AFRL-HE-WP-TR-2006-0008

AIR FORCE RESEARCH LABORATORY

Personality-enabled Architecture for Cognition (PAC): Architecture and Initial Implementation

Wayne Zachary
Jean-Christophe Le Mentec
Vassil Iordanov
Andrew Rosoff
James Stokes
James Eilbert

CHI Systems, Inc.
1035 Virginia Drive, Suite 300
Fort Washington, PA 19034

Lynn Miller
Stephen Read

University of Southern California
University Park Campus
Los Angeles, CA 90089

March 2005

Interim Report for January 2004 to March 2005

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them

This report was cleared for public release by the Air Force Research Laboratory Wright Site Public Affairs Office (AFRL/WS) and is releasable to the National Technical Information Service (NTIS). It will be available to the general public, including foreign nationals.

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies and their contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2006-0008

This technical report has been reviewed and is approved for publication.

FOR THE DIRECTOR

//SIGNED//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* March 2005	**2. REPORT TYPE** Interim	**3. DATES COVERED** *(From - To)* January 2004 - March 2005

4. TITLE AND SUBTITLE
Personality-enabled Architecture for Cognition (PAC): Architecture and Initial Implementation

5a. CONTRACT NUMBER
FA8650-04-C-6439

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63231F

6. AUTHOR(S)
[1]Wayne Zachary, [1]Jean-Christophe Le Mentec, [1]Vassil Iordanov, [1]Andrew Rosoff, [1]James Stokes, [1]James Eilbert, [2]Lynn Miller, [2]Stephen Read

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
28300411

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

[1]CHI Systems, Inc.
1035 Virginia Drive, Suite 300
Fort Washington, PA 19034

[2]University of Southern California
University Park Campus
Los Angeles, CA

8. PERFORMING ORGANIZATION REPORT NUMBER
04007.050329

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HECS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TR-2006-0008

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited. AFRL/WS-06-0370, 10 Feb 06

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This research is a collaboration between CHI Systems and the University of Southern California, to develop a new technical capability to create human behavioral representations (HBRs) with pre-defined and specific personality traits and cultural characteristics. This capability meets a current and growing need for human models that exhibit personality and cultural variability. The need arises from multiple sources, but primarily from the increased frequency and complexity of military operations involving coalition forces with great cultural and personality diversity, and the growing trend toward asymmetrical conflicts involving adversaries with poorly understood cultural values, characteristics, and behavior patterns. The research integrates theory and empirical data from personality psychology, social psychology, cognitive science, and neuroscience to create a new software environment called the Personality-enabled Architecture for Cognition (PAC). Unlike existing cognitive architectures that attempt to build affective and personality factors as customizations to an underlying formally rational symbolic architecture, PAC uses dimensions of personality, emotion, and culture as foundations for the cognitive process. This report covers the first year of effort in the program, which focused on developing the initial PAC architecture and software implementation.

15. SUBJECT TERMS Human Behavioral Representation (HBR), Personality-enabled Architecture for Cognition (PAC), Cognitive Architectures, Human Performance Modeling

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Terry W. Stanard
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED	SAR	94	**19b. TELEPHONE NUMBER** *(include area code)* 937-255-9938

THIS PAGE LEFT INTENTIONALLY BLANK

EXECUTIVE SUMMARY

CHI Systems, in collaboration with researchers from the University of Southern California, is developing a new technical capability to create human behavioral representations (HBRs) with pre-defined and specific personality traits and cultural characteristics. This capability meets a current and growing need for human models that exhibit personality and cultural variability. The need arises from multiple sources, but primarily from the increased frequency and complexity of military operations involving coalition forces with great cultural and personality diversity, and the growing trend toward asymmetrical conflicts involving adversaries with poorly understood cultural values, characteristics, and behavior patterns.

This research integrates theory and empirical data from personality psychology, social psychology, cognitive science, and neuroscience to create a new software environment called the Personality-enabled Architecture for Cognition (PAC). Unlike existing cognitive architectures that attempt to build affective and personality factors as customizations to an underlying formally rational symbolic architecture, PAC uses dimensions of personality, emotion, and culture as foundations for the cognitive process.

This report covers the first year of effort in the program, which focused on developing the initial PAC architecture and software implementation. The PAC is specifically design for application in synthetic agents in simulations and game applications for training, mission rehearsal, and analysis uses. The structure of PAC allows it to function as a personality/emotional layer that can be used stand-alone or integrated with existing constrained-rationality cognitive architectures. In addition, a preliminary set of tools was developed to support the authoring, analysis, and testing of PAC HBRs.

The PAC software development tools were then used to construct initial HBRs that served to test and debug the PAC systems, and to demonstrate the viability of the approach. The initial HBRs were based on characters from VECTOR, a game-based cultural familiarization trainer for the US Army. Analysis of these initial HBRs showed that the PAC engine can be used to create synthetic characters with tunable personality parameters that lead to different behavioral outcomes. Future phases of the effort will

enhance the initial software to incorporate emotion and cultural factors, and will begin to calibrate parameter sets to allow specific personalities to be created and stored.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

1.1 Prior Cognitive Modeling Research

The last fifteen years has seen a burst of research into computational systems that simulate or emulate human information processing. These systems, commonly called cognitive architectures (References 1, 2), have made remarkable first steps toward generative simulation of human behavior, a problem which Pew and Mavor (Reference 1) termed "possibly the most difficult task humans have yet undertaken" (*ibid*, p. 8). In addition to human behavioral simulation, existing cognitive architectures have also proven useful in developing cognitive agent applications such as intelligent interfaces (Reference 3) and intelligent tutoring systems (Reference 4). The most highly-developed systems (e.g., ACT-R/PM, COGNET/iGEN®, EPIC/GLEAN, OMAR, and Soar; see Reference 5, forthcoming) all share several common features. They each focus on the representation of knowledge, and specifically on the symbolic knowledge representation, and each contains a similar set of mechanisms for manipulating symbolic knowledge -- perceptual, cognitive, and motor processors, various kinds of memory, sensory transducers, and motor effectors. The few architectures (particularly ACT-R/PM) that pay attention to sub-symbolic processes do so primarily as supporting mechanisms for learning or manipulating symbolic information.

Despite their successes with well-defined specific work tasks or human-computer interactions (References 6, 7), fundamental problems remain. For example, the models are unable to know when to give up a task or approach and try something else, just as they are unable to recover from unexpected events or (their own) errors. They have a general inability to flexibly shed and gain tasking and adapt to group-situated processes. They are unable to respond to pressure, particularly by 'stepping up' and increasing performance, just as they are unable to demonstrate boredom, frustration, or even self-preservation. In short, they behave more like smart automata than like real people with underlying biologically-based motivational and emotional systems guiding a more dynamic system impacting cognition and behavior (Reference 8). It is almost as if the emotive and motivational factors have been viewed as epiphenomena of some central but symbolically rational processes -- as 'icing' on the cognitive cake.

We believe that much the opposite is really the case. More fundamental motivational and emotive forces are in fact the building blocks from which the most interesting, adaptive, and flexible parts of human behavior arise, as well as the source of individual and group variability commonly termed 'personality' and 'culture.' This report documents that the first stage of research is to identify and formalize these building blocks, and integrate them with the traditional rational purposive symbolic framework to create a new and more general architecture for cognition. The product of this research is a substantially new cognitive architecture, which we call the Personality-enabled Architecture for Cognition (PAC). The PAC represents a new approach to cognitive architecture, one that dramatically opens up and expands the scope by integrating constructs from psychology and neuroscience into conventional symbolic cognitive architectures.

1.2 Requirements for PAC

In today's multi-national military endeavors, factors of personality, affect, and culture influence decision-making, mission performance, and communication in many ways. The post-Cold War adversary has become a regional or non-governmental force, driven by a cultural/religious (rather than political) ideology, and led by 'strong man' leaders whose personalities directly affect military and political strategy. At the same time, the globalization of allied and coalition forces is increasingly creating the need for military systems and military operations to be staffed by personnel with highly varied cultural (value and personality system) backgrounds. There is, thus, a very strong need for more realistic representations of these factors in the human behavioral representations (HBRs) that are used in military simulations for training, mission rehearsal/analysis, and acquisition. It is against this general need that the PAC research has been undertaken, and that the following requirements for PAC were derived:

1. Be based in psychological data and theory. The overall organization of PAC and its core set of concepts, constructs, and operating principles must be explicitly linked to specific psychological theories of personality, culture, and cognition in

the scientific literature, to provide it with an explicit and principled scientific foundation.

2. Provide a generative, parameterized modeling capability – PAC, as an architecture, must be implemented at a level of abstractness that:

   - allows it to execute HBR models that *generate* behaviors as the (external) situation unfolds;
   - incorporates parameters that allow a given HBR to exhibit different personality traits leading to different behavioral sequences depending on the values set for various of those parameters.

   This is needed so that a different cognitive agent can be executed under the same conditions but with different personality-based behaviors.

