Cognitive Social Simulation and Collective Sensemaking: An Approach Using the ACT-R Cognitive Architecture

Paul R. Smart

Electronics & Computer Science
University of Southampton
Southampton, UK
Email: ps02v@ecs.soton.ac.uk

Katia Sycara

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
Email: katia@cs.cmu.edu

***Abstract*—Cognitive social simulation is a computer simulation technique that aims to improve our understanding of the dynamics of socially-situated and socially-distributed cognition. Cognitive architectures are typically used to support cognitive social simulation; however, the most widely used cognitive architecture – ACT-R – has, to date, been the focus of relatively few cognitive social simulation studies. The current paper reports on the results of an ongoing effort to develop an experimental simulation capability that can be used to undertake studies into socially-distributed cognition using the ACT-R cognitive architecture. An ACT-R cognitive model is first presented that demonstrates one approach to solving a task previously used to investigate sensemaking performance within teams of human subjects. An approach to the implementation of an ACT-R cognitive social simulation capability is then described. The approach relies on the use of a variety of custom ACT-R modules and memory-resident Lisp databases. The custom modules enable ACT-R agents to exchange information with each other during the course of their sensemaking activities. The Lisp databases, in contrast, are used to store information about communicative transactions, the experimental setup and the structure of the communication network. The proposed solution provides the basic elements required to run cognitive social simulation experiments into collective sensemaking using the ACT-R architecture; however, further work needs to be undertaken in order to address a number of limitations associated with agent communication capabilities and the ability of agents to interact with the task environment.**

***Keywords*–collective cognition; sensemaking; distributed cognition; team sensemaking; cognitive architecture.**

I. INTRODUCTION

In recent years, there has been a growing interest in the socially-distributed or socially-situated nature of human cognition across a number of scientific disciplines [1][2][3][4][5]. Cognitive processes that were typically studied at the level of individual agents, such as memory, are now being re-examined within a more social context [6], and increasing attention is being paid to the factors that enable groups to function as the processors of information [7]. This interest in the social dimension of cognition is, in part, a reflection of the growing popularity of embodied, extended and situated approaches within the sciences of the mind [8][9][10]. However, the research is also motivated by an attempt to engineer systems that harness the collective cognitive potential of groups of individuals. The advent of global information and communication networks, such as the World Wide Web, has clearly been one of the drivers in such research; however, systems that support socially-distributed cognition are also important in more restricted organizational contexts. This is particularly so as advances in sensor technology lead to a significant expansion in the scale and scope of available data assets. As organizations move into this 'Big Data' era, so they are under increasing pressure to distribute cognitive effort and harness the collective cognitive potential of their workforces.

In order to improve our understanding of the factors that affect the performance of teams of individuals, researchers have relied on the use of both human experimental studies and multi-agent simulation techniques. Multi-agent simulation techniques are of particular interest given the efficiency with which experimental studies can be undertaken and the control that can be exerted over experimental factors of interest. However, while such techniques have proven useful in investigating a number of social psychological phenomena, most notably social influence [11], they have sometimes been criticized in terms of their cognitive sophistication and fidelity. Recently, Sun [12] has advocated the use of cognitive architectures in multi-agent simulation as a means of improving the cognitive sophistication of agent implementations and enhancing the fidelity of computational models of human social behavior. Cognitive architectures are frameworks that make particular commitments about the kind of mental representations and computational procedures that are sufficient to explain important aspects of human cognition, such as problem solving, memory and learning [13]. Although a cognitive architecture can be implemented using connectionist schemes, some of the most influential cognitive architectures, such as ACT-R (Adaptive Control of Thought-Rational) [14][15] and SOAR (State, Operator and Result) [16][17] rely on rule and symbol forms of processing. Of particular note is ACT-R, which has been the focus of a sustained research and development effort for more than 30 years and which has been used to model cognitive performance in a wide variety of experimental settings. The integration of cognitive architectures into social simulation results in what Sun [12] refers to as *cognitive social simulation*.

By incorporating cognitive architectures into multi-agent simulations, we are provided with the opportunity to study the interaction between social and cognitive factors; for example, we can study the effect that different cognitive factors (such as memory decay rates, learning rates, attention, and so on) have on aspects of collective performance. Cognitive architectures thus enrich the range of experimental opportunities that are open to investigators. In addition, because cognitive architectures provide a framework for detailed cognitive modeling, cognitive social simulations may yield results of greater predictive validity compared to multi-agent simulations that assume only rudimentary cognition on the part of agents.

This paper describes an ongoing effort to use ACT-R as a platform for cognitive social simulation in respect of a particular form of socially-distributed cognition, namely collective sensemaking. Following a brief overview of ACT-R in Section II, a specific sensemaking task is described in Section III. This sensemaking task has been the focus of previous experimental work (involving both human and synthetic agents), and it has been used to advance our understanding of the factors that affect performance in team-based situations. These features make the task particularly attractive as a starting point for the current modeling and software development effort. Section IV describes the implementation of an ACT-R agent that can perform the aforementioned sensemaking task, and Section V outlines the approach taken with respect to the implementation of a multi-agent simulation capability in ACT-R. This multi-agent simulation capability serves as the basis for performing cognitive social simulations in ACT-R. In particular, it establishes the basis for future experimental work that can systematically vary factors at the cognitive, social, technological, and informational levels in order to observe the effect of these factors on collective cognitive performance. Section VI outlines areas of further work that are needed to support these experimental efforts.

The main aims of the current paper are to 1) present an ACT-R cognitive model that be used to perform a sensemaking task, and 2) illustrate how the ACT-R model can be exploited in the context of cognitive social simulations via the use of extensions to the core ACT-R architecture. The paper also describes one means by which sensemaking capabilities can be implemented in ACT-R by adopting conventional knowledge engineering techniques. The main contributions of this work are to advance our understanding of how to model distributed cognitive processes using a popular pre-existing cognitive architecture, namely ACT-R. Such models serve as an important focus for experimental work that seeks to predict and explain the impact of social, technological and psychological factors on team-level performance.

II. ACT-R

Sun [12, p. 33] defines a cognitive architecture as “a domain-generic computational cognitive model that captures essential structures and processes of the individual mind for the purpose of a broad (multiple domain) analysis of cognition and behaviour”. A cognitive architecture is thus a framework that captures some of the relatively invariant features of the human cognitive system – those features that are deemed to be more-or-less constant across domains, tasks and individuals. One example here would be the mechanisms that support the storage and retrieval of information from long-term memory. Although a number of features of the task environment may affect the ability of subjects to recall information, the mechanisms that actually realize the recall process are unlikely to change from one task to another.

