Cognitive Modeling as a Method for Agent Development in Artificial Intelligence

by

Kate Dudzik

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master of Cognitive Science

In

Cognitive Science

Carleton University
Ottawa, Ontario

Abstract

This research aims to expand the applications of cognitive modeling by exploring the use of a testbed approach to modeling human behaviour. Newell's complex task analysis method (1990) was applied to model an agent completing a complex task through incremental and iterative design to test the agent's ability to fluidly monitor and react to internal and external interruptions and perform the tasks at an expert level successfully through simulated challenge sets within different environments. A single agent is developed incrementally over five stages of distinct simulated challenge sets, with testing for backwards compatibility. The testbed approach and incremental development provides insight to agent-specific cognitive structure functionality, and to the process of combining microcognition and macrocognition in cognitive modeling.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Robert L. West, of the Institute of Cognitive Science (ICS) at Carleton University for his guidance and support that made this endeavour possible. I would also like to thank Dr. Jim Davies, ICS, Carleton University, for being the second reader of my thesis, and for providing insight, time and guidance during all stages of research and writing. Thank you for always having an open door, and for providing helpful and constructive feedback throughout every stage of my thesis.

My sincerest gratitude to the professors of the ICS, Carleton University for their guidance, including Dr. Raj Singh, Dr. Ida Toivonen, and Dr. Jo-Anne Lefevre. Thanks to Dr. Brian Greenspan, Department of English, Carleton University, for designing such a helpful Directed Reading course and for encouraging my multi-disciplinary research. Thank you to Dr. David Lebovitz, ICS, Carleton University, and my late teacher, Howard S. Needham (*ave atque vale*), for teaching me how to develop my coding skills with patience.

I would also like to acknowledge the efforts of Emily Greve, for hosting the Bartender Wiki on Github and for her assistance during data collection and presentation of results. It has been an honour to mentor and work with such a bright and kind individual.

My heartfelt thanks to my supportive family, including my strong, encouraging mother Lisa, and my tenacious father Greg. Thank you to my partner, Alexis Gosselin, for always supporting, inspiring, and encouraging me, as well as the pursuit of my goals. I am so grateful for you, and all you are. I would also like to thank my community and friends, especially Dr. Adam Benn, Department of English, Carleton University, and Dr. Christopher Mong, M.D., for always being available for a kind chat, advice, and to share food.

Table of Contents

1. Introduction......5
1.1. The Common Model 5
1.2. Macrocognition 7
1.2.1. Microcognitive Architectures 9
1.2.2. SGOMS 12
1.3. Newell's Complex Task Analysis Method......14
1.4. Thesis Overview......15
2. Test Bed Development 15
2.1. Test Bed Components 18
2.2. Goals and Task Structures......18
2.3. Test Bed Environment......20
2.4. The Agent......22
2.5. Cognitive Test Bed Model Simulation......23
2.5.1. Simulation 1: Walking in the Park......23
2.5.2. Simulation 2: The Annoying Light Task......25
2.5.3. Simulation 3: Daydreaming at Work......26
3. Model of Expertise 27
4. The Environment 29
5. Scaling-up Evaluation of the Model......30
5.1. Simulation Results......30
5.1.1. Simulation 4: Free Planning......31
5.1.2. Simulation 5: Fixed Planning 31
6. Conclusion 32
7. References......33

1. Introduction

In this thesis, I expand upon the distinction between microcognitive modeling and macrocognitive modeling through the development of a single agent using a common model and test bed approach. Specifically, I was interested in whether a microcognitive model approach was sufficient to model macrocognition. Newell's complex task analysis method (1990) was applied to model an agent completing a complex task through incremental and iterative design. The goal was to test the agent's ability to fluidly monitor and react to internal and external interruptions, as well as perform the tasks at an expert level successfully through simulated challenge sets within different environments. Finally, I tested the agent's ability to scale-up from a microcognitive task to a macrocognitive task. Results demonstrate that the test bed approach was successful for the development of agent-specific cognitive structures, as opposed to task-specific cognitive structures. The results of this simulation demonstrate that a solely microcognitive modeling approach was insufficient to model a complex task in its entirety as it is limited to the performance of highly focused tasks in controlled environments.

1.1. The Common Model

There is a strong amount of agreement about what cognitive components and processes are present in humans for the functioning of cognition, and thus are required for a functional cognitive model, including declarative memory and procedural memory. However, the dominant methodological approach to the development of cognitive modeling often includes the use of extremely similar, yet distinct architectures. They are distinct in that they have different names and creators, and they are similar in that many of the components that are included within one are the same as many others, and the way that the components and mechanisms of the architecture are combined are also often very similar (Laird, Lebiere, & Rosenbloom, 2018;

Cooper, 2007). There is a great deal of overlap in the content of the cognitive modeling method and architecture, yet there is seldom much collaboration or communication between model developers and researchers. This has created an issue within the field of cognitive modeling, as there are many individually designed cognitive architectures and modeling designs with little acknowledgement about what makes them so similar and distinct. To propose a solution to this issue, authors Laird, Lebiere and Rosenbloom had proposed a more communal approach to the field of cognitive modeling, which has come to be known as the Common Model of Cognition (formerly known as the Standard Model of the Mind). The proposed method suggests that there is sufficient research in the area of cognitive modeling defining which components and processes are deemed critical to have in all simulated models. The common model approach suggests a more community based, inclusive approach to the development of models by acknowledging the similarities from the more validated and commonly applied architectures. As of now, this includes components of Soar, ACT-R, and Sigma, though this is expected to expand to include more architectures in the discussion as the field progresses as a community (Laird, Lebiere, & Rosenbloom, 2017).

A central challenge for the common model and for Artificial General Intelligence (AGI) is the development of functional human-like agents capable of performing complex tasks in dynamic real-world environments. The model will be developed using the SGOMS ACT-R cognitive architecture to ensure adherence to human cognitive limitations, restrictions and abilities. The cognitive test-bed model approach is a method of applying Allen Newell's Single Complex Task Analysis method to create a human-like AGI. In addition to the environment, expertise was included in the test-bed. The expertise is placed in the agent, and the agent is placed within the environment and provided with complex tasks to perform. The agent development process and agent task performance were gauged for backwards compatibility in the performance of open-ended tasks.

