Improving Usability and Integration of Human Behavior Representation Engineering across Cognitive Modeling, Human Factors, and Modeling and Simulation Best Practices

Sylvain Pronovost
Human Factors Dept., CAE Professional Services
Institute of Cognitive Science, Carleton University
Ottawa, ON, Canada
spronovost@gmail.com

Dr. Robert West
Institute of Cognitive Science, Carleton University
Ottawa, ON, Canada
robert_west@ccs.carleton.ca

Keywords:
ACT-R; cognitive architecture; cognitive modeling; CDB; GOMS; human behavior representation; human factors; IPME; knowledge representation; macrocognition; microcognition; sociotechnical systems; SoHBeR; task analysis; XML.

ABSTRACT: *Models of human behavior and cognition differ greatly in breadth, level of detail, and ultimately on the features and criteria of interest relative to the intents and goals of the modelers and their field of expertise. On the one hand, cognitive modeling in general, and cognitive architectures specifically, are interested in microcognitive models of mental processes and fine-grained behavioral outcomes, pitched at a fundamental level of theoretical interest, whereas human factors and cognitive ergonomics modelers focus on performance and workload measures at a coarse macrocognitive level of interaction between multiple agents and their sociotechnical environment. There has traditionally been a gap between micro- and macrocognitive modeling endeavors, reinforced by skepticism on the possibility of reconciling what is seen as fundamental differences between their respective levels of description. The purpose of this paper is to present the progresses of the authors' research project aiming to bridge microcognitive and macrocognitive models of cognition, from cognitive architectures to task analysis. Herein are presented a methodology and a conceptual framework aimed at streamlining the process of cognitive and behavior modeling, focusing on the issues of usability and integration in the development and use of models.*

1. Introduction

The research presented here endeavors to integrate human factors models and other cognitive/behavioral modeling efforts, focusing on knowledge representation (KR hereafter), as well as on linking theoretical and applied research issues. On the issue of *knowledge representation*, the aim is to establish necessary and sufficient conditions for (i) satisfying the constraints of known design and processes concerning brain, cognition, and behavior on the one hand, and (ii) for satisfying the integration of such KR with other types of representations used in modeling and simulation (M&S) practices. The second focus is on *linking theoretical issues with applied issues*, with an emphasis on what features of models of individual agents are necessary to model their interactions with technologies, environments, and other agents, and what additional requirements are needed to make them scalable to such larger complexities.

Two interrelated solutions that are currently in development to address the aforementioned objectives are presented in this paper: the first is the development of a concept for the integration of scalable cognitive models (where scalability is meant as an architecture design bridging micro- and macro-level cognition and behavior) with human behavior representation (HBR) models, which are engineering models designed for M&S products and services. There have been numerous attempts to link low-level cognitive architectures to human-technology interaction (HTI) and multi-agent interaction models – all such models now generally fall under the label of sociotechnical systems (STS) modeling. We propose SoHBeR (Sociotechnical Human Behavior Representation), a tripartite model combining the ACT-R cognitive architecture, a sociotechnical systems model bridging ACT-R with a macro-cognitive framework, and task network models obtained from human factors best practices used in discrete-event simulations of performance and workload.

The second solution is the automated re-use of modeling data in HBR via the standardization of HBR taxonomy and structure. This research interest stems from the idea of reusing human factors models generated via all sorts of task analyses, to be translated as direct extensions of HBR models of synthetic agents. This amounts to transferring the knowledge gathered from human factors analyses into working models of intelligent agents. Some compromises have to be made by the concerned subject matter experts, such as in the way human factors analyses are conducted and data is compiled, as well as how HBR-specific programming is conducted. On the human factors side, knowledge representations of goals, tasks, functions, etc. will have to follow a strict language to satisfy formalism constraints such as explicitness, completeness, and decidability, while on the HBR programming side, extensions will have to be created to accommodate higher-level constructs such as goals, operators to reach such goals, selection rules, planning schemas for networks of subgoals and subtasks, etc. The end product would be an automated human factors model-to-HBR script to generate on-the-fly intelligent agents in synthetic environments, fulfilling roles, functions, and goals gathered from human factors analyses. The extensions for the HBR modeling specification would be a candidate choice for inclusion in the Common Database (CDB) standard in the M&S community, such as XML metadata files to be seamlessly accessed via CDB development and use.

1.1 From Micro to Macro Cognition

There are multiple approaches to modeling human behavior and cognition, from artificial intelligence (AI) to cognitive modeling, to engineering models. While such approaches exhibit considerable variability in the features and techniques they select to further their ends, it is mostly through such ends that they can be established as distinct research endeavors. The widespread use of production rules ("if-then" or "condition-action" clauses) and artificial neural networks, for example, may obfuscate what roles and functions such specific algorithms are meant to implement.

