Socially-Distributed Cognition and Cognitive Architectures: Towards an ACT-R-Based Cognitive Social Simulation Capability

Paul Smart*†, Yuqing Tang‡, Paul Stone§, Katia Sycara‡, Stefano Bennati‡, Christian Lebiere‡,
David Mott§, Dave Braines§ and Gavin Powell‖
*Electronics & Computer Science, University of Southampton, Southampton, SO17 1BJ, UK
‡Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
§Emerging Technology Services, IBM United Kingdom Ltd., Hursley Park, Winchester, Hampshire, SO21 2JN, UK
‖Airbus Group Innovations, Newport, Gwent, NP10 8FZ, UK

†Corresponding author – ps02v@ecs.soton.ac.uk

***Abstract*—ACT-R is one of the most widely used cognitive architectures, and it has been used to model hundreds of phenomena studied in the cognitive psychology literature. In spite of this, there are relatively few studies that have attempted to apply ACT-R to situations involving social interaction. This is an important omission since the social aspects of cognition have been a growing area of interest in the cognitive science community, and an understanding of the dynamics of collective cognition is of particular importance in many organizational settings. In order to support the computational modeling and simulation of socially-distributed cognitive processes, a simulation capability based on the ACT-R architecture is described. This capability features a number of extensions to the core ACT-R architecture that are intended to support social interaction and collaborative problem solving. The core features of a number of supporting applications and services are also described. These applications/services support the execution, monitoring and analysis of simulation experiments. Finally, a system designed to record human behavioral data in a collective problem-solving task is described. This system is being used to undertake a range of experiments with teams of human subjects, and it will ultimately support the development of high fidelity ACT-R cognitive models. Such models can be used in conjunction with the ACT-R simulation capability to test hypotheses concerning the interaction between cognitive, social and technological factors in tasks involving socially-distributed information processing.**

I. INTRODUCTION

The social context in which human cognition occurs has been the focus of considerable scientific and philosophical attention in recent years. Much of the interest has centered on the extent to which cognition can be seen as socially-distributed, or the extent to which groups can function as cognitive systems capable of implementing processes that are recognizably cognitive in nature [1, 2, 3, 4]. The social context of cognition has also been an important focus of attention in the human factors and cognitive engineering literatures. Here, the main thrust of the research has been on the design of technological systems to support collaborative problem-solving and decision-making [5]. There has also been an effort to understand the factors that drive and influence performance in the context of small task-oriented teams [6]. Finally, in the network sciences, there has been a growing interest in how information and communication networks can be used to exploit the collective cognitive potential of large numbers of networked individuals [7]. This interest is evident in research into social computation [8] and collective intelligence systems [9]. It is also apparent in recent research relating to so-called ‘social machines’, where the socially-embedded nature of the World Wide Web has been seen as important not only for the augmentation of human cognitive capabilities but also for the realization of new forms of socially-scaffolded machine intelligence [10].

What unites all these areas of research is an interest in the way in which cognitive, social and technological factors work to influence performance outcomes at both the individual and collective levels. These issues are, of course, of considerable importance in many organizational settings, where many forms of problem solving and decision-making activity rely on the coordinated interaction of multiple individuals. Military command and control capabilities, for example, are often ones that are influenced by factors that extend across the cognitive, social and technological domains, and this underlies the need to understand the effect of these factors on collective performance outcomes (e.g., [11, 12]). Insights into the interaction between cognitive, social and technological factors in these situations can assist with the attempt to actively engineer the social and technological environment in ways that enhance the decision making capabilities of military command and control teams.

In the attempt to gain a better understanding of the factors that affect performance in social contexts, researchers have resorted to a variety of mathematical modeling and computer simulation techniques [13]. Among these, the use of multi-agent systems to study the dynamics of collective behavior has emerged as a particularly important paradigm. For the most part, these multi-agent studies (commonly referred to as social simulation studies) have relied on agents that feature little or no internal cognitive processing capabilities. Most of the commonly available social simulation tools (e.g., Swarm [14] and RePast [15]) assume that individual agents have very rudimentary cognitive processing capabilities, and in some cases, aspects of agent cognition (e.g., beliefs or attitudes) are represented by simple numerical values that do not respect the constraints imposed by the human cognitive system.

In order to improve the cognitive sophistication and fidelity of social simulation experiments, Sun [16] has advocated the use of what are called cognitive architectures. Cognitive architectures are computational frameworks that make particular commitments about the kinds of mental representations and computational procedures that are sufficient to explain important aspects of human cognition, such as problem solving, memory and learning [17]. As a result, cognitive architectures can be used to impose realistic constraints on the cognitive capabilities of synthetic agents in the context of social simulation experiments. The integration of cognitive architectures into social simulation studies results in a form of computer simulation technique that Sun [16] refers to as *cognitive social simulation*. By factoring cognitive architectures into social simulation experiments, Sun argues that 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 variables (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, and they also support the attempt to better understand the interaction between cognitive, social and technological factors in the context of socially-distributed cognition.