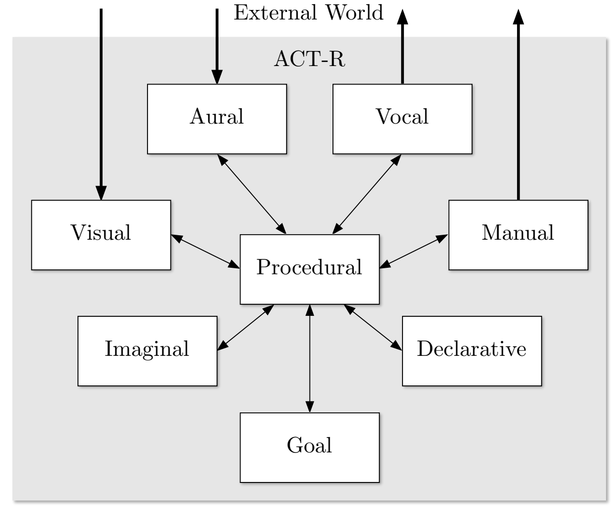

Figure 1: The core modules of the ACT-R v.6 cognitive architecture.

ACT-R is one of a number of cognitive architectures that have been used for cognitive modeling [14][15]. It is primarily a symbolic cognitive architecture in that it features the use of symbolic representations and explicit production rules; however, it also makes use of a number of subsymbolic processes that contribute to aspects of performance [15].

ACT-R consists of a number of modules (see Figure 1), each of which is devoted to processing a particular kind of information. Each module is associated with a capacity-constrained buffer that can contain a single item of information, called a chunk. The modules are assumed to access and deposit information in the buffers, and coordination between the modules is achieved by a centralized production system module – the procedural module – that can respond to the contents of the buffers and change buffer contents (via the execution of production rules). Importantly, the procedural module can only respond to the contents of the buffers; it cannot participate in the internal encapsulated activity of modules, although it can influence such processes. As shown in Figure 1, there are eight core modules in the latest version of ACT-R:

- **Input and Output Modules.** There are four input/output modules (Visual, Aural, Vocal and Manual). These provide support for modeling agent-world interactions.
- **Goal Module.** Actions within ACT-R are often dependant upon the current goal being pursued. The goal module is a specialized form of memory, with its own buffer, and it stores the current state of these goals.
- **Imaginal Module.** The imaginal module is responsible for manipulating intermediate representations of a problem when working towards a goal. For example, when calculating x in $x - 4 = 7$, the intermediate stage $x = 7 + 4$ can be stored in the imaginal module before being evaluated.

- **Declarative Module.** This module implements the memory system of the agent. It stores information in the form of chunks, each of which is associated with activation levels.
- **Procedural Module.** The procedural module is responsible for coordinating between the other modules. It contains rules that fire in response to the contents of the module buffers. The contents of the various modules are typically changed as a result of rule execution.

These modules (and their associated buffers) tend to form the basis of most ACT-R models. Cognitive modelers are not, however, restricted to the use of these modules, and new modules can be added to implement additional functionality. As an example of this kind of extension of the default ACT-R architecture, Rodgers et al. [18] added a total of nine buffers to the ACT-R architecture as part of their effort to implement a situation model (corresponding to a "mental model of the objects, events, actions, and relationships encountered in a complex task simulation" [18, p. 313]).

The ACT-R architecture has been used to model human cognitive performance in a wide variety of experimental contexts. It has generated findings of predictive and explanatory relevance to hundreds of phenomena encountered in the cognitive psychology and human factors literature, and this has earned it a reputation as the cognitive architecture that is probably the "best grounded in the experimental research literature" [19, p. 24]. ACT-R has also been used to model performance in a range of complex task settings. For example, ACT-R has been used to model driver behavior [20], including the effects of concurrent activities (such as cell phone usage) [21] and sleep deprivation on driver performance [22]. These features make ACT-R a compelling target for cognitive social simulation. To date, however, very few studies have sought to apply ACT-R to situations involving socially-distributed information processing (recently, however, Reitter and Lebiere [23] have demonstrated the use of ACT-R in a social information foraging task). The aim of the current work is to develop a generic framework for using ACT-R in cognitive social simulation experiments, and to then apply this framework to a particular kind of socially-distributed cognitive processing, namely collective sensemaking.

III. Collective Sensemaking and the ELICIT Framework

Sensemaking has been the focus of sustained research attention over the past 10-20 years [24][25][26][27]. It has been defined as a "motivated, continuous effort to understand connections (which can be among people, places, and events) in order to anticipate their trajectories and act effectively" [24]. Sensemaking is clearly something that individuals engage in as part of their attempt to explain and predict the features of some object, event or situation. This does not mean, however, that sensemaking is *only* something that individuals engage in. There is, in fact, a growing appreciation of the prevalence and importance of what might be called 'collective sensemaking' [3] or 'team sensemaking' [28], namely, the activities that are performed by groups of individuals in order to develop understanding at both the individual and collective levels. Collective sensemaking is a phenomenon of considerable importance in a number of different task contexts, such as intelligence analysis [25][29], military planning [30] and healthcare provision [31], and it is deemed to be of generic relevance to the problem-solving capabilities of military coalition organizations [3]. This highlights the importance of collective sensemaking as a focus area for cognitive social simulation experiments.

In order to support the effort to develop an ACT-R framework to study issues in collective sensemaking, it helps to have a concrete example of a sensemaking task on which to focus. For the purposes of this exercise, a particular sensemaking task was selected called the ELICIT sensemaking task. ELICIT, in this case, is an acronym that stands for the Experimental Laboratory for Investigating Collaboration, Information Sharing and Trust. It denotes an ongoing effort to provide a common experimental framework to investigate issues in group-level problem-solving [32]. The ELICIT sensemaking task is a particular activity that is performed by subjects within the context of the ELICIT framework. In essence, the task involves the selective presentation of information items – called factoids – to experimental subjects. Each factoid provides a limited amount of information about a situation, and the aim of the subject is to assimilate enough information in order to make a decision regarding the features of an impending terrorist attack (these features are typically referred to as the dimensions of the sensemaking task). The particular features the subject needs to resolve are as follows:

- **who:** the group that will attempt to perform the attack
- **where:** the country in which the attack will take place
- **what:** the kind of target the attack will be against (e.g., an army base)
- **when:** the date and time of the attack

A number of studies have been undertaken with different factoid sets (i.e., collections of factoids) in order to investigate the factors that affect performance in this task (e.g., [33]). A synthetic agent has also been developed to support multi-agent simulations involving the ELICIT sensemaking task [34]. This agent is, however, not based on a cognitive architecture, and it does not therefore provide access to the kinds of cognitive parameters that ACT-R makes available (e.g., the ability to run simulations in which agents use different cognitive strategies or possess different mnemonic capabilities).

