Micro and Macro Predictions: Using SGOMS to Predict Phone App Game Playing and Emergency Operations Centre Responses

Robert West[1(✉)], Lawrence Ward[2], Kate Dudzik[1], Nathan Nagy[1], and Fraydon Karimi[1]

[1] Institute of Cognitive Science, Carleton University, Ottawa, Canada
{robert_west, Kate.dudzik, Nathan.nagy}@carleton.ca, Fraydon.karimi@Carleton.ca

[2] Department of Psychology, University of British Colombia, Vancouver, Canada
lward@psych.ubc.ca

Abstract. In this study, we examine the ability of SGOMS models to predict human behaviour on two different scales, in micro cognitive task performance and in high level problem solving roles to better understand strategy use and training. To do this, two experiments were designed to isolate the role of knowledge structures in task performance. The first experiment involves modelling an application-based game, played on mobile phones. Results were compared to two models: the SGOMS model that matched the knowledge structures the players had learned during training, and a model optimized for speed, resulting in the fastest game play possible using ACT-R. In the second experiment we examined SGOMS predictions in a high level problem space of an Emergency Operations Center (EOC) simulation, with many interruptions and communication demands, comparing professional EOC managers and undergraduate performance. By comparing results between tasks, HCI design can be augmented using predictive modeling to inform the design to produce efficient and effective training programs.

Keywords: HCI · Training · SGOMS · ACT-R · App

1 Introduction

In this paper, we discuss the macro architecture hypothesis (West et al. 2013) and its significance for understanding the role that the cognitive sciences can play in designing systems for use in business, services, and institutions. Specifically, we are concerned with the extent that cognitive modelling can be meaningfully applied to real world systems design. In this study, we explored this by modeling two very different tasks, a simple memory game and a simulated Emergency Operations Center (EOC) response to disasters. By doing this we highlight the differences between high and low level tasks and how they impact modeling.

The idea of the macro architecture hypothesis came out of a debate within the macro cognition research community concerning the value of cognitive psychology

D. Harris (Ed.): EPCE 2018, LNAI 10906, pp. 501–519, 2018.
https://doi.org/10.1007/978-3-319-91122-9_41

research. The distinction between micro and macro cognition (Cacciabue and Hollnagel 1995; Klein et al. 2003) was created to distinguish complex, real world cognition (macro cognition) from the artificial and simplified scenarios used in Cognitive Psychology experiments (micro cognition). The basic idea was that real world (macro) cognition needs to be studied and understood on its own terms. Since the goal of Cognitive Psychology is to isolate and study fundamental cognitive functions, we naturally assume that the study of macro cognition should be based on the findings of micro cognition. However, some (e.g., Klein et al. 2002) have questioned the value of micro cognition, claiming it does not scale up and is therefore of limited use in macro level system design.

The question underlying this skepticism is whether artificial experiments that do not represent the full complexity of real world cognition tell us anything useful about cognition in the real world. In fact, this general concern is not unique to Macro Cognition. It has been floated by numerous groups concerned with scaling up from lab based experimental results (e.g., Gregson 1988; Kingstone et al. 2003; Turvey and Carello 2012; van Gelder and Port 1995). The macro architecture hypothesis is a response to this criticism that seeks to maintain the use of traditional cognitive psychology, but also addresses concerns about real world complexity.

In his famous paper, *You can't play 20 questions with nature and win*, Newell (1973) praised experimental psychology for producing clear scientific data on cognitive functions. However, he also criticized the field for lacking a way to unify the data. His solution was to create cognitive architectures (Newell 1973, 1990), which are unified, integrated models of cognition, usually specified as computer code. According to Newell's (1990) system level theory, the neural level implements the (micro) cognitive level, which is described by the (micro) cognitive architecture. The level above the (micro) cognitive level is the knowledge level. The knowledge level is unconstrained by the (micro) cognitive architecture, except for the provision that errors will occur if the processing capabilities of the (micro) cognitive architecture are exceeded (Newell 1990).

Micro cognitive architectures, such as ACT-R (Anderson and Lebiere 1998), SOAR (Laird 2012), and EPIC (Kieras and Meyer 1997), have been successfully used to model macro level tasks, demonstrating that the cognitive mechanisms derived from experimental psychology can scale up beyond the experimental paradigms that produced them (see West and Nagy 2007). However, it is possible to model the same high level task in significantly different ways using the same micro cognitive architecture by changing the knowledge entered into the model (Cooper 2007). So, although these models demonstrate that existing cognitive functions can model macro level tasks, micro cognitive architectures put very few constraints on the model and provide little guidance other than avoiding overload.

The macro architecture hypothesis proposes that Newell's system level scheme be modified to include a macro systems level in between the micro level and the knowledge level (see Fig. 1). Like the micro cognitive architecture, the macro cognitive architecture is proposed to be more or less constant across individuals and across tasks. However, this does not mean that everyone will use the same strategies and procedures. Instead, it means that everyone will use the same general system for managing and integrating different strategies, as well as for other common macro level functions such as dealing with interruptions, re-planning, sense making, and coordinating with others.

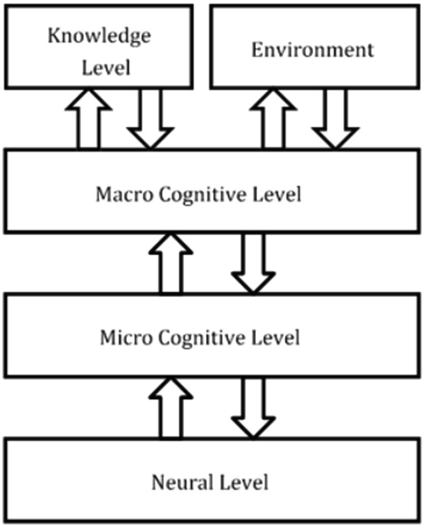

Fig. 1. Modified system levels accommodating a macro cognitive architecture (MacDougall et al. 2014)

To illustrate the difference between a macro architecture and a micro architecture we can examine the relationship between the SGOMS (macro) and ACT-R (micro) architectures. ACT-R describes human cognition in terms of parallel modules, each responsible for an element of cognition (procedural memory, declarative memory, motor system, visual system, etc.). Although parallel, the system is driven by production rules, stored in procedural memory, that fire serially and coordinate the parallel activities of the modules.