1.2. Macrocognition

Human cognition, or the study of human cognition is often categorized by using distinguishing terms to better describe the focus of the research, the method of analysis and the environment in which it is being studied. Though widely debated, the use of terms to distinguish between types of human cognition provides researchers with an understanding of the traits of cognition that are relevant to the research and the environment in which the cognition is being studied. Microcognition is most often used as a term to describe the methods and results of human cognition studied in highly specified and controlled settings and tasks. Studies of microcognition are informative, as they enable researchers to focus upon specific mechanisms and components of human cognition without excessive noise from external factors impending on task performance and often yield results that better define and explore specific cognitive components, processes and mechanisms (Cacciabue & Hollnagel, 1995; Klein et al., 2003; MacDougall, West, & Hancock, 2015; West & MacDougall, 2015). Somewhat in contrast, the term macrocognition is most often used as a term to describe the methods and results of human cognition studied in chaotic, unpredictable environments, focusing upon human cognitive behaviours in mostly broad, combinatory ways to see how humans combine and apply cognitive processes, mechanisms and components.

There is much debate surrounding the relevance of microcognitive research, macrocognitive research, and the ways in which microcognition and macrocognition are related and relevant to one another (Gregson, 1998; Kingstone et al, 2003; Klein et al., 2003; Turvey & Carello, 2012;

van Gelder & Port, 1995). Though microcognition and macrocognition are distinct, it does not necessarily mean that they are not complementary to one another. Most research in macrocognition references and includes results of microcognitive research, as it often can validate the inclusion of specific human cognitive components, processes and mechanisms, including the way in which they function, within theory and application (Cacciabue & Hollnagel, 1995; Klein et al., 2003; MacDougall, West, & Hancock, 2015; West & Nagy, 2007; West & Pronovost, 2009; West et al., 2013).

The focus of this research is how humans cognitively process task performance when internally and externally generated interruptions occur. Within the context of this research, interruptions are defined as naturalistic, unexpected events or stimuli that can occur when the agent is involved in another task, and they are in no way planned or that people can predict. Due to the dynamic and unpredictable nature of naturalistic environments navigated in everyday life, interruptions are extremely common during human task performance and are often addressed, ignored or processed seamlessly by humans (Cacciabue & Hollnagel, 1995; Klein et al., 2003; MacDougall, West, & Hancock, 2014; Newell, 1990; Somers & West, 2011; West & Nagy, 2007; West & Pronovost, 2009). Furthermore, this definition of interruption includes internally generated thoughts that are both context and non-context specific, such as planning the next task appropriate for the current situation, or a daydream that is irrelevant to the current context. It is often the case that real-world interruptions do not occur at optimal times during task performance. To account for the distinct complexity of fluid interruption navigation, our hypothesis is that additional cognitive structures must exist beyond what is currently represented within microcognitive architectures to enable an agent to track contextual changes, and fluidly return to the last known stage in task performance following an inopportune and unpredicted interruption (Somers & West, 2011; West & MacDougall, 2015; West & Nagy, 2007; West & Pronovost, 2009).

Studies analyzing interruptions in macrocognitive environments involve the use of a microcognitive architecture model navigating a dynamic problem space. Existing studies pertaining to decision making within complex environments have reliably produced results suggesting that the Declarative Memory (DM) occupies a critical role in solution development. Furthermore, context-based categories are suggested to be a supplementary component for cognitive reasoning within complex task performance (Mehlhorn, Taagen, Lebiere, & Krems, 2011; Rutledge-Taylor, Lebiere, Thomson, Staszewski, & Anderson, 2012; Somers & West, 2011; Thomson, Lebiere, Anderson, & Staszewski, 2015; van Maanen & Verbrugge, 2010; West & Nagy, 2007). Applying the approach of agent-based cognitive modeling supplemented by microcognitive simulation results will produce a better understanding of the benefits and limitations of employing microcognition in the study of macrocognition.

1.2.1. Microcognitive Architectures

The field of cognitive modeling arose from the concept of creating a full, computational model of the human mind using computational methods (Card et al., 1987; Newell, 1973; 1990). In order to achieve this accomplishment, researchers have proceeded to develop and test models of isolated components of cognition in controlled settings in order to hone and improve the human-likeness of the models. By creating multiple isolated, mechanisms-specific models that focus on emulating human likeness and cognitive functioning under controlled settings, researchers in cognitive modeling have gained insight into how to better produce a model of human cognition using computational methods (Laird, Lebiere, & Rosenbloom, 2017). The human components represented within cognitive models relies upon the research and experimentation completed within the fields of psychology and neuroscience to inform simulation design and functionality (Anderson & Lebiere, 2004; Card et al., 1983; MacDougall, West, & Hancock, 2014).

GOMS is a microcognitive architecture developed upon the theoretical basis that all expert behaviour is comprised of the following cognitive components: Goals, Operators, Methods, and Selection rules (Card et al., 1983). GOMS was developed to represent the human cognitive system as a high-functioning information processing system, comprised of symbols and productions. Furthermore, the human processing system is subdivided into three integrated subsystems: perceptual, motor, and cognitive (Card et al., 1983). The inclusion of subsystems enables the architecture to model complex and dynamic functions of human cognition by accounting for subsystem-specific abilities, symbols, processes, and productions.

As humans consistently produce goal directed behaviour, the goal component is representative of the current state of the system. The goal function is a feature that is present within levels of the architecture, enabling the component to elicit action within a specific subsystem, producing action-states within each subsystem. Furthermore, the goal implemented as a buffer enables interconnectivity within the system, as communication is required to occur between subsystems for cohesive action to be produced (Card et al., 1983; West & Somers, 2011). Operators are small units of a task that are unconscious actions which are often combined in various orders to complete a larger goal. Methods are sequences of operators. Methods are frequently reused and are always ordered in the same sequence each time, which does not require a module, such as Declarative Memory, to guide the sequence. This results in methods being combined and executed as very fast sequences due to high reuse (West & Somers, 2011).

ACT-R is a microcognitive architecture that is heavily grounded in cognitive and neuroscientific research (Anderson & Lebiere, 2003; Anderson & Lebiere, 2004). The GOMS architecture is often developed within the ACT-R architecture, as they are compatible systems with overlapping structural features (Anderson & Lebiere, 1998; West & Somers, 2011). The vast amount of overlap in architecture structure of GOMS enables compatible integration of the ACT-R sub-symbolic system. Productions within ACT-R are limited in size, requiring one production to be assigned to each operator and method within the system. Productions communicate within the system through goal buffers, which guides the order of productions and enables the productions to communicate with modules. Modules in ACT-R include the Intentional Module, Declarative Module, Manual Module, and Visual Module. As each module is associated with a specific brain region, it requires a specified buffer that is module specific. At the core of the architecture is the Procedural Module, which receives information from buffers of each module. The Procedural Module uses a production rules to interpret incoming information from the buffers, by selecting the best matching production.