One of the main advantages of the ELICIT sensemaking task is that it provides access to collections of factoids that have been used in a variety of experimental studies. This supports the attempt to develop a cognitive model in ACT-R because the factoids provide insight into the kind of knowledge structures that an agent needs in order to solve the designated problem. In addition, the availability of empirical results from previous studies (particularly those with human subjects) enables us to compare the performance of the model and assess how the results differ from those obtained with human subjects.

The first 10 factoids from one particular ELICIT factoid set (namely 'factoidset4aGMU') are shown in Table I (there are 68 factoids in the full factoid set). As can be seen from this subset of factoids, each factoid provides information about the entities and relationships associated with the situation, and

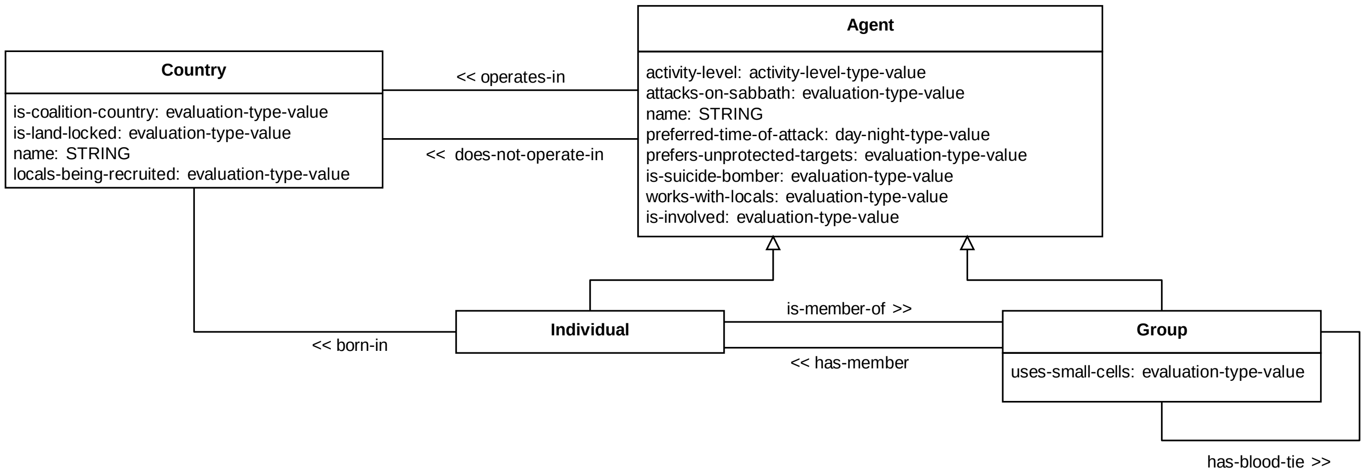

Figure 2: Part of the domain schema for the ELICIT 'factoidset4aGMU' factoidset, focusing on the Country and Agent concepts.

TABLE I: Subset of factoids from one of the factoid sets (namely 'factoidset4aGMU') used in ELICIT experiments. The characters in the 'Type' column specify the type of the factoid: E = Essential, K = Key, S = Supportive and N = Noise.

#	Type	Factoid
1	E	The Gray and Teal groups do not employ suicide bombers
2	E	There will be a suicide bomber attack at a school
3	E	The Silver group does not work in Pi
4	E	The Silver group only attacks during the day
5	N	The Rose group may be involved
6	K	The Sienna and Rose groups only target the military
7	S	Reports from the Teal group indicate standard levels of activity
8	N	There is a lot of activity involving the Rose group
9	N	The Gray group is recruiting locals – intentions unknown
10	K	The Turquoise group focuses on destroying energy infrastructure

at least some of these factoids support inferences that enable particular kinds of suspect entities (e.g., groups, countries and targets) to be eliminated. An example is provided by factoids 1 and 2 from Table I. Factoid 2 states that the impending attack will be a suicide bombing attack, and we learn from factoid 1 that Gray and Teal groups do not employ suicide bombing tactics. As a result of being presented with these two pieces of information, we can infer that neither the Gray nor the Teal group can be involved in the attack, and they can thus be eliminated from our list of suspect groups. We can also see from Table I that factoids come in four basic types: essential, key, supportive and noise [32] (these are indicated by the letters 'E', 'K', 'S', and 'N' in Table I). Expert and key factoids provide important information that is relevant to the process of resolving the who, what, when and where aspects of the task; supportive factoids provide information that tends to support the information contained in the key and essential factoids; and noise factoids contribute nothing to an agent's ability to solve the problem – the problem can be solved even if these factoids are ignored. In a typical ELICIT experiment, subsets of factoids are distributed to the members of an analysis team, and the profile of factoid sharing is controlled to mimic the features of different organizational environments. For example, the communication network can be configured so as to investigate the impact of hierarchically structured versus decentralized military command structures [33].

In order to support the development of a simulation capability in which ACT-R agents can process the ELICIT factoids and solve the sensemaking problem, a knowledge analysis of the ELICIT factoid set 'factoidset4aGMU' was undertaken. This analysis was performed in order to better understand the conceptual structures that were required by an agent tasked with processing the ELICIT factoids and to also enumerate the various inferences that were supported by each of the factoids. This analysis yielded a knowledge model that was represented using the modeling formalisms associated with the CommonKADS methodology [35]. Part of the domain schema associated with the knowledge model is illustrated in Figure 2 using UML (Unified Modeling Language) notation [36]. It highlights some of the properties and relationships that exist between two of the main components of the situation model, namely 'Agent' (a supertype of 'Group') and 'Country'. The knowledge analysis also yielded a knowledge base that contained all of the rules necessary to support factoid-related reasoning. In particular, the rules captured the essential inferences that were necessary to progressively eliminate suspect entities and identify the who (group), where (country) what (target) and when (month, day, hour) of the terrorist attack. One of these rules is shown in Figure 3. The rule, in this case, implements the inference that groups using suicide bombing tactics cannot be involved in an attack that is known to be a suicide bombing attack.

