“Consciousness” And Conceptual Learning In A Socially Situated Agent

Myles Bogner[1], Uma Ramamurthy[2], and Stan Franklin[3]
Conscious Software Research Group[4]
Institute of Intelligent Systems
The University of Memphis

1. Introduction

Wheeler notes, “orthodox cognitive science claims that situated (world-embedded) activity can be explained as the outcome of in-the-head manipulations of representations by computational information processing mechanisms” (1994). He points out the difficulty AI encountered “in moving from toy worlds to dynamic unconstrained environments,” and further argues that such difficulties are inevitable within the stated paradigm. Instead, he advocates systems that “exhibit dynamical profiles comparable to those displayed by biological neural networks, and … play the same adaptive role as biological networks, i.e., to function as the control systems for complete situated agents.” This view seems to us particularly relevant when applied to socially situated agents. Here we offer “Conscious” Mattie[5] as a prototype of the type of biologically motivated system Wheeler spoke of, able to interact, adapt and learn in a social environment comprised of human agents. CMattie should be equally at home in a society of agents of her own type, or in a mixed society. In this paper we will describe CMattie, a “conscious,” socially situated, software agent, paying particular attention to her “consciousness” and conceptual learning mechanisms.

An *autonomous agent* (Franklin and Graesser 1997) is a system situated in, and part of, an environment, which senses that environment, and acts on it, over time, in pursuit of its own agenda. It acts in such a way as to possibly influence what it senses at a later time. In other words, it is structurally coupled to its environment (Maturana 1975, Maturana and Varela 1980). Biological examples of autonomous agents include humans and most animals. Non-biological examples include some mobile robots, and various computational agents, including artificial life agents, software agents and computer viruses. We will be concerned with an autonomous software agent, "living" in a real world computing system.

Autonomous software agents, when equipped with cognitive (interpreted broadly) features chosen from among multiple senses, perception, concept formation, attention, problem-solving, decision making, short and long-term memory, learning, emotions, etc., are called *cognitive agents*. Though ill-defined, cognitive agents can play a synergistic role in the study of human cognition, including consciousness (Franklin 1997). In this article, cognitive features such as attention are used both in the folk-psychological and technical senses.

Here, we are particularly concerned with cognitive software agents that implement global workspace theory, a psychological theory of consciousness (Baars 1988, 1997). Global workspace theory postulates that human cognition is implemented by a multitude of relatively small, special purpose processes, almost always unconscious. It is a multiagent system with a society of its own. Coalitions of such processes, when aroused by novel and/or problematic situations, find their way into a global workspace (and into consciousness). This limited capacity workspace serves to broadcast the message of the coalition to all the unconscious processors, in order to recruit other processors to join in handling the current novel situation, or in solving the current problem. All this takes place under the auspices of contexts: goal contexts, perceptual contexts, conceptual contexts, and cultural contexts. Each context is itself a coalition of processes. There is much more to the theory, including attention, learning, action selection, and problem solving.

We will refer to cognitive agents that implement global workspace theory as *"conscious" software agent*s. "Conscious" software agents are domain-specific entities; very little of their architectures is domain-independent. They adapt and learn by reacting to the changes in their domain, and through their interaction with other agents in their domains, be they human or artificial. Due to this extensive interaction, "conscious" software agents tend to be social creatures, and exhibit some socially situated intelligence.

CMattie is such a "conscious" agent (Franklin and Graesser, forthcoming). Designed for a specific, narrow domain, she functions in an academic setting, "living" in a UNIX-based system. She gathers information from humans regarding seminars and seminar-like events such as colloquia, theses defense, etc. Using this information, she composes an announcement of the next week's seminars, and mails this announcement weekly to members of a mailing list that she maintains, again by email interactions with humans. CMattie uses short-term, intermediate and long-term memories. Her emotion module enables her to react to both internal and perceived events. Her sense of self-preservation underlies her concern about her resource needs and about the status of the UNIX-based system in which she "lives." By interacting with seminar organizers, CMattie learns new concepts and behaviors. Due to the nature of her domain, this learning occurs mainly through case-based reasoning. Such learning mechanisms contribute to CMattie's socially situated intelligence (SSI). She learns to react differently in different social situations.

Following the tenets of the action selection paradigm (Franklin 1995) as expanded into design criteria (Franklin 1997), CMattie is designed using a multiplicity of artificial intelligence mechanisms. Her modular architecture, as illustrated in Figure 1, implements and integrates these several diverse mechanisms. They include behavior networks (Maes, 1990) for action selection, sparse distributed memory (Kanerva, 1988) for long-term, associative memory, pandemonium theory (Jackson, 1987) for agent grouping, Copycat architecture (Mitchell, 1993; Hofstadter and Mitchell, 1994) and natural language understanding (Allen, 1995) for email comprehension, and case-based memory (Kolodner, 1993) for intermediate term, episodic memory. Each of these mechanisms has been significantly extended in order to merge with the others, and to meet the needs of this domain. CMattie is the first software agent intended as an implementation of global workspace theory.

CMattie is different from other email and scheduling agents. For example, the Calendar Agent automates a user's scheduling process by observing the person's actions and receiving direct feedback (Kozierok, 1993). The Maxims system is an email filtering agent which learns to process a user's incoming mail messages (Lashkari, 1994). These two systems employ other agents that collaborate to overcome the problem of learning from scratch. Re:Agent is an email management system (Boone, 1998). This agent routes email to handlers that delete, download, sort, and store these messages on palmtop computers and pagers. Re:Agent learns the emails' features in order to learn how to appropriately classify the messages. The Visitor-Hoster system is aimed at helping a human secretary organize a visit to an academic department (Sycara, 1994). The secretary is presented with a user interface where she inputs relevant information to the agent about the incoming visitor. The agent then plans the visit, and returns to the secretary for confirmation. In addition to differences in tasks, CMattie's architecture, method of communication, and degree of autonomy make her relatively unique among these types of agents. Her architecture combines numerous artificial intelligence techniques to model the human mind. She communicates entirely via the natural language found in email messages. CMattie is designed to fully function as a seminar coordinator.

