

## Toward a Hybrid Cultural Cognitive Architecture

Glenn Taylor ([glenn@soartech.com](mailto:glenn@soartech.com)), Michael Quist ([quist@soartech.com](mailto:quist@soartech.com)),  
Steve Furtwangler ([furtwangler@soartech.com](mailto:furtwangler@soartech.com)), Keith Knudsen ([knudsen@soartech.com](mailto:knudsen@soartech.com))

Soar Technology, Inc., 3600 Green Court, Suite 600  
Ann Arbor, MI 48103 USA

### Abstract

This paper presents work to date in developing a cognitive architecture for modeling cultural behaviors. The implementation is based on the Soar cognitive architecture, with extensions in terms of commitments to particular representations (schema) and particular information processing procedures that go beyond the standard Soar procedures. We present some motivation for this work, the methodology and implementation, and a simple example to demonstrate capabilities, with an eye toward evaluation.

**Keywords:** cognitive modeling; cognitive architectures; cultural models; Soar; appraisal theory; cultural schema theory.

### Introduction

Culture is a ubiquitous phenomenon in human behavior. It affects how we perceive our environment, how we organize our thoughts and our relationships, and how we decide to act. In short, culture is very much a cognitive phenomenon. Yet, culture has received only a surface treatment in the computational cognitive modeling community. Some narrow efforts have developed models of culture, but few efforts have attempted to develop models of culture within integrated behavior architectures.

This paper first details our motivations for our work, as well as some background in cultural theories. We then describe how we operationalize some of these theories in a computational cognitive architecture, and describe a prototype model built in that architecture. We discuss the limitations of the architecture and the model, as well as some lessons learned in this process.

### Motivation for This Work

Most computational models of culture attempt to capture the evolution of culture in simple multi-agent models (see, for example (Axelrod, 1997)). These efforts tend to be non-cognitive – the interest is in modeling theories cultural change over time, rather than the influence of culture on the behavior of the individual. While researchers in cognitive architectures have spent a good amount of effort in developing models of human cognition, these models do not typically take into account variations in cognitive styles or knowledge due to cultural influences.

In this vacuum, it is our desire to develop a cognitive architecture that explicitly supports modeling culture-

specific knowledge and cognitive processes, such that models developed within this architecture exhibit “culturally appropriate behaviors.” Applications of this kind of architecture could include cultural training systems, or model that help predict how a person of a particular culture would respond to different situations or stimuli.

### Background

There is an increasing body of theory-based work in cognitive science and cultural psychology that have attempted to identify cultural influences on behavior, and the kinds of knowledge associated with those influences. Triandis and others have investigated the importance of cultural factors such as Individualist-Collectivist, as a means to categorize and describe kinds of cultures (Hofstede, 1980; Triandis, 1994). Cultural Psychologists have identified differences in cognitive styles, such as field dependence in perception and causal attribution in eastern versus western cultures (Nisbett, Peng, Choi, & Norenzanan, 2001; Peng & Nisbett, 1999). Cultural Anthropologists have begun to describe kinds of cultural knowledge and how that knowledge is encoded and used in perception and decision-making (D’Andrade, 1992; Quinn & Strauss, 2001).

In this paper, we focus on Cultural Schema Theory as a basis for understanding how culture is manifested in cognition. We complement this with Appraisal Theory to further refine how behavior is generated from cultural knowledge and, specifically, interactions with others. These theories are described in more detail below, as is the basic underlying computational framework upon which we have built our architecture.

### (Cultural) Schema Theory

One enduring notion in cognitive science is that of cognitive schema. While there has been much debate about the applicability of Schema Theory, and while there does not yet seem to be a unified theory regarding schema (Alba & Hasher, 1983) (instead there are somewhat fractionated micro-theories), it seems still to be a fruitful avenue for building models of cognition.

From the perspective of Cultural Schema Theory, culture is epiphenomenal – it piggybacks on top of normal cognitive processes, in particular those described by Schema Theory. Schema have a number of properties relevant to the study of culture, as described by (D’Andrade, 1992):

- Schema serve as organized representations of knowledge about events, things, relationships, etc. (there are many kinds of schema)
- Schema serve as processors of information, recognizing events or objects in the environment, and assigning meaning to those events or objects
- Schema can be composed hierarchically – that is, the output interpretation of one schema is passed to another schema as input
- Schema are learned over time, either by direct experience, or by being taught

Schema Theory makes some predictions regarding cognitive processes, such as the relationship between schema and priming effects. In cultural terms, some of these schema may have cultural influences (e.g., they are learned as part of the enculturation process deliberately or as a matter of course).

One facet of Schema Theory is Script Theory (Schank & Abelson, 1977), which defines a particular kind of schema called an Event Schema to capture common occurrences as sequences of actions. The canonical Event Schema is the Restaurant Script, which describes roles (patron, waiter), props (table, dishes), and responsibilities (order food, bring bill). Scripts are divided into discrete steps called scenes that describe small subsets such as the “ordering food” scene or the “pay and go” scene, and sequences of these scenes may be explicitly ordered (e.g., pay before you go). Related is Frame Theory (Minsky, 1975), which espouses a particular representation (frames) for representing schema structures.