Artificial intelligence's stakes in cognitive modeling have been the most diverse, considering its pragmatically-driven nature. Simulation of cognition and behavior have been accomplished in "game AI", via anything from physics engine algorithms (such as line of sight and collision detection algorithms) to scripting and heuristics, and are nowadays reaching sophisticated levels akin to the implementation of techniques borrowed from theoretical and applied AI research as found in Russell and Norvig (2009). Orkin's (2006) review of the state of the art AI algorithm in the F.E.A.R game engine exemplifies this transition, from traditional finite state machines scripts to the more elegant STRIPS framework, the Stanford Research Institute Problem Solver for intelligent planning.

Cognitive modeling, in its purest academic and theoretical endeavors, uses biologically- and psychologically-inspired algorithms to simulate neural and mental processes in order to test theories of cognition. Production systems, neural networks, and hybrid cognitive architectures represent decades of research in an open community where a crosspollination of ideas helps fine-tune simulations in order to achieve more descriptive and predictive matches between experimental data and model outputs. The most successful and popular cognitive architectures are Anderson, Matessa and Lebiere's ACT-R (1997), Kieras and Meyer's EPIC (1997), and Laird, Newell and Rosenbloom's SOAR (1987).

Engineering Models of "human behavior representations" (Pew & Mavor, 1998; Zacharias, MacMillan & Van Hemel, 2008) are pitched at task-level, human-environment interactions, by approximating through mathematical parameters and variables the impact of cognition and perception on agent performance and behaviors. By using discrete-event simulations, i.e. process simulations of state changes in a complex system, coupled with such mathematical constructs, commonly referred to as performance-shaping factors (Blackman, Gertman & Boring, 2008), task flows are simulated with degrees of input variability, and a range of process and output data are generated in order to assess human and technology interactions with regards to performance, effectiveness, workload, etc.

Some attempts at hybridization of various cognitive and behavioral modeling approaches have yielded a certain degree of success, promising more constraints and credibility in their claims by bridging gaps between agent-level model, component models such as neural networks for visual perception, and synthetic environment models. One such remarkable success story is SAL (figure 1), the Synthesis of ACT-R and LEABRA, a cognitive architecture and an artificial neural network programming architecture (Jilk, Lebiere, O'Reilly et al, 2008). The SAL model was successful in modeling multi-agent tactical activities in the UNREAL Tournament™ video game environment, by combining high-level planning and low-level perceptual elements of cognitive and neural architectures.

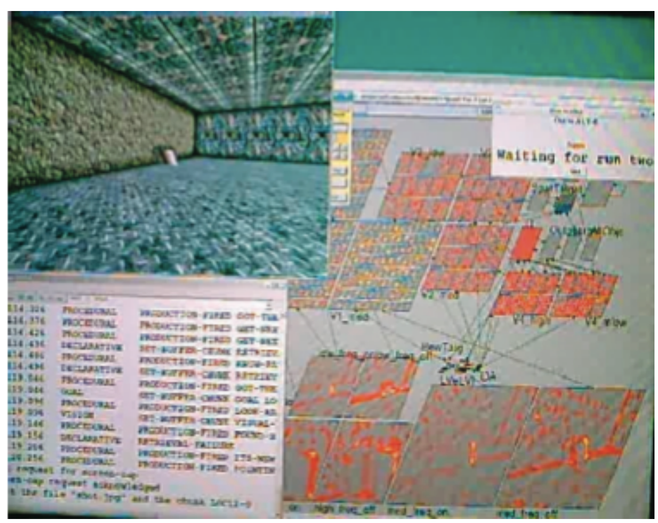

Figure 1: SAL (ACT-R architecture with a LEABRA visual perception module) in Unreal Tournament

Various attempts at integration between cognitive architectures and engineering models have also been made, from ACT-R and IPME – the Integrated Performance Modeling Environment, a discrete-event simulator modeling operator performance via task network models (Archer, Lebiere, Warwick, et al, 2002), to Kieras' combination of EPIC and the GOMS approach (the HCI methodology of Card, Moran, and Newell, 1983, explained in section 2) into GLEAN, a tool to evaluate user interface design usability (Kieras, Wood, Abotel et al, 1995).

1.2 Limitations

Crystal and Ellington (2004) reviewed task analysis models and techniques in the area of human-computer interaction and observed two majors issues shared by modeling approaches when it comes to human activity: they require increased *usability* and a higher degree of *integration*. The former is necessary because traditional task analyses are too long and/or complex to learn, difficult to perform, and once data is generated, it is hard to analyze and interpret. The latter issue concerns the tradeoff between efficiency (factoring usability, among other criteria) and effectiveness (factoring breadth and depth) of modeling techniques, with the assumption that specialized models could be combined to yield richer data than in isolation, yet having to remain tractable and usable. Those two sources of criticism of models of human activity can be leveled at the present topic of micro- and macro-cognitive modeling endeavors. We propose four problem areas for current practices in computational modeling of human behavior and cognition:

Scope Traditional modeling approaches are pitched at a specific level, whether neural, cognitive, behavioral, physical interactions with environment, swarm behavior, sociotechnical systems, or even models involving economics and politics. Trespassing on some of those boundaries would allow richer representations and more heuristic models to produce more realistic individual and multi-agent performances and predictive data.