Just as micro cognitive architectures are ultimately meant to be built on neural architectures, macro cognitive architectures are meant to be built on micro cognitive architectures. In our case, we have implemented SGOMS (described below) in ACT-R (see West and Pronovost 2009; Somers and West 2013). As noted above, there are many different ways a macro level task could be modelled in a micro cognitive architecture. SGOMS applied to ACT-R, provides a systematic way to model macro tasks in ACT-R (see Ritter et al. 2006, for a review of other attempts at systematizing model building in micro architectures). To do this the basic constructs and functions of SGOMS are deconstructed and built in ACT-R. SGOMS can stand on its own, without ACT-R, or it could be implemented in a different micro cognitive architecture, which would change some of the bottom up constraints on SGOMS (for example, EPIC allows productions to fire in parallel).

Existing ACT-R models related to aspects of macro cognition can be treated as modules within the SGOMS architecture. However, by modules we do not mean dedicated brain areas. Rather we mean set ways of integrating different brain areas to produce specific higher level functions (e.g., Varela et al. 2001). For example, the ACT-R/SGOMS architecture incorporates Salvucci and Taatgen's (2008) ACT-R model of multitasking. The SGOMS architecture also makes use of the ACT-R model of instance-based reasoning (see Thomson et al. 2015) for high level heuristic decision making.

The macro architecture hypothesis represents a way of systematizing the relationship between micro and macro cognition, with a clear systems level where the study of macro cognition fits in (see Fig. 1). It is also a hypothesis as it makes two important claims: (1) there is a macro cognitive architecture that is relatively invariant across people and across tasks, and (2) it is meaningfully constrained and fully accounted for by micro cognition. If there really is a macro cognitive architecture, then it has major implications for systems design. Specifically, task structures that fit naturally with the architecture should be easier to learn and less error prone, whereas tasks structures that go against the architecture should be harder to learn and more error prone.

1.1 Unit Tasks

Although Newell did not have a separate systems level to mediate between (micro) cognition and real world tasks, he did have a control mechanism. The *unit task* was hypothesized to mediate between the structure of the task and the abilities of the (micro) cognitive system (Newell 1990). Specifically, the unit task defined how the task was mentally broken up to avoid both overloading the cognitive system and down time. For example, a task that involves remembering would be broken down into parts so that the capacity of short-term memory would not be overloaded. Likewise, parts of the task that will not necessarily follow each other would be stored as separate unit tasks, so that the agent can be released in between to do other things if there is time (i.e., avoid downtime).

In cognitive modelling, it is common practice to first determine the unit tasks used in a task, then to model each unit task and some sort of system for choosing which unit task to use next (e.g., see Gray et al. 1993). Models built in this way can be viewed as implicitly embodying two hierarchically arranged systems - the processes contained within the unit tasks and the system for selecting and coordinating the unit tasks. However, most psychology experiments fit within a single unit task. Likewise, most applied modelling projects examine only part of the project, usually corresponding to one or a few unit tasks. Consequently, there has been very little work on the system for selecting which unit task to do next.

1.2 SGOMS

SGOMS is designed to model expert behaviour. The dominant approach in the study and modeling of expertise is to treat each domain of expertise separately (Ericsson et al. 2006; Kirlik 2012). In contrast, SGOMS assumes that expertise is based on a macro architecture that is relatively invariant across different types of experts.

SGOMS is an extension of GOMS (Card et al. 1983). GOMS analyzes tasks in terms of the agents *Goals* and the motor and perceptual *Operators* that the agent uses to accomplish their goals. Frequently repeated strings of operators are represented as *Methods* and *Selection Rules* are used to choose which method or operator to do next. Similar to ACT-R, SOAR, and EPIC, the selection rules are production rules. GOMS is a family of modelling systems that follow the GOMS principles (see John and Kieras

1996). GOMS models can also be implemented in ACT-R, SOAR, or EPIC. The main limitation of GOMS compared to other architectures is that it assumes a task is well learned and does not account for learning.

SGOMS was created when we found that GOMS was unable to handle the frequent interruptions, task switching, and re-planning in real world tasks (see Kieras and Santoro 2004; West and Nagy 2007). To fix this, we modified the definition of the unit task by adding the criterion that a unit task should be small enough so that it will most likely not be interrupted. That is, we defined the unit task as a control structure that functions to avoid overload, downtime, *and interruptions*. This modification allows the unit task to continue to serve its original function to define islands of work that can be executed in a well-defined way.

We also added a second control structure, called the *planning unit*. In SGOMS, the unit task mediates between the micro cognitive level and the macro cognitive level, while the planning unit mediates between the macro cognitive level and the real world, as represented in our perceptions and knowledge of it. In contrast to unit tasks, planning units are designed for interruptions and task switching. Planning units also allow efficient communication and coordination between agents by functioning as the building blocks for creating plans and modifying them. For example, planning units are theorized to have names that are used in communication to establish common ground (Klein 2004) between agents.

The simplest form of planning unit is an ordered list of unit tasks. If a planning unit is interrupted, the current unit task is either finished or abandoned and the situation is assessed. The task can be resumed, or a new planning unit can be chosen based on the current constraints. When a planning unit is interrupted, progress on the planning unit is stored in memory so that it can be resumed, and a new planning unit is chosen. The highest level of decision-making is choosing which planning unit to work on based on the current context, which is constantly updated during the execution of the task. If there is a plan, then that is also part of the context. In addition, each planning unit is associated with a set of constraints.