According to Schema Theory, individuals have many such schema available to be activated in any situation, which are learned over a lifetime of learning and experience. This implies a broad range of knowledge is required for culturally appropriate behavior as we seek to accomplish here, and so imposes some requirements on the kinds of solutions we might consider.

### Appraisal Theory

Appraisal Theory is a cognitive account of how emotions arise, based on fast, automatic interpretation of perceived events with respect to the situation and to one’s goals, with the result of this interpretation being a set of emotions. Appraisal Theory describes patterns of appraisals that distinguish emotions (Lazarus, 1991). For example, anger might arise because someone or something has threatened one’s goals.

Appraisal Theory and Schema Theory offer complements to each other – where schema define the structure and processing of events at a basic level, resulting in first-level interpretations, appraisals tie those interpretations to goals. Schema also provide expectations, and appraisal provides a mechanism to relate events to expectations. Where expectations are broken or goals are hindered, one result is an emotional response. Schema can also provide learned

coping strategies that incorporate culturally relevant behavior (and even culturally appropriate emotional responses – see (Matthews, 1992)).

We have considered Appraisal Theory and Emotion in our architecture largely for reasons of believability. Having a virtual agent behave in culturally believable ways means having those agents respond appropriately to events they perceive. Where cultural schema help define expectations and goals, believable responses to events tie directly to aspects of emotion where those culturally-relevant goals are affected. Outward “cultural behavior” that does not include emotional elements will not be believable to human observers.

### Soar

The Soar cognitive architecture has been developed as a framework for general problem solving, incorporating knowledge-based approaches to decision-making, and demonstrating a wide range of cognitive phenomena (Laird, Newell, & Rosenbloom, 1987; Newell, 1990). The role of a cognitive architecture is to provide a theory-based framework for building models, and in particular that the framework provides constraints in which to guide model building to stay faithful to the theory. The Soar architecture is an executable version of the Soar theory, which pertains to memory organization and access, goal-directed problem solving, and learning.

Soar exhibits a number of strengths with respect to Cultural Schema Theory, for which we think it is highly relevant as the basis for a cultural cognitive architecture:

- Highly scalable long-term memory, enabling a means to represent and quickly access schema
- Associative memory activation, allowing for schema to be built up incrementally in memory
- Mixed automatic and deliberative reasoning, allowing for automaticity of schema activation and processing, as well as goal-directed, problem-solving behavior
- Ability to incorporate multiple reasoning methods, possibly allowing for cultural variations in cognition
- Graph-based working memory, which maps well to Scripts/Frames approaches to schema representation

Regarding Appraisal Theory, recent work by (Marinier & Laird, 2006) describes a principled implementation of Appraisal Theory within Soar. We take this as inspiration for the work here, rather than an explicit re-use of that model.

Another facet of Soar is that it is purely a symbolic architecture, inspired by the Physical Symbol System Hypothesis (Newell & Simon, 1963). However, much of the Schema Theory literature talks about phenomena such as priming and framing effects, which seem most easily explained in non-symbolic terms of recency and activation. Recent extensions to the Soar architecture have begun to expand into “sub-symbolic” approaches, such as for enabling

working memory decay (Nuxoll, Laird, & James, 2004), reinforcement learning (Nason & Laird, 2004), and episodic memory (Nuxoll & Laird, 2004). We adopt some of these ideas of sub-symbolic processing in the system described here in an attempt to capture some of these phenomena, especially activation of schema.

### Related Work

Cultural Models are fairly commonplace within the cultural anthropology literature, though these are descriptive rather than computational models (D’Andrade, 1992; Quinn & Strauss, 2001). There are a few theory-based models of cultural behavior in computational cognitive models (e.g., (Marsella & Pynadath, 2007)), though for the most part research in this area is quite new. There are a few computational models of schema, the most notable being (Rumelhart, Smolensky, McClelland, & Hinton, 1986) and (Schank & Abelson, 1977). While we would not describe our system as a parallel distributed system in same the way Rumelhart does, we adopt some of the same qualities, such as their notion of independent processing units, as well as the idea an on-the-fly construction of a schema based on inputs, rather than representing schema statically in memory. We differ primarily in that we represent schema symbolically with a simple spreading activation model.

There are a few, richer implementations of computational models of Appraisal Theory within cognitive architectures (Gratch & Marsella, 2004; Marinier & Laird, 2006; Wehrle & Scherer, 2001). However, each of them differ in the specific appraisal dimensions used, the specific emotion types, and the computations used to arrive at those emotions.