The current paper describes an ongoing effort to develop a cognitive social simulation capability based on a particular cognitive architecture, called Adaptive Control of Thought-Rational (ACT-R) [18, 19] (see Section II). Although ACT-R is one of the most widely used cognitive architectures, there are relatively few studies that have used ACT-R to study cognition in social contexts. Given the importance of socially-distributed cognition as a focus area for scientific study, coupled with the need to incorporate cognitively realistic design constraints into multi-agent simulations, there is a compelling need to better understand how to apply cognitive architectures to tasks involving socially-distributed information processing. The specific aim of our research is to develop a generic capability for running cognitive social simulation experiments and then apply this capability to a specific task. The capability we aim to develop is hereafter referred to as the ACT-R Cognitive Social Simulation Capability or ACT-R CSSC (see Section IV), and the task that is the focus of our current modeling and simulation efforts is a task that has previously been used to investigate socially-distributed problem solving as part of what is called the ELICIT experimentation framework [20] (see Section III). ELICIT, in this case, is an acronym that stands for the Experimental Laboratory for Investigating Collaboration, Information Sharing and Trust. It represents a sustained effort to advance our understanding of the factors that affect collective performance in a complex task that draws on factors spread across the cognitive, social and technological domains. These features make the task suitable for testing the ability of the ACT-R CSSC to support experimental studies into socially-distributed cognition.

The main aim of the current paper is to briefly describe the core features of the ACT-R CSSC (see Section IV), as well as a number of supporting applications and services (see Sections V and VI). The paper also describes recent work to develop a human experimentation platform for undertaking experiments with human subjects (see Section VII). The results of studies using this platform will support the development of detailed (high fidelity) ACT-R cognitive models that can be used as the basis for cognitive social simulation experiments using the ACT-R CSSC.

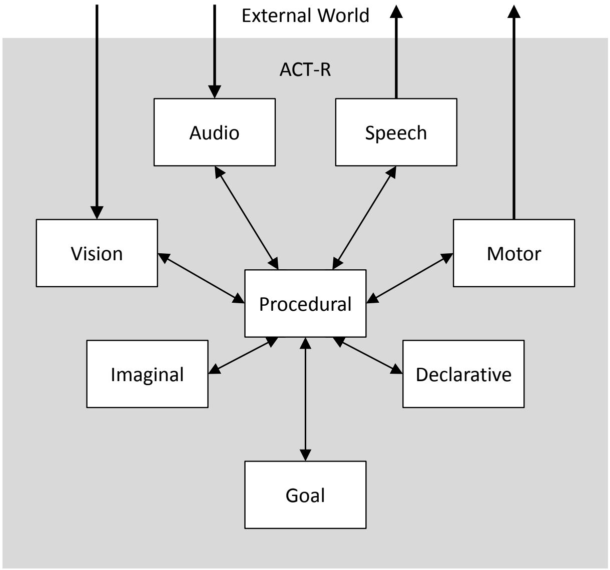


Fig. 1. The core modules of the ACT-R v.6 cognitive architecture.

II. ACT-R

Sun [16, 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 concerns 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.

ACT-R consists of a number of modules (see Fig. 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.

The core ACT-R modules (and their associated buffers) tend to form the basis of most ACT-R cognitive 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. [21] 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" [21, 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" [22, p. 24]. For the most part, however, ACT-R has been used to model the performance of individual subjects engaged in relatively simple cognitive tasks. Despite the growing interest in the social context of cognition (see Section I), there have been relatively few studies that have sought to use ACT-R for the purposes of cognitive social simulation. One example of a study that does use ACT-R to explore the interactions between cognitive and social factors in a group problem solving task is a study by Reitter and Lebiere [23]. This study serves to demonstrate the feasibility of using ACT-R to model human performance in tasks requiring social interaction and engagement. In addition, ACT-R has been used to model aspects of, what might be described as, *socially-situated cognition*. In this case, the emphasis is not so much on collective cognitive performance as on the ability of individual agents to adapt their cognitive and behavioral responses to the demands of the social environment. As an example of this kind of work, Best and Lebiere [24] describe the attempt to integrate ACT-R into a multi-player virtual environment in order to support Military Operations on Urban Terrain (MOUT) training simulations. Individual ACT-R agents or models[1], in this case, interact with other agents (which may be human or synthetic) in order to engage in coordinated tactical manoeuvres based on military doctrine.

III. ELICIT EXPERIMENTATION FRAMEWORK

In order to support the effort to develop a cognitive social simulation capability based on ACT-R, it helps to have a concrete task to focus on. The task that has been selected in the context of the current work is a task that forms part of the ELICIT experimentation framework [20]. The ELICIT experimentation framework consists of the ELICIT experimentation platform, which corresponds to the software environment used to run ELICIT experiments; the ELICIT task, which is the actual task performed by experimental participants; and the ELICIT factoids, which are the pieces of task-relevant information that are processed by experimental participants during the course of task execution. The ELICIT experimentation platform consists of the various computer interfaces presented to human subjects during the ELICIT task. These interfaces comprise what we refer to as the ELICIT task environment.