Interoperability The isolated development of oftentimes proprietary algorithms aiming to model a subset of phenomena related to HBR hinders not only the transfer of knowledge from one modeling paradigm to another, but also that possibility of sharing data and bridging systems to be syntactically and semantically interoperable. A unified modeling approach, coupled with data format, validation, and interchange standards, specifically aimed at HBR interoperability, is needed to overcome the isolation of current and future HBR modeling practices.

Reusability HBR modeling paradigms are pursued in a fashion whereby models and data are tightly coupled together, thereby lacking "plug-and-play" capabilities: the overall architectures and algorithms, as well as the more specific models engineered through them, and data structures used to specify inputs are amalgamated or fused together, lacking modularity. In the words of Jones, Crossman, Lebiere, et al (2006), this could be done by "*creating a clean distinction between the parts of a model that depends on the unique aspects of the architecture and those that do not*", among other strategies.

Ergonomics The learning curve to develop sufficient skills to understand, analyze, and tweak cognitive models is steep, let alone to develop one's own model. One needs to learn the capabilities and limitations of all aspects of the modeling architecture, the subtle differences between modeling paradigms, and comparing how a model fares with regards to other architectures requires the researcher to rewrite models from one modeling language to another**.**

1.3 Solutions Under Development

Our research proposes two solutions to overcome the limitations of current modeling approaches: (i) a unified modeling taxonomy and modeling framework, and (ii) the technological means to standardize such endeavors. The SoHBeR framework, Sociotechnical Human Behavior Representation, is aimed at multi-agent, flexible, and scalable HBR modeling, and is presented in section 2. A standardized, computational knowledge representation approach is presented in section 3, detailing SoHBeR XML data representation, validation and tools for interoperability. The modeling framework and standardization techniques rely on existing technologies and concepts from the literature in cognitive science, human-computer interaction, and human factors and ergonomics. Of interest to us are the ACT-R cognitive architecture, the GOMS modeling approach, the IPME software, the extensible markup language (XML), and the common database standards (CDB), some of which are also detailed below.

2. SoHBeR Modeling

The SoHBeR modeling framework is a conservative extension of the original GOMS technique to model operator tasks and behaviors from Card, Moran, and Newell's seminal work in the study of HCI, as presented in *The Psychology of Human-Computer Interaction* (1983). The scientists had developed a framework to analyze routine, expert-level use of a technology for a human operator by breaking down the task flow in goals, operators, methods, and selection rules (figure 2). Note that in GOMS, "operators" were merely a label to refer to a task or activity, while methods referred to compound tasks.

While HCI benefited greatly from GOMS models and analyzes for user interfaces and other workstation studies, with an emphasis on human error, performance, etc., the modeling framework has significant limitations: it does not address unpredictability in less straightforward and non-routine tasks, it is very much oriented towards the study of usability, not focused on functionality, and it requires extensive training to learn GOMS analysis. GOMS is thus geared towards routine, sequential tasks modeling, with a single operator, and does not fare well in the pursuit of HBR involving dynamic, uncertain, and cooperative/competitive human activity, which involve decision-making, learning, task scheduling and prioritizing, and coordination between agents.

```
GOAL: DELETE-FILE
.   GOAL: SELECT-FILE
.   .   [select:  GOAL: KEYBOARD-TAB-METHOD
.   .             GOAL: MOUSE-METHOD]
.   .   VERIFY-SELECTION
.   GOAL: ISSUE-DELETE-COMMAND
.   .   [select*: GOAL: KEYBOARD-DELETE-METHOD
.   .             .   PRESS-DELETE
.   .             .   GOAL: CONFIRM-DELETE
.   .             GOAL: DROP-DOWN-MENU-METHOD
.   .             .   MOVE-MOUSE-OVER-FILE-ICON
.   .             .   CLICK-RIGHT-MOUSE-BUTTON
.   .             .   LOCATE-DELETE-COMMAND
.   .             .   MOVE-MOUSE-TO-DELETE-COMMAND
.   .             .   CLICK-LEFT-MOUSE-BUTTON
.   .             .   GOAL: CONFIRM-DELETE
.   .             GOAL: DRAG-AND-DROP-METHOD
.   .             .   MOVE-MOUSE-OVER-FILE-ICON
.   .             .   PRESS-LEFT-MOUSE-BUTTON
.   .             .   LOCATE-RECYCLING-BIN
.   .             .   MOVE-MOUSE-TO-RECYCLING-BIN
.   .             .   RELEASE-LEFT-MOUSE-BUTTON]

*Selection rule for GOAL: ISSUE-DELETE-COMMAND
  If hands are on keyboard, use KEYBOARD-DELETE-METHOD,
    else if Recycle bin is visible, use DRAG-AND-DROP-METHOD,
    else use DROP-DOWN-MENU-METHOD
```

