules. Figure 1b is defined by the IRG rule:

$$\begin{aligned}
 &\text{click mouse} \\
 \rightarrow &\text{init} && - \text{INIT}, \\
 &\text{transmit on INIT} && - \text{TRANSMIT}, \\
 &\text{click on TRANSMIT.} && (3)
 \end{aligned}$$

The rule states that: the task “click mouse” can be expanded into three processes. Uppercase words are variables.



Variables after a minus sign are bound to the identifier of the process. A right-hand element of an IRG rule that refers to the identifier of another process receives information from that process. Rule (3) states that transmit requires information from INIT and that click requires information from TRANSMIT.

For every information flow defined in a rule, the IRG interpreter generates constraints of the form specified in (2). In the case of click mouse, the interpreter generated:

$$S_{init} < S_{transmit} < E_{init}$$

$$S_{transmit} < S_{click} < E_{transmit}$$

These constraints determined the temporal relationships between the processes. The fact that init, transmit, and click are ordered in the “click mouse” rule has no consequence for their temporal relationships. In fact the ordering in the rule could be different, perhaps [transmit, init, click] without consequence for the constraints generated from rule expansion (cf. GOMS).

Figure 2 presents seven rules describing a set of APCs for simple motor and perceptual operators, e.g. seeing, hearing, pressing a mouse button. Each APC (left-hand side of a rule) maps into a set of information-flow-constrained architectural processes (right-hand side). The processing commitments made in these rules concern hypotheses about the nature of the human cognitive architecture.

The specification of each APC includes not only the details of the information flows within the operator, as described

for click-mouse above, but also the information flows into and out of the structure. These information flows are represented in the parameters on the left-hand side of each rule. For example, the second rule “check ATTENDED to FIELD : RESULT” in Figure 2 takes information from an attentional process, bound to ATTENDED, and returns RESULT, which is the identifier of the “hold verified in wm” process.

The rules in Figure 2 are not pseudo-code; they are presented in the exact syntactic form required for input to a tool, described below, that given an IRG expands task descriptions into a set of processes representing cognitive, perceptual and motor behavior.

Task-Level Rules

In the previous section, we illustrated how IRG can be used to represent a theory of the temporal properties of a human cognitive architecture. That in and of itself is a potentially powerful tool, but here it is a precursor to our primary goal in this paper which is to demonstrate how to specify theories of the knowledge required to perform particular tasks and to thereby show how IRG solves the problems identified with existing task description languages.

Figure 3 shows the major part of the IRG specification of the PI task. The rules in the figure are hierarchically structured, such that method 1 describes the overall task, which decomposes into the lower-level methods represented by subsequent rules. The decomposition of the