The ELICIT 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 (e.g., the likely location and target of the attack). In order to correctly identify the features of the attack, subjects need to share factoids with one another. This sharing occurs either as a result of subjects posting messages directly to one another, or by subjects posting factoids to shared information repositories, called 'websites'. The profile of information sharing in the task is a function of both subject characteristics (e.g., the subject's willingness to share information), as well as the structure of the organizational environment in which the task is performed. For example, a subject's access to particular websites can be controlled through the setting of 'access rights'. In addition, the structure of the communication network can be configured to limit the channels of communication between individual subjects. In the case of a fully-connected communication network, each team member can send messages to every other member of the team; however, more restrictive communication policies can be created to limit the profile of inter-agent communication. In general, the accessability of the websites (website access policy) and the structure of the communication network (communication policy) are configured so as to investigate the effect of different organizational environments on the team's ability to solve the ELICIT problem (e.g., [11]).

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

The decision to use the ELICIT task in the current context is reflected in our recent cognitive modeling and software development activities. Firstly, an initial ACT-R cognitive model was developed to generate test data and to also assess the feasibility of implementing socially-distributed processes in ACT-R (see [25] for more details). Secondly, we have developed a human experimentation platform based on a variant of the ELICIT task (see Section VII). This platform supports the effort to gather detailed bodies of human performance data as the basis for the developing high fidelity cognitive models.

IV. ACT-R COGNITIVE SOCIAL SIMULATION CAPABILITY

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. Together these extensions constitute what we refer to as the ACT-R Cognitive Social Simulation Capability, or ACT-R CSSC. The ACT-R CSSC consists of a number

---

[1] In the context of cognitive social simulation studies, each ACT-R agent typically corresponds to an instantiated ACT-R model. In this paper, the terms 'ACT-R agent' and 'ACT-R model' are synonymous.

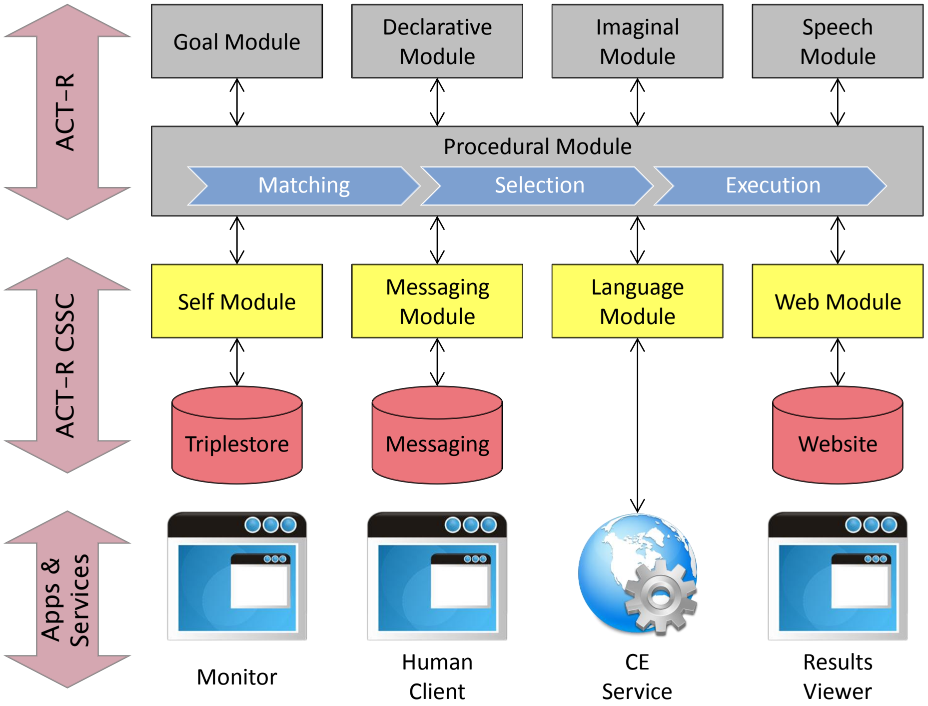


Fig. 2. Components of the ACT-R CSSC. The modules and databases that comprise the ACT-R CSSC are built on top of the core ACT-R architecture and occupy the middle layer of the diagram (note that not all of the modules of the core ACT-R architecture are shown in this diagram). The applications and services listed on the bottom layer are intended to support the use and operation of the ACT-R CSSC. These are described in Sections V and VI.

of custom modules that build on the existing modules of the core ACT-R architecture (see Section II), as well as a set of databases that are responsible for storing information relevant to a particular simulation.