3. Produce empirically testable behavioral predictions. PAC must give developers of PAC-based HBRs the ability to form behavioral predictions, in the form of observable, testable expectations of how the HBR should behave under different circumstances and personality parameter settings. This will allow both the specific HBR and PAC in general to be empirically tuned, verified and validated against human behavioral data.

4. Be compatible with emerging simulation paradigms, particularly use of game-engine based applications. The HBRs constructed and executed with PAC should be able to be inserted and executed in the broadest range possible of simulation environments, but including at least standards-based (e.g., HLA-based) constructive simulations and game-engine based simulations. This will give PAC the greatest flexibility to meet the range or required uses by the Air Force.

5. Support model-development needs. Beyond the software engine that embodies the PAC architecture, PAC also needs to include components that structure and support the development, testing, and debugging of PAC-based HBR models. This allows the PAC to be used efficiently and effectively by other researchers, engineers, and analysts.

The research described here was organized to achieve these requirements for PAC.

1.3 Organization of the Report

Section 2 of this report describes the psychological theory of personality that forms the scientific foundation for PAC. It also details the research methods used to move from this theoretical basis to a functioning PAC system. The primary method is a spiral development strategy, which produces an evolving sequence of implementations of PAC over the life of the project. Section 3 reports the results of the effort, in four parts.. Subsection 3.1 describes the conceptual architecture and organization of PAC, including its relationship with existing cognitive architectures. Subsection 3.2 describes, in both general and specific terms, the knowledge representation created to accompany the overall processing architecture described in Subsection 3.1. Subsection 3.3 then details the specific computational system that was designed and implemented to operationalize the conceptual architecture and knowledge representation of PAC. The set of development tools that was created to support development and testing of PAC-based HBRs is then described in Subsection 3.4.

Section 4 discusses the results from a variety of perspectives. Subsection 4.1 describes and discusses two simple HBRs that were built with the initial versions of PAC. Subsection 4.2 analyzes the current PAC implementation in terms of its psychological goals. Subsection 4.3 discusses the usefulness and role of the PAC development tools, and Subsection 4.4 discusses the significance of the results of the first year of the PAC effort in terms of its overall requirements. Finally, Section 5 provides some overall conclusions.

2. METHODS, ASSUMPTIONS, AND PROCEDURES

This section describes the foundations for the research. Subsection 2.1 reviews the psychological theories and data from which PAC was developed. Subsection 2.2 then details the methods used in the first year of the research program.

2.1 Underlying Theory of Personality

Personality can be simply defined as enduring tendencies to think, feel, and behave in consistent ways. Work on the lexical analysis of trait language (e.g., Reference 9) and work on the structure of a variety of different trait scales (e.g., Reference's 10, 11, and 12) have given rise to what is commonly called the Five-Factor Model of personality. The Five-Factor model, also called simply the 'Big Five', is a hierarchical model of personality, in which relatively narrow and specific traits are organized in terms of five broad factors: Neuroticism, Extraversion, and Openness to Experience, Agreeableness, and Conscientiousness (Reference 13). As the overall evidence in support of the Big Five taxonomy has grown over the past two decades, FFM has become the personality model of reference for personality psychologists. However, even though the Big Five model has demonstrated generalizability across different cultures and languages, it remains a psycholinguistic, and not a cognitive framework. Thus, the Big Five model can not simply be translated into software for PAC. Rather, a cognitive foundation that is consistent with the Big Five model had to be developed separately.

The development of this cognitive foundation for PAC is based on prior work by the research team (and others), which is summarized in the following paragraphs. This past research suggests that it is possible to effectively capture personality by differential configurations and relative activation of goals, plans, resources, and beliefs (References 14, 15). Personality traits, such as ‘helpful’, ‘gregarious’, and ‘dutiful’ can be directly translated into configurations of goals, plans, beliefs, resources and stylistic behaviors. For example, the trait “helpful” can be decomposed into the goal to help others, beliefs about the value of helping others and whether others deserve help, the plans to help, and the resources to do so.

Recent findings in neuroscience and temperament (References 16, 17, 18, 19, 20) suggest that these goals central to personality are organized into two levels; namely, specific (or level one) emotional/motivational systems and broader, overarching (or level two) motivational systems. Mapping of brain circuits and neurotransmitter systems (References 21, 22), and evolutionary analyses (References 23, 24, 25, 26) provides evidence for a set of *level one emotional/motivational systems* that handle the variety of major adaptive strategies that people must incorporate and pursue in everyday life. Among these adaptive strategies are: (1) social bonding, (2) fear of social separation, (3) dominance and the development of authority relations in groups, (4) exploration and play, (5) caring and parenting, (6) mating, and (7) self-preservation and concerns for physical safety. Each of these strategies corresponds to a motivational system that organizes a set of *individual goals*; these individual goal sets are the basis of the specific traits discussed above.

At a more general level, are *level two overarching motivational systems* -- a Behavioral *Approach* System (BAS), which governs sensitivity to reward and *approach* to rewarding stimuli (and active exploration) and a Behavioral *Inhibition* System (BIS), which governs sensitivity to punishment and *avoidance* of threatening stimuli (References 17, 26).

There is evidence that these motivational systems can be mapped onto the personality structure that has been revealed by the lexical analysis of trait language (e.g., Reference 9) and work on the structure of a variety of different trait scales (e.g., References 10, 11, 12). The work on personality structure provides evidence for what is now termed the Big-5 dimensions of personality: Extroversion, Neuroticism, Agreeableness, Conscientiousness and Openness to Experience. There is considerable evidence, to date, that the broad *level two motivational systems* provide a biological basis for at least two of these dimensions: Extroversion and Neuroticism. A third brain system, to be discussed presently, may provide the biological basis for Conscientiousness. In addition, there is also tentative evidence (e.g., References 21, 27) that aspects of the *level one* motivational systems provide the biological bases for Agreeableness and for Openness to Experience.

Work on theories of temperament (e.g., References 16, 17, 18, 19, 20) suggests that the biological basis for the extroversion dimension is the sensitivity to reward in the BAS (Reference 26). Similarly, underlying Neuroticism is the sensitivity to punishment or the desire to avoid threat that BIS mediates (Reference 26). A number of other authors, working on affect, have similarly argued for such approach and avoidance systems (References 28, 29, 30).

Davidson (see References 31, 32) has provided extensive evidence that the left and right prefrontal cortexes (PFC) are differentially involved in the BIS and BAS systems. The left PFC seems to process positive, approach-related or appetitive emotions, whereas the right PFC processes withdrawal related emotions, such as fear, disgust, and sadness. Interestingly, anger is more typically related to the approach system and seems to be processed in the left PFC, with more approach related emotions (e.g., anger associated with not achieving goals towards which one is striving) (Reference 32).

Thus, recent neuropsychological research suggests that these systems are mapped into the left and right Prefrontal Cortex, respectively, and may integrate and provide a "read-out" from the lower level *emotional/motivational* systems (Reference 28). These two levels are highly and bi-directionally interconnected such that the lower level systems send activation to the higher level systems; while at the same time the higher level systems can influence the activation of the lower level goal systems, as illustrated in Figure 1. For example, there are feedback pathways from the prefrontal cortex to the lower level motivational systems, so that global changes in the BAS and BIS can affect a wide range of goals that are integrated by the corresponding system. Thus, changes in the behavior of higher-level systems can influence the behavior of all the goal systems that are integrated by the higher-level system. Note also that changes in the lower level motivational systems can influence all of the individual goals that are organized by that system. Thus, changes in the sensitivity or activation of the caring system or the dominance system can influence the individual goals within all those systems.

The goal systems represented in Figure 1 represent an evolution in our thinking about the specific goals that can be found in the *level one* motivational systems, based on recent cluster analyses of a large sample of trait terms (Reference 33). These analyses suggested a modification in the major goals that will be represented in the initial implementations of the PAC program.