It is important to note that although the nature of the inferences associated with the processing of ELICIT factoids can seem straightforward, as is exemplified by the rule in Figure 3, the problem-solving process itself is by no means simple from the perspective of a human subject. The number of factoids to be processed ($N = 68$) imposes a heavy cognitive burden on the subjects. In addition, the subjects have no way of knowing at the outset of the process which factoids are relevant (key and essential factoids) as opposed to those that are not (i.e., the noise factoids). The number of suspect entities in each dimension of the problem also creates difficulties.

```
attack.is-suicide-bombing-attack = yes AND
group.is-suicide-bomber = no
    IMPLIES
group.is-involved = no
```

Figure 3: An example of a rule that resulted from a knowledge analysis of the ELICIT factoidset.

In total there are 7 groups, 5 countries, and 8 target types mentioned in the factoids, any of which could be involved in the attack. The result is that even when supportive cognitive artefacts are used (e.g., paper and paper) the challenge of identifying the correct suspect entities can seem formidable, and not all human subjects are able to solve the problem. Partly as a result of this complexity, it was difficult to validate the integrity of the aforementioned knowledge model in terms of its scope and accuracy. In fact, it is by no means clear simply by looking at all the rules in the knowledge base whether a reasoning system that implemented all the rules would be able to derive the correct solution to the problem. In order to evaluate this, a reasoning system was developed using the CLIPS (C Language Integrated Production System) expert system shell [37] (the CommonKADS knowledge model, in this case, served as a specification for the CLIPS-based implementation). This implementation effort served to identify a number of shortcomings in the original knowledge model specification. One particular shortcoming relates to factoid 51, which reads 'Sigma has closed all its schools'. This factoid is actually intended to rule out Sigma (a country) as a suspect entity. Given that we know an attack will be against a school (factoid 2), we know that a country must contain schools in order for it to be a suspect. The problem with factoid 51, in this case, is that the closure of schools is intended to mean that there are no schools in Sigma that are *viable* targets. The original (CommonKADS) knowledge model assumed that a country was a suspect entity irrespective of whether the targets located within that country were 'open' or 'closed'. This error perhaps serves to highlight one of the shortcomings of the current ELICIT factoid sets: they require subjects to make particular kinds of interpretations; however, not all of these interpretations are enforced by the semantics of the statements themselves. Obviously, further work is required to investigate this issue. Ideally, one would like the ACT-R agent to interpret factoids in the same way as human subjects, and thus one strategy could be to modify the text of the factoid statements in order to reduce semantic ambiguity. A second strategy could involve an effort to record the kinds of interpretational errors humans make when reading the sentences and then ensure that ACT-R agents make the same sort of errors with similar frequency. In both cases, one can imagine collecting the required data with questionnaires that present each of the factoids and solicit input from respondents in the form of (for example) multiple choices.

IV. Instantiation of an ACT-R Agent for Individual Sensemaking

In order to develop an ACT-R system to investigate collective sensemaking, it is necessary to develop an ACT-R cognitive model that implements the sensemaking process itself. In the present case, it is necessary to develop a cognitive model that can process each of the factoids in the aforementioned factoid set, build a mental model of the prospective situation, and then engage in reasoning processes that progressively eliminate suspect entities. In fact, this is just a *minimum* requirement. ACT-R is intended to model human cognitive processes in a way that is cognitively realistic. This means that an ACT-R sensemaking agent should not just be able to reason over the ELICIT factoids, it should also do so in a way that mimics the strategies adopted by human agents: this is what enables us to gain an explanatory and predictive toehold over human performance in particular experimental contexts. For the purposes of the current modeling and development effort, this constraint was relaxed, and the aim of simply developing a cognitive model that could solve the ELICIT sensemaking task was adopted. The motivation for this departure from standard cognitive modeling practice was based on a number of factors. Firstly, a cursory analysis of a small sample of human individuals ($N = 3$) engaged in the sensemaking task revealed a variety of different strategies. A notable difference was the way in which subjects made use of external resources, particularly pencil and paper. These enabled subjects to create paper-based lists and tables that were modified as each sentence was encountered. These bio-external representations appeared to function as cognitive aids in the problem-solving process, serving as a durable trace of task-relevant information (e.g., reminders as to which groups were not involved in the attack). This suggests that there may not be a uniform way to solve the ELICIT problem, and multiple kinds of model may be required depending on the nature of the task environment that subjects are confronted with (e.g., the kinds of representations and visualizations that are made available by a computer interface). Practical issues also governed the decision not to engage in detailed cognitive modeling for the purposes of this initial development effort. The primary goal of the current activity is to develop an experimental simulation capability that can accommodate multiple cognitive agents (i.e., agents that incorporate at least some of the constraints, characteristics and limitations associated with the human cognitive system). The details of the actual cognitive processing performed by the agents is largely irrelevant to this development effort, although it is clearly an important focus of attention when simulation experiments are being performed.

As with all ACT-R models, the implementation of the ELICIT sensemaking process in ACT-R draws on the use of production rules that match against the contents of the buffers associated with each of the ACT-R modules (see Figure 1). For the purposes of testing the cognitive model, chunks encoding the information content of each of the factoid statements were pre-loaded into declarative memory using the ACT-R `(add-dm)` command. This meant that agents were not required to engage in low-level perceptual processing of the factoid statements, neither were they required to engage in the linguistic analysis of those statements. Instead, the agents began the simulation having effectively 'memorized' all the facts implied by the factoids. Although this is clearly unrealistic relative to the kind of experimental contexts in which the ELICIT task has been investigated, it constituted an important simplifying assumption in the context of the early stages of the modeling effort. In particular, it was important to avoid a situation where assumptions were being made about the specific nature of the task environment (e.g., computer interfaces and communication equipment) that would then influence the modeling effort. In addition, the implementation

```
(P evaluate-group-suicide-bombing-tactics-3
=goal>
    isa                 sensemaking-goal
    selected-dimension  who
    status              evaluate-property
    selected-object     =object
    target-attribute    is-suicide-bomber
=imaginal>
    isa                 statement
    object              =object
    attribute           is-suicide-bomber
    value               no
=retrieval>
    isa                 statement
    object              attack
    attribute           is-suicide-bombing
    value               yes
==>
+imaginal>
    isa                 statement
    object              =object
    attribute           is-suspect
    value               no
    source              self
=goal>
    status              check-suspect-status
)
```

Figure 4: One of 124 productions that enables an ACT-R agent to solve the ELICIT sensemaking task.

of simulated perceptual processing and agent-world interaction mechanisms would have served to complicate what was already a challenging modeling activity.