Both GOMS and ACT-R architectures have been modeled to produced reliable results indicating a strong ability to describe and predict human expert capabilities across domains, such as medicine, aviation, robotics. The low-level specificity expressed within the GOMS cognitive components enables the architecture to provide a strong and detailed account for task performance (Card et al., 1983; Ramkumar, Stappers, Niessen, Adebahr, Schimek-Jasch, Nestle & Song, 2017). ACT-R provides a broader account of human behaviour, specializing in the production of descriptive models of psychology experimental conditions (see ACT-R CMU for database of literature).

1.2.2. SGOMS

Sociotechnical GOMS (SGOMS) is a macrocognitive architecture that is a theory of how human cognition functions in macrocognitive environments and tasks (West & Nagy, 2007; West & Pronovost, 2009; West et al., 2013). The primary component that distinguishes SGOMS from a micro cognitive architecture is the planning unit. The planning unit is a structure that enables the organization of unit tasks required for goal completion (West & Nagy, 2007; West, MacDougall, & Hancock, 2014). This acts as a way to bridge a microcognitive architecture to the knowledge level to perform macrocognitive tasks that are more complex and dynamic than microcognitive tasks (West & Nagy, 2007, West et al., 2015). There can often be multiple planning units involved with the performance of larger goals to manage order of unit task completion and interruptions. Planning units are critical in the performance of higher-level goals, as they can be organized to order unit tasks, to order planning units, as well as manage interruptions in the case of a constraint-based issue (West & Nagy, 2007). This means that if a person is completing a planning unit and there is a reason to stop or change the plan, they can interrupt the planning unit between unit tasks to select a different, more suitable planning unit given the situation. This is a critical distinction between planning units and unit tasks, as unit tasks may not be interrupted unless it is an extreme case (West & Nagy, 2007; West & Pronovost, 2009).

SGOMS has been successfully used to model human expertise in environments that include macrocognitive features such as multiple agents and interruptions (MacDougal et al., 2015; West, Ward, Dudzik, Karimi, & Nagy, 2017). The SGOMS theory of expertise is that the cognitive components of expertise are largely similar across experts, differing only in application and content. In other words, the general cognitive structure for the performance of expert behaviour is the same across forms of expertise and individuals, and only differs with regards to task-specific information and environmentally specific behaviours and strategies (West et al., 2017).

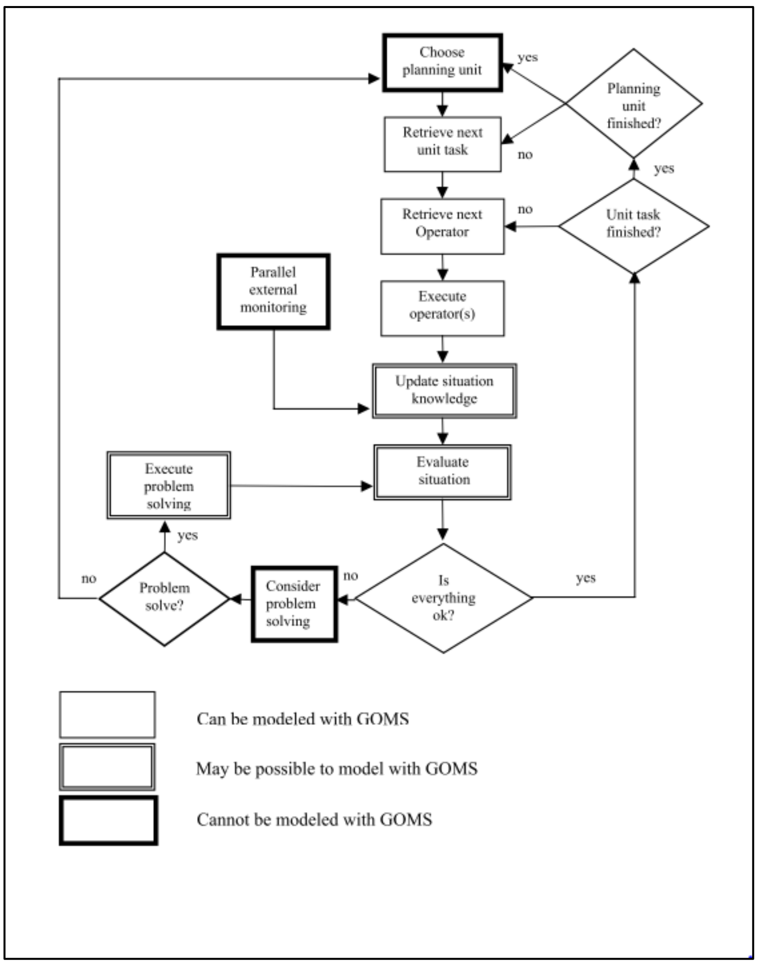

Figure 1. SGOMS Framework. West & Nagy (2007).

1.3. Newell's Complex Task Analysis Method

Newell (1973) famously proposed the development of a functional, unified cognitive architecture. The unified architecture is to be implemented as a control structure for a full agent, requiring the linking of bands and system levels of cognition within a single cognitive model (Newell, 1973, 1990; see Anderson & Lebiere, 2003, for a review). In addition to the unified cognitive architecture, Newell described methodologies for researching unified cognition. One method, which has rarely been used, is to model a complex task in its entirety (Newell, 1973). Focusing on the analysis of a complex task can provide insight into what Newell defines as the control system for human cognition, which cannot be attained from the exploration of multiple small tasks performed within highly controlled, and most often artificial settings (Newell, 1973; 1990). The complex task approach to cognitive model development is also applicable to the design and testing of artificial general intelligence (AGI) as it focuses upon the development of human-like systems (Newell, 1973; 1990).

As described in section 1.1, the study of cognition can be subdivided into two categories: microcognition and macrocognition (Cacciabue & Hollnagel, 1995; Klein, Woods, Bradshaw; Hoffman, & Feltovich, 2004). Critical features of macrocognitive tasks include factors such as dynamically changing environments, unexpected events, team work, multiple conflicting goals, social and emotional reactions, and so on. Our interpretation of modelling a task in its entirety is that the task should be situated in a realistic macrocognitive environment that can support an expert level task. In particular, this will involve modelling high level, strategic decision making in real world expertise.