CMattie has several drives, some corresponding to her tasks (sending seminar announcements, reminding organizers to send information, and acknowledging messages). These drives are explicitly built-in to the agent, and operate in parallel. Some drives vary in urgency, an extension to Maes' work. For example, the urgency level for sending out a seminar announcement will increase as the time to send the announcement approaches. Each drive activates behaviors that work to fulfill the drive.

Behaviors in CMattie (Song, 1998) correspond to global workspace theory's goal contexts. Each behavior has an activation level affected by drives, other behaviors adjacent to it in the behavior net, internal conditions, and environmental inputs (the perception registers' contents). Only one behavior can be active at a time. A behavior's activation is spread to those behaviors that can fulfill its unmet preconditions and to behaviors whose preconditions can be satisfied by this behavior. Each behavior can thus be considered part of a behavior stream. For example, there's a behavior stream that composes the seminar announcement. One behavior in that stream might fill the Cognitive Science Seminar's portion of the seminar announcement.

CMattie's emotions play two roles (McCauley and Franklin,1998). First, emotions indirectly affect a behavior stream's activation level by affecting the strength of drives. Emotions allow CMattie to be pleased about sending out a seminar announcement on time and to be anxious about an impending system shutdown. In these cases, emotion might increase a behavior stream's activation level since it is pleasing for CMattie to complete these streams promptly. Second, emotions influence the suggested actions that are the output of Sparse Distributed Memory. Therefore, CMattie may be more or less inclined to actively pursue a suggested action based on the action's associated emotional level.

2. Overview of an Architecture for Supporting SSI

CMattie's architecture is quite complex as seen in Figure 1. Our references point to several papers where specific modules of the architecture have been described in depth (Ramamurthy, Bogner, and Franklin, 1998; Bogner, 1998; McCauley and Franklin, 1998; Zhang, Franklin, and Dasgupta, 1998; Ramamurthy, Franklin, and Negatu, 1998). Here we present a brief overview of CMattie's architecture so that the reader can follow our discussion.

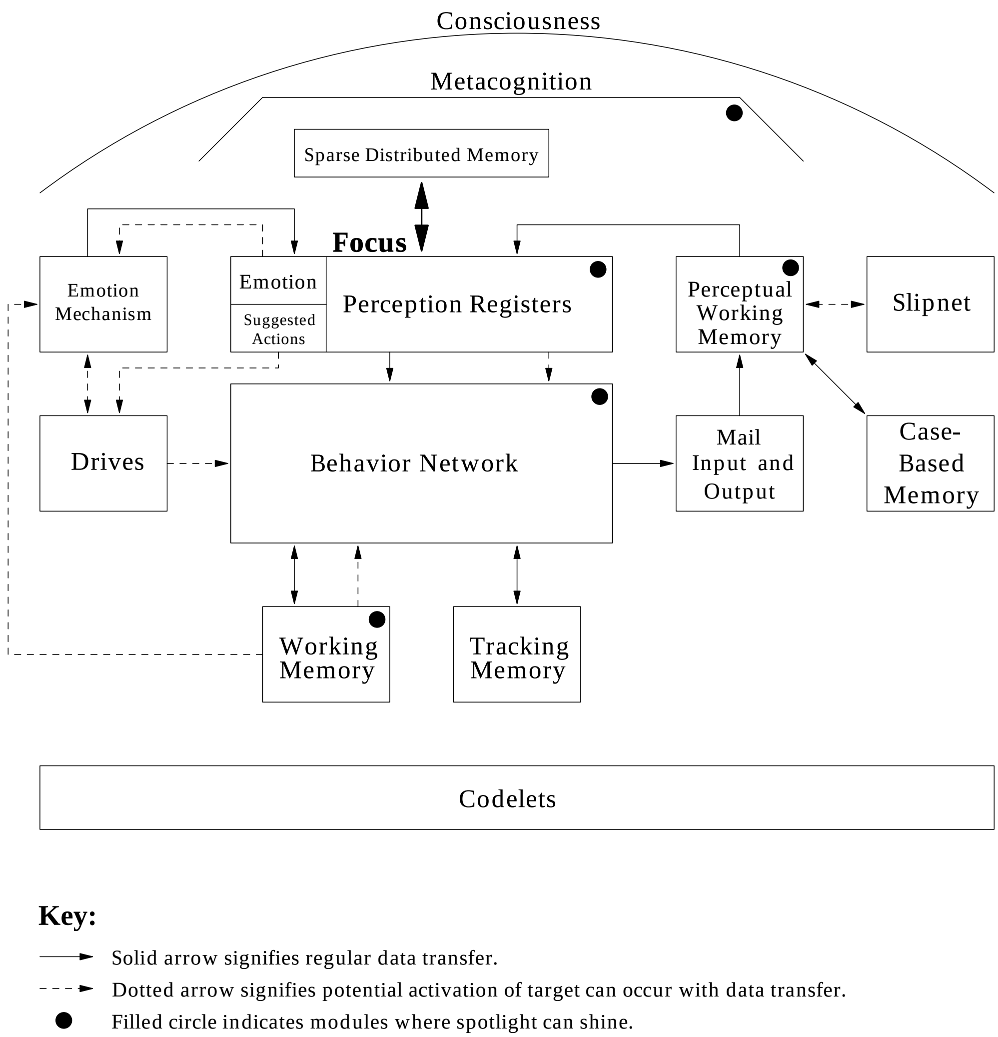

Figure 1: CMattie's Architecture

Incoming email messages are first received by the mail-input portion and are then moved to the perception module. CMattie’s sensory data are, for the most part, the incoming email messages she receives. Perception for the agent occurs when she comprehends such a message. Comprehended messages are placed in the Focus. CMattie’s perception is described in more detail in the next section.

The Focus serves as an interaction point for several of CMattie’s modules, including Sparse Distributed Memory. Sparse distributed memory is a content addressable memory that serves as long-term, associative memory for CMattie (Anwar and Franklin, forthcoming). This memory stores the contents of the perceptual registers as well as her emotions and actions. Default information, such as time and room can often be recovered, contributing to the understanding of incoming messages. Recovering remembered emotions and actions helps with action selection in the new situation.