Planning unit choice is based on either memorized rules (for fast, emergency situations), or memory-based heuristics (for slower, more complex decisions, see West and Nagy 2007, for an SGOMS example). Both of these can be modelled in ACT-R (see Thomson et al. 2015, for a discussion of using ACT-R to model heuristics). SGOMS does not specify what heuristics should be used, this is up to the modeler, instead SGOMS is a system for managing this process. This involves coordinating: (1) low level and parallel functions, such as bottom up and top down perceptual and motor actions, (2) expert knowledge representations, such as production rules (representing procedural memory) and expert knowledge (represented in declarative memory), (3) specific plans for coordinating agents, (4) updating and maintaining representations of all the factors and parameters relative to the task (i.e., context or situation awareness), and (5) Using heuristics, knowledge of the task and the context to re-plan and adjust to unexpected events.

SGOMS has the following hierarchical structure of representations. Each is associated with a different set of cognitive mechanisms:

- Rules and heuristics for selecting planning units based on context
- Planning units - sets of unit tasks to execute, can be interrupted and re-started
- Unit tasks - expert systems for choosing methods, smart but brittle
- Methods - fixed set of actions, executed ballistically
- Operators - basic units of perceptual and motor actions
- Bottom up monitoring - when not busy with top down commands, the system checks the environment and memory for relevant information

The level above controls the level below, but the resources are shared. So, for example, different planning units can call on the same unit task. Figure 2 shows the cycle of operations. Interruptions can occur at any level and may be solved on any level. For example, unit tasks can solve expected or common interruptions related to that unit task because it is part of the routine process. Only if an interruption percolates to the top does it result in re-planning.

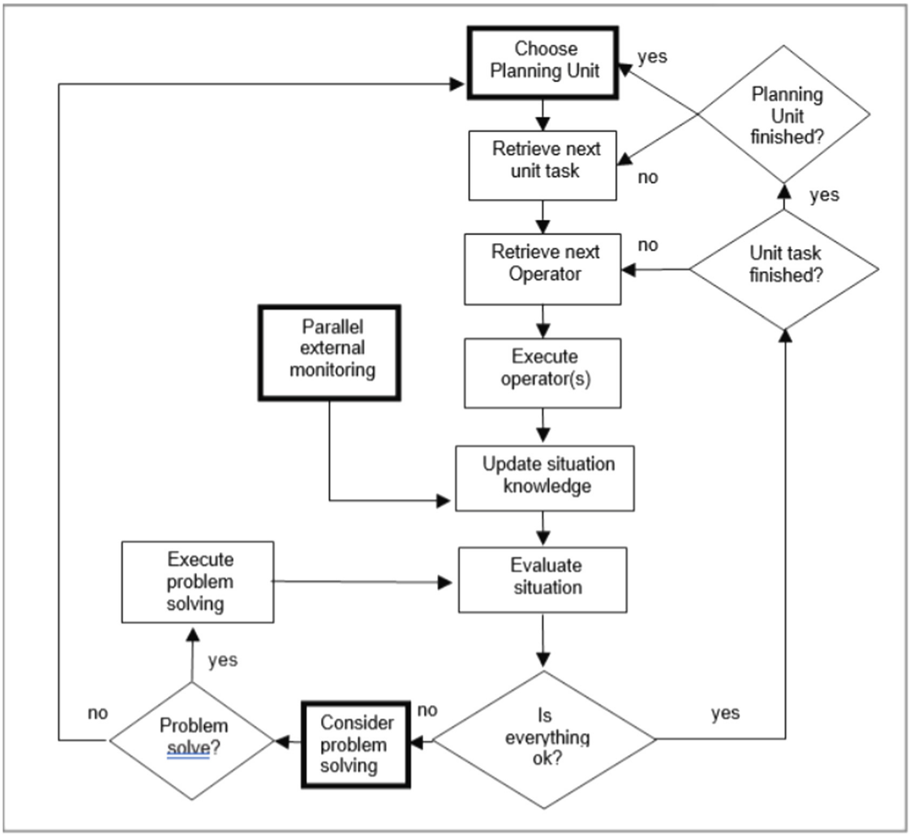

Fig. 2. SGOMS cycle of operations (West and Nagy 2007)

1.3 Model Development

SGOMS provides a way of analyzing the macro portion of a task in terms of planning units and unit tasks. To get to the micro level the model is implemented in a micro cognitive architecture, ACT-R in this case. For this, SGOMS imposes a specific way of modelling within ACT-R, which consists of a set of: (1) task generic production rules for managing planning units, dealing with interruptions, and updating context, (2) task generic ways of representing information in declarative memory (required to interact with the generic production rules), and (3) a set of goal buffers instead of just one (similar to Salvucci and Taatgan 2008). An SGOMS analysis of the planning units and unit tasks can be entered directly into this framework. To make it into a functioning micro cognitive model, the unit tasks and the rules and heuristics for selecting planning units must be modelled in detail. Also, an appropriate model of the task environment must be created (we use Python ACT-R for this, see Stewart and West 2006). Implementing SGOMS in ACT-R puts major constraints on SGOMS and also provides some difficult challenges for ACT-R.

So far, we have tested the SGOMS architecture on telecommunication network maintenance workers (West and Nagy 2007), video game team play (Pronovost and West 2008a, 2008b; West et al. 2013), aviation (Somers and West 2012, 2013) and professional mediation for disputes (West et al. 2013). For mediation, video games, and network maintenance, model tracing showed that there were no cases where the model could not reasonably predict the human behaviour, although in some cases some minor adjustments to productions rules were required.

More generally, in terms of evaluating the SGOMS architecture, or any other macro architecture, the key to is to show that it works across the all examples of the class of behaviours it is supposed to model. SGOMS is meant to model expertise, therefore, SGOMS should work across all forms of expertise. As noted above, this is contrary to current practices and theory concerning expertise, as they treat each area of expertise separately.

2 Experiment 1: Alphabet Expert Task

Within Experiment 1, we analyzed the two main ways that planning units are organized in SGOMS. The first is an ordered list of unit tasks and the second is a set of unit tasks that are cued by the environment. These are known as ordered planning units and situated planning units respectively. The Alphabet Expert task, which is described below, was a speeded stimulus-response task. Subjects were taught three different stimulus response patterns, which represented unit tasks in the SGOMS system. Then subjects were told three separate ways to order the unit tasks. The orders represented planning units in the SGOMS system. However, without imposing the SGOMS structure, these instructions can simply be viewed as describing a hierarchically organized task.