### Architecture Implementation

In terms of a cognitive architecture for modeling culture, we have extended the Soar architecture to accommodate an implementation of Schema Theory. This includes representation of both schema definitions (potential schema) and schema instances (active schema), and mechanisms for processing schema. This works within the basic organization of the Soar architecture, in that schema instantiation is implemented as rules that fire under the appropriate conditions (e.g., after perceiving an event in the environment), and result in manipulations of working memory elements in Soar. These modifications to the architecture can be best described as knowledge-level extensions at this point, rather than as changes to the underlying Soar architecture. Here, the introduction of schema imposes constraints on representation and processing, beyond those that the basic Soar architecture provides. The rest of this section describes these extensions in detail, including the basic representations and processing of schema.

### Schema Representation

Within a model, schema are represented in two forms, first as rules in long term memory, and second as working memory elements (essentially, short term memory). In Soar, all long term memory is represented as rules. When a rule representing a piece of a schema fires based on input from the environment (an event), a working memory element is created representing an “active” elements of that schema. This section defines the knowledge definitions in the implementation. To ground these ideas, a hypothetical Restaurant Schema inspired by Schank & Abelson’s original work is shown in Figure 1.

**Schema Definitions** A schema definition is a template for either a single event (event template) or a sequence of scenes (script template); each scene is defined, recursively, as a disjunction of schema. Each schema definition specifies a set of role variables, which will be bound to specific actors and props (other non-actor objects) when the schema is instantiated. In addition, schema definitions specify mappings between these roles and those appearing in subschema, and may specify additional activation conditions that assign numerical preferences to the allowed role bindings.

**Schema Instances** A schema instance is formed from a schema definition by assigning concrete values to all roles, and by binding an event template to a specific event or the initial scene of a script template to a specific instance of a suitable subschema. Script templates will progress through various levels of completion, as their scenes become bound over time. A numerical activation level, the initial value of which depends on which activation conditions are met by the bindings, represents the prominence of a particular schema instance in the agent’s working memory.

**Events** Events represent occurrences in the environment that are relevant to schema. The leaf nodes on all schema definitions are event templates, and match against events to activate the schema.

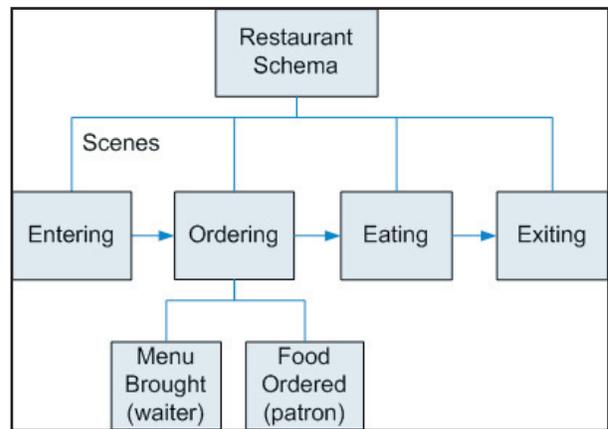


Figure 1: Restaurant Schema

**Roles** Roles list the actors and props in a schema, and the activation conditions for a given schema place additional constraints and preferences on the allowed bindings of roles. In the restaurant schema, “waiter,” “patron,” and “food” are all roles.

**Goals/Expectations** For each partially completed schema instance, the agent has a goal or expectation that the next scene will occur. When the defined actor for the next scene is the agent, the agent forms a goal to move the script forward; this will recurse downward through the scene definition until the agent finds an action that he can take, i.e., an event template with himself as the subject. For example, in the Restaurant Script defined above, one goal for the patron role is to eat a hamburger. On the other hand, when the defined actor for the next scene is another actor, the agent simply forms an expectation that other actor will act in a particular way. In the Restaurant Script, one expectation is that the waiter will bring the food after the patron has ordered.

**Universe (ontology)** Within the model is a definition of all the things (actors, objects, places) that appear in the simulated world, and the relationships between them – effectively an ontology. These serve as features that can be used to define events and activation criteria for schema. In the restaurant example, the waiter, chef, and patron would be actors defined in the universe, locations such as seating area and kitchen would be defined, and each would have propositional properties such as “chef is in the kitchen.”

**Schema Processing**

Schema undergo a process of activation, decay, and deactivation based on relevance to events perceived in the environment. Schema also generate expectations about future events and goals that one should pursue. A basic illustration of this process is shown in Figure 2, supporting the description of the process in this section.