The real work of almost all of CMattie’s modules is performed by codelets (Hofstadter and Mitchell, 1994). Codelets lie underneath CMattie’s modules including her behavior network, emotion, metacognition, perception, and portions of “consciousness”. Each codelet can be thought of as a small distinct agent designed to perform a single task. For example, one perceptual codelet’s task is to find the seminar speaker’s name in the incoming email message. CMattie’s codelets correspond to processors in global workspace theory and to the demons of pandemonium theory. Codelets coalesce into coalitions, become “conscious”, broadcast their information to all other codelets in the system, and receive the “conscious” broadcast. CMattie, following yet another tenet of the action selection paradigm, is very much a multi-agent system.

CMattie contains a global workspace based on Baars’ theory of consciousness. This allows the agent to focus attention on a specific situation. The agent’s “consciousness” module will be described in detail below.

The drives are the high level motivators and are based on Maes’ goals. All of CMattie’s drives are built-in, and they operate in parallel. CMattie’s behaviors are activated by the drives and work to fulfill them. Each behavior is comprised of codelets.

CMattie’s tracking memory stores templates used in composing outgoing email messages of different types. It also keeps track of the current seminar announcement mailing list. Tracking memory is external to CMattie, acting as a cognitive prosthesis for the agent. As of now, this memory also stores default information on seminars, such as the day of the week each one occurs. This function will probably be subsumed by associative memory.

All outgoing messages are composed in the composition workspace. Message composition consists of filling the fields of an outgoing message template. The information used to fill these fields comes from the perception registers and any of associative, case-based, or tracking memories. A current seminar announcement template is always being generated in the composition workspace. As new information is perceived and placed in perception registers, the template fields are filled. When a seminar announcement is moved to mail output and mailed, a new announcement template is placed in the composition workspace.

By monitoring what is in "consciousness", the activation of drives, emotional states, parameters in the behavior network, and the perception module, metacognition keeps track of CMattie's internal conditions (Zhang, Franklin and Dasgupta, 1998). Using a classifier system (Holland, 1986), metacognition makes inferences about CMattie's state. If necessary, it can influence "consciousness", perception, learning, and the behavior network. For example, metacognition can change the behavior network's activation level threshold to make the agent more goal-oriented or more opportunistic. It can cause voluntary attention by influencing the activation levels of certain coalitions of processors. Metacognition plays the role of an overseer, trying to keep CMattie's action selections on a productive track.

Learning via several types of mechanisms allows CMattie to become more closely coupled to her environment. She can learn new behaviors, for example, a new step in preparing for a system shutdown. She might also learn a new strategy for sending out reminders to seminar organizers. Much of her learning uses case-based reasoning. She learns new concepts in her slipnet allowing her to better understand incoming messages. This learning will be described in detail later. CMattie creates (learns) new codelets by modifying existing codelets enabling her to perform the newly learned behaviors and perceptual techniques. Coalitions of codelets are learned via association a la pandemonium theory (Jackson, 1987). This allows the agent's codelets greater ease in communicating and recruiting other codelets to help in performing tasks. Associative learning also occurs in sparse distributed memory as actions, events, and emotions are associated with one another when placed in this memory.

3. Perception

The perception module in CMattie (Figure 2) was inspired by and can be thought of as an extension of the Copycat architecture (Mitchell, 1993).

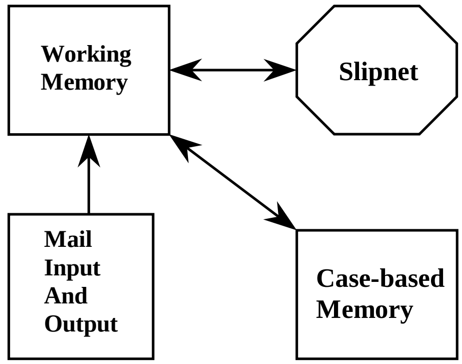

Figure 2: CMattie's Perception Module

Copycat is based on the premise that analogy-making is a process of high-level perception, and that analogy-making lies at the core of understanding. Copycat makes and interprets analogies between situations in a microworld of letter-string analogy problems. Copycat's domain is predefined and fixed; therefore, there is no learning. Since CMattie "lives" in a dynamic domain, her perceptual learning mechanism enables her to perceive this dynamism.

CMattie's perception involves building instances of known concepts in her domain; learning, detecting and creating new concepts; and making appropriate relations between those concepts. Her perception module consists of mail input and output, the slipnet, working memory and case-based memory.

Mail Input and Output. These provide CMattie's interface to her domain. Using this unit, she receives and sends out email messages related to seminars, seminar-like events such as colloquia, and maintenance of the recipient mailing list. Mail input and output can process more than one email message at a time, enabling the perception module to perceive and understand emergency events in CMattie's world. This aids in maintaining her sense of self-preservation as she proactively reacts to her changing resource needs. She immediately reacts to the status of the UNIX-host system wherein she "lives".

Slipnet. The slipnet is a network of nodes and links representing CMattie's permanent perceptual concepts. A concept has a core and a set of features representing its basic characteristics. In a given context, a feature might have a specific value. In CMattie, concepts are often defined by a region of nodes and links in the slipnet. Each of the concepts in the agent may be an individual node or a group of nodes. The various nodes are connected to each other through weighted links.

One of the built-in concepts in the perception module is the Seminar concept with the following features:

Name of the seminar
Organizer of the seminar
Location where the seminar is to be held
Date of the seminar
Day of the week of the seminar
Time at which the seminar is to be held
Speaker of the seminar
Title of Talk for the seminar
Periodicity of the seminar

Name and *Day* are features of the Seminar concept, and they are concepts themselves, each with a separate set of features. For example, as shown in Figure 3, the *Day* node in CMattie's slipnet is a concept with nodes *Monday*, *Tuesday* as its features. Seminar concept is deeper than the *Name* and *Day* concepts and, therefore, has a higher *depth value* than those two concepts. Depth values aid in the assignment of node activation level.

Each node in the slipnet has one or more codelets associated with it. When an email message is received by mail input, these codelets aid in understanding the message, which is written in natural language (Zhang et al, 1998). They recognize relevant words and phrases in the received message, and send activation to the appropriate slipnet nodes. A corpus of email messages collected for two years contributed to the building of the slipnet.