We used the SGOMS template to create an SGOMS ACT-R cognitive model of this task. We also created an optimal ACT-R cognitive model of the task. As predicted, the SGOMS model was slower than the optimal model on specific parts of the task due to the overhead produced by keeping track of where it was in terms of planning units and unit tasks. This feature of SGOMS is required in order to tolerate interruptions and allow for re-planning. We predicted that subjects would take an SGOMS approach, as it is better for real world, macro level tasks, where interruptions and re-planning are common.

The concept of methods relates to a long history in human factors (Meyer and Kieras 1997). However, the term "methods" is best known as a part of the GOMS modelling system (Card et al. 1983). In the GOMS modelling system a method is a way to achieve a specific sub-goal in the task. The GOMS approach assumes that cognitive, perceptual, and motor actions can be described as distinct, independent operators. Operators describe different actions within a task, such as recall target, move hand to mouse, move eyes to icon, move cursor to icon, and click mouse. Methods are usually ways of organizing operators to achieve sub goals. For example, the chain of actions described above could be considered a method for clicking on an icon. Methods are specific to the interface and the task. They are learned and are assumed to be reused. For example, it is generally assumed that people use the same method for clicking an icon each time (parameterized to suit each instance by considering factors such as distance, target size, etc.). Therefore, for experienced users operating simple interfaces, there will only be a limited number of methods that could reasonably be used (Newell 1973; Card et al. 1983; Gray and Boehm-Davis 2000).

In CPM GOMS (John and Kieras 1996), different operators can be used in parallel to accomplish goals. Usually these models are constructed in the form of a PERT chart where the term, *templates,* is used to describe common ways of organizing and interleaving operators (Gray, John, and Atwood, 1993; John and Kieras 1996). Gray et al. (2000) also used the term, *micro strategies*, to describe what appears to be the same thing as templates. However, as Vera et al. (2005) point out, Gray and Boehm-Davis's (2000) work on this concept elevates it from a descriptive tool in CPM GOMS (templates), to an actual theory of how the cognitive system interacts with the environment (micro strategies). Vera et al. (2005) also suggest using smaller units, called Architectural Process Cascades, for describing these interactions.

We will use the term micro strategies, to refer to low-level strategy decisions for completing a task. However, in this study we were interested in *perceptual/motor* micro strategies only for purposes of controlling for them. Our main purpose was to see if we could detect the influences of *cognitive* micro strategies. By cognitive micro strategies we mean low-level strategies related to the internal processing of information. Our goal in this study was to use models to predict differences in cognitive micro strategies at specific points in the task and then test for these differences in human subjects.

2.1 Procedure

The key elements for this type of experiment are (1) having a very simple response pattern so that the perceptual/motor micro strategies can be isolated, (2) having pre-existing, models representing contrasting options for understanding the task, (3) a highly detailed, model driven analysis of the results, and (4) an analysis based on the results of individual subjects.

2.2 Subjects

Two subjects were analyzed within the Alphabet Expert task. Two of the authors, NN and FK volunteered. Neither had experience with the SGOMS ACT-R model at the time of testing. FK was an experienced video game player, while NN was not.

2.3 Method

To account for variations in methods and unit tasks between participants, we kept our experimental task as simple as possible. To this end, we created a task called the Alphabet Expert, designed to limit response method variability and produce clear unit task structures. Each trial, subjects were presented with a four-letter code and were required to respond with the appropriate, corresponding two-letter code. Therefore, each trial was identical in terms of required response actions. However, trials were designed to include sequences in which participants knew which prompt code would occur in a predictable order, while other sequences were randomized, therefore required participants to perceive the prompt code to know which code to respond with. In this study, we attempted to find evidence for the SGOMS architecture. To do this we adopted the approach used in (Gray and Boehm-Davis 2000) to study micro strategies, replicating the two conditions of the experiment within a game-play task.

2.4 Training

Phase 1: Subjects were required to learn three distinct unit tasks, which were presented individually as sequential units. Unit tasks were presented with one-second intervals between prompt code presentation and response code. Subjects trained until they had attained their best speed and accuracy of unit task performance. To do this, subjects trained at home on their personal mobile phones. Figure 3 illustrates the structure of each unit task.

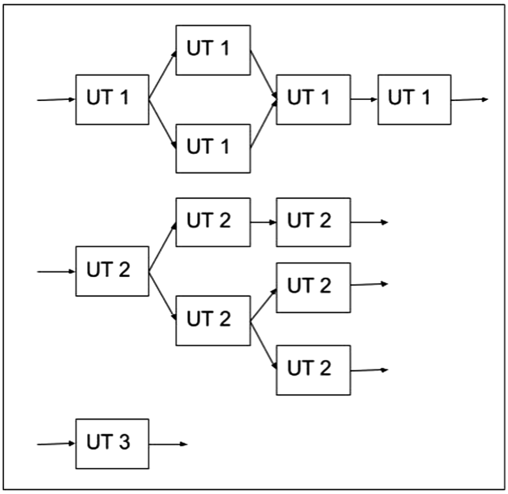

Fig. 3. Unit task structures.

Phase 2: Subjects were required to learn three distinct planning units separately. The start of each planning unit began with a distinct prompt cue consisting of a four-letter code, followed by a sequence of three unit tasks (introduced in Phase 1). Two of the planning units were ordered planning units, as they were designed to consistently present the same unit tasks in a planning unit specific order. The third planning unit learned was a situated planning unit, as each time the three unit tasks were presented in an unpredictable and random order. The planning unit structures are presented within Table 1. As training Phase 2 included a prompt code prior to planning unit commencement, participants were aware of which planning unit they were practicing each time.

Table 1. Planning unit structure.

Planning unit	Unit task order
Ordered planning unit 1	Unit task 1 Unit task 2 Unit task 3
Ordered planning unit 2	Unit task 2 Unit task 3 Unit task 1
Situated planning unit	Unit task ? Unit task ? Unit task ?