1.4. Thesis Overview

The purpose of this section is to provide an overview of the contents that comprise this thesis. Chapter 2 provides a detailed account of the common model test bed, in which macro level functions were added to ACT-R and tested the functionality of the model. This includes the test bed development process, components, environment, agent development, and the cognitive test bed model challenges: Simulation 1: Walking in the Park, Simulation 2: The Annoying Light Task, and Simulation 3: Daydreaming at Work. The agent performing the simulations described in Chapter 2 will become part of the foundation for the scaling-up evaluation (Chapter 5). Chapter 3 includes a description of the SGOMS bartending model of expertise, including the process of development, model description, and a functional link to view the pseudo-code version of the model online. Chapter 4 describes the bar environment that will be used within the scaling-up evaluation in Chapter 5. Chapter 5 provides an account of the scaling-up evaluation, which consists of combining the expert knowledge (Chapter 3) to the agent (Chapter 2) and testing the agents ability to apply the expert knowledge within the environment (Chapter 4).

2. Test Bed Development

Newell famously proposed the development of a functional, unified cognitive architecture (Newell, 1990). The unified architecture is to be implemented as a control structure for a full agent, requiring the linking of bands and system levels of cognition within a single cognitive model (Newell, 1973; 1990; see Anderson & Lebiere, 2003, for a review). In addition to the unified cognitive architecture, Newell described methodologies for researching unified cognition. One method, which has rarely been used, is to model a complex task in its entirety

(Newell, 1973). Focusing on the analysis of a complex task can provide insight into the control system for human cognition that cannot be attained from the exploration of multiple small tasks performed within highly controlled, and most often artificial settings (Newell, 1973; 1990). The complex task approach to cognitive model development also makes sense for the design and testing of artificial general intelligence (AGI).

The study of cognition can be subdivided into two categories: microcognition and macrocognition (Cacciabue & Hollnagel, 1995; Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004). Microcognition is the type of cognition that is elicited by and can be measured and studied through controlled experimental conditions and environments. Macrocognition refers to the type of cognition that is produced in the real world. The distinguishing features of macrocognitive tasks include factors such as dynamically changing environments, unexpected events, team work, multiple conflicting goals, social and emotional reactions, and so on. Our interpretation of modelling a task in its entirety is that the task should be situated in a realistic macro cognitive environment. In particular we are interested in modelling high level, strategic decision making in real world expertise.

One of the strengths of the common model architecture, as implemented in various specific architectures (e.g., ACT-R, SOAR, EPIC), is that it can simulate human expertise (Prexenski, Brechmann, Wolff & Russwinkel, 2017; Russwinkel, Urbas & Thüring, 2011; Salvucci, Taatgen & Kushleyeva, 2006; Schoelles & Gray, 2012). In these simulations the goal is to program the agent to carry out the expertise in the environment. In our opinion, the common model architecture has already been shown to be successful in this area. Our goal is different, we want to build on this success and simulate how experts manage conflicting high level goals within dynamic real world environments. As part of this we will implement a model of real world expertise in the agent, but we consider the expertise model to be part of the testbed and not the object of the testing. The goal for the testbed is to provide: (1) an environment to put the agent in, and (2) expert knowledge to put into the agent. The purpose of the testbed is to stress the agent architecture by having it execute its expertise under various challenging circumstances.

We selected the task of bartending for our simulation due to the high degree of expertise required for task performance and the naturally dynamic, interruption-filled environment. We are implementing this in an iterative way by creating a single agent whose knowledge and skills evolve over time along with the complexity of the environment and the level of expertise required to perform the bartending job within it.

Our agent was given a name, Rachel, to signify the consistency of using the same agent each time. We believe that the level of consistency provided by developing a single agent will provide a clearer sense of progress and development. Also, in addition to Rachel's bartending expertise, we are developing other aspects of Rachel profile to provide conflicting goals. For example, Rachel likes daydreaming and planning her next holiday, two goals that conflict with her bartending duties. Eventually, we want Rachel to have a backstory, a form of personality, an emotional style, and values around topics such as sports, religion, and politics. In this sense, the intention is for Rachel to be like a character in a novel, rather than a generic bartender. However, as described below, adding relatively simple conflicting goals makes significant structural demands on the architecture.

As West and MacDougall pointed out, the result of everyone building their own macrocognitive models has been a proliferation of unrelated models (West & MacDougall, 2015). This state of affairs is open to the same criticism that Newell levelled at micro models and psychology experiments, that is, there is no unification and therefore limited progress (Newell, 10973). To deal with this we propose our system as a generic macrocognitive test-bed for common model architectures to facilitate comparison, integration and sharing of knowledge bases. In this paper we describe the generic structure of our testbed, such that it can be shared and implemented in any of the common model architectures. Also, more generally, we describe a way of creating test-beds that can be used to generate other macrocognitive test-beds for different purposes and different forms of expertise.

2.1. Test Bed Components

According to Newell, to predict the behaviour of an expert agent within a real-world setting and task you need to define: “the goals of the agent, the environmental and task structures, and a complete actualization of the unified cognitive architecture within the agent,” (1973). Newell’s components are the foundation for our common model testbed. The process of creating the testbed has four parts: (1) expertise is defined in terms of the goals and the task structures required for the expert task; (2) the environment is built to allow all the actions required to execute the expertise to be carried out, (3) the agent is provided with the expertise as well as goals and activities that can conflict with the expertise, and (4), the architecture is tested to see if it has the functionality to deal with external interruptions from the environment and internal interruptions from competing goals.

2.2. Goals and Task Structure

Newell proposed that modelling an expert would be the most effective way to model the whole of a complex task, as experts have extremely well practiced knowledge pertaining to goals, task method, order and performance (Newell, 1990). GOMS was developed by Card, Moran and Newell for modelling expert tasks (usually related to a computer interface) (1983). GOMS stands for Goals, Operators, Methods, and Selection rules. GOMS can be seen as a system for hierarchically arranging goals and subgoals (into methods and unit tasks) so as to execute operators in an expert manner, according to selection rules.

In a limited sense, GOMS was the first success in fully modelling a complex task. However, GOMS is restricted to modelling individuals performing clearly defined tasks in isolation, under ideal conditions. Sociotechnical GOMS (SGOMS) is a version of GOMS that allows GOMS structures to be used in multi-agent sociotechnical environments, with interruptions and unexpected events (West & Nagy, 2007). The SGOMS macrocognitive architecture adds a higher-level control system of planning units that can be used to strategize and plan, as well as react to unexpected events and re-plan new goals and subgoals as required by the dynamic environment.

We are implementing the goals and the task structures of expert bartending within an SGOMS model, to provide the expertise for the test-bed. However, the development of the SGOMS expert bartending model is not the final product. The purpose of building in expertise is to approach a human level of complexity in terms of knowledge and skill development. To cause the agent to go beyond the execution of routine expertise we add challenges. Challenges can be of two types, external and internal.