Working Memory. This memory holds the contents of the incoming email message. It also holds the perception process' intermediate results, as codelets associated with slipnet nodes operate inside working memory to understand the received email message. The most significant inference made in this process is the categorization of the *type* of the incoming message.

Case-based Memory. Case-based memory constitutes CMattie's episodic memory. In it she stores the sequences of email messages that form episodes. This allows her to relate new events to similar past events. She understands these past events using her built-in domain knowledge. Case-based memory aids her in learning new slipnet concepts through case-based reasoning. This memory acts as an intermediate term memory, and the information stored there is used to learn domain knowledge.

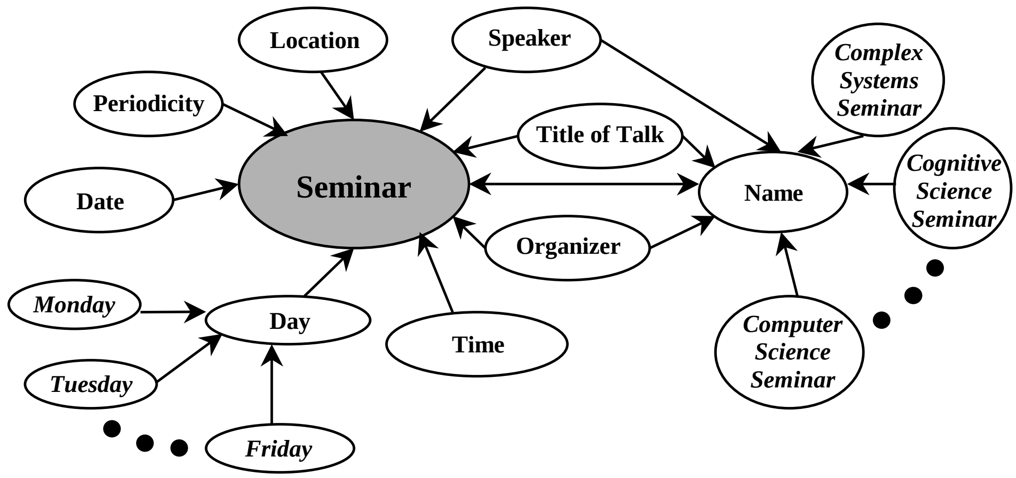

Figure 3: Segment of the Slipnet in the Perception Module

3.1. Perception Process

When an incoming message is understood, every significant word or phrase has been given a field name, and the *type* of the email message has been inferred. This information is then transferred by perceptual codelets to the perception registers in the focus. Some of the perception registers are Name, Organizer, Location, Date, Day, Time, Speaker, Title-of-Talk, Periodicity, and Message Type. Other perception registers hold previously unencountered words and phrases that occur in the received email messages and that might be relevant. The perception process is complete when the *type* of the received message has been inferred and the understood information regarding the received message has been transferred to the perception registers.

4. Bringing the Focus to "Consciousness"

Many of CMattie's components use information from the focus (Figure 4). This section describes how the focus is used to bring perceived information into "consciousness". The focus includes four vectors: the perception registers, the output of case-based memory, the output of sparse distributed memory, and the input to both case-based memory and sparse distributed memory. The Perception module places the components of the understood email message into the perception registers. That constitutes the current percept. Next, sparse

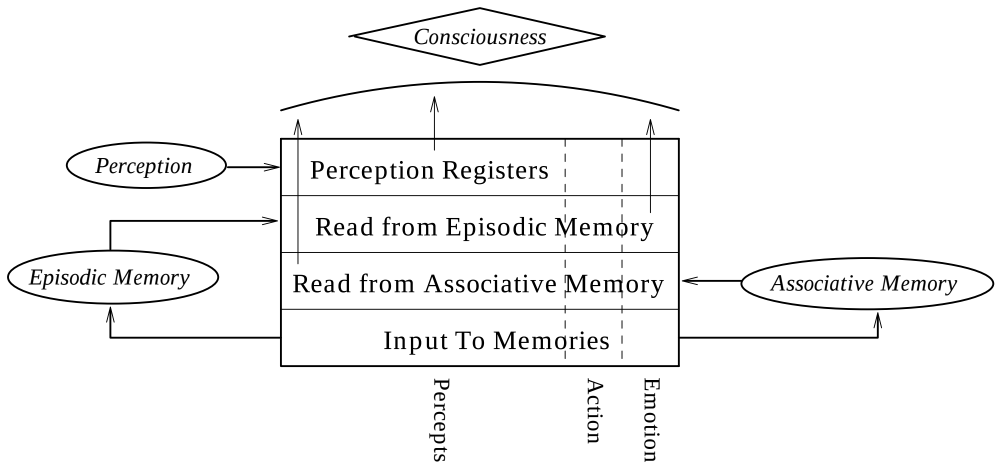

Figure 4: CMattie’s Focus

distributed memory is read with the current percept as the address. Also, case-based memory is read with the same address. These reads are designed to gather the information most relevant to what was just perceived.

A “consciousness” codelet is one whose function is to bring specific information to “consciousness” (Bogner, 1998). In particular, after the memory reads, perceptual “consciousness” codelets bring information from the focus to “consciousness”. One such “consciousness” codelet is associated with each of the perception registers and carries the specific piece of perceived information from that register. For example, one codelet carries the speaker’s name, and another carries the seminar’s time.

Specific “consciousness” codelets spring into action when the information in the perception registers is relevant to them. For example, if what is perceived is a request to be removed from the seminar announcement mailing list, the “consciousness” codelet which carries a person’s email address becomes active. It then joins the playing field on its quest for “consciousness”.

In addition, some “consciousness” codelets check for conflicts amongst the relevant items returned from the percept and the memory reads. For example, a conflict occurs if the perceived place, time and date for the Cognitive Science Seminar are the same as case-based memory’s output of these same features for the Graph Theory Seminar. The “consciousness” codelet recognizing the conflict joins the playing field and raises its activation level. Since it is associated with the other perceptual “consciousness” codelets, the coalition manager (described below) groups them together to form a coalition.