Phase 3: Subjects then practiced the whole task with one-second intervals between trials. For the ordered planning units, when subjects were cued with the code for the planning unit, they needed to recall the first unit task and type in a code to begin it. When that unit task was finished, the code for the planning unit was presented again and subjects had to type in a code for the next unit task. The sequence continued until the planning unit was finished. For the situated planning unit, subjects were cued with a code specific to indicate the situated planning unit. Subjects responded by inputting a code to start the situated planning unit. Then subjects were cued by the code for the beginning of a unit task. After completing the unit task, subjects were presented with the same prompt code indicating continuation of the situated planning unit, requiring the same appropriate response to continue. As before, the cue for the next unit task was presented. This pattern was then repeated a third time. The order of three unit tasks presented within the situated planning unit was always random. Subjects practiced on their personal devices at home until they were satisfied they were doing the task as quickly and accurately as possible.

Phase 4: For the final training stage, the one second interval between trials was removed, so as not to mask cognitive actions, and subjects once again practiced until they were satisfied they were going as quickly and accurately as possible. Once this stage was reached, we collected the data for analysis.

2.5 Model Development

Two models were developed to predict the human behaviour in this task: an SGOMS model that applied the same knowledge structures that were practiced by participants, and a model optimized for speed to produce the fastest possible game play using ACT-R. We used perceptual/motor method time estimates across all conditions for both models. With the methods determined, the only difference between the two models was the number of productions, which were determined before the experiment began. The SGOMS ACT-R model was created by writing the code for the unit tasks, the planning units and the perceptual/motor methods, and inserting them into the SGOMS ACT-R template. The optimized model worked in the following way: the code for the current planning unit was stored in the imaginal buffer (i.e., working memory) and the production representing the correct response was selected by matching with this information and the current code, which was in the visual buffer. The perceptual/motor methods were the same as within the SGOMS knowledge model.

2.6 Alphabet Expert Results

To evaluate the results, we divided the trials into distinct categories that corresponded to different predictions of the SGOMS architecture, represented in Table 2. In SGOMS, an action occurring inside a unit task occurs as it would in the optimal ACT-R model. That is, there is no overhead. These response categories were labeled Unknown Unit Task Middle (Unknown Mid UT) and Known Unit Task Middle (Known Mid UT), where Known refers to conditions in which the subject knew the next response based on the last response, and Unknown refers to conditions in which the subject had to read the new code to know the right response.

For the known response, we assumed a single perceptual/motor method, where the subject entered the next response as fast as possible. For the unknown response, we assumed the subject used two perceptual/motor methods, the first to identify the code and the second to enter the response. After removing trials with an error and outliers more than two standard deviations from the mean, the reaction times of the two conditions were very consistent, with the unknown condition taking longer, as expected. These two conditions formed the baseline for fitting the results. FK's response times were considerably faster than NN, in part because FK used two hands to type and NN used one, but also possibly because FK was an avid video game player and NN was not. We equalized the response times by subtracting the difference between FK's average RT and NN's average RT from NN's average score for both conditions. We applied this same correction to each condition based upon whether the response was known or unknown. Figure 2 shows the results with 0.05 confidence intervals for our subjects' data. For the response categories related to the ordered planning units, NN and FK were the same and not significantly different from the SGOMS model for all but one response category (Known First Unit Task) where NN matched the SGOMS model, and FK matched the optimal model (Fig. 4).

Table 2. Response category descriptions.

Response category	Description
Unknown Mid UT	A response in the middle of a unit task that is not known until the code is perceived
Known Mid UT	A response in the middle of a unit task that is determined by the response before it
Unknown PU-O Start	The response to the code to begin an ordered planning unit. This response cannot be determined by the response before it
Known PU-O Mid	The response to the code to begin the second or third unit task in an ordered planning unit. This response can be determined by the response before it
Known PU-S Mid	The response to the code to begin the second or third unit task in an unordered planning unit. This response can be determined by the response before it
Known First UT	The response to the first unit task in an ordered planning unit. This response can be determined by the response before it
Unknown First UT	The response to the first unit task in a situated planning unit. This response cannot be determined by the response before it
Unknown PU-S Start	The response to the code to begin a situated planning unit. This response cannot be determined by the response before it

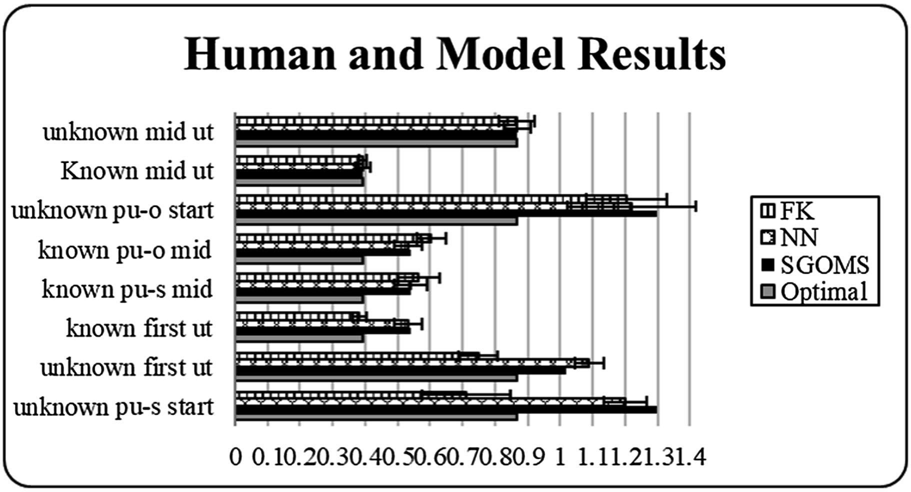

Fig. 4. Human and model results.