For the external challenges, we are using a dynamic environment with unexpected events. For the internal challenges, we are using an intermittent planning task that is unrelated to the expert task, specifically, planning a holiday. The test for the common model agent is to use the expert knowledge from the SGOMS model to simulate human behaviour in the dynamic environment, while simultaneously dealing with competing goals and interruptions. In other words, the task for the test-bed is to manage internal and external conflicts that are both complex and ongoing.

2.3. Test Bed Environment

Environments are often a secondary consideration in modeling, however, determining the validity of the agent depends on the validity of the task environment. Humans fluidly operate between multiple and distinct settings, even when doing the same task. Furthermore, real life environments are dynamic, so their features can change during the task, requiring the agent to modify goals and strategies without crashing.

The impoverished environments in psychology experiments, although they serve a legitimate scientific purpose (control and isolation of variables), have been justly criticized for creating a false sense of ordered control for human cognition (e.g. Gregson, 1988; Kingstone, Smilek, Ristic, Friesen, John & Eastwood, 2003; Klein, Woods, Bradshaw, Hoffman & Feltowich, 2004; Turvey & Carello, 2012; Van Gelder & Port, 1995). Although Newell did not discuss the complexity of the environment and, indeed, suggested the highly controlled environment of chess as a candidate (1973), we argue that his method of modelling a complex task is an effective way to address complex environments.

The process of building realistic, multi-agent environments is a highly technical and detailed endeavour that needs to be organized to succeed. Therefore, we need to be clear about the principles used to generate the environment. Fundamentally, there are two different starting points for generating an agent/environment simulation. In environment driven simulations, the steps are to build the environment and then create agents capable of operating in it. In agent driven simulations, the steps are reversed, where agents are built first and then environments are created to suite and test them.

The most extreme version of starting with an environment is to use real, physical environments and embody the agents in robots. The next most realistic way is to use a simulated physics environment and embody the agents in simulated robots. In contrast, the environment can be generated based on the needs of the agent and the specific tasks to be evaluated. These environments are highly artificial but are necessary, as current research lacks the ability to simulate complete models of perceptual/motor functions well enough to support higher level cognition in complex environments. For this reason, we are pursuing this option, although we note that the use of highly realistic environments is important for resolving perceptual/motor issues related to the common model.

The use of highly simplified environments to support agent behaviour relies on the theoretical assumption that perceptual/motor functions are modular and separate from central cognitive functions. Assuming this, perceptual/motor functions can be modelled in an artificial way, provided the outcomes are represented to the central cognitive system in a valid way. Therefore, the simplified environment approach needs a theory and a methodology to create valid representations.

Another issue of modeling a simplified environment is the danger of embedding problem solutions within the environment. As noted by Chalmers, French & Hofstadter (1992), if you provide an agent with everything it needs to solve a problem, and little else, it will solve the problem. Natural problem space environments are filled with multiple alternate paths. Therefore, a realistic environment should provide alternative paths for agents to consider. Our solutions to the issues raised are as follows:

- Complexity: To create a realistic, complex environment we are basing it on the SGOMS model of professional bar tending. The environment consists of all the objects needed for agent to perform the expert actions. It is constructed using the Python ACT-R environment system and can be updated as the agent is updated. We also plan to develop a generic way of describing the environment that can be attached to the SGOMS model, so that it can be implemented in different common model systems.
- Dynamic: The bar contains patrons and other bartenders. The patrons make decisions based on the probability of behaviour and level of alcohol consumption. The other bartenders also follow the SGOMS model, but it is necessary for them to cooperate with each other. Taken together these processes generate an environment with significant dynamic properties.
- Representation Validation: Common model architectures of expertise can be criticized for using overly simplistic structures and ignoring problems related to representation, meaning, and grounding. Expert model representations are usually derived from a Knowledge Level analysis of the task. As West & Young note, this methodology can be justified by arguing that it is sufficient to use only the parts of the representations that influence the task (2017). This argument sidesteps complex issues related to representation, but it means that representations created in this way must be understood as task-bound simplifications. This caveat is acceptable for our purposes.

2.4. The Agent

To test a common-model agent within the test-bed environment, the agent must first be loaded with the expertise model and be provided with minimal perceptual/motor abilities to perceive and manipulate the objects in the environment. The agent is then loaded with at least one alternate, competing task and placed in the environment. We selected Python ACT-R (Stewart & West, 2007) as our implementation platform, as it is compatible with the SGOMS expertise model, and because the ACT-R architecture has been successfully used to model and predict human behaviour in complex task performances (West, Hancock, Somers, MacDougall & Jeanson, 2013).

2.5. Cognitive Test Bed Model Simulation

To develop the agent and the test-bed, we are using an iterative design approach [9]. We are in the initial stages of development, but already we've found that the sort of agents that are sufficient for modelling psychology experiments fail to act like humans when faced with even the simplest macro-level tasks. Failure to produce human-like behaviour is very valuable in early development phases as it reveals fundamental limitations of the architecture, enabling a more adept and comprehensive model to be developed in future iterations. The purpose of iterative development is to ensure that each stage of development is functional before moving to the next. This provides insight in developing a unified model, as one can isolate problems of functionality to iteration stages. With each iteration of development, we add a new component to test agent performance and check backwards compatibility. That is the new agent must also be able to complete all the previous tasks.

2.5.1. Simulation 1: Walking in the Park

The bar is attached to a park where agents can walk, have a smoke, socialize, etc. The initial task for our agent, Rachel, was to search for a location in the park, based on a landmark. The competing tasks were to take in the view of the park and daydream. In other words, we wanted to evaluate the agent's ability to experience a wandering mind while managing a simple goal.

For the visual search process, we used the SOS vision system in Python ACT-R, as it is capable of replicating results of human cognitive performance in visual search experiments (West & Emond, 2004, West, Emond, & Tacoma, 2005). Similar to other visual search models, SOS vision can search for specific defining features of the target, such as object shape or colour. When a match is found SOS places the chunk representing the perceived features of the object in the visual buffer. This can then be compared to the features of the landmark goal stored in DM. If it is a match, the agent will engage their motor system, walking towards the landmark and completing the task at arrival. If it is not a match, the agent will return to the search state while wandering the park (i.e., randomly choosing locations to walk to).