5. "Consciousness"

CMattie's global workspace gives the agent several important performance features. It allows for coalitions of codelets to gain attention. Information about these codelets is broadcast to all of the agent's other codelets. Recipients of this broadcast become active themselves if enough of the information is understood, and if it is applicable. In this way, the broadcast recipients have the potential to contribute towards solving the problem raised by the "conscious" coalition. This broadcast also allows metacognition a view of the events taking place in the system. Learning also uses the information in "consciousness" to learn to associate codelets as a coalition. In addition to "consciousness" codelets, the "consciousness" module consists of four major components: the playing field, coalition manager, spotlight controller, and broadcast manager.

5.1. CMattie's Playing Field

Artificial Minds (Franklin, 1995) contains a detailed summary of pandemonium theory first described by Oliver Selfridge in 1959 for perceptual uses and extended by John Jackson to an "idea for a mind" (Jackson, 1987). Pandemonium theory's components interact like people in a sports arena. Both the fans and players are known as demons. Demons can cause external actions, they can act on other internal demons, and they are involved in perception. The vast majority of demons are the audience in the stands. There are a small number of demons on the playing field. These demons are attempting to excite the fans. Audience members respond in varying degrees to these attempts to excite them, with the more excited fans yelling louder. The loudest fan goes down to the playing field and joins the players, perhaps causing one of the players to return to the stands. The louder fans are those who are most closely linked to the players. There are initial links in the system. Links are created and strengthened by the amount of time demons spend together on the playing field and by the system's overall motivational level at the time.

CMattie uses pandemonium theory's notion of a playing field. A collection of codelets which act as demons are instantiated when the program first runs. Each of these is a generator codelet of a specific codelet type. If a codelet of one of these types is to become active as a result of having received information broadcast from the "conscious" coalition, the appropriate generator codelet instantiates a copy of itself with the relevant information. This allows for multiple codelets of the same codelet type to run in parallel, each working with different information. These generator codelets can be considered fans in pandemonium theory's arena.

All codelets, other than these generator codelets and the "consciousness" codelets which have not detected relevant information, are considered active and are performing their functions. These active codelets are pandemonium theory's players on the playing field. The playing field is a shared space in memory; all active codelets exist in this shared memory space.

Codelets on the playing field may be associated with one another. Some of these links are built-in. For example, codelets underlying the same higher level concept, such as a behavior, are likely to be associated with one another. Codelet associations also develop when codelets are together in "consciousness". This illustrates one point of difference with pandemonium theory. There, association arises or is strengthened from being together in the playing field. Here, it's from being in "consciousness" together.

Codelets have a two-part name. The first portion signifies from where a codelet on the playing field is derived, such as a particular behavior. Since there can be multiple codelets of the same type active, codelets also carry a unique identification number. Codelets on the playing field have an activation level, which may come from the higher level construct from which they were instantiated, for example from a behavior, a slipnet node or an emotion. "Consciousness" codelets provide their own activation. Activation normally decays over time. The activation level of codelets is an important factor in deciding which coalition gains "conscious" attention.

Figure 5 illustrates CMattie's playing field. Two components of her global workspace implementation, the coalition manager and the spotlight controller, play important roles on the playing field.

5.2. Coalition Manager

The coalition manager groups active codelets into coalitions, and keeps track of them. To make coalitions, the coalition manager groups codelets according to the strength of the associations between them. If a collection of codelets is associated above a certain threshold level, these codelets are considered to be in a coalition. Coalitions are capped on average to a maximum of seven codelets. Therefore, all of the codelets associated with a single higher level concept may or may not be in the same coalition.

The playing field provides an active dynamic environment. The activation levels of codelets continually decay. Newly activated codelets join existing coalitions. Codelets leave one coalition and possibly join another. Codelets

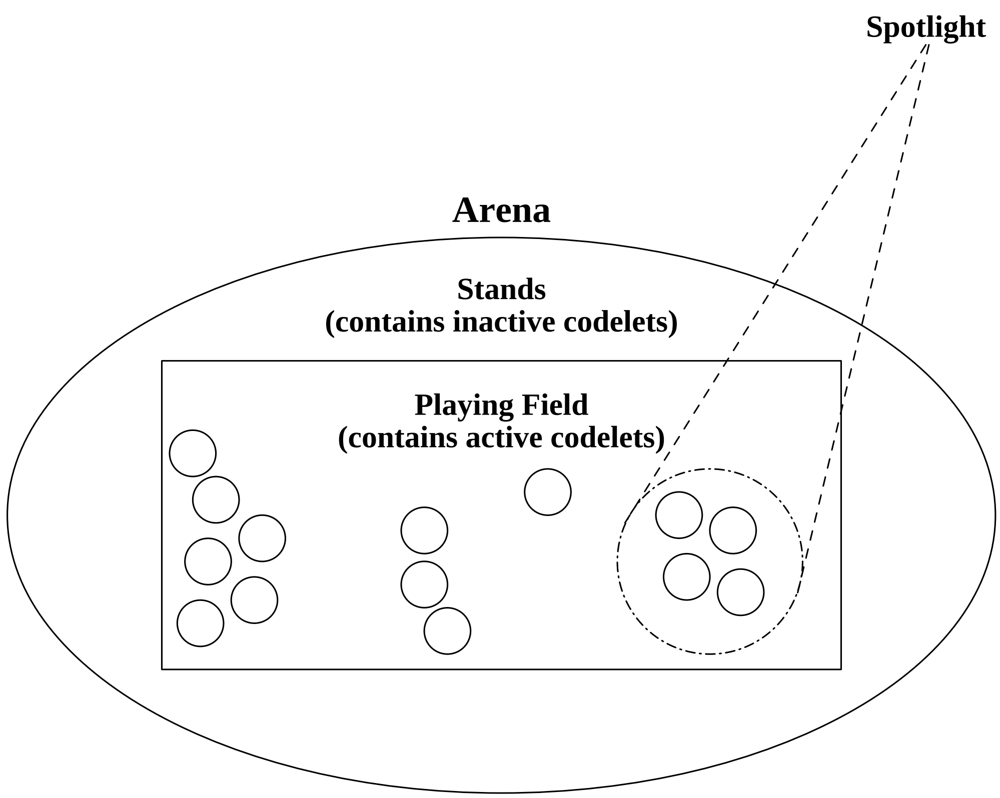

Figure 5: CMattie's Playing Field

leave the playing field when their actions are complete. Due to this dynamic environment, the coalition manager must continually and efficiently survey the playing field to keep its record of coalitions up to date.