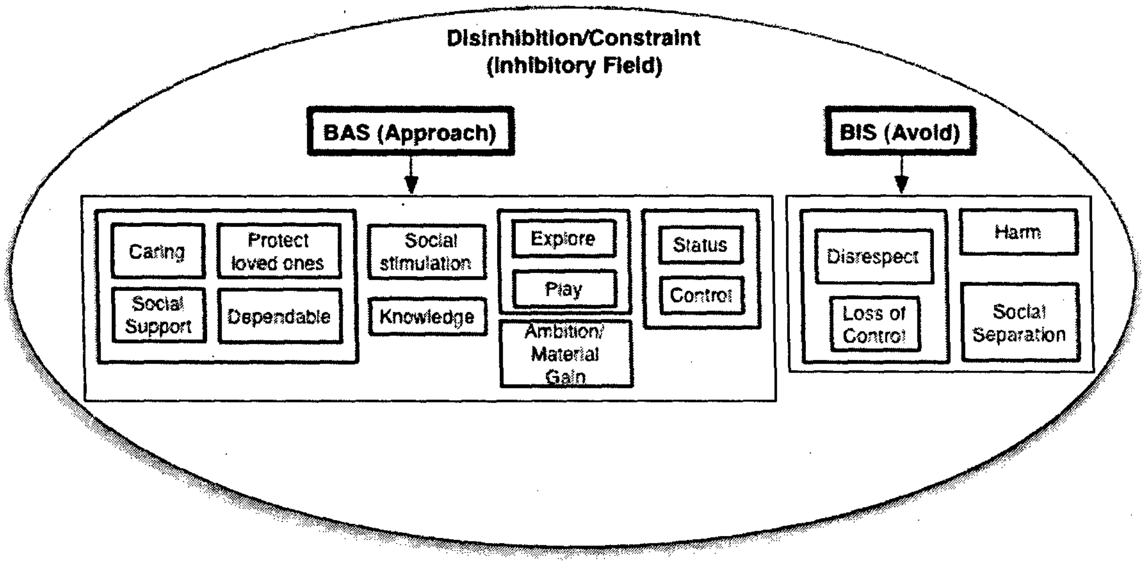


Figure 1. Interconnections of PAC Motivational Systems

In addition to the two overarching motivational systems, there is evidence for a third brain system, the Disinhibition/Constraint system (DCS) that provides for an even more general level (level three) of inhibitory control for the other systems (References 34, 35). Inhibition acts to enforce selectivity among activated concepts by enhancing the differences in their activations (see Reference 36). Higher levels of inhibition result in greater differentiation among concepts, as only the most highly activated concepts will remain active. As a result, variability in strength of inhibition affects the likelihood that various concepts will play a role in cognition, motivation, and behavior. Thus, DCS may govern the extent to which the system is goal-focused (resulting in enacting more goal-directed behavior) versus highly reactive to changing environments (resulting in an individual appearing more prone to distraction). This system may provide part of the biological underpinnings for the broad trait of Conscientiousness.

There is tentative evidence that the other two major Big Five dimensions, Agreeableness and Openness to Experience, are also to be found in this neuro biological model. Specifically, Reference 27 (in press) reviews evidence for a biological system that underlies Agreeableness, which corresponds to the cluster of goals in

Figure 1 involving: Caring for others, Protecting Loved Ones, Being Dependable, and Seeking Social Support. In addition, the cluster of goals involving Exploration and Play, which have been identified in recent cluster analyses (e.g., References 21, 37), may play a central role in Openness to Experience.

Furthermore, each of the broad traits in the Big-5 has many specific facets or subcomponents (References 11, 38). For example, Extroversion seems to have separate components for energy level, gregariousness, and dominance. And, Neuroticism seems to have separate components corresponding to hostility, anxiety, and fear of social rejection. These facets will map onto the underlying dynamics of the PAC model.

The PAC system should also be able to capture many aspects of culture and emotion, through its emphasis on motivational systems (goals), plans and beliefs. Culture has been viewed as "personality writ large" (Reference 39). That is, culture may reflect the underlying personality propensities in a given population – some of which may be biologically-based and some of which will be based on the goals and beliefs of local populations. One central aspect of culture is the goals and values of that culture (References 40, 41, 42, 43). For example, many cultures emphasize rank, status, and competition, while others emphasize cooperation, sharing, and equality (Reference 44). Related to this, some cultures have been described as agentic (focused on individual outcomes) and others as communal (focused on group outcomes) (Reference 45). (This has also been referred to as the difference between individualistic and collectivistic cultures (References 46, 47, 48).

There are a number of other ways in which culture may impact behavior in the PAC model (for examples see References 49, 50, 52).

(1) culturally-based beliefs and norms might constrain (or afford) actions;

(2) plans (and knowledge of them) may differ across cultures for achieving particular goals;

(3) emotional and other stylistic displays may be constrained by cultural norms and values;

(4) cultural differences in temperament (e.g., Reference 52) (and its reflection in the model, e.g., differential sensitivity of the BIS and BAS) may impact variability in behavior; and

(5) although there are cross-cultural universals in the attachment/bonding systems, there are cross-cultural differences in the extent to which individuals are emotionally secure versus anxious and easily threatened in their interactions with others versus more avoidant/dismissing and less trusting of others (Reference 53).

2.1.1 The Resulting PAC Functional Organization and Architecture

The PAC framework integrates results from multiple sources, ranging from psychometric work on trait scales (References 12, 54) and lexical analyses of trait language (References 9, 55) to recent work in neuroscience, identifying specific brain systems for different motivational domains (References 21, 27, 56). From these, an articulated, general PAC model is being constructed in which personality is based on a hierarchically organized set of motivational systems, ranging from individual goals to higher order approach and avoidance systems (Reference 57). The PAC model integrates symbolically and biologically-based theories to yield a hybrid cognitive/neural/biological framework that provides a generative model of personality and individual behavioral plasticity.

Personality is represented in terms of the way in which individual differences in goal-based structures affect a person's (or HBR's) interpretation of a situation and the behavior of other agents, and how the agent responds, given that interpretation. As part of the current research, a trait ontology is being developed in which traits are decomposed into conceptual components which can be translated into the planning and goal structures of an intelligent agent. This ontology should both provide a principled way to program specific personality traits into PAC, as well as making it easier for the user to do so.

The underlying dynamics of personality in the PAC system can be conceptualized as involving potentially “tweakable” parameters, set to initial starting positions. For example, in the current version of the PAC one can set the parameters in the model for a given “personality” such that there might be a greater initial sensitivity of the BAS or the BIS system, greater initial activation of specific goals, or stronger versus weaker activation of the inhibitory/constraint system. Concurrently, cues from situational features can also activate particular systems more so than others (e.g., situational cues suggesting threats would activate the BIS rather than BAS system), feeding into the overall activation of competitive forces in the system (e.g., relative activation of BIS and BAS systems) driving behavior.

Thus, emergent behavior in the PAC model would be a true interaction of initial starting positions and input from situational features, yielding "trait-like" personality by agents that is apt to exhibit many of the patterns of personality observed in the personality literature (Reference 58). By modifying the configuration of goal-based structures, PAC should be able to create different 'personalities' — computational agents who respond differently to similar input.

2.2 Research Methodology

The initial year of the PAC effort has been approached primarily as computational research. That is, it has focused on design, development, and demonstration of an initial PAC software system. The initial PAC software is, of course, a necessary precondition for any subsequent validation, testing, and HBR application development.

2.2.1 Spiral Software Development Method

The PAC development process, as most research and development efforts, consists mainly of analytical and software implementation components. Because of its investigative nature, the analytical process can't be fully completed without proper validation, often performed with the use of the system's software modules. Subsequent adjustments to the system design usually also affect the software implementation. The conventional, top-down "design-then-implement" development model does not ensure proper feedback for the investigative process, thus, presenting serious risks. Therefore, a development process coupling the two components in an iterative, feedback sensitive manner seems more appropriate.

The spiral model is a software development model defined by Barry Boehm (Reference 58) combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. In the scope of the PAC project, the iterations are 2-3 months long (Figure 2).

Each phase starts with a design goal (such as a user interface prototype as an early phase) and ends with an investigator reviewing the progress thus far. Analysis and engineering efforts are applied to each phase of the project, with an eye toward the end goal of the project. The Spiral model has influenced the modern day concept of agile software development (Reference 59), and is rarely used alone for software development anymore. Agile development addresses a broader set of software issues than the spiral model and is particularly effective for small teams working with rapidly changing requirements. In the PAC development process, a spiral model has been applied to both investigative and implementation processes, while the latter also takes advantage of numerous agile development techniques.

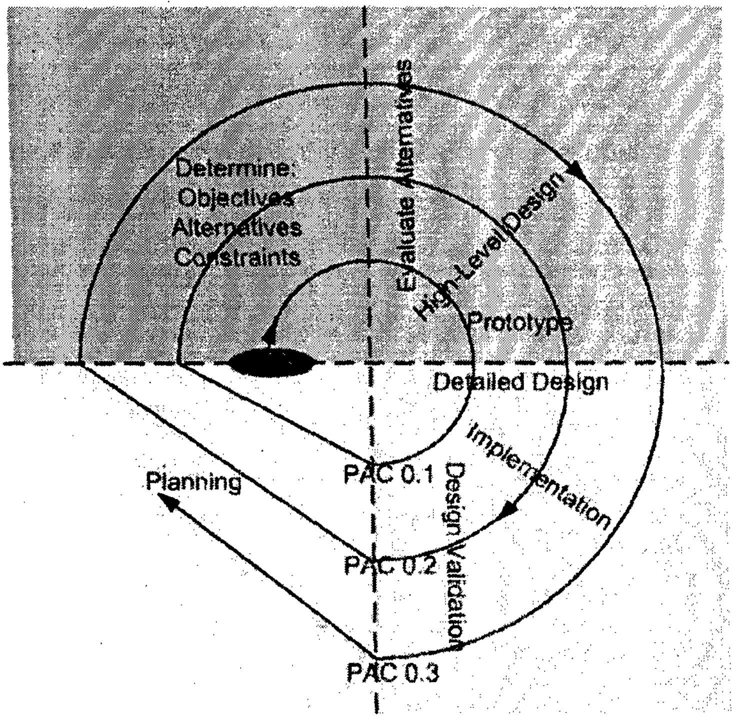


Figure 2. PAC Spiral Cycles

(Reference 60) describes communication, simplicity, feedback and courage as the driving values of extreme programming, one of the most prominent agile development methodologies. The PAC team has retained some of its programming principles, such as:

- Continuous integration
- Small Releases

- Simplicity in Design
- Refactoring
- Collective Ownership
- Testing

During the project's first 9 months, 3 iterations were performed starting with a rough-cut of representational, executable and GUI elements as an operable application, while introducing additional features in the following iterations.