However, this approach resulted in Rachel searching the park in a very intense, highly focused way. To allow Rachel to take in the scenery while she searched, we augmented the visual module to process bottom up vision which perceives random objects in the environment. To avoid interference with the top down search we augmented this process by adding in the rule of bottom-up when not top-down, meaning that when the vision module is not following top down instructions to search for a specific object or environmental feature, it is driven in a bottom up fashion by the salience of objects in the environment. This allowed Rachel to see and remember other things as she wandered through the park and consequently, to map her environment in memory.

To allow Rachel to daydream we used the same logic with declarative memory as we did with the visual system. That is, we applied the bottom-up when not top-down rule. This means that whenever the declarative memory is not busy with a request from the procedural module, it will retrieve various thoughts, stored as chunks, based on spreading activation.

The purpose of the walking in the park simulation task was to test the ability of the agent to run multiple cognitive functions in parallel. We found the simplicity of the task was extremely useful in this simulation, as we could analyze how each additional component modified the behaviour of the agent. The architectural assumptions are illustrated in Fig. 1.

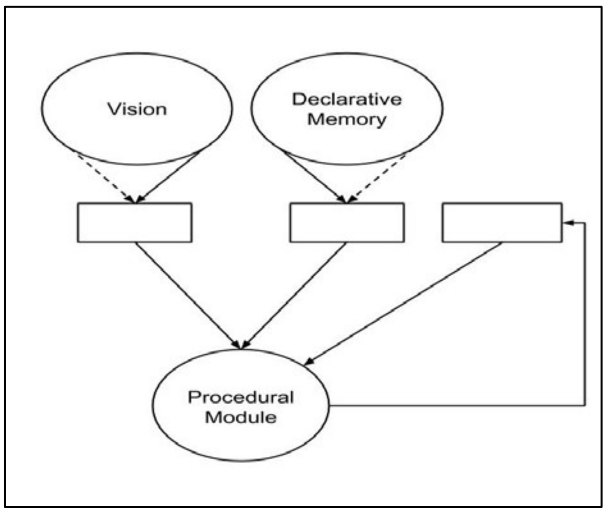

Fig. 1. Top-Down (solid lines) and Bottom-Up (dotted lines) processing of internal and external information.

2.5.2. Simulation 2: The Annoying Light Task

The next simulation was meant to test Rachel's ability to deal with interruptions. To create this ability, we implemented the architectural modifications to the common model structure described by West & Young (2017). This approach uses an emotional module to represent the alarm function associated with amygdala. The emotional module functions as a shadow production system. It can see everything that is in all the buffers, but it cannot make requests from the other modules. Instead it places an alarm (low, medium, or high) in the emotional buffer, which alerts the procedural module to the issue (Figure 2). The interruption is then appropriately handled by the productions related to the SGOMS task management system.

The annoying light task involves serving food when the light from the kitchen comes on. Rachel is working alone, serving at the bar, so when the light comes on (medium interruption) she must finish whatever she is doing at the moment, get the food, deliver it, and go back to what she was doing. In the simulation the light comes on at random and quite frequently, so the task feels like it would be very annoying. However, the simulation was successful, demonstrating the agent's ability to quickly and effectively switch tasks.

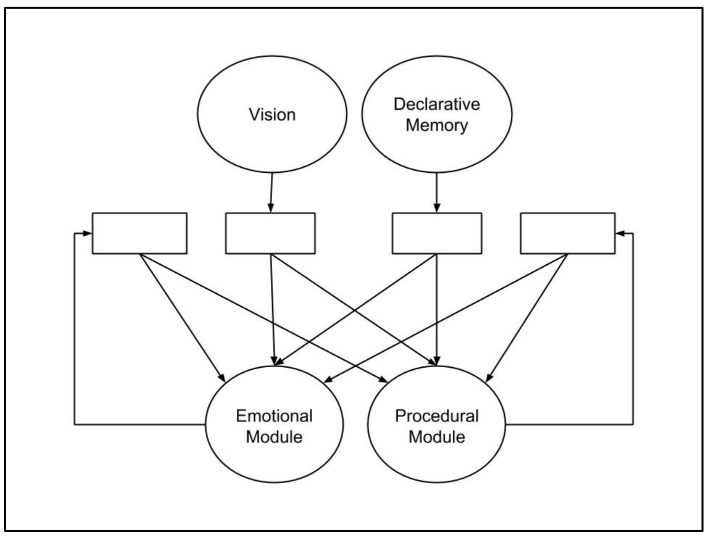

Fig. 2. Cognitive test-bed model with alternate top-down and bottom-up pathways.

2.5.3. Simulation 3: Daydreaming at Work

Our next simulation was done to ensure backwards compatibility with the walk in the park simulation. In this simulation, the tasks and information provided to the agent in Simulation 1 and Simulation 2 were combined, allowing Rachel to daydream about her holiday while working alone at the bar and responding to the light for food pickup.

The simulation could not run without crashing, as the architecture could not distinguish between bottom-up and top-down chunks of information. Specifically, the retrieved chunk from a top-down request from declarative memory could get replaced in the buffer by a bottom-up holiday memory before the appropriate top-down production could match to it. To fix the problem we altered the architecture to create separate bottom-up and top-down pathways, with their own buffers (see Fig. 3). In this system, unexpected bottom up information, such as a spider in a wine glass or remembering that you left the stove on, are handled by the emotional module by placing an alarm in the emotional buffer.

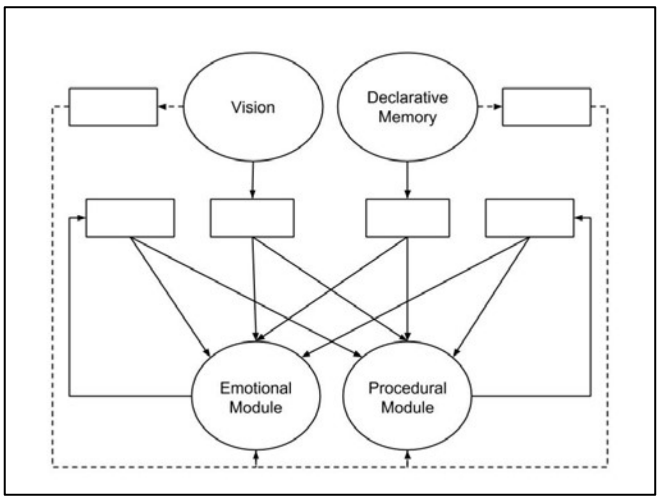

Fig. 3. Cognitive test-bed model with alternate top-down and bottom-up pathways.