5.3. Spotlight Controller

The spotlight controller determines which coalition becomes "conscious". It calculates the average activation level of each of the coalitions by averaging the activation levels of the coalition's codelets. The spotlight shines on the coalition with the highest average activation level. Average activation among a coalition's codelets, not the total activation, is taken to prevent larger coalitions from having an advantage over smaller ones. In the same way as the coalition manager's, the spotlight controller's domain is extremely dynamic. Here are some instances. An activation level goes to zero when an instantiated codelet's work is complete. A "consciousness" codelet may greatly increase its activation when it, say, finds a conflict. A behavior being chosen sends new activation to each of it underlying codelets. And so on.

5.4. Broadcast Manager

Once the spotlight controller has determined a "conscious" coalition, it notifies the broadcast manager who is responsible for gathering information from the "conscious" coalition, and sending it to all of CMattie's codelets. As in global workspace theory, messages are small and understood by only some of the agent's codelets.

Specifically, from the "conscious" coalition the broadcast manager gathers objects labeled for broadcast. These objects contain information needed for specifying the current novelty or problem. This information is then broadcast to all of CMattie's generator codelets.

In addition to being broadcast, information gathered from a coalition is placed on the blackboard, implemented as a shared memory space. This blackboard containing at most the last seven broadcasts, or the last seven items in "consciousness", serves to implement short-term working memory. Codelets can poll this blackboard and search for parameters which they understand.

6. "Consciousness" as a Facilitator for Learning

The "conscious" broadcast recruits codelets that understand the message and for which it is relevant. This causes their activation to increase, motivating them to begin performing their respective tasks. These tasks might include activating their overlying higher construct, say a behavior, an emotion, a slipnet node, or a learning mechanism. Figure 6 illustrates the significant role of "consciousness" in perceptual learning, highlighting global workspace theory's premise that "consciousness" is sufficient for learning. This section focuses on the perceptual learning that results from the "conscious" broadcasts.

CMattie has a limited number of seminars already defined in her slipnet. She "knows" about these seminars through the built-in seminar concept and its features. In particular, she knows that:

- A seminar is held once a week
- It has an organizer and a name
- Each week, there might be a different speaker
- It has a different title-of-talk

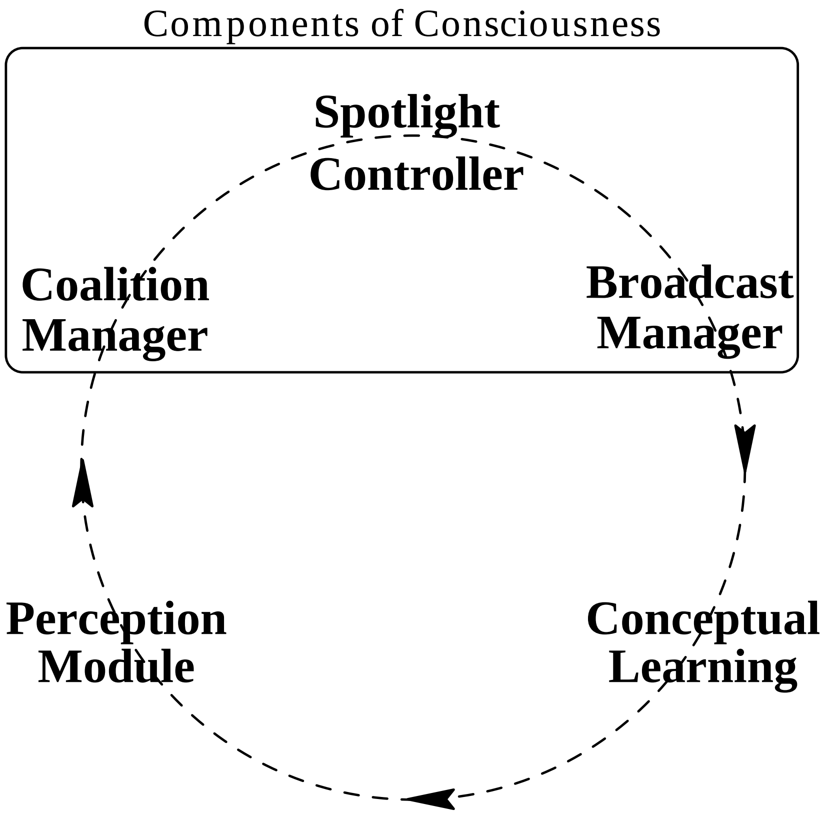

Figure 6: "Consciousness" As A Facilitator For Conceptual Learning

- It is usually held at the same location, on the same day of the week, and at the same time.

Suppose a seminar organizer sends her a message announcing a seminar with a seminar name that she has never seen before. CMattie attempts to treat such a message in a way similar to seminars that she already knows. The learning mechanism described here is based on the premise that any agent, including humans, learns based on what it already knows. When the message understanding mechanism attempts to understand this message, the agent recognizes that it is an *initiate-seminar-message* for a seminar, but the *name* of this seminar is not part of the built-in knowledge. This information is placed into the perception registers, brought to "consciousness", and broadcast. CMattie has codelets that understand this broadcast and can activate behavior streams that act to converse with the sender of the message to determine if the sender wishes to initialize a new seminar. She sends an acknowledgement to the sender stating that a new seminar with that seminar name will be initialized, with the sender as its organizer and requesting confirmation. Reinforcement of a sort is provided to CMattie by the response she might or might not get. Based on the “conscious” broadcast of this feedback, if any, a new slipnet node is created for this seminar name, and it is linked to the *name* node, which is also a feature of the seminar concept. When this new node is generated, the associated codelets for it are generated as well, a quite straightforward process. The new codelets are based on similar, existing codelets for the other *name* nodes. Once the process is complete, CMattie has understood the incoming confirmation message, and the perception module sends the relevant fields to the perception registers.