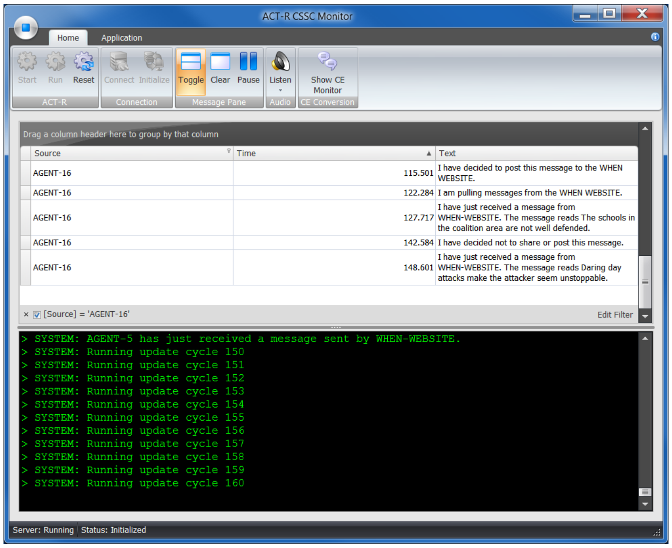
Fig. 2 shows some of the modules and databases that are currently implemented as part of the ACT-R CSSC. The most important module, from the perspective of cognitive social simulation, is the 'messaging' module. This module enables agents to exchange text messages with other agents. For example, ACT-R agents can use the module to make requests to retrieve text messages sent by other agents. They can also use the module to post messages to other agents within their immediate social network. Each message takes the form of an ACT-R chunk that consists of `source`, `target` and `text` slots. The `source` and `target` slots specify the agent who posted the message and the agent that is the intended recipient of the message. The `text` slot contains the content of the message. For the purposes of the current work, the use of messages is limited to the communication of ELICIT factoids. As a result, the content of each message corresponds to the text of a particular factoid. Given that the ELICIT factoids are expressed in natural language, and ACT-R is only able to process information expressed in the form of chunks, additional processing is required to enable ACT-R agents to interpret the content of messages. This capability is provided by the language processing components of the ACT-R CSSC, namely, the language module (see Table I) and the Controlled English (CE) Service (see Section VI).

TABLE I. CUSTOM ACT-R MODULES IMPLEMENTED AS PART OF THE ACT-R CSSC.

Module	**Description**
Messaging	The messaging module enables agents to exchange messages with their immediate peers in the social network defined for a particular experiment.
Self	The purpose of the self module is to make information about the self, such as agent personality characteristics, available to the procedural module in order to influence the dynamics of rule execution. In the case of the current work, this module is used to store information about the characteristics that are typically associated with multi-agent simulations involving the ELICIT task (see [26, 27]).
Web	The web module is intended to emulate the functionality of the ELICIT experimentation platform in respect of the ability to interact with ELICIT websites. The more generic use of the module is to support the access of agents to shared network-accessible information repositories.
Language	The language module provides the functionality to interpret the textual content of messages. The aim of the language module is to convert messages formulated in natural language into a set of chunks that capture the meaning of the message. These chunks can then be used by ACT-R models to drive domain-relevant reasoning processes.

In addition to the custom modules, a number of memory-resident Lisp 'databases' were created to support the storage of information relating to both the task and experimental simulation. These databases are implemented within the same Lisp environment as that hosting ACT-R. The result of this implementation strategy is that relevant information in the databases can be accessed directly from within ACT-R without relying on external third-party components. A complete description of the databases implemented as part of the ACT-R CSSC is presented in Smart et al. [28].

Together, the custom modules and databases provide the infrastructure to run a range of cognitive social simulation exper-



Source	Time	Text
AGENT-16	115.501	I have decided to post this message to the WHEN WEBSITE.
AGENT-16	122.284	I am pulling messages from the WHEN WEBSITE.
AGENT-16	127.717	I have just received a message from WHEN-WEBSITE. The message reads The schools in the coalition area are not well defended.
AGENT-16	142.584	I have decided not to share or post this message.
AGENT-16	148.601	I have just received a message from WHEN-WEBSITE. The message reads Daring day attacks make the attacker seem unstoppable.

☑ [Source] = 'AGENT-16'

```
> SYSTEM: AGENT-5 has just received a message sent by WHEN-WEBSITE.
> SYSTEM: Running update cycle 150
> SYSTEM: Running update cycle 151
> SYSTEM: Running update cycle 152
> SYSTEM: Running update cycle 153
> SYSTEM: Running update cycle 154
> SYSTEM: Running update cycle 155
> SYSTEM: Running update cycle 156
> SYSTEM: Running update cycle 157
> SYSTEM: Running update cycle 158
> SYSTEM: Running update cycle 159
> SYSTEM: Running update cycle 160
```

Server: Running | Status: Initialized

Fig. 3. Screenshot of the ACT-R CSSC Monitor application.