**Schema Activation** Schema instances are initially created by the occurrence of a suitable event (for event templates) or by

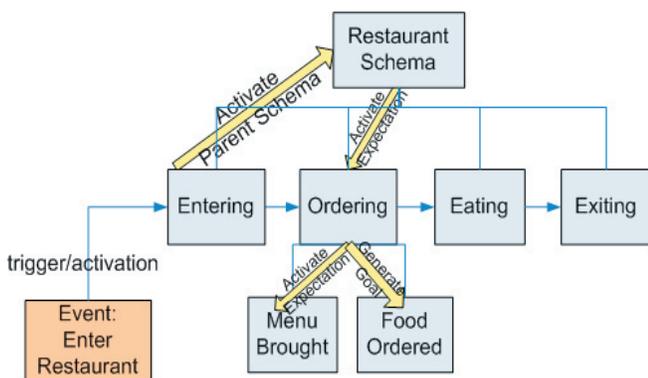


Figure 2: Event-triggered Schema Activation

the instantiation of a suitable first scene (for script templates). In these cases, the activation level of a schema is set according to the activation conditions satisfied by its bindings. This is potentially a mixed bottom up (event-driven) and top-down (goal and expectation activation) process. Because many schema may refer to the same type of event, or contain compatible scenes, a single event can lead to the activation of multiple schema. A new event or scene can also extend an existing script that is partially completed, provided that it is compatible with the next scene. This creates an extended instance with a new, generally higher, activation level.

**Schema Deactivation** The activation levels of all schema instances erode over time, and are refreshed only when a schema takes another step toward completion. This allows for some measure of interruption and resumption of schema, while still removing “roads not taken” from the agent’s working memory after a decent interval has passed. Our implementation decrements all activation levels by a constant factor after each new event. A more nuanced approach would be to allow the schema definitions themselves to specify expected intervals between scenes, or based more deliberately on “clock time.”

**Event Matching** In our implementation, events are described by their thematic roles, such as “agent,” “experiencer,” and “content.” Event templates are defined so as to bind their roles to some of these entries in the event description and restrict other entries to specified values. Event generation uses the same templates: when the agent wants to make a particular event happen with a particular set of bindings, the agent uses the appropriate template to see which thematic roles these bindings should play.

**Goal Generation** As discussed earlier, the agent creates a goal to take any action that will move an active script forward. To resolve conflicts between proposed actions, the agent prefers to take the action that arises from the script with the highest activation level (and selects randomly to break any remaining tie). This conflict resolution scheme is a weakness of the current implementation, because it conflates activation level (belief that “this is happening”) with the agent’s interests (“I want this to happen”). If the agent had goals beyond simply following the script, it would be straightforward to include them within the conflict resolution phase.

**Appraisal**

Appraisal in this architecture is a process for interpreting events from the environment with respect to one’s goals, values, and ability to cope with the event. From this appraisal, one generates an emotional response, when then may generate coping behavior in response to an event. This perspective is borrowed from the work of (Scherer, Schorr, & Johnstone, 2001) and (Ellsworth & Scherer, 2003).

Table 1: Appraisal dimensions.

Appraisal Dimensions	
Intrinsic Pleasantness	Urgency
Outcome Probability	Control
Conduciveness	Adjustment
Compatibility with Internal Standards	Goal Relevance
Compatibility with External Standards	Causal Motive
Causal Agent	

**Appraisal Dimensions** In this implementation, we adopt directly the appraisal dimensions described in (Scherer et al., 2001), as shown in Table 1. Other researchers have identified (or implemented) other sets of dimensions. We have attempted to be faithful to Scherer’s work where possible, though it is not always obvious how to map an event to an appraisal (this holds also for other researchers’ dimensions as well).

### Emotional Response

According to emotional appraisal theory (Scherer et al., 2001), each appraisal dimension influences multiple components of emotion; the direction of this “influence” may be positive, negative, or neutral, depending on the appraisal/emotion pair. In our implementation, we multiply the appraisal dimensions by a fixed matrix, then apply a nonlinear transformation to restrict the range to  $[-1,1]$ , to yield the components of the emotional response (ER):

$$ER_i = \text{arctanh}(\sum \{j\} M_{ij} * A_j)$$

This emotional response is then averaged with the agent’s previous emotional state to give the agent’s new emotional state. As with the activation levels of scripts, we multiply all emotional components by a constant factor after each event; this returns the emotional state to the neutral position after enough time has passed. The result is that the emotional state at any given time is a weighted average of the agent’s

Table 2: Emotional components.

Emotional Component	
Happiness	Pride
Joy	Guilt
Disgust	Shame
Anxiety	Boredom
Despair	Rage
Sadness	Irritation
Contempt	Fear

emotional responses to all past events, with more recent events weighted more strongly, multiplied by a decaying function of the time since the last emotionally charged event.

The emotional dimensions we used are shown in Table 2 (adapted from (Scherer et al., 2001)). These arise directly from specific patterns of appraisal dimensions.

An emotional response can also be a triggering condition for further schema that define the responses to emotional situations. Specifically, emotional attributes are first-class predicates in the universe ontology, and can be tested within activation conditions for schema. For example, one could define a schema for actor X having an argument with actor Y whose initial activation level was proportional to actor X’s level of irritation.

### Prototype

We have developed a prototype cultural model based on the architecture we have described. The prototype represents a first attempt at exercising the architecture in a relevant domain, to see where it works and does not. We have yet not conducted a formal evaluation of the model.