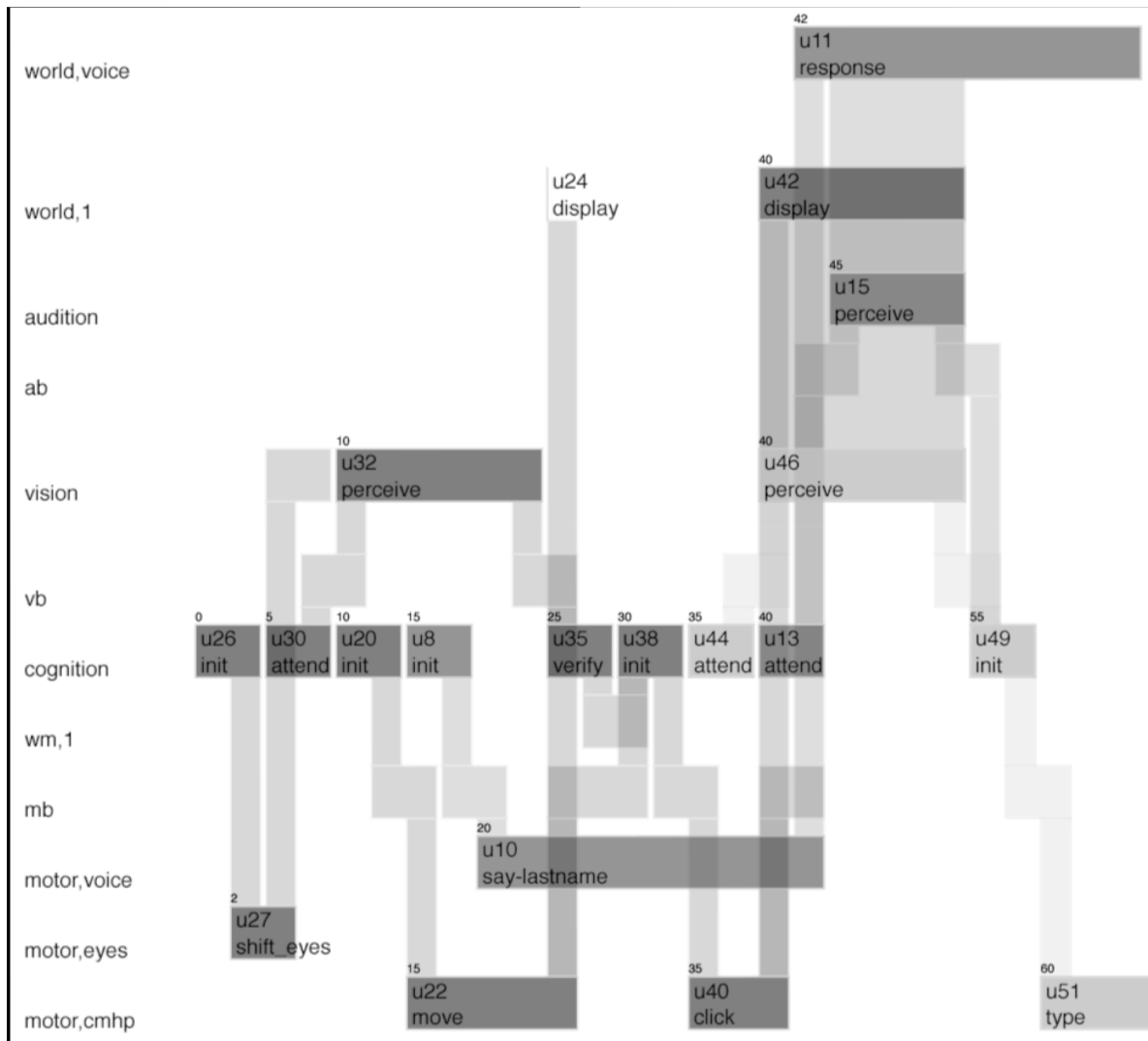


Figure 4: Part of the behaviour graph for the IRG model of the PI task, showing (from left to right) moving to and clicking on the name field (u22, u40), the request for the customer's name (u10), perceiving the feedback that the cursor is in the name field (u46), typing the name (u51). Many of the processes are conducted concurrently.

task continues until all tasks have been expanded into the cognitive, perceptual or motor processes and their cascaded information flows that are presented in Figure 2.

The methods in Figure 3 illustrate how higher-order tasks can be composed from subtasks, which are eventually composed of the APCs provided by the cognitive architecture (see Figure 2). At all levels, information flows between tasks are defined using the parameter passing mechanism described in the previous section.

Generating a Prediction

Using a tool called CORE (Howes et al., 2004; Vera et al., 2004; Lewis et al., 2004), temporal constraints generated through expansion of the IRG are posted to a Sicstus Prolog CLP FD (Constraint Logic Programming for Finite Domains) constraint store. They are elaborated with

constraints that determine typical durations for each process (e.g. as articulated in Card et al.'s Model Human Processor (MHP)). Constraint satisfaction is used to determine a prediction of the optimal adaptation to the task constraints. The output from CORE is a *behavior graph*, representing the start times and durations of each process (e.g. Figure 4). In Figure 4, each row represents a processor and each box represents a process. The names of each processor are shown on the left. The figure illustrates two key aspects of IRG.

1. *Cascaded information flows permit theory-congruent concurrency.* Serial order in IRG rules does not impose a temporal order. Although control constraints specified at the higher levels of IRG task descriptions (e.g., Figure 3) may look as though they define a strict linear sequence (as GOMS methods do), this is in fact not the case. For example, in Figure 4, it can be seen that the request for the

customer's last name (u8 & u10) is predicted to occur concurrently with the movement of the mouse to the name field (u20, u22, u38 & u40). The IRG specification ensures that process scheduling is consistent with information requirements but does not prevent the concurrent scheduling of what are otherwise autonomous processes.