Figure 2: original GOMS modeling example

2.1 SGOMS, and S$_2$GOMS

The study of sociotechnical systems, i.e. the complex interactions between humans and technological environments, is a natural extension of HBR research endeavors, albeit a far more complex one. Macrocognitive models have barely been explored outside of the kingdom of artificial intelligence (see Sun, 2005, for a recent account of the state of the art in macrocognitive modeling research). West and Nagy (2007) set out to explore the possibility of reconciling micro- and macrocognitive modeling approaches by laying out a framework extending GOMS into the world of macrocognition, an endeavor which would combine the analytic power of GOMS concepts, methods, and results on the microcognitive level, with the potential of sociotechnical systems-level analysis of complex, multi-agent interactions.

Their SGOMS model (Sociotechnical systems GOMS, see figure 3) resulted in the realization that additional concepts and an extended theoretical framework were needed to bridge micro- and macrocognitive levels of analysis. Most significant of these concepts were that SGOMS requires the analysis of complex human activity in terms of planning units and unit tasks (where a planning unit is a super-ordinal construct via which unit tasks are organized and sequenced), with theoretical extensions for scheduling and coordination. Also worth noting is that the SGOMS model only makes accurate predictions when planning units may be interrupted, shed, and resumed, for coordinated activities.

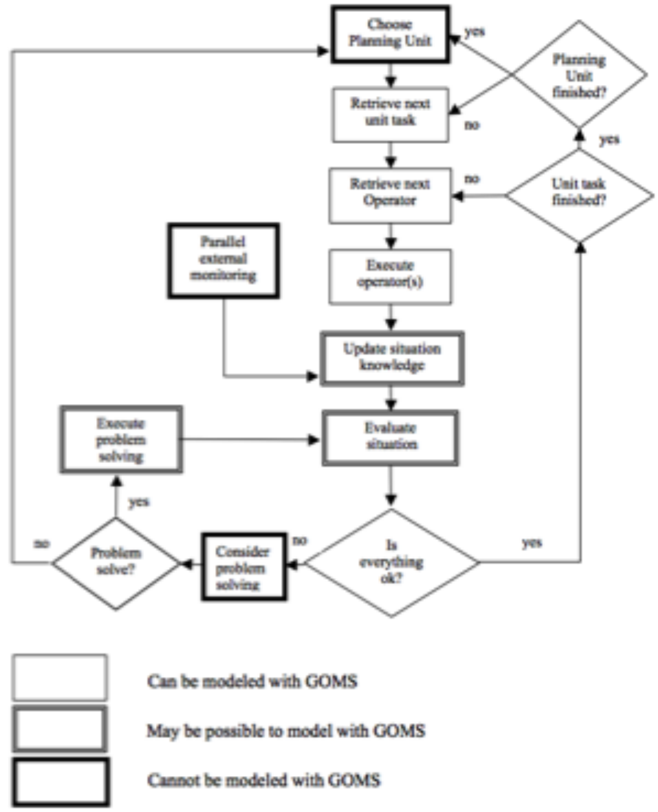

Figure 3: the SGOMS framework

Pronovost and West (2008ab) extended the SGOMS framework to account for strategic activities. The S$_2$GOMS model (Strategic Sociotechnical systems GOMS) is applicable to strategic multi-agent interactions modeling, including cooperative and competitive interactions modeling, decision-making under uncertainty, and was tested in a low-fidelity synthetic environment in the form of a World of Warcraft™ video game scenario (with the additional goal of validating and promoting low-fidelity synthetic environments as computationally viable testbeds for academic research in HBR).

S$_2$GOMS not only confirmed the theoretical claims and conceptual extensions of SGOMS by predicting the performance of unit tasks within planning units (figure 4), it also deliberately reduced the complexity of modeling decision-making processes by including decisions as planning units, following the rationale of Schultz (1997) in mapping the “estimate process” as specified in the military decision-making process (MDMP) of the US Armed Forces Joint Doctrine for Joint Operations, with the theoretical constructs of prospect theory in the cognitive psychology of decision-making (Kahneman & Tversky, 1977). West and Pronovost (2009) further demonstrated that it was theoretically possible for SGOMS models to be translated into ACT-R models, thereby allowing a microcognitive theory in the form of a cognitive architecture to model macrocognitive, sociotechnical systems-level phenomena.