The model represents a hypothetical Iraqi male head of household (HoH). The model application is intended to be a hypothetical training system for soldiers learning to understand the implications of their actions in a house-to-house search scenario. The schema we have developed are from anecdotal sources in talking with cultural anthropologists and soldiers about the target domain, rather than from any detailed ethnographic study. We have also consulted literature and documents for some details where they might apply (Baker, 2003; USArmy, 2003, 2004). The model is meant to respond to soldier actions in “culturally appropriate ways.” The user interface includes ways for a trainee to interact with the model in a kind of “choose your own adventure” style – selecting at each step one action from a set of allowable actions, and observing the responses from the model.

### Example Schema

In this domain and model, we have developed a few schema that are appropriate to the domain task of house searches, from the perspective of a male head of household whose house is being searched. We have surmised that even if this person’s house had never been searched before, he might have heard of such incidents and so might have developed a schema based on those descriptions. These schema include what we describe as a “respectful house search” and a “disrespectful house search,” and the trainee essentially navigates through some variations of these two extremes by selecting actions with respect to interacting with the head of household, his family, and searching the house. Figure 3 illustrates the “respectful house search” schema, in which the head of household greets the soldier at the door, and invites him in to conduct business.

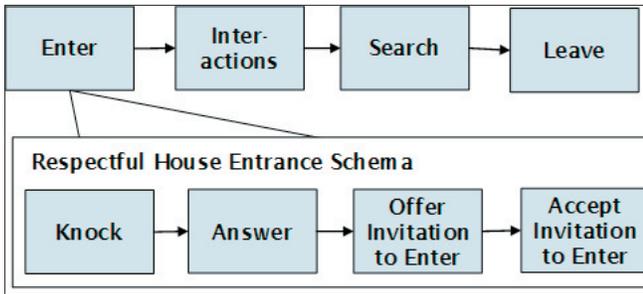


Figure 3: A hypothetical house search schema with “respectful” entrance scene.

Figure 4 illustrates two variations on the “disrespectful house search schema” in which the soldier enters the house forcefully (kicking down the door) or knocks during prayer time. These generate negative emotional responses because they are both unexpected and because they break the expectations of the head of household.

While we present these schema as neatly constructed hierarchies to give a sense of their contents, in the implementation, their definition is in the form of multiple rules that do not fire until an event activates subsets of them. This is similar to (Rumelhart et al., 1986), where we have defined the units and relationships in advance, and where an activated schema is constructed in working memory incrementally with new data from the environment.

Figure 5 shows a selected action schema from the head of household in response to a disorderly house search by the soldiers, in which the head of household exhibits “confrontational” behavior by shouting at the soldiers. This Confrontation schema represents a coping strategy on the part of the HoH to offset the emotional response generated from the disrespectful house search. This response might be selected versus a more physical confrontation because the man’s appraisal indicates that he has little power with respect to the armed soldiers.

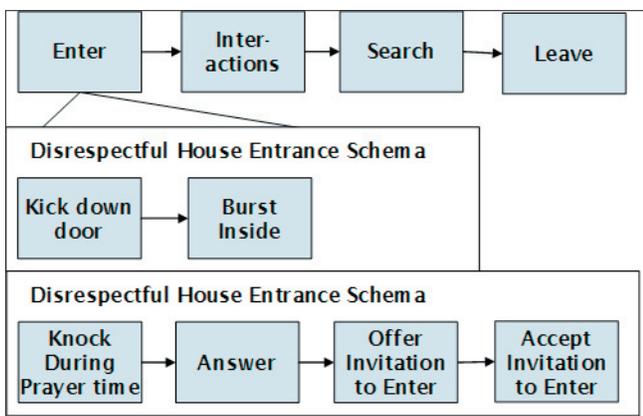


Figure 4: A variation of the house search schema with “disrespectful” ways to enter a house.

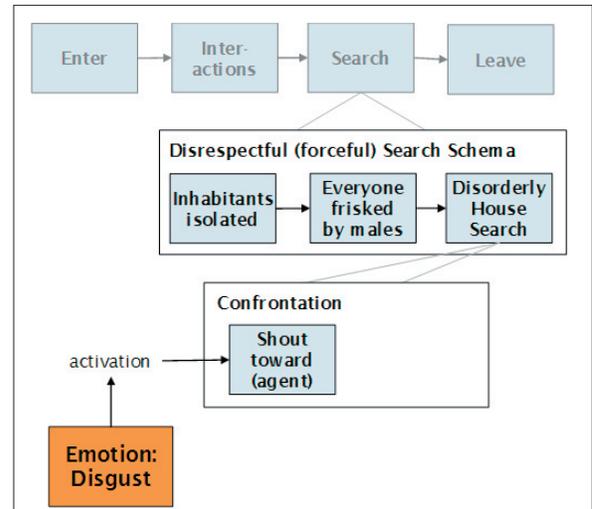


Figure 5: Schema (Confrontation) activated based on emotional response.