2. *Cascaded information flows prevent theory-incongruent interleaving.* For example, in Figure 4 there are five init processes each with a corresponding motor process. The motor processes are predicted to be scheduled in the same order as the cognitive init processes. The init processes transmit information to the motor processor, through a cascaded buffer. The fact that the buffer resource can compute only one process at a time, combined with the need to receive and send information, ensures the consistent ordering of the cognitive and motor processes. In contrast, with a language that requires temporal dependencies between each init and its corresponding motor process no systematic relationship between the order of the inits and the order of the motor processes is imposed.

Discussion

We have described a theory, called IRG, of the structure of competence for interactive task performance. The theory addresses shortcomings with existing task description languages, which are either too restrictive (e.g., GOMS) or under constrained (e.g., UAN, CPM-GOMS, PDL). IRG demands specification of the information requirements of each task in the hierarchy. Expansion of the hierarchy and deduction of the optimal strategy given the cascade-based constraints results in the generation of a prediction of the time-course of interactive performance.

A second contribution of the paper is that we have also shown how it can be used to express theories of the processing capabilities of a cognitive architecture (APCs). One might well ask why we chose to do this: Why did we describe how APCs, such as click mouse, map into primitive processes? Could we not have treated APCs as black boxes and pieced these together in the fashion of GOMS? The answer is no; doing so would miss a point fundamental to the approach: It is precisely the fact that we describe the basic resource and information requirements of the elemental architectural processes that provides the required constraint on performance. There is no avoiding this level of detail when parallelism matters.

One response to our critique of UAN and PDL is that they offer a set of mechanisms sufficiently rich as to enable the expression of whatever a theorist requires. Indeed successful efforts have been made in this direction (Vera, John, Remington, Matessa, Freed, in press). It may even be possible to capture the inequalities that represent cascaded information flows. However, such a response would miss the point that what cognitive science needs is computational expressions of theories that are not sufficiently rich to express whatever a modeler wants but rather sufficiently

constrained as to make commitments to the nature of the underlying human information processing system.

Acknowledgment

The work was supported by the NASA Aviation Operations Safety Program and the Intelligent Systems Program.

References

- Card, S.K., Moran, T.P., and Newell, A. (1983), *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA.
- Freed, M., Matessa, M., Remington, R., and Vera, A. (2003). How Apex Automates CPM-GOMS. In *Proceedings of the 5th International Conference on Cognitive Modeling*, April 10-12. Bamberg, Germany.
- Gray, W.D., John, B.E., & Atwood, M.E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world task performance. *Human-Computer Interaction*, 8, 237-309.
- Hartson, H.R., & Gray P.D. (1992). Temporal aspects of task in the User Action Notation. *Human Computer Interaction*, 7, 1-45.
- Howes, A., Vera, A., Lewis, R.L., & McCurdy, M. (2004). Cognitive Constraint Modeling: A formal approach to supporting reasoning about behavior. In *Proc. 26th Annual Meeting of the Cognitive Science Society*.
- Kieras, D.E. (1999). *A Guide to GOMS Model Usability Evaluation Using GOMSL and GLEAN3*. Technical Report. University of Michigan, Ann Arbor, MI.
- Lewis, R.L., Howes, A., & Vera, A. (2004). A constraint-based approach to understanding the composition of skill. In *Proc. Sixth International Conference on Cognitive Modeling*. LEA, Inc., Mahwah, NJ, USA.
- McClelland, J. (1979). On the time-relations of mental processes: An examination of systems of processes in cascade. *Psychological Review*, 86, 287-330.
- Paterno, F., Mancini, C., & Meniconi, S. (1997). ConcurTaskTrees: A diagrammatic notation for specifying task models. In *Proc. INTERACT'97*. Chapman & Hall, London, UK.
- Salvucci, D.D., & Lee, F.J. (2003). Simple cognitive modeling in a complex cognitive architecture. In *Proc. CHI 2003*. ACM Press, New York, USA.
- St. Amant, R., & Ritter, F.E. (2004). Automated GOMS-to-ACT-R model generation. In *Proc. Sixth International Conference on Cognitive Modeling*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA.
- Vera, A., Howes, A., McCurdy, M., & Lewis, R.L. (2004). A constraint satisfaction approach to predicting skilled interactive performance. In *Proc. CHI 2004*. ACM Press, New York, USA.
- Vera, A., John, B., Remington, R., Matessa, M., & Freed, M. (in press). Automating Human-Performance Modeling at the Millisecond Level. *Human-Computer Interaction*, in press.