The result of the modeling effort was an ACT-R cognitive model capable of solving the ELICIT sensemaking task. As with all ACT-R models, the key components of the model are the various chunk-types, chunks and production rules used as part of the solution. A chunk-type, in this case, specifies the structure of a chunk in terms of the slots that it can contain (a chunk-type essentially provides a template for a chunk, in the same way that a class in object-oriented programming provides a template for the data structure of an object). The main chunk-type for the model is the `statement` chunk-type:

```
(chunk-type statement
   object
   attribute
   value
   (is-true yes)
   (source self)
   (confidence 100))
```

The `statement` chunk-type represents a basic fact about the sensemaking situation, which is captured as an `<object, attribute, value>` triple, similar to the triples seen in the Resource Description Framework [38]. The chunk type contains a number of slots, the most of important of which are the `object`, `attribute` and `value` slots. The function of these and other slots is described in Table II.

Aside from chunk-types, the model contained a total of 124 production rules, one of which is shown in Figure 4. Rule execution was controlled by task state information that was stored in the ACT-R goal buffer. In particular, the process of evaluating and eliminating suspect entities was decomposed

TABLE II: Description of the slots associated with the `statement` chunk-type.

Slot	Description
object	Contains the name of an object that features as part of a situation model.
attribute	Contains the name of an attribute associated with the object named in the 'object' slot.
value	Contains the value of the attribute named in the 'attribute' slot.
is-true	Indicates whether the statement made about the object in question is true (yes) or false (no) (default is yes).
source	Specifies the source of the statement. This could be the name of a particular agent, the name of a physical information source, such as a sensor, or the name of an information repository. In cases where the statement exists as the result of an inference made by the agent, the value of the slot will be `self` (this is the default value).
confidence	Indicates the level of confidence the agent has in the truth or falsity of the statement. A value of 100 (default) indicates maximum confidence, whereas a value of 0 indicates no confidence. At present, this slot is not used as part of the reasoning process; however, it could be used in experiments that aim to study the influence of uncertainty on processes at both the individual and collective (team) levels.

into a number of inference steps, which are illustrated in Figure 5 and described in Table III. These inference steps correspond to particular stages in the larger reasoning process, and each one is associated with a particular subset of rules. By recording which inference step is 'active' at any given time, an agent can track its progress and limit the number of rules that can apply. The rule depicted in Figure 4 exemplifies this: the rule features a conditional element that matches to the `status` slot of a `sensemaking-goal` chunk that is contained in the goal buffer. This rule can only be selected for execution when the value of the `attribute` slot corresponds to 'evaluate-property', and this identifies one of the inference steps associated with the larger reasoning process (see Figure 5).

Together, the productions work to enable an ACT-R agent to eliminate suspects and yield the correct answer to the ELICIT sensemaking puzzle. Using a PC with an Intel Quad Core 2.93 GHz processor, 3 GB of RAM and running a 64-bit version of Windows 7, the entire process runs in a simulated time of 16.25 seconds. This, of course, in no way resembles the performance of human subjects in actual experimental contexts – the human subjects take much longer to find the solution to the problem. This serves to highlight the fact that the assumptions made by the current model (e.g., perfect initial memory of all facts, adoption of a robust suspect elimination strategy, and knowledge about what evaluative criteria to apply to each kind of suspect entity) do not apply to the situation seen in many of the experimental studies performed in the context of the ELICIT framework. An additional concern relates to the use of a variety of Lisp functions in order to realize the reasoning process. Although not shown in Figure 4, some of the rules rely on the use of '!bind!' and '!eval!' evaluations in order to invoke Lisp functions. As is noted by the ACT-R 6.0 reference manual [39, p. 165], the use of '!bind!' and '!eval!' in these situations may cause the model to depart from ACT-R theory, and it also creates potential problems for ACT-R's production compilation or rule learning mechanism. In spite of these shortcomings, the ACT-R model presented here does provide a working example

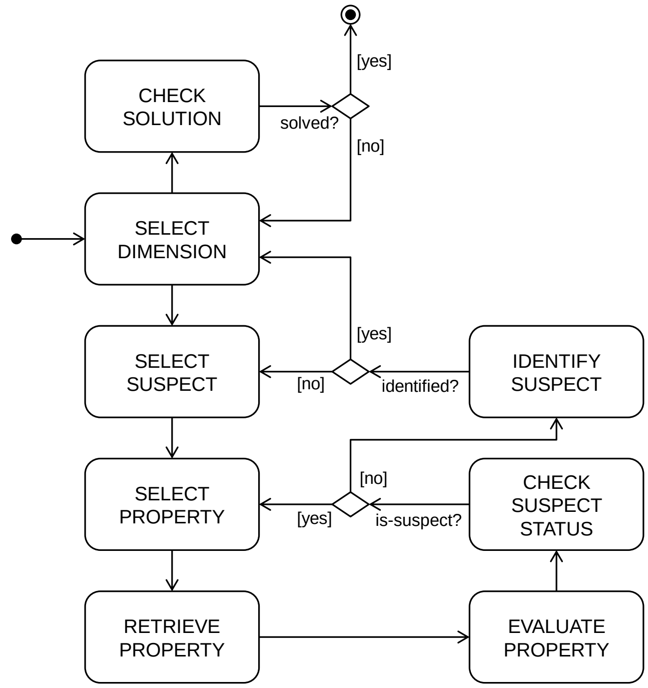

Figure 5: Activity diagram showing the inference steps implemented by an ACT-R agent engaged in the ELICIT sensemaking task.

TABLE III: Stages of the reasoning process that are implemented by the ACT-R cognitive model designed to solve the ELICIT sensemaking task.

Name	**Description**
CHECK-SOLUTION	Determines whether the sensemaking problem has been solved, i.e., whether the who, what, when and where dimensions of the sensemaking task have been resolved.
SELECT-DIMENSION	Selects one of the dimensions (who, what, when or where) of the sensemaking task to focus on.
SELECT-SUSPECT	Selects a particular suspect entity (e.g., a particular group) to focus on.
SELECT-PROPERTY	Selects a particular property (e.g., 'is-suicide-bomber') of the selected suspect entity to evaluate.
RETRIEVE-PROPERTY	Retrieves information about the selected property from declarative memory.
EVALUATE-PROPERTY	Evaluates the information retrieved from memory and assesses whether the suspect status of the selected entity should be modified.
CHECK-SUSPECT-STATUS	Determines whether the selected suspect entity is still a suspect following evaluation of the property retrieved in the 'RETRIEVE-PROPERTY' inference step.
IDENTIFY-SUSPECT	Assesses whether a particular suspect entity (e.g., a particular group) for a particular dimension (e.g., who dimension) can be identified. If so, the agent will proceed to select a different dimension to focus on. If not, another suspect entity will be selected until all suspect entities of a particular type have been evaluated. At this time, control passes back to the 'SELECT-DIMENSION' inference step.

of how a process previously glossed as sensemaking [34] can be implemented in a cognitive architecture that incorporates the kinds of constraints and characteristics presumed to apply in the case of human cognition.