### Prototype Graphical User Interface

As mentioned earlier, the GUI is framed in part as a simple entry into a “training application” for soldiers learning how an Iraqi head of household might respond to different actions on the part of the soldier. In this way, the trainee can interact with the model via a point-and-click interface, and see responses to his or her actions. The GUI is also designed to demonstrate how the model is working, not as part of the training application, but rather as a debugging tool at the architecture level. The rest of this section describes each component of the GUI, which is illustrated in Figure 6.

**Activated Schema** The upper-left pane shows a list of the agent’s schema instances. When these are script templates, they can be expanded to show the scenes that have completed (which, as schema instances, can be further expanded, etc.), as well as the pending scene within the script.

**Events** The middle-left pane describes the events that have taken place thus far in the model. The current selection is linked to the view in the Event Appraisal pane (middle-right).

**Environment Inhabitants** The environment consists of a building with rooms, locations, doors, and persons, which are assigned a color based on their affiliation (blue, red, green). Appraisals and emotional responses shown pertain to the Iraqi head of household.

**House View (Plan View)** The upper-right pane shows the physical state of the simulation environment, including the locations of all entities (red and blue dots) and the states of all doors.

**Available Trainee Actions** Trainee actions allowed by the

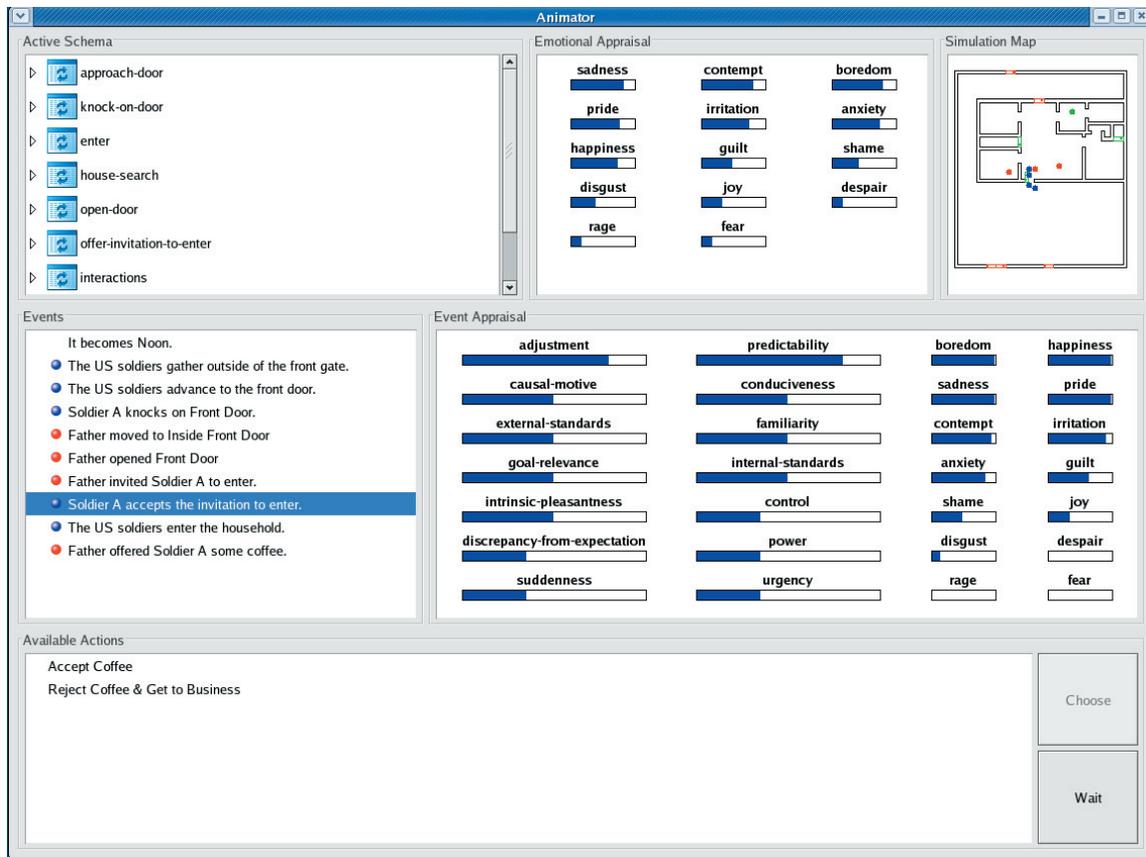


Figure 6: Prototype screen shot, showing the active schema, current emotional appraisal, events, and appraisal of events

current situation are listed in the lower left pane, and can be selected by pressing the 'Choose' button in the lower right. The option to take no action, 'Wait,' is always available.

**Event Appraisal** Clicking on events in the event pane brings up the appraisal and the emotional response to those events. (Not all events are appraised, and in such cases the responses are blank.) Response dimensions range from -1 (strong

negative) to 0 (neutral) to +1 (strong positive).