- Iteration 1: Development focus is on obtaining a functional PAC system with a limited implementation of most major components. This iteration has produced an executable system - starting point for PAC modeling experimentation/evaluation - and a test-bed/demonstration environment.
- Iteration 2: Development focus is on PAC model development tools. This iteration has produced a PAC version with an improved user interface.
- Iteration 3: Development focus is on introducing new features such as emotions. Iteration 3 is the currently active activation.

Agile software development has allowed the PAC software team to minimize expenses in the form of rationale, justification, documentation, reporting, meetings, and permission to sustain a constant development pace while cooperating with the PAC investigative team. Adopting a spiral model has enabled both teams to work tightly together while maintaining risks and overhead costs at a reasonable level.

2.2.2 Use of Case Examples

The notion of "use case" has been a key element in software development methodologies since the 80's. In general a use case is a prose account of the purpose of a system under development: a description of an actor trying to achieve a goal by using the system (an Actors and Goals model of functionality). In practice, use cases vary from informal to formal, from general to highly detailed. Despite attempts, over a period of two decades, to introduce highly formal use cases into software development, day to day employment of use cases remains relatively informal, or at most, semi-formal. When Ivar Jacobson (a pioneer of the use case approach – see Reference 61) was asked about formal models of use cases, he replied, "Oh, I have a formal model for use cases, all right. The only problem is that no one wants to use it" (Reference 62).

More recently, Kent Beck (Reference 60), who created Extreme Programming, also avoided formal structure, preferring the notion of "user story." The continuing difficulty with formalisms may be related to the fact that use cases pose the core user-centric problem. That is, the representation of system use must be accessible to individuals with significantly different backgrounds and viewpoints, specifically individuals who understand the user's task and individuals who understand the software implementation issues. At the same time, a use case is being used to express early thoughts on design which may evolve quickly, giving a more informal medium an advantage in overall efficiency. Whatever the limitations of the many different formulations of the use case approach, it is clear that use case analysis is essential to moving development forward in a user-centric fashion. Given the range of formulations available, it is also clear that the approach needs to be adapted to the development problem at hand.

PAC represents a general use case that is familiar in the arena of cognitive modeling. This is perhaps best described in what can be termed a Stakeholders and Interests model, an aspect of use cases which compliments the original Actors and Goals model (Reference 63). Taking a familiar everyday example to illustrate the approach, a vending machine use case could be described entirely in terms of the behavior of the machine and the individual (primary actor) attempting to purchase an item from the machine. However, this would overlook the fact that a number of other stakeholders (the seller of the items, the site hosting the machine, etc.) are important in determining the functional requirements of the vending system. From this perspective, PAC stakeholders include the individual who must construct the agent using PAC, the cognitive scientist who uses PAC agents to explore theoretical issues, the developer of a larger system (e.g., training system) which incorporates agents for a specific application purpose, and the human who uses the larger system and is the direct "consumer" of the agents' behavior. These stakeholders drive the need for various types of functionality including a modeling tool or editor for the agent specification/construction process, an execution testbed for both algorithm (modeling solution) and agent development, execution inspection facilities for developers and researchers, and the software interfaces required to integrate agents with larger general purpose simulations to provide application context. As functional components, these are relatively well understood and familiar in various IDEs (Integrated Development Environments) and simulation-based virtual environments, many of them far removed from the issues of cognitive modeling.

In a very real sense, the primary actor in a PAC use case is the human interacting with the agent. By means of this interaction, the human is trying to accomplish some goal, such as a goal specified as part of a training exercise. The utility of the interaction (e.g., for training purposes) is generally considered to be dependent on the validity of the model, that is, the validity of the agent as a representation of human behavior. In the larger world of software development, the use case approach normally involves the acquisition of the business knowledge of the user. In the PAC case, it is the human's knowledge of and expectations for the behavior of other humans which must be captured as system requirements. Specifically, PAC requirements include the human-like expression of personality and cultural characteristics and emotions on the part of the agent. Clearly the acquisition of user knowledge for PAC is quite different from the common business example. In fact, very little of this knowledge can be provided directly by the user, and PAC must rely on current scientific theories of personality and culture to characterize expected agent behavior.

Ideally, a detailed PAC use case would specify how a modeler constructs an agent which behaves or interacts in specific ways in specific contexts. However, theories of personality, culture, and emotion are themselves general constructs and often incomplete as related to the phenomena of interest. There is, for example, little data on how the expression of fear might vary in a specific behavioral context (e.g., a specific battlefield situation), depending on personality or cultural traits. A use case, or comprehensive set of cases, for the final version of PAC cannot therefore be expected at the start of development. Despite this fact, each spiral implementation in PAC development must embody a use case with considerable contextual specificity in order to avoid the familiar trap of addressing a toy problem. In each PAC spiral iteration, the development is being guided by one or more behavioral scenarios or vignettes. Each vignette is an exemplar of real world behavior central to a use case of interest, used to focus discussion and concept formation in a particular modeling area related to personality and culture. Modeling solutions are posed against this realistic target in order to assess their ability to account for behaviors of interest and to identify limitations of the solutions which must be addressed. Each vignette is specified informally, allowing discussions to range through personality- and culture-based interaction variants based on the same context.

The Spiral 1 vignette (see Section 4 below) is drawn from the VECTOR (Virtual Environment Cultural Training for Operational Readiness) training scenario. VECTOR is a game-based training system which allows a human user to interact with members of another culture (agents) to conduct cultural-familiarization. VECTOR includes an Arab cultural model and the training scenario allows the human trainee to interact with "Arab" agents playing various roles in a military context. The vignette of interest consists of a soldier approaching a shopkeeper in order to gather information. The interaction can play out in a number of ways, depending on the actions of the actors, as they move from greeting, to information request, and finally request resolution and disengagement. Development discussions proceeded by examining the mapping of possible personality motives (the focus of Spiral 1 efforts) to behavioral variations within the vignette's interaction sequence. For example, how would a shopkeeper highly motivated by concerns for personal safety react to an information request compared to a shopkeeper highly motivated by material gain, and how could this difference in motivation be implemented in a systematic way? (The general question of how theoretical constructs regarding personality and culture can/should be made computationally tractable to support modeling of human behavior has been focused through a use case scenario.) These discussions lead directly to the Spiral 1 design and implementation.

The Spiral 2 effort focused on implementing emotion within a PAC agent. Initial discussions have been organized around a vignette similar to that for Spiral 1, in the sense that the vignette represents a brief but complete (resolved) interaction episode. In this vignette, panhandlers approach an individual on a subway and request/demand money. The panhandlers may be more or less aggressive, depending on their own motives and emotional states, engendering more or less fear on the part of the individual approached. The situation can easily be recast in a military context with a similar opportunity to address questions related to the modeling of fear. This vignette, however, may not be especially appropriate for exploring the modeling of other emotions and multiple vignettes may be required. The work of Shaver, et al. (Reference 63) is particularly interesting in this context. They assert that "real emotional events, or exemplars of emotion, are perceived and understood with reference to emotion prototypes or scripts" (p. 1072). Based on a collection of accounts of emotional events, they are able to identify separate prototypical stories for fear, sadness, anger, joy, and love. As Spiral 2 development proceeds, further vignette development will be addressed in the light of these findings.

3. RESULTS

The method described in Subsection 2.2 has produced four inter-related results to date, each of which is discussed in a separate subsection below. The core result is the development and definition of the PAC architecture, as discussed in subsection 3.1 below. The architecture breaks out the structure and flow of cognitive processing from both a purely purposive perspective and a personality-based perspective, and also defines how these two are integrated. Complementing this processing architecture is the development of representations for the kinds of knowledge that are specifically used in the personality-based aspects of the architecture. The PAC knowledge representation is discussed in Subsection 3.2. The definition of the processing architecture and knowledge representation enables development of the third major result, the software engine that embodied the PAC architecture and knowledge representation. The PAC execution engine, discussed in detail in Subsection 3.3, is used to run HBR models expressed in the PAC knowledge representation. The final result of the current effort is a set of development tools that structure and support the creation and execution of PAC-based HBRs, as discussed in Subsection 3.4.