For the response categories related to the situated planning units, the differences between NN and FK were greater. NN was similar to the SGOMS model. In two cases, there was a significant difference between NN and the SGOMS model (Unknown First UT and Unknown PU-S Start), however, NN's results were still closer to the

SGOMS model than the optimal model. FK matched the SGOMS model in some cases, and the optimal model in other cases. In two cases FK was significantly faster than the optimal model by a small amount (again, these were Unknown First UT and Unknown PU-S Start). FK reported experimenting with different strategies to speed up the task. The pattern of FK's results indicated that heightened training lead to the elimination of the SGOMS processes for some parts of the task, as performance matched the optimal model. FK's response times within the situated planning unit and for the first response for each unit task in the situated planning unit were due to his strategies implemented to focus on speeding up this part of the task, therefore achieving faster response times. However, FK's responses to the prompt cue to begin the second and third unit tasks within the situated planning unit were as predicted by the SGOMS model (Known PU-S Mid), indicating that vestiges of the SGOMS process remained.

Based upon self-report data, consistent with FK's background as an avid video game player, we believe that FK was attempting to reorganize his understanding of the task so that actions were faster. NN also reported testing different strategies to improve performance. Both NN and FK focused most of their strategic thinking on the situated planning unit. This is also where the significant differences occurred, which may indicate that situated planning units are a more obvious candidate for optimization. As noted above, we are not claiming that people cannot learn the optimal way to do a task. Our research hypothesis claiming that the optimal way to perform a task is not the default starting point for most real-world tasks is supported by the results of the Alphabet Expert task.

2.7 Alphabet Expert Conclusion

SGOMS knowledge model provided a better fit to the data for six out of six response categories for NN, and for three out of six response categories FK. For FK, in the three cases where the SGOMS model did not match the data, the optimal model provided a reasonable match. The pattern of results supports our claim that people use the SGOMS architecture as their default system, and only later convert to an optimal form if the task can be performed without interruptions and they have the motivation for thinking about it and practicing it. This could potentially be understood in terms of Lebiere and Best's (2009) strategy evolution and strategy discovery levels. More broadly, our results support the idea that macro level factors systematically shape default strategies for using our micro cognitive architecture.

Macro cognitive architectures are descriptions of these regularities and how they are related to dealing with specific classes of real world problems. For example, SGOMS describes how people execute expert knowledge in real world environments, where interruptions and re-planning are common. Given that people rarely take part in psychology experiments, it is not surprising that they would default to a strategy that works well in their daily life. As we have shown, the Gray and Boehm-Davis (2000) methodology works well and can be employed to study the effects of macro cognitive architectures using micro cognitive experimental and modeling methods.

3 EOC Management

In the second study, we examined SGOMS predictions on a very different scale. This study is part of an ongoing project to use SGOMS to model the behaviour of Emergency Operations Centre (EOC) workers during disaster simulations, with the goal of gaining insight into training, policy, and systems design. The following is a working definition of EOCs:

> *"An **emergency operations center** (**EOC**) is a central command and control facility responsible for carrying out the principles of emergency preparedness and emergency management, or disaster management functions at a strategic level during an emergency, and ensuring the continuity of operation of a company, political subdivision or other organization.*
> *An EOC is responsible for strategic direction and operational decisions and does not normally directly control field assets, instead leaving tactical decisions to lower commands. The common functions of EOCs is to collect, gather and analyze data; make decisions that protect life and property, maintain continuity of the organization, within the scope of applicable laws; and disseminate those decisions to all concerned agencies and individuals." (Emergency operating center, Wikipedia.)*

EOC managers can only make recommendations and pass on information, they cannot force the field commanders to do things. The EOC manager, who is free from ongoing distractions in the field, can maintain a detailed and accurate model of the situation and take the extra time to make strategic recommendations. The recommendations and the reasons for the recommendations are passed onto the field commanders, which frees them from time-consuming data consolidation and provides them with a broader context. However, at the same time the field commanders can see if there is anything locally wrong with the recommendations based on their more immediate and detailed knowledge of the situation on the ground.

EOC management is obviously an important function, where failures have very serious consequences (e.g., consider the response to the Fukushima nuclear disaster). To get an overview of EOC management skills, here is a summary of the abilities an EOC manager should have from the U.S. Federal Emergency Management Agency (FEMA). Note that the descriptions are very vague but mostly refer to high-level decision-making:

- Comprehensive – emergency managers consider and take into account all hazards, all phases, all stakeholders and all impacts relevant to disasters.
- Progressive – emergency managers anticipate future disasters and take preventive and preparatory measures to build disaster-resistant and disaster-resilient communities.
- Risk-driven – emergency managers use sound risk management principles (hazard identification, risk analysis, and impact analysis) in assigning priorities and resources. Integrated – emergency managers ensure unity of effort among all levels of government and all elements of a community.
- Collaborative – emergency managers create and sustain broad and sincere relationships among individuals and organizations to encourage trust, advocate a team atmosphere, build consensus, and facilitate communication.

– Coordinated – emergency managers synchronize the activities of all relevant stakeholders to achieve a common purpose.
– Flexible – emergency managers use creative and innovative approaches in solving disaster challenges.
– Professional – emergency managers value a science and knowledge-based approach; based on education, training, experience, ethical practice, public stewardship and continuous improvement.

Macro level analyses of real world tasks are more similar to anthropology or sociology than to experimental methodology. Cognitive psychology comes into it mainly in terms of a language for describing principles and heuristics for strategic decision making, where the data sources are usually interviews and observations. Using a macro cognitive architecture provides a principled framework for understanding a task, which provides a basis for creating design improvements. In this spirit, we applied SGOMS to understanding EOC management.

3.1 SGOMS and EOC Professionals

For this study we used observations from actual EOC professionals in disaster simulations, as well as inexperienced undergraduates in similar simulations. In the simulations, performance did not depend on being fast, which is what GOMS traditionally focuses on. Instead, success was related to sense making and the ability to support decision-making. The process of choosing planning units within SGOMS provides a good way of understanding this process. In SGOMS, planning units are chosen using constraint-based decision making. This involves understanding the current context, or constraints, and using various heuristics for bounded rationality decision making. The agents in the field must do this for whatever part of the disaster they are working on. So, using the SGOMS structure, we can conceptualizer the EOC manager as providing context (constraints) and sometimes suggesting solutions (heuristic based rational analysis) to help the agents in the field choose appropriate planning units. Essentially, the EOC manager functions as a planning unit recommender system.