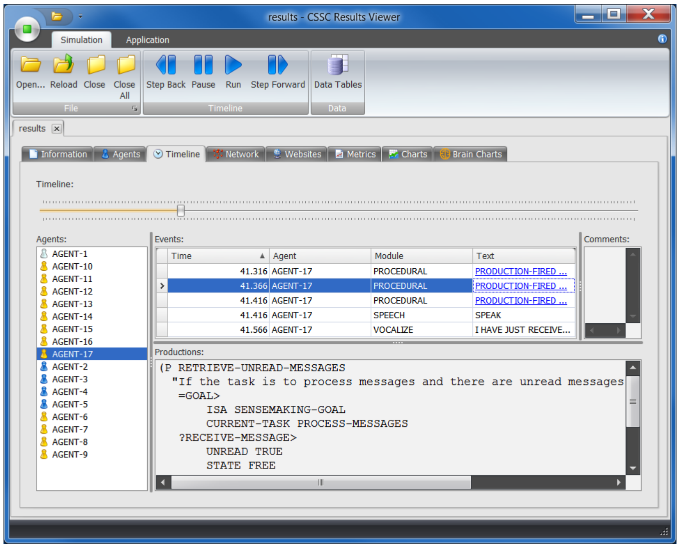
iments using ACT-R. ACT-R cognitive models that implement task-specific cognitive processing routines can make use of this infrastructure in order to support inter-agent communication and control the access of ACT-R agents to specific network-accessible information repositories (e.g., ELICIT websites). This system therefore implements the basic requirements of a system designed to support the execution of cognitive social simulation experiments: each of the agents engages in task-relevant cognitive processing and is also able to communicate information to other agents at appropriate junctures in the larger (socially-distributed) problem-solving process.

V. ACT-R CSSC APPLICATIONS

The ACT-R CSSC is implemented entirely in Lisp and is intended to be run with the standalone versions of the ACT-R system that are available for download from the ACT-R website [29]. This is intended to minimize the overhead associated with installation and configuration. In order to simplify the process of running experiments with the ACT-R CSSC, a number of Windows desktop applications have been implemented. These can be used to initiate simulation experiments, monitor the progress of simulations and view the results of completed experiments. Subsequent sections describe two of the applications that are currently available to support the use of the ACT-R CSSC.

A. *ACT-R CSSC Monitor*

The ACT-R CSSC Monitor application is a conventional Windows desktop application implemented on top of the .NET framework (see Fig. 3). It provides support for the execution and monitoring of simulation experiments. Using the application, a user can launch the ACT-R CSSC in the background and run experiments with a variety of different simulation parameters. The application communicates with the ACT-R CSSC using Transport Control Protocol (TCP) sockets. Bidirectional communication through these sockets enables the Monitor application to initiate experimental simulations, and it also enables the ACT-R CSSC to provide information regarding the progress of ongoing simulations. The main advantage of the Monitor application is that it abstracts away from the use of the ACT-R CSSC and reduces the need for the user to understand Lisp. As an added benefit, the Monitor application enables the user to attach a speech synthesis process to selected ACT-R agents in order to generate an audible trace of agent reasoning activity[2].



Agents:
AGENT-1
AGENT-10
AGENT-11
AGENT-12
AGENT-13
AGENT-14
AGENT-15
AGENT-16
AGENT-17
AGENT-2
AGENT-3
AGENT-4
AGENT-5
AGENT-6
AGENT-7
AGENT-8
AGENT-9

Events:

Time	Agent	Module	Text
41.316	AGENT-17	PROCEDURAL	PRODUCTION-FIRED ...
41.366	AGENT-17	PROCEDURAL	PRODUCTION-FIRED ...
41.416	AGENT-17	PROCEDURAL	PRODUCTION-FIRED ...
41.416	AGENT-17	SPEECH	SPEAK
41.566	AGENT-17	VOCALIZE	I HAVE JUST RECEIVE...

Productions:

```
(P RETRIEVE-UNREAD-MESSAGES
  "If the task is to process messages and there are unread messages
   =GOAL>
       ISA SENSEMAKING-GOAL
       CURRENT-TASK PROCESS-MESSAGES
   ?RECEIVE-MESSAGE>
       UNREAD TRUE
       STATE FREE
```

Fig. 4. The ‘Timeline Tab’ of the ACT-R CSSC Results Viewer application.

B. *ACT-R CSSC Results Viewer*

The ACT-R CSSC stores the results of simulations as a set of plain text files. These files can be loaded by another application, called the ACT-R CSSC Results Viewer. This application was developed to support the visualization and analysis of simulation results. It provides access to all of the data generated by a simulation experiment in the form of a tabbed display. The ‘Timeline Tab’ (see Fig. 4), for example, displays (among other things) all the rules that were executed by particular agents at each step of the simulation, and it therefore enables an experimenter to gain a better understanding of the behavior of agents and the reasons for their actions. Other tabs provide access to information concerning agent properties (‘Agent tab’), the structure of the communication network (‘Network Tab’), the accessibility of particular websites to particular agents (‘Websites Tab’), the value of specific metrics that are typically encountered in the context of ELICIT experiments (‘Metrics Tab’) (e.g., the cognitive self-synchronization values reported by Manso [27]), and a set of graphical charts that summarize the results of the simulation (‘Charts Tab’).

VI. LANGUAGE PROCESSING

As mentioned in Section IV, the factoids used in the ELICIT task are expressed in natural language. This presents a challenge for ACT-R cognitive models because ACT-R does not come equipped with a natural language processing capability. In order to coordinate cognitive processes with

---

[2] The input for this auditory output is provided by the ACT-R model. Any output routed through ACT-R’s speech module (see Fig. 1) can be listened to via the speech synthesis engine.