V. An ACT-R Framework for Cognitive Social Simulation

In order to use ACT-R as a platform for cognitive social simulation, a number of extensions were made to the core ACT-R architecture. The most important extension, from the perspective of cognitive social simulation, relates to the creation of a specialized 'messaging' module that enables agents to exchange text messages with other agents. The messaging module features two buffers: 'send-message' and 'receive-message'. The send-message buffer enables agents to post a message to other agents in the simulation, while the receive-message buffer enables agents to check for unread messages and retrieve any messages that are available. A memory-resident Lisp database – the 'message database' – is used to store all the messages sent by agents, and the messaging module interfaces with this database to both create new messages and retrieve existing ones. Together, the messaging module and the message database implement the capability for agents to communicate with one another as part of a socially-distributed, collaborative problem-solving process.

The messages that get communicated by agents are represented by particular kinds of chunks, called `message` chunks. These chunks contain the following slots:

- **source:** Specifies the name of the agent that posted the message.
- **target:** Specifies the intended target of the message. If the value of this slot is 'any', the message will be posted to any agent that the originating agent can communicate with based on the structure of the communication network (see below).
- **text:** Specifies the text of the message to be communicated.

Whenever a `message` chunk is asserted in the send-message buffer as a result of rule execution, the messaging module first determines whether the target agent is a peer of the originating agent (this information is stored in another memory-resident Lisp database, called the 'connection database'). If this is the case, the messaging module will create a new record in the message database that reflects the information content of the `message` chunk. All such records are initially marked as 'unread', reflecting the fact that they have not been processed by the intended recipient.

Whenever an agent wants to retrieve new messages from the database, they can make a request to the receive-message buffer. This will cause the messaging module to check the message database for any unread messages that have been posted to the agent. If any such messages are available, one of the messages will be retrieved and a new `message` chunk will be created in the receive-message buffer. The manner in which unread messages are selected from the database can be controlled by a particular parameter, called the 'message-selection-mode' parameter, which is associated with the messaging module (for example, the most recently posted messages can be selected by setting the message-selection-mode parameter to 'newest'.). Figure 6 presents two production rules that exemplify the process of sending and receiving messages via the buffers of the messaging module.

Once a message is available in the receive-message buffer, the agent needs to interpret the message and translate the contents of the message into one or more of the `statement`

```
(p retrieve-unread-messages
=goal>
    isa       sensemaking-goal
    task      process-messages
?receive-message>
    unread    true
    buffer    empty
==>
+receive-message>
    isa       message
)

(p interpret-and-send-received-message
=goal>
    isa       sensemaking-goal
    task      process-messages
=receive-message>
    isa       message
    text      =text
?send-message>
    state     free
    buffer    empty
==>
+parse-message> =receive-message
+send-message>
    isa       message
    text      =text
    target    any
)
```

Figure 6: Two production rules that exemplify the use of the messaging module to send and receive messages via the send-message and receive-message buffers. The request to the parse-message buffer in the 'interpret-and-send-received-message' rule causes the language module to interpret the received message and create `statement` chunks in the agent's declarative memory module (see main text for details).

chunks that were described in Section IV. In order to accomplish this, the agent makes use of a second module, called the 'language' module. Like the messaging module, this module is a new custom module that does not form part of the core ACT-R architecture (see Figure 1). The language module exposes a single buffer, called 'parse-message' that can receive `message` chunks, 'interpret' the text content of the chunks (i.e., the text contained in the `text` slot of the `message` chunk), and create `statement` chunks reflecting the information content of the message. At the present time, the interpretation process is a simple one involving a string-based pattern matching mechanism. The language module makes the results of the interpretation process available to the agent by using the `(add-dm)` command to assert the `statement` chunks directly into the agent's declarative memory module. This means that the information content of any messages passed to the language module can be accessed by the agent by making retrieval requests to its declarative memory module. The use of the `(add-dm)` command, in this context, allows for situations in which multiple `statement` chunks must be asserted to reflect the factual content of a message – unfortunately, there is not always a one-to-one mapping between the messages that represent a factoid and the `statement` chunks that represent the semantic content of the factoid.

Aside from the message database, the current framework incorporates a number of additional databases. One of these is the connection database, which is used to define the structure of the communication network that exists between agents. This database contains records that specify the direct channels of communication that exist between agents. The database is queried by the messaging module to determine whether an originating agent can post a message to a particular target agent. Another database is the 'triple database'. This is used to store general information about the experimental simulation. For example, the triple database can be used to store information about the number of agents to create, the properties of agents, the particular factoid set to use, the time allowed for agents to complete the sensemaking task, and so on.

VI. Limitations

The approach described in Section V yields a system in which multiple ACT-R agents, each associated with a distinct cognitive model can engage in the sort of reasoning process described in Section IV. Each agent can thus engage in sensemaking in parallel with other agents. In contrast with the situation in Section IV, there is no need to represent all of the factoid-related information in an agent's memory at the outset of the simulation. Instead, agents can be provided with an initial subset of factoids (by creating messages in the message database), and then additional factoids can be made available as the simulation progresses (either as a result of a centralized information distribution mechanism, or, more interestingly, as a result of agents posting messages to each other via their respective messaging modules). As such, this system implements the basic requirements of a system designed to support the execution of cognitive social simulation experiments: each of the agents engages in a sensemaking process and is also able to communicate task-relevant information to other agents at specific junctures in the problem-solving process. As it stands, however, the current system possesses a number of limitations, and these are the focus of future research and development efforts. These limitations are described below.