Planning units are also used as a quick efficient way to communicate and coordinate, but it requires common ground (MacDougall et al. 2014). One prediction that arises from this is that this will function efficiently to the degree that the EOC managers and agents in the field use a common set of planning units. In terms of the professional EOC management teams that we observed, this seemed to play out when observing the difference between teams composed of ambulance, police, and fireman versus teams composed of trained EOC managers. The ambulance, police, and fireman seemed to be more efficient in their reactions. This appeared to be due to their shared labeling system for planning units related to emergency response.

3.2 Model of Novices

Our approach for modelling the undergraduates was to assume that people have a cognitive template for understanding and implementing instructions in a task. We further assumed that this template amounts to using an SGOMS-like structure to interpret the instructions. Half way through the task and at the end of the task the participants were asked to report all the events. The main finding for the undergraduates was that they forgot a surprising number of events. We focused on that as it indicated a lack of situation awareness, which is critical for EOC management.

The model for the undergraduate participants was relatively simple as their task was limited. Also, because this was not a speeded task, the ability of GOMS to predict completion times was not useful. Participants had more than enough time to complete the components of the task, so minor differences in speed would not impact performance. Therefore, we did not model the low level perceptual motor components of the task.

Another issue was that some components of the task could not be handled by GOMS. Specifically, GOMS is not designed to model sense making or the composing of notes and reports. However, task components that cannot be directly modelled in GOMS can still be represented by unit tasks.

For this model we used unit tasks for sense making, writing, and information gathering. We also used the interrupt mechanism in SGOMS to deal with new incoming information. SGOMS is capable of ignoring irrelevant information but in this case, we assumed, due to the task, that participants would pay attention to all of the incoming information. The model operated in the following way: when new information was received, it triggered an interruption to the current planning unit. The model would then switch to the *new-information* planning unit. This was an ordered planning unit consisting of, first, the *attend-new-information* (ANI) unit task followed by the *select-incident-to-work-on* (SITWO) unit task. The SITWO unit task represents the metacognitive process of selecting what to work on next. For the student volunteers we modelled this using the availability heuristic. This was implemented using the ACT-R declarative memory system, which has been shown to be accurate in predicting human forgetting across a wide range of experiments.

Another unique feature of the model was the use of instances. SGOMS does not need a pre-existing planning unit for each instance of a task component. Instead, SGOMS can use a generic planning unit to generate a specific instance of that planning unit. In this case, when a new event occurred, a planning unit for that event would be generated by attaching the identity of the event to a copy of the generic planning unit for processing the events (i.e., writing reports and issuing advice in this case). Thus, the model could create new planning units to represent different events.

After an interruption, the SITWO unit task would use the availability heuristic to decide what to work on next. This was modeled using the ACT-R declarative memory retrieval mechanism to retrieve the most active memory chunk representing an ongoing event. Running the model predicted that, without rehearsal, the participants would forget to report almost everything. However, rehearsal occurred in the model every time an event was worked on. The key to remembering was to work on an event.

3.3 EOC Simulation Results

Most of the failures to recall an event could be placed into three categories that could all be accounted for with the model. The first was completion errors. These occurred when an event seemed to be completed, in that it was appeared to be no longer a problem. This can be accounted for in the model by assuming that coding an event as being completed blocks it from being retrieved when the availability heuristic is used. This could be modeled by making one of the retrieval criterions that the event be active (i.e., not completed), or it could be modeled using the spreading activation mechanism in ACT-R. Either way, this would prevent the event from being retrieved and worked on further, allowing the activation level to decay below threshold.

The second category of forgetting occurred when new event information was received during the processing of incoming information from another event. This includes when a new event was presented, or when there was a change in the narrative of a previously presented event (e.g., an ice storm has knocked out power but then causes the roof of an ice rink to collapse). This type of forgetting can be modeled by assuming that when the novice participants were interrupted they did not rehearse or further process the interrupted task before switching from it. This would prevent the interrupted task from getting a memory boost from having been worked on.

The third category of forgetting occurred when two events with nearly identical features (e.g. resource required, emergency type) at different locations were presented within a close time frame in the simulation. In the model, this type of error would be a natural outcome of the ACT-R partial matching mechanism. If a chunk representing one event had a higher activation and was also very similar to a chunk representing another event, the higher activation chunk would always be retrieved, thus making the less active chunk unavailable to the availability heuristic.

3.4 EOC Training Recommendations

Based on using SGOMS to frame our understanding of EOC management, we derived the following recommendations for training. The first is to have a clear labeling system for the planning units used by EOC management and the field assets. Second, these labels need to be practiced, we recommend the use of practice simulations for this. Finally, when training EOC operators there should be a focus on maintaining situation awareness through deliberate memory storage strategies. For example, we noticed that the professional teams used maps to organize themselves, suggesting that they were using location to encode knowledge in memory. This is also a place where better technology could help. For example, having automatic transcription of information from telephone, radio, and TV might help, as these sources leave no trace except in memory.

4 Conclusion

In this paper we showed how SGOMS could be used to model two very different tasks. In the alphabet expert task we showed how SGOMS could model the effects of training and strategy, and make highly accurate reaction time predictions. In the EOC task we showed how SGOMS can be used as a framework to understand high level tasks and to model particular parts of the task, memory in this case. More generally, the ability of SGOMS to usefully model both of these tasks supports our argument that SGOMS is appropriate for modelling any expert task.