The second type of learning that takes place in the perception module occurs when CMattie learns concepts which are not completely identical to the built-in seminar concept, but slightly different from it. In her domain, colloquia, dissertation defenses, dissertation committee meetings, and faculty meetings, all fall into this category. This second learning mechanism is based on viewing every new situation in terms of a previously solved problem (analogy-making). When CMattie receives a message about such a non-seminar event, say a dissertation defense, she treats it as a *speaker-topic message* for a seminar. This understanding is disseminated through “consciousness”. The agent sends an acknowledgement to the sender stating that she is initializing a new seminar by the name “Dissertation Defense Seminar” with the sender as organizer. This misunderstanding can be expected to result in one or more of the following events, depending on the sender.

- The acknowledgement elicits a negative response from the sender, starting an episode. The resulting “conversation” between CMattie and the sender is stored in case-based memory. This episode provides information that allows CMattie, even with her limited natural language understanding, to learn that dissertation defense is similar to the seminar concept, but with slightly different features. In this case, the *periodicity* feature has a different value. CMattie learns this through case-based reasoning.
- The sender ignores the acknowledgement, and CMattie includes the Dissertation Defense Seminar in her weekly seminar announcement. In this case nothing is learned at this time, but perhaps later.
- CMattie includes the Dissertation Defense Seminar in seminar announcement. This action is likely to elicit a negative response from the sender, starting an interaction with CMattie. This episode again is stored in case-based memory to aid her in learning what a Dissertation Defense is.
- The sender might also ignore the incorrect weekly announcement, but respond to the reminder sent by CMattie the following week, when she doesn’t

receive a *speaker-topic message* for the Dissertation Defense Seminar. This, again, generates an episode allowing CMattie to learn about a Dissertation Defense.

- The sender ignores all the reminders. In this case, ignoring the reminders itself acts as feedback to CMattie, giving rise to a new concept that is similar to the seminar concept with its periodicity feature modified.

Regardless of which of these scenarios occur, CMattie eventually learns a new concept called *dissertation defense* that is closely related to the seminar concept. Note that this conceptual learning takes place through the internal interaction between "consciousness" and perception. Each of these possible situations becomes "conscious" before any changes to the perception module occur. Also note the crucial role in this conceptual learning played by external social interaction with a human.

How is all this done? There are two main capabilities. First, CMattie has codelets in her slipnet that look for words and phrases that she has not previously encountered. The perception module tracks such new words and phrases that occur with any regularity by keeping statistics and recognizing novelty. This aids in CMattie's natural language understanding. Second, even with her limited natural language understanding, CMattie can understand messages from organizers that have negative connotations. Her slipnet has nodes and codelets that detect words and phrases with such negative connotations.

Consider a possible path CMattie might take during her conceptual learning. CMattie misunderstands the first dissertation defense message, and sends an acknowledgement to the organizer for a Dissertation Defense Seminar. Suppose the organizer responds with a negative message saying, "It is not a seminar, but a dissertation defense." CMattie understands the negative connotation in "not a seminar" and the repeated occurrence of the phrase "dissertation defense" activates her questioning capability to send a message to the organizer with the question, "How does a dissertation defense differ from a seminar?" The organizer might reply with a simple explanation such as, "A dissertation defense is like a seminar, but it might not occur regularly" or "Dissertation defenses do not ordinarily occur every week". CMattie understands the negative connotations in relation to the words "regularly," "every week" or "weekly". These are keywords in the slipnet related to the *periodicity* feature of the seminar concept. CMattie uses her case-based memory and natural language understanding to reason that a dissertation defense has a periodicity different from that of a seminar. This interaction with the organizer and her reasoning effects the creation of a new concept, *dissertation defense,* in the slipnet with related codelets that search for it in future messages. This "conscious" learning enables her to correctly perceive and understand a dissertation defense message when she next encounters one. Of course, CMattie must also learn to behave differently when faced with a dissertation defense message than she does with a seminar message. That is a subject for another chapter.

A trace of this learning stored in her case-based memory serves to enhance her case-based reasoning capabilities. Later, CMattie might encounter a colloquium message, and in response to her incorrect acknowledgement of a Colloquium Seminar, be told that "It is a colloquium, not a Colloquium Seminar." CMattie's case-based reasoning depends on (a) past experiences she has had, and (b) her ability to understand new situations in terms of her past experiences. She recalls her experience with the first dissertation defense message from her case-based memory, and reasons that *colloquium* might be similar to *dissertation defense*. She sends a message to the organizer with the question, "Is a colloquium similar to a dissertation defense?" Her understanding and reasoning, based on the organizer's reply to her query, aid her in learning about colloquia.

Thus CMattie's conceptual learning is socially situated through her "conscious" interaction with seminar organizers. This interaction enables her to acquire a measure of socially situated intelligence.

7. Criteria for Performance Analyses

Design and development of CMattie has been ongoing for several years. As of this writing, CMattie is actively being implemented. CMattie is a successor to the successful Virtual Mattie. CMattie's task set is a superset of VMattie's, a "non-conscious" autonomous agent. Here we present a portion of VMattie's test results to illustrate what tasks CMattie is expected to perform (Song, 1998).

VMattie was tested over a period of four weeks. These tests were designed to simulate real world settings. During testing, she received 55 messages comprised of 10 message types. The majority of messages received fell into the categories of Seminar Initiation, Speaker-Topic, Seminar Conclusion, and Add to Mailing List. She received 5 messages which were irrelevant to her domain.

VMattie was able to correctly fill all of the perception registers for 96.4% of the messages she received. She chose the date of seminar and title of talk incorrectly for only two Speaker-Topic messages as two words were collapsed together without a space in the incoming messages. Even with this misperception, she correctly composed acknowledgement messages and sent them to the senders of each received message.

The behavior network used this perceived information to generate seminar announcements. VMattie was 100% accurate in generating and sending out the seminar announcements. This included correctly recovering missing information from her tracking memory for default values with full accuracy. VMattie was able to correctly change her mailing list upon receipt of Add to Mailing List and Remove from Mailing List messages.

VMattie sent 7 reminder messages to seminar organizers on time during this testing. She received 5 replies to her reminders before the seminar announcement distribution date. She correctly inserted "TBA" for the remaining 2 instances in the seminar announcements.

Rigorous testing of CMattie is planned. CMattie's implementation of Global Workspace Theory makes her an agent significantly more complex than VMattie. At the moment, even without the test results, CMattie's role as an implementation of Global Workspace Theory makes her valuable as a conceptual model of mind.