TABLE II. EXAMPLES OF ELICIT FACTOIDS EXPRESSED IN BOTH NATURAL LANGUAGE AND THE 'EXTRACTED CE' FORMAT.

Natural Language	Extracted CE
The Gray group is recruiting locals – intentions unknown	the group Graygroup recruits the group locals.
The Silver group does not work in Pi	the group Silvergroup does not operate in the country Pi.
The Blue group has been known to use suicide bombers	the group Bluegroup uses the suicide bomber 'a suicide bomber'.

respect to the semantic content of the ELICIT factoids, the factoids need to be re-represented in a form that ACT-R agents can 'understand'; i.e., the factoids need to be converted into ACT-R chunks.

The ACT-R CSSC incorporates two mechanisms for dealing with natural language expressions, both of which are encapsulated within the language module component of the ACT-R CSSC (see Table I). The first mechanism is a simple string-based pattern matching mechanism that matches against the text of a particular factoid and asserts chunks corresponding to the information content of the factoid. This solution strategy requires that appropriate chunk specifications be manually hard coded into a Lisp function that is called by the language module prior to the execution of any experiments. The main drawback of this approach is that it limits the range of messages that can be exchanged by ACT-R agents: all messages need to conform to a structure that is consistent with the pattern matching mechanism. The approach is viable (albeit impractical) for situations in which a finite number of messages will be exchanged and the structure of those messages is known in advance (as is the case with the ELICIT task); however, it is unsuitable for situations in which ACT-R agents need to deal with less constrained forms of information exchange, as might be the case where ACT-R agents are required to inter-operate with human agents and no constraints are placed on the structure of exchanged messages. Ideally, what is required is a natural language processing capability that can derive the semantic content of textual messages and convert this into a chunk-based format that ACT-R agents can incorporate into their cognitive processing routines.

In the context of the current work, we have established an interface between the language module component of the ACT-R CSSC and a language processing service that is provided by a system called the CE Store [30]. The CE Store is a system developed by IBM to support the creation, storage and manipulation of semantically-rich knowledge models expressed in a controlled natural language, called Controlled English (CE). Such models have been used for a variety of purposes, most notably to support information extraction [31] and enable human-machine communication [32]. In the current context, the CE model for natural language processing [31] was extended to support the processing of a small subset ($N = 3$) of ELICIT factoids. This model was used by a natural language processing agent to transform the natural language statements of the ELICIT factoids into a more formal CE-based representation, called 'extracted CE'. This intermediate CE representation has the advantage of being both human-readable and amenable to services that convert the 'extracted CE' statements into semantically-equivalent ACT-R chunks (Table II presents some examples of ELICIT factoids expressed in natural language and 'extracted CE'). The conversion service is implemented as part of the CE Store and uses an additional CE model to effect the actual transformation from 'extracted CE' to ACT-R chunks. By using this service, the ACT-R CSSC is able to support experiments in which ACT-R agents (possibly in conjunction with human subjects) communicate with one another using CE. This provides considerable flexibility in terms of the integration of natural language processing capabilities into ACT-R-based experiments. For example, the CE models that implement the various transformational steps (from natural language through to ACT-R chunks) can be adapted to support different problem domains and different kinds of collaborative activity.

The ACT-R CSSC communicates with the CE Store via a Web service interface, referred to as the CE Service (see Fig. 2). The language module of the ACT-R CSSC exploits this service by making requests to the ACT-R CSSC Monitor (see Section V-A), which then routes the request to the CE Service[3]. The chunk specifications that are returned from the CE Service are subsequently posted back to the ACT-R CSSC where they are used to create chunks within the ACT-R model that initiated the request.

Regardless of whether ELICIT factoids are processed by the string-based pattern matching mechanism or the CE Service, the chunks that result from the conversion process are ultimately made available to ACT-R agents by means of the `(add-dm)` command. This command asserts the chunks directly into the agent's declarative memory module. The use of the ACT-R `(add-dm)` command, in this context, allows for situations in which multiple chunks must be asserted to reflect the information content of a message – unfortunately, there is not always a one-to-one mapping between the messages that represent a factoid and the chunks that are used to represent the semantic content of the factoid. An alternative strategy for making chunks accessible to ACT-R agents would rely on a buffer associated with the language module; however, buffers in ACT-R can only hold a single chunk at any one time. In situations where multiple chunks are returned by the language processing components, the chunks would therefore need to be cached locally within the module and then passed to the buffer in a serial manner.

VII. HUMAN EXPERIMENTATION PLATFORM

In order to support the development of high fidelity cognitive computational models of collaborative problem solving, it is necessary to obtain detailed behavioral records of human subjects engaged in the target task. Such models can be used to run experiments that compare the performance of ACT-R agents with their human counterparts under similar experimental conditions. In the context of the current work, we have developed a human experimentation platform, called ACT-R-Web, to support the acquisition of behavioral data from human subjects in a number of experimental contexts. The platform is intended to support experiments that investigate the various information sharing strategies used by participants, the conditions under which these strategies are used and their effect on collective performance. It is also intended to support studies that investigate the effect of communicating

---

[3] This use of the Monitor application to route requests to the CE Store avoids the need to introduce a dependency on third party Lisp packages supporting Hypertext Transfer Protocol (HTTP) requests.