**Emotional Appraisal** The overall emotional state of the Iraqi head of household is shown in the top middle pane. As described in the text, this is an aggregate of his emotional responses to all appraised events, with more recent events weighted more highly

### Lessons Learned

One immediate lesson from attempting to develop models of social theory is that many of these theories are more descriptive in nature than causal or empirically predictive. This sometimes makes it difficult to define mechanisms that would account for the outward behavior of a human system, thus difficult to define these same mechanisms computationally. For this reason, we have tried to rely on the existing Soar architecture at the very least as a solid basis for cognitive modeling, so we are not entirely re-inventing the wheel. Further, when attempting to implement such a theory in a computational framework, one becomes immediately aware of gaps in the theory where a modeler must make assumptions in order to make the model work. In this spirit, we have been as forthcoming as possible in our description of this implementation so that others can replicate the work and,

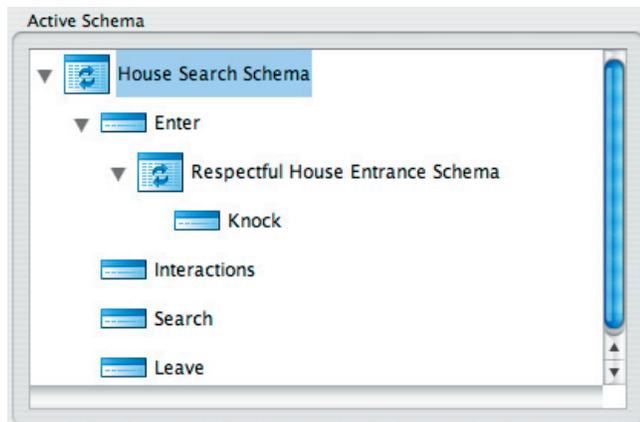


Figure 7: Prototype screen shot, showing detail of active schema

where appropriate, criticize our choices in implementation.

The processing mechanism we have developed for activation of schema is appealing in a few ways. First, schema are in a sense “floating around” in the model, awaiting an appropriate trigger event that would activate them. This allows for multiple schema to be independently activated given the same trigger event. In a sense, this comes “for free” within the Soar architecture -- with schema represented partially in rules, the rules will only fire when appropriate events occur to trigger a schema. Currently the system will track along with each active schema so long as events keep each schema relevant. When an event occurs that makes one schema more likely, its activation is raised and the other schema activations are lowered, bringing the one to the fore. Where there are no relevant external events to activate new schema, “background schema” can remain active to maintain a basic level of behavior.

Where we are lacking in this regard is in some mechanism to provide “preponderance of evidence” effects such as might be found in a Bayesian solution. For example, consider a schema for “answer phone call from X” that begins with the event “phone rings” and that has an activation condition “caller ID says X”. If the phone rings and no caller ID is observed, we would instantiate this schema many times, once for each potential caller; but each instance would have a low activation level. Therefore, even though the next action may be “pick up the phone” in all of these schema, we will still prefer to take the action suggested by any single schema with a moderate activation level, and may end up turning on the TV instead of answering the phone. More generally, we are not able to represent relationships between parallel active schema; each evolves independently of all the others.

### Conclusions and Future Work

We have presented the beginnings of a cognitive architecture, extended from the Soar architecture, as the basis for building knowledge-based models of culture. We have used Cultural Schema Theory and Appraisal Theory as the building blocks of this computational architecture.

The hybrid symbolic-numeric model we have developed is a shortcut to a more fully-integrated hybrid architecture built within Soar. Some of the choices made here for activation processing were purely pragmatic given the constraints of the project. However, similar work is being done to implement activation levels within the Soar architecture proper (Nuxoll et al., 2004). We would expect to leverage that work when it becomes available.

The schema we have describe here are of course hypothetical schema, based only on anecdotal conversations with soldiers and cultural anthropologists. Validation of this framework, and the schema in the prototype, is not a trivial task. Collecting data to populate schema is essentially an exercise in ethnography. Some methods have been proposed to conduct ethnographic studies in order to develop a cultural

model, and also to evaluate the resulting model (D’Andrade, 2005). We hope in the future to look more at existing ethnographic studies, or conduct our own to gather data.

We have taken the stance that, in large part, “culture is knowledge,” and our implementation here reflects that mindset. With this, we are bound to come up against the knowledge acquisition bottleneck. Certainly we could consider automated means of capturing this kind of cultural knowledge, but there is a lack of robust technology to do this in well-defined domains, let alone in cultural arenas, where the standard method of collecting and capturing data is by ethnographic means. Addressing this problem will be one rich avenue of future work.

### Acknowledgments

We would like to thank Dr. Scott Atran for his deep insight into culture and cultural studies. Thanks also to Bob Marinier of the University of Michigan for many fruitful discussions about appraisal that influenced this work. Thanks also to the soldiers/instructors West Point and ROTC at University of Michigan for discussions about cultural behavior. We would also like to thank Dr. Linda Pierce of Army Research Labs, and Dr. John Allen, formerly of DARPA, for support on this contract.