8. Conclusions

This paper presents an overview of CMattie's architecture focusing on "consciousness" and conceptual learning. The two modules implementing these contribute to the implementation of global workspace theory, and allow her to interact intelligently with seminar organizers. This interaction succeeds due to the unique integration of these two modules. Perceptual output enters the focus, which is brought to "consciousness". The global workspace broadcast allows for "conscious" conceptual learning, completing the cycle. This cycle allows CMattie to acquire her socially situated intelligence. In particular, we hope to show that "conscious" software agents can be capable of essentially one-shot learning through interaction with a human. In future stages, extensions such as unlearning are planned.

CMattie is the first software agent designed to implement global workspace theory. As such, she can be considered the first "conscious" agent. It is hoped that the implementation decisions both provide testable hypotheses to neuroscientists and cognitive scientists, and that successful results will lead to more intelligent "conscious" agents.

9. Notes

1. Supported in part by NSF grant SBR-9720314.
2. Supported in part by Cancer Center Support CORE grant, P30 CA 21765 and by American Lebanese Syrian Associated Charities (ALSAC).
3. Supported in part by NSF grant SBR-9720314 and by ONR grant N00014-98-1-0332.
4. Including Art Graesser, Zhaohua Zhang, Aregahegn Negatu, Ashraf Anwar, Lee McCauley, and Scott Dodson.
5. "Conscious" Mattie is intended to implement a psychological theory of consciousness, hence her name. This theory is described later in the introduction. We make no claims that she is conscious in the sense of being sentient.

10. References

Allen, James. (1995). Natural language understanding. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Anwar, Ashraf & Franklin, Stan. (Forthcoming). *Sparse distributed memory for conscious software agents*. http://www.msci.memphis.edu/~cmattie.

Baars, Bernard. (1988). A cognitive theory of consciousness. New York: Cambridge University Press.

Baars, Bernard. (1997). In the theater of consciousness. New York: Oxford University Press.

Bogner, Myles. (1998). Creating a conscious agent. Memphis: Master’s thesis, The University of Memphis.

Boone, Gary. (1998). Concept features in Re:Agent, an intelligent email agent. Proceedings of the Second International Conference on Autonomous Agents. New York: ACM Press, 141-148.

Franklin, Stan. (1995). Artificial minds. Cambridge, MA: The MIT Press.

Franklin, Stan. (1997). Autonomous agents as embodied AI. Cybernetics and Systems, Special Issue on Epistemological Issues in Embodied AI, 28, 499-520.

Franklin, Stan & Graesser, Arthur. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. Berlin: Springer-Verlag, 21-35.

Franklin, Stan & Graesser, Arthur. (Forthcoming). *Models of Consciousness*. http://www.msci.memphis.edu/~cmattie.

Hofstadter, Douglas & Mitchell, Melanie. (1994). The copycat project: A model of mental fluidity and analogy-making. Holyoak, K. & Barden, J. (Eds.). Advances in connectionist and neural computation theory, 2. Norwood, NJ: Ablex.

Holland, J. H. (1986). A mathematical framework for studying learning in classifier systems. Farmer, D., et al. (Eds.). Evolution, games and learning: Models for adaption in machine and nature. Amsterdam: North-Holland.

Jackson, John. (1987). Idea for a mind. SIGGART newsletter, 101, 23-26.

Kanerva, Pentti. (1988). Sparse distributed memory. Cambridge, MA: MIT Press.

Kolodner, Janet. (1993). Case-based reasoning. Morgan Kaufmann Publishers.

Kozierok, Robyn & Maes, Pattie. (1993). A learning interface agent for scheduling meetings. Proceedings of the 1993 International Workshop on Intelligent User Interfaces. Orlando, FL, 81-88.

Lashkari, Yezdi, Metral, Max, & Maes, Pattie. (1994). Collaborative interface agents. Proceedings of AAAI ’94 Conference. Menlo Park, CA: AAAI Press.

Maes, Pattie. (1990). How to do the right thing. Connection science.

Maturana, H. R. (1975). The organization of the living: A theory of the living organization. International Journal of Man-Machine Studies, 7, 313-32.

Maturana, H. R. & Varela, F. (1980). Autopoiesis and cognition: The realization of the living. Dordrecht, Netherlands: Reidel.

McCauley, Lee & Franklin, Stan. (1998). An architecture for emotion. AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition. Menlo Park, CA: AAAI Press, 122-128.

Mitchell, Melanie. (1993). Analogy-making as perception. Cambridge, MA: MIT Press.

Ramamurthy, Uma, Franklin, Stan, & Negatu, Aregahegn. (1998). Learning concepts in software agents. From Animals to Animat 5: Proceedings of The Fifth International Conference on Simulation of Adaptive Behavior. Cambridge, MA: MIT Press, 372-377.

Ramamurthy, Uma, Bogner, Myles, & Franklin, Stan. (1998). Conscious learning in an adaptive software agent. Proceedings of The Second Asia Pacific Conference on Simulated Evolution and Learning (SEAL98). Canberra, Australia.

Sycara, Katia, & Zeng, D. (1994). Visitor-hoster: Towards an intelligent electronic secretary. CIKM94 Workshop on Intelligent Information Agents. http://www.cs.cmu.edu/afs/cs/user/katia/www/visit-host.html.

Song, Hongjun. (1998). Control structures for software agents. Memphis: Doctoral dissertation, The University of Memphis.

Wheeler, Michael. (1994). For whom the bell tolls? The roles of representation and computation in the study of situated agents. Cognitive Science Research Paper 320, School of Cognitive and Computing Sciences, University of Sussex.

Zhang, Zhaohua, Franklin, Stan, & Dasgupta, Dipankar. (1998). Metacognition in software agents using classifer systems. Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98). Menlo Park, CA: AAAI Press, 83-88.

Zhang, Zhaohua, Franklin, Stan, Olde, Brent, Wan, Yun, & Graesser, Arthur. (1998). Natural language sensing for autonomous agents. Proceedings of the IEEE Joint Symposia on Intelligence and Systems. Rockville, Maryland, 374-81.