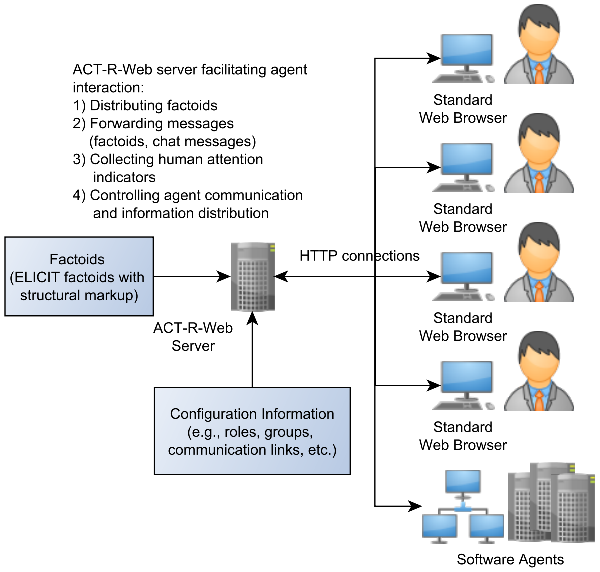


Fig. 5. The ACT-R-Web human experimentation platform.

guesses (i.e., provisional attempts to identify the features of the terrorist attack as part of the ELICIT task) to teammates. A particular focus of attention concerns whether a participant's knowledge of the guesses of their teammates influences the emergence of cognitive biases (e.g., anchoring effects). We have begun a series of human experiments using the ACT-R-Web platform as the basis for future cognitive modeling efforts. These experiments are based on a variant of the ELICIT task in which the original ELICIT factoids are enriched with 'structural markup', e.g., keywords.

A. *ACT-R-Web Platform*

The architecture of the ACT-R-Web platform is depicted in Fig. 5. The platform consists of 1) an ACT-R-Web server, 2) a collection of Web browser clients, and 3) an optional collection of software agents. Human subjects interact with each other through the ACT-R-Web server using their Web browser clients. The software agents are intended to support the integration of ACT-R agents into experiments with human subjects (see Section VII-B).

At the beginning of an experiment, the ACT-R-Web server loads a set of factoids with structural markup, and then distributes these factoids to participants. Upon receiving these factoids, human subjects can organize and share their received factoids and chat with other human participants. The ACT-R-Web server is capable of regulating the interaction between the participants according to a configuration file that specifies the roles of participants, the grouping of participants, the communication links between participants, and so on. The server is also able to exert control over the timing of inter-agent communication and the distribution of information. For example, an experimenter can specify when factoids should be distributed, when to allow message forwarding, when to allow chatting, when to enable the submission of answers, and so on.

The Web browser client of the platform allows human subjects to participate in the experiments using any device with a modern Web browser capable of supporting HTML5 and JavaScript. The Web interface itself is composed of four major components: 1) the 'Inbox' component, which displays the received factoids; 2) the 'MyList' component, which allows human subjects to store and organize factoids using drag-and-drop operations; 3) the 'Guessbox' component, which allows human subjects to share his/her guesses with other participants; and 4) the 'Outbox' component, which allows human subjects to share factoids with other participants.

The Web browser client collects various types of data relating to cognitive function. For example, attention to particular factoids is indicated by mouse clicks on factoid messages. By default, factoids are displayed in an abridged form, appearing as a list of keywords, such as 'Cyan ⋯ Fuschia ⋯ year round'. This abridged form is expanded to the complete message – e.g., 'The Cyan and Fuchsia groups are capable of attacking year round' – whenever the factoid is selected using the mouse. Other behaviors that are logged include those associated with the forwarding of messages, chatting to other participants, organizing factoids in the 'MyList' component, and searching for specific factoids using keyword searches.

B. *Software Agents*

A simple software agent architecture has been implemented to support the integration of ACT-R agents into experiments using the ACT-R-Web platform. These agents can run on separate machines or separate processes and can communicate with human subjects through the ACT-R-Web server using HTTP connections. In future work, we intend to use this architecture to validate the ACT-R cognitive models with respect to human performance data.