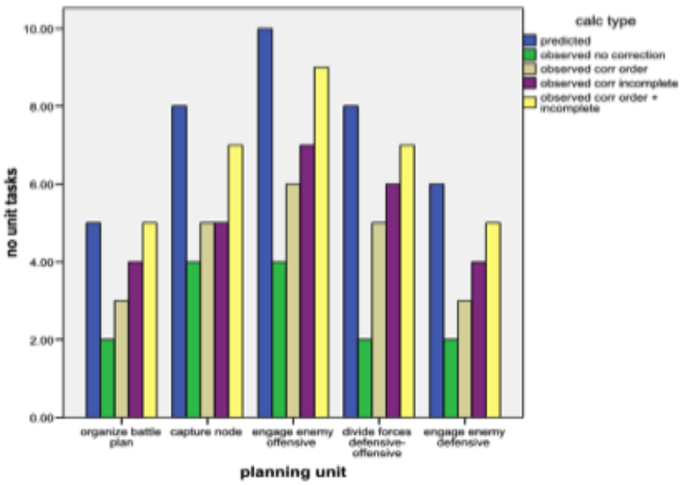

Figure 4: an example of S_2GOMS' predictive power

2.2 SoHBeR

SoHBeR, the Sociotechnical Human Behavior Representation modeling framework under development, is an attempt to unify traditional cognitive modeling with a sociotechnical systems (STS) theory and human behavior representation (HBR) engineering approaches. By bridging and combining the ACT-R cognitive architecture and the IPME task network modeling suite, guided by the S_2GOMS framework presented above, it is hoped that HBR best practices would satisfy the requirements laid out in section 1.2, namely *scope*, *interoperability*, *reusability*, and *ergonomics*. The following section details how SoHBeR may provide the conceptual and technological means to implement this HBR modeling framework.

3. SoHBeR Standardization

While HBR models from all approaches achieve ever-increasing levels of complexity, augmenting in breadth and depth, we argue, along with other scientists (Crystal & Ellington, 2004, Jones et al, 2006) that they still don't play well together because of *taxonomical* issues. All three approaches (AI, cognitive modeling, and engineering models) do not possess the necessary and sufficient theoretical framework and taxonomy to produce coordinated, multi-agent behavior in total interoperability, or even allow the transfer of a specific model and its data (inputs and outputs) from one modeling approach to another. How do we get various models of routine-like, expert, individual agency to scale up to models of dynamic and strategic, multi-agent behaviors under uncertainty?

What we need is to streamline the efforts towards integration and interoperability by means of establishing a common, abstract taxonomy to account for complex behavior (Jones et al, 2006), and we argue that this should be done via standardization across modeling and simulation (M&S) communities (Pronovost, 2009). Let us address the first question of interest raised by this previous statement: what are those taxons, exactly, and where do we find them? In artificial intelligence, they are broad in scope, vague in conceptualization, and scattered heterogeneously – from the procedural finite state machines consisting of sets of conditions-actions, to the planning AI incorporating goals, hierarchical structures for complex actions, etc. (Orkin, 2006). Cognitive Modeling generally yields more principled taxonomies and sets of "primitives" by virtue of being dependent on cognitive theories that are the underlying assumptions of cognitive architectures like ACT-R, EPIC, and SOAR. They use a mechanistic model where production systems determine behavioral outcomes based on productions rules coupled with inputs and past experience (declarative and procedural memories) (see Polk & Seifert, 2002, for a comprehensive overview of cognitive modeling). And engineering models, as we have seen in section 1.1, possess abstractions dealing with performance, workload, operator resources, and performance-shaping factors to express behavioral variability (Zacharias et al, 2008).

How do we go from there to achieve SoHBeR standardization? The commonalities in abstract, conceptual primitives found in modeling paradigms can be reduced to a small set of universals spanning from latencies, workload metrics, conditions and actions, goal-oriented behavior, etc., all of which can be in turn subsumed via hierarchical structures as found in human factors best practices, e.g. HGA (hierarchical goal analyses), MFTA (mission-functions-tasks analyses), unsurprisingly similar to HCI techniques such as GOMS. Once we decide which primitives are necessary and sufficient for a common modeling framework, as well as on a common structure to organize them, we can then move on to a translation of this taxonomy and this framework into XML data structures.