A. *Question/Answering Capabilities*

At present, ACT-R agents can only communicate factual information to each other; they cannot pose questions for their peers to answer. In order to support question/answering capabilities in the context of the current approach, agents need to be able to 1) recognize the conditions under which a question should be asked, 2) formulate the question in a way that other agents can understand, 3) post the question to other agents and 4) recognize what information is being sought in a question posed by another agent. Of these, only points 1, 2 and 4 are significant challenges – the posting of information to other agents is already addressed by the solution presented in Section V. The challenge related to recognizing the conditions under which a question should be asked could perhaps be addressed by the possibility of what are known as retrieval failures in ACT-R. These are essentially situations in which a request to the retrieval buffer fails to retrieve information from declarative memory. A solution to the fourth challenge could rest on the structure of the `statement` chunk used for knowledge representation (see Section IV). In particular, a question could be encoded as a specialized form of the `statement` chunk in which the `value` slot is either empty or absent. The values of the `object` and `attribute` slots would then signal what information was being requested, and this could be used by a recipient to implement a retrieval request against their own declarative memory module.

B. Communication Strategy

In addition to question/answering capabilities, agents will need to make decisions about what to communicate, when to communicate and who to communicate with. A number of different strategies are possible here. For example, agents could adopt a somewhat passive strategy and only communicate facts when requested to do so, perhaps in response to a particular question. Alternatively, they could adopt an active strategy in which all information is readily communicated. Agents also need to make decisions about who to communicate with. They can thus adopt strategies that might be described as directed or undirected in nature. Directed strategies, in this case, are strategies where agents send messages to particular individuals within their social network; undirected strategies, on the other hand, are ones in which a message is broadcast to all connected agents. A final decision that agents need to make concerns the nature of what should be communicated. Should agents, for example, limit to their communication to information that has been received from external sources (e.g., other agents), or should they include information that has been inferred as a result of their own sensemaking activity? A number of previous studies using cognitive architectures (including ACT-R) have begun to explore these issues in some detail [23][40], and a closer examination of the solutions used in those particular studies is likely to be helpful in the present context.

C. Cognitive Strategy

As discussed in Section IV, the particular strategy adopted by agents to solve the ELICIT sensemaking task is unlikely to be the same as that used as human subjects. In order to better understand the effect of specific manipulations on the performance of human subjects in the task it will thus be necessary to pay closer attention to the kinds of strategies adopted by human subjects. In all likelihood, this will require studies involving the use of protocol analysis techniques [41].

D. Task Environment

Related to the previous point is the need to consider the role that material elements of the task environment play in shaping and influencing cognitive processes at both the individual and collective levels. In cases where humans are tasked with the ELICIT sensemaking problem, they typically resort to using bio-external artefacts, such as pen and paper (see Section IV). The external resources, in these cases, appear to be functioning as a form of environmentally-extended working memory that potentially enhances the cognitive capabilities of the human subject [42]. Inasmuch as ACT-R attempts to duplicate the cognitive constraints and limitations of the human brain, then ACT-R sensemaking agents may need to rely on the same sort of cognitive scaffolding as is seen in the case of their environmentally-situated human counterparts. In particular, it may be that a successful model of complex task performance in naturalistic situations will need to avail itself of computational analogues of the specific aspects of the physical problem-solving environment that are exploited by human subjects. In situations where human subjects are attempting to solve the ELICIT sensemaking problem via computer interfaces, it may thus be necessary to represent elements of the computer interface within the ACT-R simulation. For example, in situations where human subjects are able to view a list of messages that have been posted by other agents, it may be necessary to equip ACT-R agents with an ability to 'see' such messages in order to replicate the kind of mnemonic support that is provided by the real-world list. Recent versions of ACT-R provide support for modeling agent-world interactions through the use of what is called a 'device' object. The use of this object to represent features of the task environment within which agents are situated constitutes an important focus area for future research efforts.

E. Task Features

In addition to these issues, it should be noted that all the factoids in the ELICIT sensemaking task are assumed to be true. This is clearly unlike the situation in real-world sensemaking contexts where agents have to deal with conflicting, uncertain and dynamic information from a constantly evolving situation. Therefore, in addition to the aforementioned issues, future work will need to consider modifications to the ELICIT sensemaking task in order to establish a closer alignment with the kinds of situations faced by real-world sensemakers.

VII. Conclusion and Future Work

The aim of cognitive social simulation is to improve our understanding of the complex inter-play between factors that are spread across the psychological, social and technological domains. This makes cognitive social simulation techniques particularly appealing as a means to undertake experiments into socially-distributed cognition. Cognitive social simulation studies typically rely on the use of cognitive architectures; however, to date, the most widely used cognitive architecture – ACT-R – has seen only limited use in computational studies exploring group-level cognitive dynamics. The current paper reports on the results of an ongoing effort to develop an experimental simulation capability that can be used to undertake studies into socially-distributed cognition using the ACT-R architecture. Using an existing experimental task as a starting point, a cognitive model was developed to show how sensemaking processes could be accommodated within ACT-R. A cognitive social simulation capability was then implemented in ACT-R by relying on the use of a combination of custom modules and memory-resident databases in order to enable agents to exchange information during the course of their sensemaking activities. This solution provides the basic ingredients required to undertake cognitive social simulation experiments into collective sensemaking; however, further research needs to be undertaken in order to improve the communicative capabilities of agents, as well as the task environment in which they are situated (see Section VI). Our future work in this area will aim to improve the sophistication of the ACT-R cognitive model so that agents are able to adopt a variety of communication strategies concerning what information (i.e., factoids) to share and who to share the information with. We will also seek to enable better forms of agent-environment interaction by implementing a custom ACT-R device object. These extensions will support cognitive social simulations that aim to replicate and extend previous results obtained with human subjects in the context of the ELICIT sensemaking task [33].

ACKNOWLEDGMENT

Research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] N. J. Cooke, J. C. Gorman, C. W. Myers, and J. L. Duran, "Interactive team cognition," Cognitive Science, vol. 37, no. 2, 2013, pp. 255–285.

[2] G. R. Semin and E. R. Smith, "Socially situated cognition in perspective," Social Cognition, vol. 31, no. 2, 2013, pp. 125–146.

[3] P. R. Smart and K. Sycara, "Collective sensemaking and military coalitions," Intelligent Systems, vol. 28, no. 1, 2013, pp. 50–56.

[4] G. Theiner, C. Allen, and R. L. Goldstone, "Recognizing group cognition," Cognitive Systems Research, vol. 11, 2010, pp. 378–395.

[5] M. Kearns, "Experiments in social computation," Communications of the ACM, vol. 55, no. 10, 2012, pp. 56–67.

[6] J. Sutton, C. Harris, P. Keil, and A. Barnier, "The psychology of memory, extended cognition, and socially distributed remembering," Phenomenology and the Cognitive Sciences, vol. 9, no. 4, 2010, pp. 521–560.