VIII. CONCLUSION

The aim of cognitive social simulation is to improve our understanding of the complex inter-play between factors that are spread across the cognitive, 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, the most widely used cognitive architecture – ACT-R – has, to date, seen only limited use in studies involving social interaction and inter-agent communication. 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. The simulation capability consists of a number of custom ACT-R modules and memory-resident databases that together comprise what we have referred to as the ACT-R CSSC. This system inter-operates with a number of additional applications and services that are intended to support the execution and monitoring of simulation experiments, the visualization and analysis of simulation results and the transformation of natural language expressions into ACT-R chunks. The ACT-R CSSC and supporting applications have been tested with an initial ACT-R cognitive model (see [25]) that captures some of the key cognitive and behavioral dynamics of the ELICIT task. This model is a low-fidelity cognitive model in the sense that it is not based on detailed analyses of human behavior. In order to support the development of high-fidelity cognitive models, it is important to gather detailed behavioral data from human subjects that are actively engaged in the target task. This is the purpose of the ACT-R-Web human experimentation platform. The ACT-R-Web platform incorporates the functionality needed to monitor and record the behavior of human subjects engaged in a task that closely resembles that used in the case of ELICIT experiments. The results of studies using this platform will support the development of high-fidelity cognitive models that can be used as the basis for future cognitive social simulation experiments using the ACT-R CSSC.

ACKNOWLEDGEMENT

This research was sponsored by the U.S. Army Research Laboratory and the U.K. 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 author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] E. Hutchins, *Cognition in the Wild*. Cambridge, Massachusetts, USA: MIT Press, 1995.

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

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

[4] G. Theiner, “Varieties of group cognition,” in *The Routledge Handbook of Embodied Cognition*, L. A. Shapiro, Ed. New York, New York, USA: Routledge, 2014.

[5] G. Convertino, D. Billman, P. Pirolli, J. Massar, and J. Shrager, “The CACHE study: Group effects in computer-supported collaborative analysis,” *Computer Supported Cooperative Work*, vol. 17, no. 4, pp. 353–393, 2008.

[6] N. J. Cooke, J. C. Gorman, and W. J. L, “Team cognition,” in *Handbook of Applied Cognition*, 2nd ed., F. T. Durso, R. S. Nickerson, S. T. Dumais, S. Lewandowsky, and T. J. Perfect, Eds. Chichester, UK: John Wiley & Sons, 2007.

[7] W. Mason and D. J. Watts, “Collaborative learning in networks,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 3, pp. 764–769, 2012.

[8] M. Kearns, “Experiments in social computation,” *Communications of the ACM*, vol. 55, no. 10, pp. 56–67, 2012.

[9] T. W. Malone, R. Laubacher, and C. Dellarocas, “The collective intelligence genome,” *MIT Sloan Management Review*, vol. 51, no. 3, pp. 21–31, 2010.

[10] P. R. Smart and N. R. Shadbolt, “Social machines,” in *Encyclopedia of Information Science and Technology*, M. Khosrow-Pour, Ed. Hershey, Pennsylvania, USA: IGI Global, in press.

[11] 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.

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

[13] M. Prietula, K. Carley, and L. Gasser, *Simulating Organizations: Computational Models of Institutions and Groups*. Boston, Massachusetts, USA: The MIT Press, 1998.

[14] “SWARM - Summary,” URL: http://savannah.nongnu.org/projects/swarm [accessed: 2014-04-21].

[15] “Repast Suite,” URL: http://repast.sourceforge.net/ [accessed: 2014-04-21].

[16] R. Sun, “Cognitive social simulation incorporating cognitive architectures,” *Intelligent Systems*, vol. 22, no. 5, pp. 33–39, 2007.

[17] 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.

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

[19] 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, pp. 1036–1060, 2004.

[20] 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.

[21] 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, pp. 313–345, 2013.

[22] 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.

[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] B. J. Best and C. Lebiere, “Teamwork, communication, and planning in ACT-R agents engaging in urban combat in virtual environments,” in *Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, Acapulco, Mexico, 2003.

[25] P. R. Smart and K. Sycara, “Cognitive social simulation and collective sensemaking: An approach using the ACT-R cognitive architecture,” in *6th International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE’14)*, Venice, Italy, 2014.

[26] D. M. Wynn, M. Ruddy, and M. E. Nissen, “Command & control in virtual environments: Tailoring software agents to emulate specific people,” in *15th International Command and Control Research and Technology Symposium*, Santa Monica, California, USA, 2010.

[27] M. Manso, “N2C2M2 validation using abELICIT: Design and analysis of ELICIT runs using software agents,” in *17th International Command and Control Research and Technology Symposium*, Fairfax, Virgina, USA, 2012.

[28] P. R. Smart, D. P. Richardson, K. Sycara, and Y. Tang, “Towards a cognitively realistic computational model of team problem solving using ACT-R agents and the ELICIT experimentation framework,” in *19th International Command and Control Research Technology Symposium (ICCRTS’14)*, Alexandria, Virginia, USA, 2014.

[29] “ACT-R,” URL: http://act-r.psy.cmu.edu/ [accessed: 2014-04-20].

[30] “IBM Controlled Natural Language Processing Environment,” URL: http://ibm.co/RDIa53 [accessed: 2014-07-23].

[31] D. Mott, S. Poteet, P. Xue, and A. Copestake, “Natural language fact extraction and domain reasoning using controlled english,” in *10th Annual DELPH-IN Summit*, Tomar, Portugal, 2014.

[32] A. Preece, D. Braines, D. Pizzocaro, and C. Parizas, “Human-machine conversations to support multi-agency missions,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 1, pp. 75–84, 2014.