3.1 PAC Architecture

Although there have been active lines of research into the architecture of cognition for almost as long as there has been computational technology, the current generation of cognitive architectures can be traced back to the work on the Human Operator Simulator or HOS system, developed by the US Navy in the 1960s and 1970s (see Reference 1 for more details on HOS and the various systems discussed below). The SOAR architecture was developed and began its evolution in the mid 1980s, following the integration of new theories of reasoning and theories of memory by Newell and colleagues. By the 1990s, several additional cognitive architectures had been developed, including ACT-R, EPIC and COGNET/iGEN®. Each of these placed differing emphasis on the level of granularity in their representation both of knowledge and processing mechanisms, as well as focusing on different stages of the novice-toexpert continuum of cognitive skill acquisition. In addition, they varied in the types of applications or purposes for which they had been built, some, like ACT-R, were built as a means to formalize and test psychological/cognitive theories. Others, like EPIC, were created to enable engineering analyses of human performance in the context of system design. And yet others, like COGNET/iGEN®, were focused on use in training and/or decision support environments. Over time, most of these architectures began to be used for multiple purposes.

As the 1998 National Academy of Science review (ibid) pointed out, these various systems did, in fact, share a common underlying conceptual organization that focused on systems of symbols maintained in and manipulated from memory, which can be traced back to the seminal work of Broadbent in the 1950's. They also all share a common focus on rational but situationally-constrained behavior that can be traced back to the development of game theory and micro-economics.

This common framework includes the following general features, as shown in Figure 3. The person/model:

- is constantly building and maintaining an internal symbolic representation of the external world that provides awareness of the external situation, as filtered by the sensory, perceptual, and interpretive processes (and knowledge);
- recognizes opportunities that the current situation (as internally represented) affords for accomplishing work goals and tasks;
- acquires or activates localized work goals and from opportunities afforded by the current situation;
- recognizes constraints and conflicts between and among the various activated goals/strategies, and prioritizes or otherwise deconflicts them;
- tailors remembered or learned strategies for accomplishing the prioritized work goals to the current situational context; and
- undertakes physical actions (including verbal actions) in the environment as needed to carry out the tailored strategies.

The resulting behaviors are typically consistent with theories of constrained rationality, and show how a fixed body of knowledge can generate a range of behaviors that accomplish abstractly defined work goals across a range of different situations.

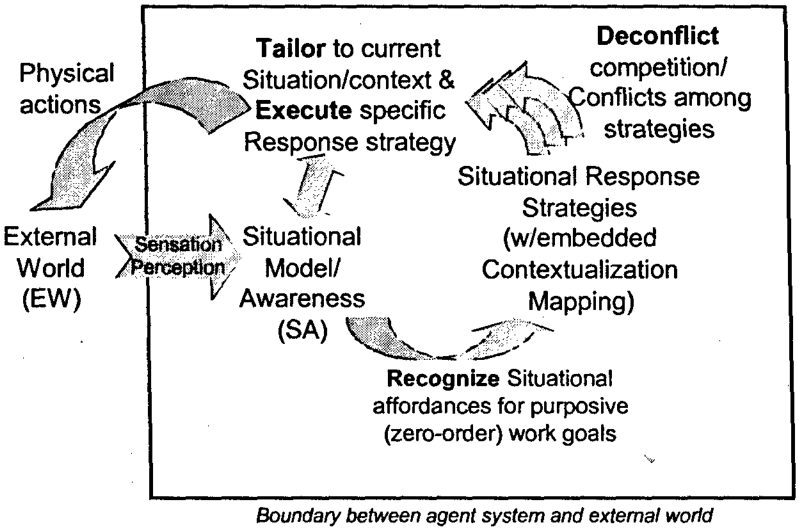

Physical actions
External World (EW)
Sensation
Perception
Tailor to current Situation/context & **Execute** specific Response strategy
Deconflict competition/ Conflicts among strategies
Situational Response Strategies (w/embedded Contextualization Mapping)
Situational Model/ Awareness (SA)
Recognize Situational affordances for purposive (zero-order) work goals
Boundary between agent system and external world

Figure 3. Structure of Situationally-constrained rational, Cognition

As noted in Section 1, these constrained rationality models do not deal with many psychological tendencies, commonly called personality traits, which differentiate individual performance. At the same time, however, this general architecture has been shown to be highly consistent with a large body of psychological theory, and to be predictive within the limits of its scope. A challenge, then, for the PAC research was to define a cognitive architecture that incorporated personality-based effects without attempting to vitiate or to deconstruct the validated prior work on constrained rationality.

At the conceptual level, this was done by defining a deeper level process which followed the same general processing structure as that used in the constrained rationality models. This two-level process is shown in Figure 4. The process originally shown in Figure 3 is retained in Figure 4 (the light blue portion), but here it is encompassed by a larger, personality-based process in light green.

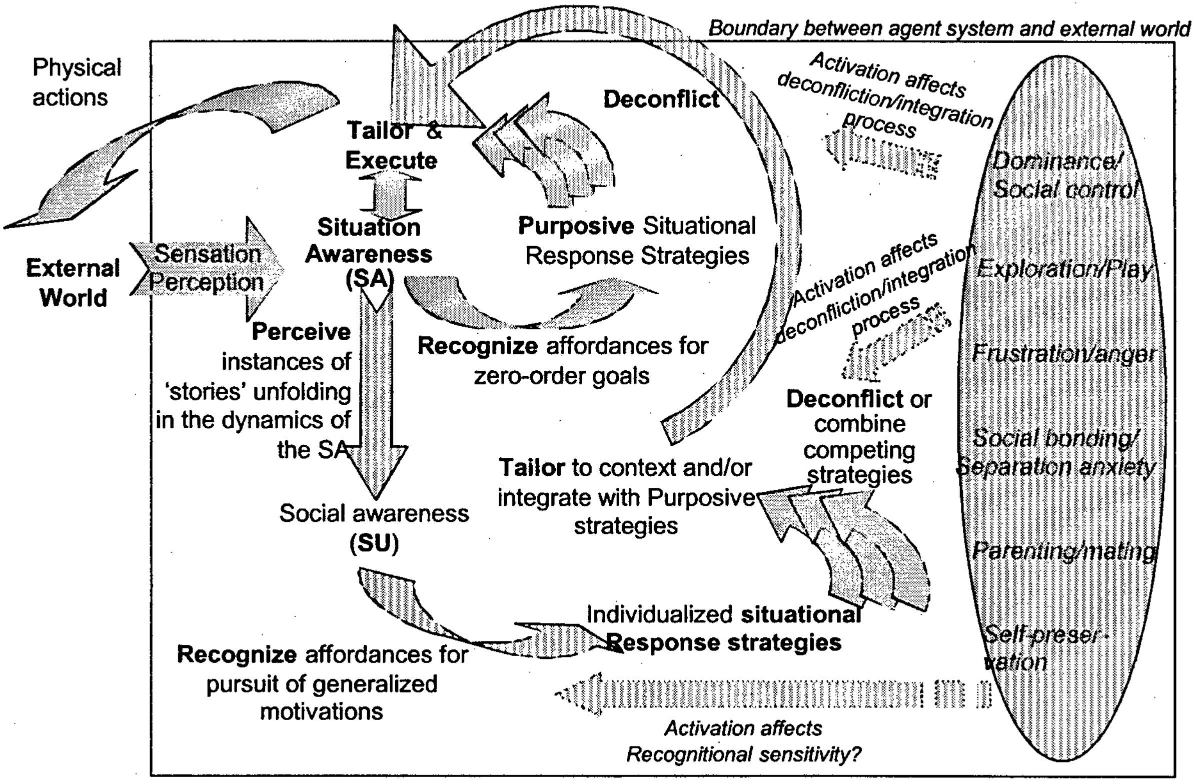

Boundary between agent system and external world
Physical actions
External World
Sensation Perception
Tailor & Execute
Situation Awareness (SA)
Deconflict
Purposive Situational Response Strategies
Activation affects deconfliction/integration process
Dominance/ Social control
Exploration/Play
Frustration/anger
Social bonding/ Separation anxiety
Parenting/mating
Self-preser-vation
Perceive instances of 'stories' unfolding in the dynamics of the SA
Recognize affordances for zero-order goals
Activation affects deconfliction/integration process
Deconflict or combine competing strategies
Tailor to context and/or integrate with Purposive strategies
Social awareness (SU)
Individualized situational Response strategies
Recognize affordances for pursuit of generalized motivations
Activation affects Recognitional sensitivity?

Figure 4. Structure of Personality-based Cognition in PAC

This process has a similar structure to the constrained rationality process. The person/model:

- is constantly building and maintaining a social awareness by perceiving instances of generally understood stories that are unfolding across the series of symbolic knowledge structures that make up the situation awareness;
- recognizes opportunities that the current social situation (as internally represented) affords for acting on the deeper level motivations (as in Figure 1 above) that make up the individual's personality;
- acquires or activates localized goals or strategies for action on those general motivations, given the opportunities afforded by the current situation;
- recognizes constraints and conflicts between and among the various activated individual/personality strategies, and prioritizes or otherwise deconflicts them; and
- tailors remembered or learned strategies for accomplishing the prioritized work goals to the current situational context.