3.1 XML Knowledge Representation

SoHBeR representations, i.e. the data about goals, tasks, performance metrics, operator allocation, latencies, etc., need to be standardized in one format or another, and multiple options are available to this end. XML, the eXtensible Markup Language, already has more than a decade of history as a standard used to structure, store, and transport information. XML doesn't "do" anything, it merely specifies a set of guidelines to follow to encode documents in a structured, digital representation of data, where the structure of the knowledge domain itself is arbitrarily defined hierarchically, with properties and relations, but has to make use of XML constructs such as markup notation and operators. Its syntax is simple, and

XML happens to be a candidate format for many types of software architecture outputs used across a variety of scientific and engineering applications. For our intents and purposes, XML happens to be the format of IPME outputs, of metadata in Common Database (CDB, reviewed in section 3.4 below) compliant files, is compatible with various tools used in human factors modeling such as Microsoft Visio, mind-mapping software, and finally, can be accessed by existing programming language libraries for Python, Java, and LISP, for which three different implementations of the ACT-R cognitive architecture have been produced. Figure 5 is an example of three tasks framed in an XML-compliant format using an XML editor.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Task_Flow xmlns="SoHBeR"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="SoHBeR [redacted]
    <Task xmlns="">
        <ID>0001</ID>
        <Label>Setup Operator Workstation</Label>
        <Operator>Radar Operator</Operator>
        <Goal>Picture Compilation</Goal>
        <Latency>5.0</Latency>
        <TaskType>Discrete</TaskType>
        <Verbal>0</Verbal>
        <Auditory>1</Auditory>
        <Cognitive>6</Cognitive>
        <Psychomotor>7</Psychomotor>
    </Task>

    <Task xmlns="">
        <ID>0002</ID>
        <Label>Communicate Contact to Officer of the Watch</Label>
        <Operator>Radar Operator</Operator>
        <Goal>Picture Compilation</Goal>
        <Latency>0.5</Latency>
        <TaskType>Discrete</TaskType>
        <Verbal>7</Verbal>
        <Auditory>5</Auditory>
        <Cognitive>5</Cognitive>
        <Psychomotor>1</Psychomotor>
    </Task>

    <Task xmlns="">
        <ID>0003</ID>
        <Label>Track Contact</Label>
        <Operator>Radar Operator</Operator>
        <Goal>Picture Compilation</Goal>
        <Latency>0.5</Latency>
        <TaskType>Discrete</TaskType>
        <Verbal>0</Verbal>
        <Auditory>1</Auditory>
        <Cognitive>6</Cognitive>
        <Psychomotor>6</Psychomotor>
    </Task>

</Task_Flow>
```

Figure 5: a SoHBeR-compliant XML data file

3.2 XML Schema

A very dire consequence of creating knowledge representations for reusability, interoperability, ergonomics, and augmenting the scope of HBR models would be to have to manually validate the datasets to be input into another HBR model or architecture, or to have to manually verify the consistency and legitimacy of their outputs. This is where XML Schema comes into play. In order to validate not only the compliance of data to XML standards, but to further validate any HBR data in XML format, one needs only create a template XML Schema to automatically verify whether data is missing or is improperly formatted. This will be the very core of the SoHBeR standardization effort: compliance validation through an XML Schema, called the SoHBeR XML Schema, part of which can be seen in figure 6 below. An XML Schema specifies how an XML data file should be formatted with regards to a Document Type Definition (DTD), a set of markup declarations determining the syntax of a document. In the case of SoHBeR, the elements and attributes of various data types refer to the expected labels, types, and values of the taxonomy established through the SoHBeR modeling framework. For example, an element tagged as being a “Goal” in any HBR XML file that purports to be compliant to SoHBeR standards would have to be of the type “string”, and this would be automatically validated by the SoHBeR XML Schema, as seen by comparing figures 5 and 6.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="SoHBeR" targetNamespace="SoHBeR">

<!-- comments format -->

<xs:element name="Task_Flow">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Task" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>

                        <!-- BASIC DATA -->

                        <xs:element name="ID" type="xs:string" />
                        <xs:element name="Label" type="xs:string" />
                        <xs:element name="Operator" type="xs:string" />
                        <xs:element name="Goal" type="xs:string" />
                        <xs:element name="Latency" type="xs:decimal" />

                        <!-- TASK DESCRIPTION -->

                        <!-- Effects data -->
                        <xs:element name="IDInboundTask" type="xs:string" minOccurs="0" />
                        <xs:element name="IDOutboundTask" type="xs:string" minOccurs="0" />
                        <xs:element name="Queue" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="FIFO" />
                                    <xs:enumeration value="LIFO" />
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="DecisionNode" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Multiple" />
                                    <xs:enumeration value="Probability" />
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>

                        <!-- Timing data -->
                        <xs:element name="Distribution" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Normal" />
                                    <xs:enumeration value="Gamma" />
                                    <xs:enumeration value="Exp" />
                                    <xs:enumeration value="Rectr" />
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="TaskType" minOccurs="0">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Discrete" />
                                    <xs:enumeration value="Continuous" />
                                    <xs:enumeration value="Repetitive" />
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                        <xs:element name="MeanTime" type="xs:decimal" minOccurs="0" />
```

Figure 6: the SoHBeR XML Schema (fragment)

3.3 XML Data Binding, Queries, and Transformations

An even greater benefit of the XML format is the capabilities for integration with programming interfaces that have been created to take full advantage of the data structures represented. Such application programming interfaces (APIs) are worth noting here, with regards to the capabilities that we anticipate will be of great use for HBR modeling. The *Document Object Model* (DOM) API allows the navigation of an XML document as a radial structure (a tree-like outline), treating XML entities as objects and properties, which in turn allows the binding of XML elements to object-oriented programming declarations for scripting. *XQuery* allows users to retrieve information from XML data in the form of collections, a useful tool for database creation and maintenance. Should there be a need to alter the very structure of any or all of the HBR XML-compliant datasets or even the SoHBeR XML Schema itself, *XSLT* allows alterations of XML structures into novel syntax and data.