3. Model of Expertise

The proceeding stage of Newell's complex task analysis method included the requirement to build a model of human expertise. The purpose of building in expertise is to approach a human level of complexity in terms of knowledge and skill development within the agent. The SGOMS model consists of planning units and unit tasks that are required to complete the task of bartending at an expert level. The SGOMS bartending model is distinct from the agent, as it is a model of the expertise required for the task of bartending, not of the cognitive mechanisms and components required to parse the information. It is a model of the actions and knowledge required for expert level task performance.

The model was developed in consultation with expert bartenders to ensure applicability and realizability of the model at the planning unit and unit task level. Cue-cards were used to document each core unit task, including the steps required to complete the task and the unit task structure. This method was selected as the medium of cue-cards allowed field performance testing, as each card could be selected by researchers and placed to predict the next task that would be performed by an expert bartender. Furthermore, by using the cue-card model of task representation the same structure could be uploaded of the SGOMS bartending model as pseudo-code on the online platform Github. Electronic copies of the cue-cards were made to create a basic, interactive online model that illustrates a pseudo-code version of the planning units and unit tasks of the SGOMS bartending model. The purpose of loading the SGOMS bartending model to Github is to allow for others to contribute to the evolution of the agent knowledge base and skillset as it currently exists, and as it grows in complexity.

The SGOMS model represents a theory of the information required to perform the task, and this information and knowledge of task performance exists entirely internal to the agent and is not found within or bound to the task environment. As context categorization is highly subjective and task based, the bartender role was categorized based upon the criteria of whether the context was opening the bar, running the bar, and closing the bar. For our model, the tasks focused upon were based entirely within the context of running the bar. This required reporting of all tasks that may be performed within the context of service hours within the bar, though it is acknowledged that bartending also may include “opening” and “closing” duties beyond what was studied for the purposes of our research.

The steps of task performance in the bartending model were created using knowledge of expert task performance and were mapped to SGOMS unit task templates. Previous research has validated the form of SGOMS unit task structures that are utilized within the bartending model (West, Ward, Dudzik, Nagy, & Karimi, 2017). The SGOMS model of bartending expertise can be found at: https://github.com/emgrev/Bartender_Model/wiki.

4. The Environment

Expertise can only function optimally within the correct environment, as described in Chapter 2. This is to say that an expert bartender cannot bartend without a bar environment that includes the required items and environmental features. As the SGOMS model is distinct from the environment and from the agent, it signifies that the cognitive components and mechanisms are structured unit tasks required to perform task, and as such, must be compatible with the structure of unit tasks that are not specifically bound to the act of bartending.

The environment that is introduced will be simplified, as object representation is limited to what was sufficient for the current model. The environment includes objects that are required to perform the expert tasks. The environment is distinct from the agent, therefore exists outside of the cognitive structure of the agent and SGOMS model of expertise. This is consistent with previous research in expertise, as the consistency and reliability of the environment plays a critical role in the ability for expert performance (Kahneman & Klein, 2009; West et al., 2017).

5. Scaling-up Evaluation of the Model

One criticism of micro cognitive architectures, such as ACT-R, is that they will not scale up from modelling microcognitive tasks to modelling macrocognitive tasks. Newell's complex task methodology is a way of testing this. To test the SGOMS/ACT-R model, all the abilities discussed above were combined to model a task involving multiple planning units and unexpected interruptions. The task was based on the SGOMS bartender model and thus captures, at least in part, the sort of complexity that people face in real world tasks. The purpose of the simulation was to evaluate whether the model would scale up to a more complex task that involved all of the model's abilities.

5.1. Simulation Results

The agent was provided with four different planning units, comprised of multiple unit tasks. This included one planning unit with two unit tasks, two planning units with three-unit tasks, and one planning with one unit task (for examples of two and three unit task planning units, see Bartender model at https://github.com/emgrev/Bartender_Model/wiki/List-of-Planning-Units).

Two simulation were done to test two different ways that interruptions can be treated. In Simulation 1: Free Planning, after dealing with an interruption the agent was free to randomly select any planning unit and start it at the beginning. In Simulation 2: Fixed Plan. after dealing with an interruption the agent had to return to the interrupted planning unit and pick up where it left off. In both cases the agent also had to ignore intrusive thoughts about going on vacation.

According to Ritter, Schoelles, Quigley, and Klein (2011), when a model is evaluated by whether or not it can complete a task, as opposed to whether it fits human data, the stability of the model should be the focus. For this it should be noted if the model failed, where it failed, and why it failed (similar to the simulations above). There is no magic number for the number of runs needed to evaluate the model. The model was run a total of 200 times, which is higher than most simulations (Ritter et al, 2011).

5.1.1. Simulation 4: Free Planning

The simulation run starts with the environmental set-up, and initialization of the agent. The agent begins the simulation run by selecting a planning unit to complete. When the agent is interrupted, they are required to address the interruption by switching to a planning unit to address the interruption. After addressing the interruption, they selected another planning unit to start at random. If no interruption occurs the agent finishes the planning unit and chooses another planning unit at random. When the agent randomly selects the next planning unit, this is the end of one run. The simulation continued until 200 runs were completed. All runs were successfully completed.

5.1.2. Simulation 5: Fixed Plan

As simulation 1 ran successfully without crashing, the model was expanded to have the agent pick up where they left off in a planning unit after an interruption. An additional production was added to create this functionality, so this model was distinct from the previous simulation. The interruption management production was created to be universal, so this function was the same across planning units. The simulation run started the same way as Simulation 1 and the agent selected a random planning unit to complete. In Simulation 2, when the interruption occurred the agent addressed the interruption after completing the current unit task and noting in memory where it was interrupted. After the interruption is addressed, the agent returns to their place within the planning unit to pick up where they left off. A run was complete when they finished the planning unit. The simulation was run for 200 runs and found that the model did not crash.

6. Conclusion

Our goal for this series of simulations was to develop a cognitive model over multiple iterations of an agent that is capable of successfully managing common macrocognitive environmental factors, including internal and external interruptions, while performing an expert-level task. Applying a microcognitive architecture approach was insufficient to model agent task performance in dynamic environments when interruptions and multiple, conflicting goals were introduced. The test-bed approach enabled the isolation of the model and architectural constraints that prevented the agent from completing the challenges presented by maintaining a consistent log of the progress and issues faced during the development of Rachel. This method was selected over the method of creating isolated models for each task. The goal was to implement principles of Newell's Complex Task Analysis Method (1990), and to attempt to model an agent performing a task in its entirety, as opposed to performing restricted unit tasks that can be isolated within a microcognitive tasks.