References

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

Cacciabue, P.C., Hollnagel, E.: Simulation of cognition: applications. In: Hoc, J.M., Cacciabue, P., Hollnagel, E. (Eds.), Expertise and Technology: Issues in Cognition and Human–Computer Cooperation, pp. 55–74. NEA, Hillsdale (1995)

Card, S., Moran, T., Newell, A.: The Psychology of Human– Computer Interaction. Lawrence Erlbaum Associates, Hillsdale (1983)

Cooper, R.P.: The role of falsification in the development of cognitive architectures: Insights from a lakatosian analysis. Cogn. Sci. **31**(2007), 509–533 (2007)

Emergency Operating Center: Wikipedia online (n.d.). https://en.wikipedia.org/wiki/Emergency_operations_center

Ericsson, K.A., Charness, N., Hoffman, R.R., Feltovich, P.J. (eds.): The Cambridge Handbook of Expertise and Expert Performance. Cambridge University Press, New York (2006)

Federal Emergency Management Agency, Department of Homeland Security: Exercise Simulation System Document [ESSD] (2014). https://training.fema.gov/iemc/exercisesimulationdocument.aspx

Gray, W.D., Boehm-davis, D.A.: Milliseconds matter: an introduction to microstrategies and to their use in describing and predicting interactive behavior. J. Exp. Psychol. **6**(4), 322–335 (2000)

Gray, W.D., John, B.E., Atwood, M.E.: Project ernestine: validating a GOMS analysis for predicting and explaining real-world task performance. Hum. Comput. Interact. **8**(3), 237–309 (1993)

Gregson, R.A.M.: Nonlinear Psychophysical Dynamics. Erlbaum Associates, Hillsdale (1988)

John, B.E., Kieras, D.E.: The GOMS family of user interface analysis techniques: comparison and contrast. ACM Trans. Comput. Hum. Interact. **3**(4), 320–351 (1996)

Kieras, D.E., Meyer, D.E.: An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. J. Hum. Comput. Interact. **12**(4), 391–438 (1997)

Kieras, D., Santoro, P.: Computational GOMS modeling of a complex team task: lessons learned. In: Proceedings of HCI, 24–29 Apr 2004, Vienna, Austria, pp. 97–104 (2004)

Kingstone, A., Smilek, D., Ristic, J., Friesen, C.K., John, D., Eastwood, J.D.: Attention, researchers! It is time to take a look at the real world. Curr. Dir. Psychol. Sci. **12**, 176 (2003)

Kirlik, A.: The emerging toolbox of cognitive engineering models. In: Paper presented at the International Conference on Cognitive Modeling, Berlin, Germany, 13–15 Apr 2012

Klein, G., Ross, K.G., Moon, B.M., Klein, D.E., Hoffman, R.R., Hollnagel, E.: Macrocognition. IEEE Intell. Syst. **18**(3), 81–85 (2003)

Klein, G., Woods, D.D., Bradshaw, J.D., Hoffman, R.R., Feltovich, P.J.: Ten challenges for making automation a “team player” in joint human-agent activity. IEEE Intell. Syst. **19**, 91–95 (2004)

Laird, J.E.: The SOAR Cognitive Architecture. MIT Press, Lakatos (2012)

Lebiere, C., Best, B.J.: From microcognition to macrocognition: architectural support for adversarial behavior. J. Cogn. Eng. Decis. Making **3**(2), 176–193 (2009)

MacDougall, W.K., West, R., Hancock, E.: Modeling multi-agent chaos: killing aliens and managing difficult people. In: 36th Annual Meeting of the Cognitive Science Society, pp. 2603–2608 (2014)

Meyer, D.E., Kieras, D.E.: A computational theory of executive control processes and human multiple-task processes and human multiple-task performance: Part 1. Basic mechanisms. Psychol. Rev. **104**, 3–65 (1997)

Newell, A.: You can't play 20 questions with nature and win: projective comments on the papers of this symposium. In: Visual Information Processing. Academic Press (1973)

Newell, A.: Unified Theories of Cognition. Harvard University Press, Cambridge (1990)

Pronovost, S., West, R.L.: A GOMS model of virtual sociotechnical systems: using video games to build cognitive models. In: Proceedings of the European Conference on Cognitive Ergonomics (2008a)

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

Ritter, F.E., Haynes, S.R., Cohen, M.A., Howes, A., John, B.E., Best, B., Lebiere, C., Jones, R. M., Lewis, R.L., St. Amant, R., McBride, S.P., Urbas, L., Leuchter, S., Vera, A.: High-level behavior representation languages revisited. In: Proceedings of the International Conference on Cognitive Modeling, Trieste, Italy, pp. 404–407 (2006)

Salvucci, D.D., Taatgen, N.A.: Threaded cognition: an integrated theory of concurrent multitasking. Psychol. Rev. **115**, 101 (2008)

Stewart, T.C., West, R.L.: Deconstructing ACT-R. In: International conference on cognitive modelling, Trieste, Italy (2006)

Somers, S., West, R.L.: Macro cognition: using SGOMS to pilot a flight simulator. In: Proceedings of the Annual ACT-R Workshop (2012)

Somers, S. West, R.L.: Steering control in a flight simulator using ACT-R. In: Proceedings of the International Conference on Cognitive Modeling (2013)

Thomson, R., Lebiere, C., Anderson, J.R., Staszewski, J.: A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. J. Appl. Res. Mem. Cogn. **4**, 180–190 (2015)

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

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

Varela, F., Lachaux, J.P., Rodriguez, E., Martinerie, J.: The brainweb: phase synchronization and large-scale integration. Nat. Rev. Neurosci. **2**(4), 229–239 (2001)

Vera, A.H., Tollinger, I., Eng, K., Lewis, R., Howes, A.: Architectural building blocks as the locus of adaptive behavior selection. Proc. Cogn. Sci. Soc. **27** (2005)

West, R.L., Nagy, G.: Using GOMS for modeling routine tasks within complex sociotechnical systems: connecting macrocognitive models to microcognition. J. Cogn. Eng. Decis. Mak. **1**, 186–211 (2007)

West, R.L., Pronovost, S.: Modeling SGOMS in ACT- R: linking macro-and microcognition. J. Cogn. Eng. Decis. Mak. **3**(2), 194–207 (2009)

West, R.L., Somers, S.: Scaling up from micro cognition to macro cognition: using SGOMS to build macro cognitive models of sociotechnical work in ACT-R. In: The proceedings of Cognitive Science, pp. 1788–1793. Boston, Mass: Cognitive Science (2011)

West, R.L., Hancock, E., Somers, S., MacDougal, K., Jeanson, F.: The macro-architecture hypothesis: applications to modeling teamwork, conflict resolution, and literary analysis. In: Proceedings of the International Conference for Cognitive Modeling (2013)