Since SoHBeR-compliant XML data is accessible via scripting for many types of APIs, integration with software from all modeling paradigms would be greatly facilitated. Python and LISP have their own XML DOMs, which would be directly interoperable with ACT-R, while IPME can benefit from C++, JavaScript and Python XML DOMs in a similar fashion.

3.4 CDB XML Integration

One of the ideas under review for a full-blown capability for HBR modeling interoperability is the inclusion of the SoHBeR XML Schema specification into the Common Database (CDB) initiative, a standardization effort initiated by Presagis Canada/USA Inc., a business specialized in modeling and simulation software solutions. The CDB is "*an open synthetic environment database specification*"[1], whose entities are represented via five data formats: TIFF, GEO-Tiff, OpenFlight, Shapefile, and XML. This last file format is the one of interest, where all the metadata associated with a CDB-compliant entity is stored. It is hoped that the extension of the CDB specification with the SoHBeR XML Schema as a standard for HBR modeling would allow greater interoperability with M&S technology and various defence-oriented assets such as SAFs and CGFs (Semi-Automated Forces and Computer-Generated Forces), within a common data repository.

4. Discussion

There are anticipated benefits and a few limitations to this research endeavor, some of which are readily assessable, while others are dependent on factors both theoretical and practical in nature. The benefits can be segregated in direct, anticipated, and collateral benefits. The *direct benefits* are the establishment of necessary and sufficient features for a framework bridging individual agency and sociotechnical systems modeling, thereby linking cognitive architectures, applied cognitive engineering, and even human factors best practices via a common modeling framework and common knowledge representations.

The *anticipated benefits* address the limitations and derived requirements established in the introduction: the *scope* of a common HBR modeling framework will increase, bearing scalability from simple to complex agent-environment and agent-agent interactions. Greater *interoperability* will be achieved via common data structures, used as inputs and transfers between algorithms. Algorithm- and platform-independent, modular data will yield data and model *reusability*. Finally, greater *ergonomics* will be achieved via the standardization of data structures for HBR in that there will be less to learn about for each and every new architecture or synthetic environment.

A very interesting anticipated *collateral benefit*, besides a reduction in costs, time and resources, is the increased capacity to make a more rigorous science out of HBR modeling. Indeed, by using identical inputs as independent variables, common data structures shared by the algorithms involved, and testing via some constrained variability (such as through discrete-event simulations), we could then measure and benchmark different algorithms in a much simpler way, therefore achieving a level of commensurability as of yet much harder to obtain. See Gluck & Pew's (2005) presentation of the AMBR project, the Agent-Based Modeling and Behavior Representation model comparison effort, for an in-depth account of the hardships of model comparison.

There are of course some anticipated difficulties in the pursuit of such far-reaching endeavors. One mostly controversial theoretical difficulty lies in the apparent absence of strong isomorphisms between cognitive architectures and HBR models when it comes to their taxons. Indeed, there is no easy way to decide which processes, elements, and relations at one level of description, say, the cognitive processes of interest in the ACT-R cognitive architecture, would match which other processes, elements, and relations at another level of description, such as the task-level of human factors models used in HBR engineering models. An isomorphism is a mapping representing a relationship between objects, properties or operations, and such isomorphisms must be either discovered or arbitrarily chosen in order to achieve a common modeling framework. This is precisely the aim of efforts into bridging micro- and macro-cognitive models and theories of cognition and behavior (West & Nagy, 2007, Pronovost & West, 2008ab, West & Pronovost, 2009).

The future of SoHBeR lies into the achievement of further validation in simulation models and synthetic environments, using various modeling frameworks and architectures of human behavior representation. Such validation efforts can be made using low-fidelity video game engines as experimental testbeds, as well as more sophisticated SAFs/CGFs, but they must also match the experimental data of research in cognitive psychology. Other areas of inquiry of possible interest are the development of an OWL- (Web Ontology Language) compliant specification, in order to make SoHBeR directly translatable into a markup language to share data using ontology engineering, which would be useful to manipulate knowledge representations in inference

[1] http://www.presagis.com/products/standards/cdb/

engines such as description logic-based systems, the semantic web, etc. Finally, it may turn out that XML is not the best candidate format for run-time environments, so the JavaScript Object Notation (JSON) is under consideration, a less verbose data interchange format compared to XML that reduces data entry and even data processing overhead significantly.