To this point, the process is an almost direct match with the constrained rational cognition process in Figure 3. However, this deeper level process does not presume to have a separate or parallel 'action' component. Instead, it leaves the action aspect under the control of the constrained rationality process. It, thus, ends the processing cycle with a step of integrating the execution of the individualized strategies with the tailoring and execution step of the constrained rationality process. In general, this is envisioned as happening in one of two ways, either by:

- inserting additional actions into the set of actions constructed by the purely work-focused cognitive process; or
- further tailoring the work-related goals to meet the strategies activated by the personality-based deeper loop.

A second difference between the blue and the green processing cycles is the pervasive effect of the set of baseline personality traits possessed by the individual involved. Represented in the oval to the far right of Figure 4, these correspond to baseline activations of the various factors in the underlying psychological theory of personality (shown in Figure 1 above). It should be noted that these baseline activations affect several parts of the green (i.e., personality-based cycle), such as:

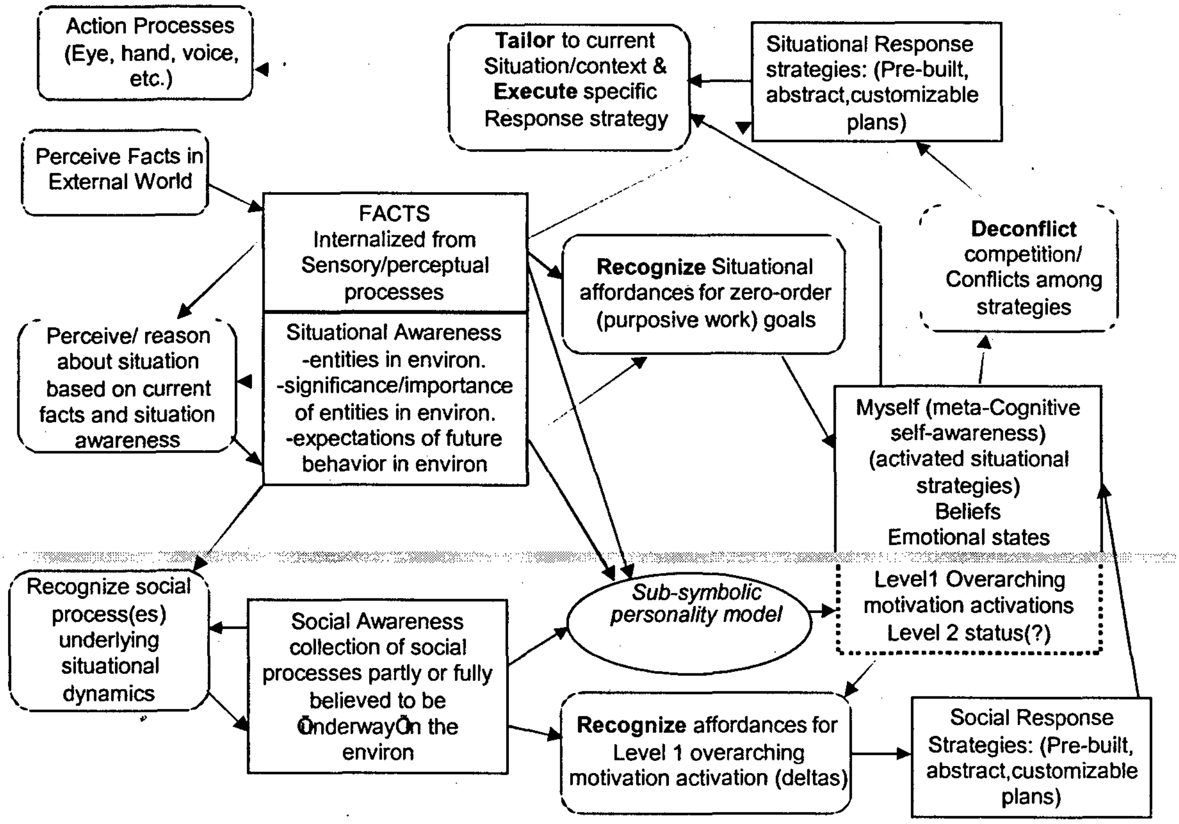
- the recognition of affordances to pursue specific motivations; e.g., a person with a low baseline for pursing dominance is less likely to recognize situations which afford an opportunity to increase social status or dominance; or
- the deconfliction of competing motivations and associated strategies.

The unfolding social situation and the person's response to it may also result in short term perturbations to activations of the various motivations and deeper-level controls (e.g., BIS/BAS/DvC) which, in turn, may temporarily change the behavior of the system. Over time, though, the activations of these personality factors would be expected to return to their baseline activation levels, because of the persistent nature of baseline personality traits.

3.1.1 Structural Organization for PAC

The general cognitive processing structure pictured in Figure 4 addresses several challenges to the development of PAC. It provides a structure that shows how personality-based sources to behavior can co-exist and integrate with purposive-work based sources. It shows how a common organization can be used for the two levels of processing. It integrates the cognitive theory of personality set forth in Section 2 with the framework used in conventional cognitive architectures. And, finally, it suggests a general but novel knowledge representation for the personality-based processes (i.e., the story-based structure.

The conceptual nature of Figure 4, however, makes it still too vague to serve as a general basis for implementation of the PAC software to remedy this, a more formal representation was developed, based on the flow in Figure 4. This representation is shown in Figure 5. It sought to decompose the processing structure into specific computational processes, shown as oblongs (rounded-corner figures), and data/knowledge stores, shown as rectangles. Each computational process was, furthermore, required to provide information, data, or knowledge to either some store or to the external world. Similarly, each data store could provide information only to the computational process (i.e., never directly to another data/knowledge store).



Action Processes (Eye, hand, voice, etc.)
Tailor to current Situation/context & Execute specific Response strategy
Situational Response strategies: (Pre-built, abstract,customizable plans)
Perceive Facts in External World
FACTS Internalized from Sensory/perceptual processes
Recognize Situational affordances for zero-order (purposive work) goals
Deconflict competition/ Conflicts among strategies
Perceive/ reason about situation based on current facts and situation awareness
Situational Awareness -entities in environ. -significance/importance of entities in environ. -expectations of future behavior in environ
Myself (meta-Cognitive self-awareness) (activated situational strategies) Beliefs Emotional states
Level1 Overarching motivation activations Level 2 status(?)
Recognize social process(es) underlying situational dynamics
Social Awareness collection of social processes partly or fully believed to be ÒnderwayÓn the environ
Sub-symbolic personality model
Recognize affordances for Level 1 overarching motivation activation (deltas)
Social Response Strategies: (Pre-built, abstract,customizable plans)

Figure 5. Conceptual Organization of PAC

The layout of Figure 5 is deliberately organized to place all processes either above or below a line (shown in yellow), which divides the overall architecture. The components above the yellow line correspond to the processes involved in conventional constrained rationality cognitive architectures, while the components below the yellow line make up the processes that are unique to the personality-based processes. Thus, the PAC software could be implemented as an entirely separate personality-based layer in a two layer cognitive system, working in tandem with a constrained rationality layer (which could, in principle, be any existing system). Importantly, the connections between processes above and below the line are limited to two specific data/knowledge stores:

1) the contents of the (current) situation awareness; and

2) the metacognitive store in which the awareness of the currently active strategies and corresponding goals is maintained.

This simple interconnection between the personality layer and the constrained rationality layer, further structures the relationship between the two layers, and suggests an integration strategy, as discussed below in Subsection 3.3

3.2 PAC Knowledge Representation

3.2.1 Generative Narrative Structures

During the initial design phase of the PAC, it became clear that if we wished to build agents with realistic human personalities, that it would not be sufficient to just design the motivational system of the agent. We also needed to capture how the agent would interpret the events and actions of other agents. This interpretation process was clearly a central part of the basis of human personality. For example, one thing that distinguishes an Anxious or Fearful individual from others is that he is much more likely to perceive situations as threatening. Thus, at the same time we started to design the motivational and behavioral systems of the PAC, we also started to think about how to capture this interpretation or attribution process.

For a number of reasons we settled on a narrative - based system, influenced by Lehnert's (References 64, 65) work on plot units. Note that the PAC system is influenced by, but not really based on Lehnert's system.

There are several reasons why we decided on a narrative – based system. First, Read and Miller have long argued that narrative or story structures play a central representational role in a variety of social concepts (see also References 66, 67), but especially traits. For example, we (Reference 68) have argued that fundamental to the representation of most traits is a simple narrative or story. Take the trait and the associated behavior helpful. For an action to be helpful, the agent must see that someone else is in need, they must intend to do something about that need, they must take action to satisfy that need, and the need must be satisfied. This forms a basic narrative with a *condition* that instigates a *goal*, which leads to the formation and enactment of a *plan*, which leads to the *outcome* of the *plan*, either success or failure. This narrative structure of traits, thus, made a narrative representation a natural fit in our attempts to capture the interpretation of social interaction and events.