### References

- Alba, J. W., & Hasher, L. (1983). Is memory schematic? *Psychological Bulletin*, 93(2), 203–231.
- Axelrod, R. (1997). The Dissemination of culture. A model with local convergence and global polarization. *Journal of Conflict Resolution*, 41 (2), 203-226.
- Baker, W. (2003). *The Cultural Heritage of Arabs, Islam, and the Middle East*: Brown Books.
- D’Andrade, R. (1992). Schemas and motivation. In R. G. D. A. C. Strauss (Ed.), *Human motives and cultural models* (pp. 23-44). Cambridge, UK: Cambridge University Press.
- D’Andrade, R. (2005). Some Methods for Studying Cultural Models. In N. Quinn (Ed.), *Finding Culture in Talk*: Palgrave Macmillan.
- Ellsworth, P., & Scherer, K. (2003). Appraisal Processes in Emotion. In R. J. e. a. Davidson (Ed.), *Handbook of Affective Sciences*: Oxford University Press.
- Ellsworth, P., Scherer, K. R., & Scherer, K. R. (2003). Appraisal Processes in Emotion. In *Handbook of Affective Sciences* (pp. 572-595). Oxford: Oxford Univ. Press.
- Gratch, J., & Marsella, S. (2004). A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Systems Research*, 5(4), 269-306.
- Hofstede, G. (1980). *Culture’s Consequences International Differences in Work-Related Values*. Newbury Park: SAGE Publications, Inc.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). *Soar: An*

- architecture for general intelligence. *Artificial Intelligence*, 33(3), 1-64.
- Lazarus, R. S. (1991). *Emotion and Adaptation*. New York: Oxford University Press.
- Marinier, R., & Laird, J. (2006). A Cognitive Architecture Theory of Comprehension and Appraisal. Paper presented at the Agent Construction and Emotions 2006, Vienna.
- Marsella, S., & Pynadath, D. (2007). When and to What Extent Is Cultural Modeling Feasible?, BRIMS. Norfolk, VA: SISO.
- Matthews, H. (1992). The Directive Force of Morality Tales in a Mexican Community. In R. G. D. A. C. Strauss (Ed.), *Human motives and Cultural Models* (pp. 127-162). Cambridge, UK: Cambridge University Press.
- Minsky, M. (1975). A framework for representing knowledge. In P. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill.
- Nason, S., & Laird, J. E. (2004). Soar-RL, Integrating Reinforcement Learning with Soar. Paper presented at the International Conference on Cognitive Modeling.
- Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.
- Newell, A., & Simon, H. A. (1963). GPS: A Program that Simulates Human Thought. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and Thought*. New York: McGraw-Hill.
- Nisbett, R., Peng, K., Choi, I., & Norenzanan, A. (2001). Culture and system of thoughts: Holistic versus analytic cognition. *Psychological Review*, 108, 291-310.
- Nuxoll, A., Laird, J., & James, M. (2004). Comprehensive Working Memory Activation in Soar. Paper presented at the International Conference on Cognitive Modeling.
- Nuxoll, A., & Laird, J. E. (2004). A Cognitive Model of Episodic Memory Integrated With a General Cognitive Architecture. Paper presented at the International Conference on Cognitive Modeling.
- Peng, K., & Nisbett, R. (1999). Culture, dialectics, and reasoning about contradiction. *American Psychologist*, 54(741-754).
- Quinn, N., & Strauss, C. (2001). *A Cognitive Theory of Cultural Meaning*. Cambridge, UK: Cambridge University Press.
- Rumelhart, D., Smolensky, P., McClelland, J., & Hinton, G. (1986). Schema and sequential thought processes in PDP models. In e. a. Rumelhart (Ed.), *Parallel Distributed Processing* (pp. 7-57). Cambridge, MA: MIT Press.
- Schank, R., & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding: An enquiry into human knowledge structures*. Hillsdale, NJ: Earlbaum Assoc.
- Scherer, K. R. (1997). The Role of Culture in Emotion-Antecedent Appraisal. *Journal of Personality and Social Psychology*, 73(5), 902-922.
- Scherer, K. R., Schorr, A., & Johnstone, T. (Eds.). (2001). *Appraisal processes in emotion: Theory, Methods, Research*. Appraisal processes in emotion: Theory, Methods, Research: Oxford University Press.
- Triandis, H. (1994). *Culture and Social Behavior*. New York, NY: McGraw-Hill.
- USArmy. (2003). *1st Infantry Division Culture Guide to Iraq*.
- USArmy. (2004). *Iraq Culture Smart Card*. In U. Army (Ed.).
- Wehrle, T., & Scherer, K. (2001). Toward Computational Modeling of Appraisal Theories. In K. Scherer, A. Schorr & T. Johnstone (Eds.). Oxford, UK: Oxford University Press.