The results show that the effect of unexpected internal and external interruptions on complex tasks cannot be modelled using a standard ACT-R approach but can be modelled using an SGOMS structure within ACT-R plus some modifications to how ACT-R works. Importantly, these extra features would not affect previous ACT-R models of cognitive experiments. Future work will push the model further toward fully simulating humans doing expert tasks.

7. References

Anderson, J. R., & Lebiere, C. (2003). The Newell Test for a theory of cognition. The Behavioral and Brain Sciences, 26(5), 587-601; discussion 601-648.

Cacciabue, P. C., & Hollnagel, E. (1995). Simulation of cognition: Applications, expertise, and technology. In J. M. Hoc, P. C. Cacciabue, & E. Hollnagel (Eds.), Cognition and human-computer cooperation (pp. 55–73). Mahwah, NJ: Erlbaum.

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

Chalmers, D., French, R., & Hofstadter, D. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. Journal of Experimental and Theoretical Artificial Intelligence, 4(3):185-211.

Dudzik, K., West, R. L., & Young, J. T. (2018). Developing a macro cognitive common model test bed for real world expertise. Procedia Computer Science, 145, 691–698. Gregson, R.A.M. (1988). Nonlinear Psychophysical Dynamics. Hillsdale, NJ.: Erlbaum Associates.

Kahneman, D., & Klein, G. (2009). Conditions for Intuitive Expertise: A Failure to Disagree. American Psychologist, 64(6), 515–526.

Kingstone, A., Smilek, D., Ristic, J., Friesen, C. K., John, D., & Eastwood, J. D. (2003). Attention, researchers! It is time to take a look at the real world. Current Directions in Psychological Science, 12, 176.

Klein, G.; Woods, D. D.; Bradshaw, J. D.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten challenges for making automation a “team player” in joint human-agent activity. IEEE: Intelligent Systems, 91-95.

MacDougall, K., Martin, M., Nagy, N., West, R. L. (2015). A Method for Building Models of Expert Cognition in Naturalistic Environments. The 13th International Conference on Cognitive Modelling.

MacDougall, W. K., West, R., & Hancock, E. (2014). Modeling Multi-Agent Chaos: Killing Aliens and Managing Difficult People. *36th Annual Meeting of the Cognitive Science Society*, 2603–2608.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. Visual Information Processing, 283–308.

Newell, A. (1990). Unified theories of cognition. Cambridge: Harvard University Press.

Ramkumar, A., Stappers, P. J., Niessen, W. J., Adebahr, S., Schimek-Jasch, T., Nestle, U., & Song, Y. (2017). Using GOMS and NASA-TLX to Evaluate Human–Computer Interaction Process in Interactive Segmentation. International Journal of Human-Computer Interaction, 33(2), 123–134.

Ritter, F. E., Schoelles, M. J., Quigley, K. S., Klein, L. S. (2011). Determining the number of simulation runs: Treating simulations as theories by not sampling their behavior. In: *Human-in-the-loop simulations*. London, UK: Springer, 2011; p. 69–95.

Prezenski, S., Brechmann, A., Wolff, S., & Russwinkel, N. (2017). A cognitive modeling approach to strategy formation in dynamic decision making. Frontiers in Psychology, 8.

Pronovost, S., & West, R. L. (2008). Bridging Cognitive Modeling and Model-Based Evaluation: Extending GOMS to Model Virtual Sociotechnical Systems and Strategic Activities. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 52(19), 1635–1639.

Russwinkel, N., Urbas, L. and Thüring, M. (2011). Predicting temporal errors in complex task environments: A computational and experimental approach. Cognitive Systems Research, 12(3-4), 336-354.

Salvucci, D. D., Taatgen, N. A., & Kushleyeva, Y. (2006). Learning when to switch tasks in a dynamic multitasking environment. In Proceedings of the Seventh International Conference on Cognitive Modeling (pp. 268-273). Trieste, Italy.

Schoelles, M. & Gray, W. (2012). SimPilot: An exploration of modeling a highly interactive task with delayed feedback in a multitasking environment. In N. Rußwinkel, U. Drewitz & H. van Rijn (eds.), Proceedings of the 11th International Conference on Cognitive Modeling, Berlin: Universitaetsverlag der TU Berlin.

Stewart, T. & West, R. L. (2007). Deconstructing and reconstructing ACT-R: Exploring the architectural space. Cognitive Systems Research. 8. 227-236.

Stewart, T., & West, R.L. (2007). Python ACT-R code. Published as open source code. https://github.com/tcstewar/ccmsuite

Turvey, M. T., & Carello, C. (2012). On intelligence from first principles: Guidelines for inquiry into the hypothesis of physical intelligence (PI). Ecological Psychology, 24(1), 3–32.

Van Gelder, T., & Port, R. F. (1995). It's about time: An overview of the dynamical approach to cognition. In R. F. Port & T. Van Gelder (Eds.), Mind as motion: Explorations in the dynamics of cognition. MIT Press.

West, R. L., & Emond, B. (2004). The environment as theory: An example using the ACT-R/SOS environment. Proceedings of the 6th International Conference on Cognitive Modeling (ICCM 2004), 398–399.

West, R. L., & MacDougal, K. (2015). The Macro Architecture Hypothesis: Modifying Newell's System Levels to Include Macro Cognition. Biologically Inspired Cognitive Architectures.

West, R. L., & Nagy, G. (2007). Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems: Connecting Macrocognitive Models to Microcognition. Journal of Cognitive Engineering and Decision Making.

West, R. L., & Young, J. T. (2017). Proposal to Add Emotion to the Standard Model. Fall Symposium Technical Report, Volume 17, Symposium 6: A Standard Model of the Mind.

West, R. L., Dudzik, K, & Greenspan, B. (2017). Generating narrative with ACT-R agents, Talk at the ACT-R Workshop. London, U.K.

West, R. L., Emond, B., & Tacoma, J. (2005) Simple Object System (SOS) for creating ACT-R environments: A usability test, a test of the perceptual system, and an ACT-R 6 version. Proceedings of the Annual ACT-R Workshop.

West, R. L., Hancock, E. Somers, S., MacDougall, K., & Jeanson, F. (2013). The Macro Architecture Hypothesis: Applications to Modeling Teamwork, Conflict Resolution, and Literary Analysis. Proceedings of the International Conference on Cognitive Modeling.

West R., Ward L., Dudzik K., Nagy N., Karimi F. (2018) Micro and Macro Predictions: Using SGOMS to Predict Phone App Game Playing and Emergency Operations Centre Responses. In: Harris D. (eds) Engineering Psychology and Cognitive Ergonomics. EPCE 2018. Lecture Notes in Computer Science, vol 10906. Springer.