Second, and related to the preceding point, several authors (e.g., References 68, 69, 70) have argued that a story or narrative structure is central to the ways in which people represent their understanding of social interaction. This makes narrative a natural way to represent our agent's understanding.

A third reason why we chose a narrative structure is that it provided a structured, constrained way to represent human social knowledge, in a way that was easy for the user of the PAC program to think about. The intent of the PAC architecture is to develop a relatively easy to use system for programming realistic human agents in a constrained set of contexts, such as in a game or game-based learning system. We were not trying to develop an abstract, completely general system for representing knowledge and reasoning about that knowledge. Instead, we wanted a way to easily represent important knowledge about social events and actions in a limited context.

A final reason became obvious only in hindsight, as we started to think about how to extend the PAC to emotions. We realized that a narrative representation might also provide a natural way to represent emotional appraisals and emotional behaviors. A number of emotion appraisal theorists have talked about the idea of emotion scripts or scenarios, which capture the sequence of events that happen in an emotional experience, from the initial interpretation or appraisal of goal-relevant events to the resulting emotional and behavioral response. For example, Lazarus (Reference 71) talks about the idea of a "core relational theme" which is akin to a simple story. And Shaver, Schwartz, O'Connor, and Kirson (Reference 63) identified the emotion "scripts" for a set of basic emotions, which start with the conditions initiating the emotional experience and terminating with the behavioral responses to the emotion. This conception is consistent with most appraisal theorists' explicit treatment of emotions as providing a readiness to behave in particular ways in response to a set of environmental conditions.

3.2.2 Plot Unit Structure in PAC

Narrative processing is a way of making sense of what is going on in the world – understanding who (including the self) did what to and with whom, when, where, why, in what fashion, and leading to what effect in the world. To make sense of the world in this way, individuals draw on pre-stored knowledge representations, especially those involving stereotyped sequences of actions and events. But these pre-existing representations are not, by themselves, narratives. Nor are the events that transpire in the world narratives. Narratives are produced during the interaction between certain kinds of events (in the world) and certain kinds of pre-existing representations (in the mind). Individuals relate internal representations and external events in this way when they code what happened/is happening in the world as a sequence(s) of deliberate, goal-directed actions, which have caused a change from an earlier state to a later state.

The definition of narrative used in PAC specifies 'events' and 'states' as core elements. States' and 'actions' can be viewed as two special types of events:

A. Stative events, or STATES, are internally uniform propositions about the way things/people are. STATES can be either *permanent* or *temporary.*

B. Nonstative events, or EVENTS, on the other hand, are heterogeneous details about the subprocesses that produce the termination of one state, the onset of the next, or the coexistence of both. EVENTS can be:

- *bounded*, involving changes from a definite initial state to a definite terminal state; or
- *unbounded,* coded as ongoing processes without clear temporal boundaries; or

- *actions*, consisting of events deliberately initiated by an active participant or agent (can be either bounded or unbounded).

All of these elements contribute to narrative processing/thought, but this does NOT mean that putting STATES and EVENTS together in just any way will form a narrative. These elements must be connected or sequenced in a very particular way (by the perceiver/narrator) to construct a narrative. The key to this special way of interrelating STATES and EVENTS is the 'action structure'. Action is at the heart of narrative, because it provides a causal explanation or meaning for why a state change has occurred. Action involves an actor/agent intentionally bringing about or preventing a change in the world. In turn, change occurs when some state of affairs either ceases to be or comes into being.

In PAC, the 'Action Structure' is the smallest building block of narrative. It forms a micro-level representation of the causal-chronological sequence, and specifies such elements as the actor/agent, the act-type, the modality of action, the setting, and the possible behavioral outcomes. Perhaps most importantly, it also specifies the opportunities that the action affords for activation or application of the overarching motivations that comprise the top level of the PAC personality model (Figure 1 above).

Action-structures are composed into higher -level structures called 'Plot-Units.' A 'Plot Unit' interrelates two or more Action Structures, and forms a macro-level representation of the causal-chronological sequence. The contingent structure of the Plot-Unit specifies the way that events typically play out, with a limited range of variation and consequences. Plot Units, in PAC, are represented from an ego-centric view of the agent/character. Thus, for even a simple interaction to occur, there must be some general mapping of the plot units that are understood by the two actors. Figure 6 shows this principle for two interacting HBRs created with PAC (see Section 4 for specifics on these HBRs). The interaction involves a coalition solider on patrol in a Kurdish village, hoping to gain information on a person-of-interest from a shopkeeper in a village café. The two plot units (one each from the shopkeeper and soldier viewpoint) are pictured in Figure 6. Although fairly linear, there action builds to the point of the major contingency, at which the shopkeeper will either provide or deny information. The way in which each component action structure is carried out determines the internal state of each agent.

The shopkeeper, in particular, may end up being motivated to be helpful and to trust the soldier (in which case helpful information may be supplied). However, other motivations may become involved, depending on the personality of two interacting people (or in this case, HBRs).

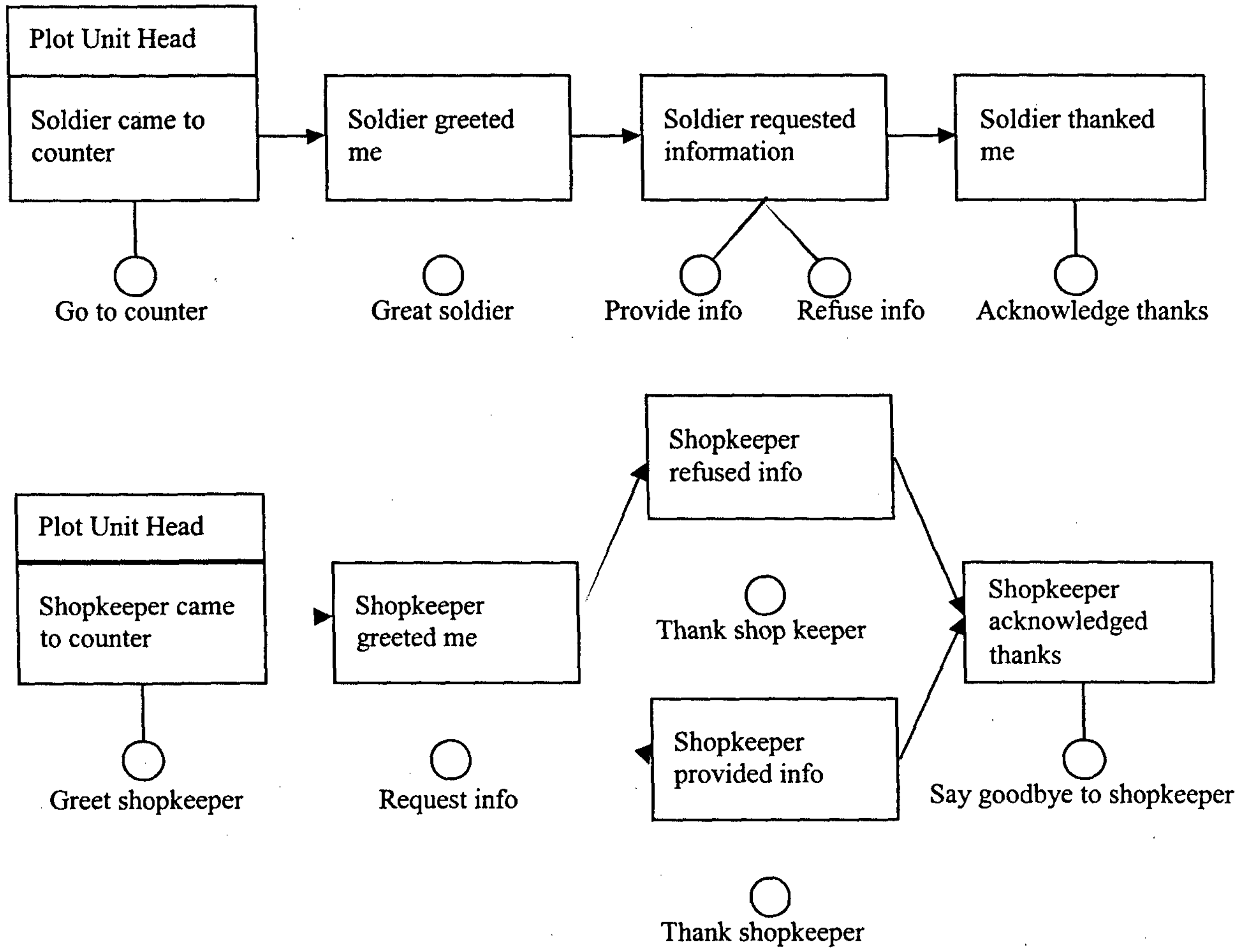


Figure 6. Matching Ego-Centric Plot Units

Thus, at any point in time, a portion of the plot unit may have been played out and recognized by each party, albeit with different interpretations, while the remaining portions may still be unplayed and contingent. This gives rise to a space of interactions and outcomes that occur at each point in the story, as pictured in Figure 7.