5. References

Anderson, J. R., Matessa, M., Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human Computer Interaction, 12*(4), 439-462.

Archer, R., Lebiere, C., Warwick, W., Schunk, D., and Biefeld, E. (2002). Integration of task network and cognitive models to evaluate system design. *1st Conference of the U.S. Army Research Laboratory Collaborative Technology Alliances*, 9 April,2002.

Blackman, H. S., Gertman, D. I., and Boring, R. L. (2008). Human Error Quantification Using Performance Shaping Factors in the SPAR-H Method. *Proceedings of the 52nd Annual Meeting of the Human Factors and Ergonomics Society*.

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

Crystal, A., and Ellington, B. (2004). Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. *Proceedings of the Tenth Americas Conference on Information Systems, New York, New York.*

Jilk, D., Lebiere, C., O'Reilly, R. and Anderson, J. (2008). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence, 20*(3), 197-218.

Jones, R. M., Crossman, J. A., Lebiere, C., & Best, B. J. (2006). An abstract language for cognitive modeling. *In Proceedings of the Seventh International Conference on Cognitive Modeling* (pp. 160-165). Trieste, Italy.

Gluck, K. A., and Pew, R. W. (2005). *Modeling human behavior with integrated cognitive architectures: Comparison, evaluation, and validation*. Mahwah, NJ: Lawrence Erlbaum Associates Inc.

Kahneman, D., and Tversky, A. (1977). Prospect Theory: An Analysis of Decision under Risk. *Econometrica 47*: 263–91.

Kieras, D. E. & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391-438.

Kieras, D.E., Wood, S.D., Abotel, K., & Hornof, A. (1995). GLEAN: A Computer-Based Tool for Rapid GOMS Model Usability Evaluation of User Interface Designs. *UISTÕ95 Proceedings of the ACM Symposium on User Interface Software and Technology*.

Laird, J. E., , Newell, A., and Rosenbloom, P. S. (1987). SOAR: An Architecture for General Intelligence. *Artificial Intelligence, 33*, 1-64.

Orkin, J. (2006). Three States and a Plan: The AI of F.E.A.R. *Proceedings of the Game Developer's Conference* (GDC).

Pew, R. W., and Mavor, A. S., (eds.) (1998). *Modeling Human and Organizational Behavior: Application to Military Simulations*. Washington, DC: National Academy Press.

Polk, T. A., and Seifert, C. M. (2002). *Cognitive Modeling*. Cambridge: MIT Press.

Pronovost, S. (2009). Bridging Cognitive Modeling and Human Behavior Representation Engineering. *Proceedings of the 9th Conference of the Australasian Society for Cognitive Science*.

Pronovost, S., and West, R. L. (2008a). A GOMS Model of Virtual Sociotechnical Systems: Using Video Games to Build Cognitive Models. *Proceedings of the European Conference on Cognitive Ergonomics*.

Pronovost, S., and West, R. L. (2008b). Bridging cognitive modeling and model-based evaluation: extending GOMS to model virtual sociotechnical systems and strategic activities. *Proceedings of the 52nd Annual Meeting of the Human Factors and Ergonomics Society*.

Russell, S. J., and Norvig, P. (2009). *Artificial Intelligence, a Modern Approach, 3rd edition*. Englewood Cliffs, NJ: Prentice Hall.

Schultz, Lt. Col. J. V. (1997) *A Framework for Military Decision Making under Risks*. Maxwell Air Force Base, Alabama: Air University Press.

Sun, R. (ed.) (2005). *Cognition and Multi-Agent Interaction From Cognitive Modeling to Social Simulation*. New York: Cambridge University Press.

West, R. L. & G. Nagy (2007) Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems: Connecting Macro Models to Microcognition. *JCEDM, 1* (2): 186-211.

West, R. L., and Pronovost, S. (2009). Modeling SGOMS in ACT-R: Linking Macro- and Microcognition. *JCEDM, 3* (2), 194-207.

Zacharias, G. L., MacMillan, J., and Van Hemel S. B. (eds.) (2008). *Behavioral Modeling and Simulation: From Individuals to Societies*. Washington, DC: National Academies Press.

Author Biographies

SYLVAIN PRONOVOST is a PhD candidate in cognitive science at the Institute of Cognitive Science, Carleton University, as well as a human factors consultant in the defence and aerospace sector, working at CAE Professional Services.

ROBERT WEST, PhD is a professor of psychology associated with the Institute of Cognitive Science, Carleton University, as well as head of the Carleton Cognitive Modeling laboratory (CCM Lab), affiliated with the Centre for Applied Cognitive Research (CACR).