[7] V. Hinsz, R. Tindale, and D. Vollrath, "The emerging conceptualization of groups as information processors," Psychological Bulletin, vol. 121, 1997, pp. 43–64.

[8] R. Menary, The Extended Mind. Cambridge, Massachusetts, USA: MIT Press, 2010.

[9] P. Robbins and M. Aydede, The Cambridge Handbook of Situated Cognition. New York, USA: Cambridge University Press, 2009.

[10] L. A. Shapiro, Embodied Cognition. Abingdon, Oxfordshire, UK: Routledge, 2011.

[11] W. Mason, F. Conrey, and E. Smith, "Situating social influence processes: Dynamic, multidirectional flows of influence within social networks," Personality and Social Psychology Review, vol. 11, no. 3, 2007, pp. 279–300.

[12] R. Sun, "Cognitive social simulation incorporating cognitive architectures," Intelligent Systems, vol. 22, no. 5, 2007, pp. 33–39.

[13] P. Thagard, "Cognitive architectures," in The Cambridge Handbook of Cognitive Science, K. Frankish and W. M. Ramsey, Eds. Cambridge, UK: Cambridge University Press, 2012, pp. 50–70.

[14] J. R. Anderson, How Can the Human Mind Occur in the Physical Universe? Oxford, UK: Oxford University Press, 2007.

[15] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind," Psychological Review, vol. 111, no. 4, 2004, pp. 1036–1060.

[16] J. E. Laird, A. Newell, and P. S. Rosenbloom, "SOAR: An architecture for general intelligence," Artificial Intelligence, vol. 33, no. 1, 1987, pp. 1–64.

[17] J. E. Laird, The SOAR Cognitive Architecture. Boston, Massachusetts, USA: MIT Press, 2012.

[18] S. M. Rodgers, C. W. Myers, J. Ball, and M. D. Freiman, "Toward a situation model in a cognitive architecture," Computational and Mathematical Organization Theory, vol. 19, no. 3, 2013, pp. 313–345.

[19] J. E. Morrison, "A review of computer-based human behavior representations and their relation to military simulations," Institute for Defense Analyses, Tech. Rep. IDA Paper P-3845, 2003.

[20] D. D. Salvucci, "Modeling driver behavior in a cognitive architecture," Human Factors, vol. 48, no. 2, 2006, pp. 362–380.

[21] D. D. Salvucci and K. L. Macuga, "Predicting the effects of cellular-phone dialing on driver performance," Cognitive Systems Research, vol. 3, no. 1, 2002, pp. 95–102.

[22] G. Gunzelmann, R. L. Moore, D. D. Salvucci, and K. A. Gluck, "Sleep loss and driver performance: Quantitative predictions with zero free parameters," Cognitive Systems Research, vol. 12, no. 2, 2011, pp. 154–163.

[23] D. Reitter and C. Lebiere, "Social cognition: Memory decay and adaptive information filtering for robust information maintenance," in 26th AAAI Conference on Artificial Intelligence, Toronto, Canada, 2012.

[24] G. Klein, B. Moon, and R. R. Hoffman, "Making sense of sensemaking 1: Alternative perspectives," Intelligent Systems, vol. 21, no. 4, 2006, pp. 70–73.

[25] D. T. Moore, "Sensemaking: A structure for an intelligence revolution," National Defense Intelligence College, Tech. Rep., 2011.

[26] P. Pirolli and D. M. Russell, "Introduction to this special issue on sensemaking," Human-Computer Interaction, vol. 26, no. 1-2, 2011, pp. 1–8.

[27] K. E. Weick, Sensemaking in Organizations. Thousand Oaks, California, USA: Sage Publications, 1995.

[28] G. Klein, S. Wiggins, and C. O. Dominguez, "Team sensemaking," Theoretical Issues in Ergonomics Science, vol. 11, no. 4, 2010, pp. 304–320.

[29] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in International Conference on Intelligence Analysis, vol. 5, McLean, Virginia, USA, 2005.

[30] E. Jensen, "Sensemaking in military planning: A methodological study of command teams," Cognition, Technology & Work, vol. 11, no. 2, 2009, pp. 103–118.

[31] S. A. Paul and M. C. Reddy, "Understanding together: Sensemaking in collaborative information seeking," in ACM Conference on Computer-Supported Cooperative Work. Savannah, Georgia, USA: ACM, 2010, pp. 321–330.

[32] M. Ruddy, "ELICIT – the experimental laboratory for investigating collaboration, information sharing and trust," in 12th International Command and Control Research and Technology Symposium (ICCRTS), Newport, Rhode Island, USA, 2007.

[33] M. Manso and M. Ruddy, "Comparison between human and agent runs in the ELICIT N2C2M2 validation experiments," in 18th International Command and Control Research and Technology Symposium (ICCRTS), Alexandria, Virginia, USA, 2013.

[34] M. Ruddy, D. M. Wynn, and J. McEver, "Instantiation of a sensemaking agent for use with ELICIT experimentation," in 14th International Command and Control Research and Technology Symposium, Washington D.C., USA, 2009.

[35] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. R. Shadbolt, W. Van de Velde, and B. Weilinga, Knowledge Engineering and Management: The CommonKADS Methodology. Cambridge, Massachusetts, USA: MIT Press, 2000.

[36] "Unified Modeling Language (UML)," URL: http://www.uml.org/ [accessed: 2014-04-17].

[37] "CLIPS: A Tool for Building Expert Systems," URL: http://clipsrules.sourceforge.net/ [accessed: 2014-04-14].

[38] "RDF – Semantic Web Standards," URL: http://www.w3.org/RDF/ [accessed: 2014-04-14].

[39] "ACT-R 6.0 Reference Manual," URL: http://act-r.psy.cmu.edu/actr6/reference-manual.pdf [accessed: 2014-04-15].

[40] M. Kang, "The effects of agent activeness and cooperativeness on team decision efficiency: A computational simulation study using Team-Soar," International Journal of Human-Computer Studies, vol. 65, no. 6, 2007, pp. 497–510.

[41] N. R. Shadbolt and P. R. Smart, "Knowledge elicitation: Methods, tools and techniques," in Evaluation of Human Work, 4th ed., J. R. Wilson and S. Sharples, Eds. Boca Raton, Florida, USA: CRC Press, in press.

[42] A. Clark, Supersizing the Mind: Embodiment, Action, and Cognitive Extension. New York, New York, USA: Oxford University Press, 2008.