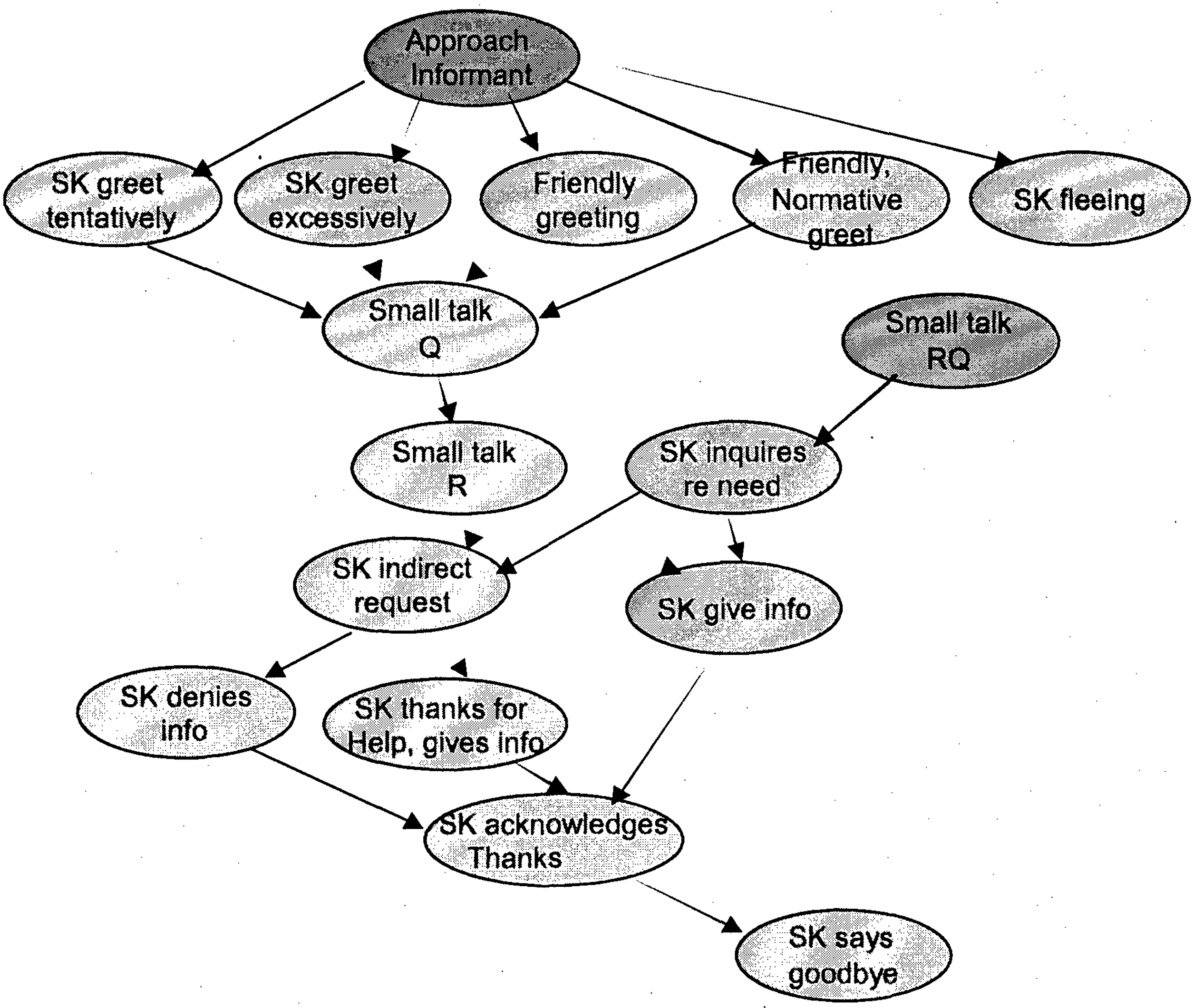


Figure 7. Instantiated Plot Unit from Soldier Perspective

The details of the formalism used to represent the plot unit and action structure are given later in the following subsection.

3.3 PAC Software Architecture and Implementation

3.3.1 PAC Architecture Design Rationale

Several factors drove the design of the PAC architecture. First, we wanted a system that can be used to generate a synthetic player exhibiting a motivation oriented behavior. The behavior was also supposed to reflect cultural and individual specificities. Finally, we wanted to be able to exhibit these properties at a minimum cost for the implementation of the architecture, and more important, for the implementation of a PAC model.

3.3.1.1 Finite State Machine Approach

Our first observation was that many games and interactive training applications rely on a heavily canned approach that can be described in many cases as a finite state machine. Each possible state is represented as pre-recorded animated sequence; for example, a scene of a soldier asking for directions, or a shopkeeper refusing to answer a question. When using a large enough number of pre-recorded scenes, it becomes possible to create a very large number of alternative scenarios. In many cases, the space of possible sequence of scenes is large enough to provide adequate training, providing each time a different scenario.

The only decision that an agent must perform in such an environment is to select what should be the next state. We are using a turn taking solution where each agent is presented in turn with a new state and has the opportunity to analyze it and select a new state accordingly, which will be presented to the next agent. A PAC agent will select the next state according to its interpretation of the presented state and the level of activations of its own motivational goals. A Human (agent) interacting with a PAC agent will look at the animation corresponding to the current state, and select one of the possible actions presented to him, possibly in the form of a multiple choice question. Each of the possible choices will correspond to another state that will be then submitted to the PAC agent. This type of setting is particularly appropriate to simulate social interactions, but would not be suitable, for example, for a first person shooting game.

3.3.1.2 Interpreting the Current Situation

To put things in a simple perspective, half of the work of an agent is to understand the current situation, the other half being to decide how to respond to it. Such understanding is a very complex process, where the new fact presented is only one part of the information available. First, the context in which the new facts are occurring plays an important part. Second, it also involves prior knowledge, which is often specific to both a cultural experience, and an individual experience. For example, the interpretation that the behavior of a person is rude can be shared by a group of people sharing the same culture, but not by another group from a different culture.

The PAC model represents cultural experiences as stories. It has been claimed that one can understand a situation by referring to past experience. Past experiences, represented as memorized stories, are associated with, what we call, motive implications. A motive implication indicates how a current given situation provides an opportunity to fulfill a particular goal. By relating a memorized story to a current situation, it becomes possible to understand its implication toward satisfying one or more motivational goal. Some stories may be shared by a large group of people, in which case they can be associated to cultural traits. Other stories may be only common to a small group or even a single individual, which is then more indicative of a personality trait.

3.3.1.3 Motivational Goals

Past experience, or story, is only one aspect that drives the behavior of an individual. Another aspect is his or her current motivations, or motive. Every individual, at any particular time, has different motives. Some motives may be part of their own personality, while others may be appearing as some opportunities arise in a given situation. A PAC model defines a set of motives that will be used during the execution. Each motive is quantified by a numerical value. Every individual represented in a PAC is associated with a predefined motive value, one value for each of the motives defined in the model. This set of values represents the personality signature of the individual.

As we have seen, each story, or more precisely each story element, can be associated with one or more motive implications. These motive implications are combined with the personality signature to determine the current level of activation of each motive. The motive interpreter, in charge of this operation in the PAC architecture, also applies a decaying effect to take into account previous implications.

3.3.1.4 Behavior Selection

In our approach of using finite state machine, the behavior, or action, of an agent is represented by the choice of a new state among all the possible states defined in the simulated environment. The selection of an action is determined by two factors. First, the current context, as provided by the story associated with the current situation, determines a set of possible behavior. Second, the level of activation of each motive is used to select the actual behavior among them.

A story is composed of elementary story units that are being matched against the current state. Each story unit is associated to a set of possible behavior options. Each of these behavior options is tagged by a motive value for each of the motives described in the model, and the behavior that most closely matches the currently highest motive is selected.

3.3.2 PAC Architecture

The following terms are used in this architecture description:

Situational State: one of the possible states describing the simulated environment,

Plot Unit: Low level story. The current architecture only uses plot units for stories,

Action Structure: one element of a plot unit,

Narrative: instantiation of a plot unit,

Action Instance: instantiation of an action structure

Motive: a motivational goal,

Motive Activation: numerical value indicating the level of activation of a motive,

Motive Implication: numerical value indicating relevance in relation to a motive,

Behavioral Option: a potential action referring to a situational state and tagged by a set of motive implications.

3.3.2.1 Execution Cycle

Figure 8 depicts the Spiral 1 PAC architecture.

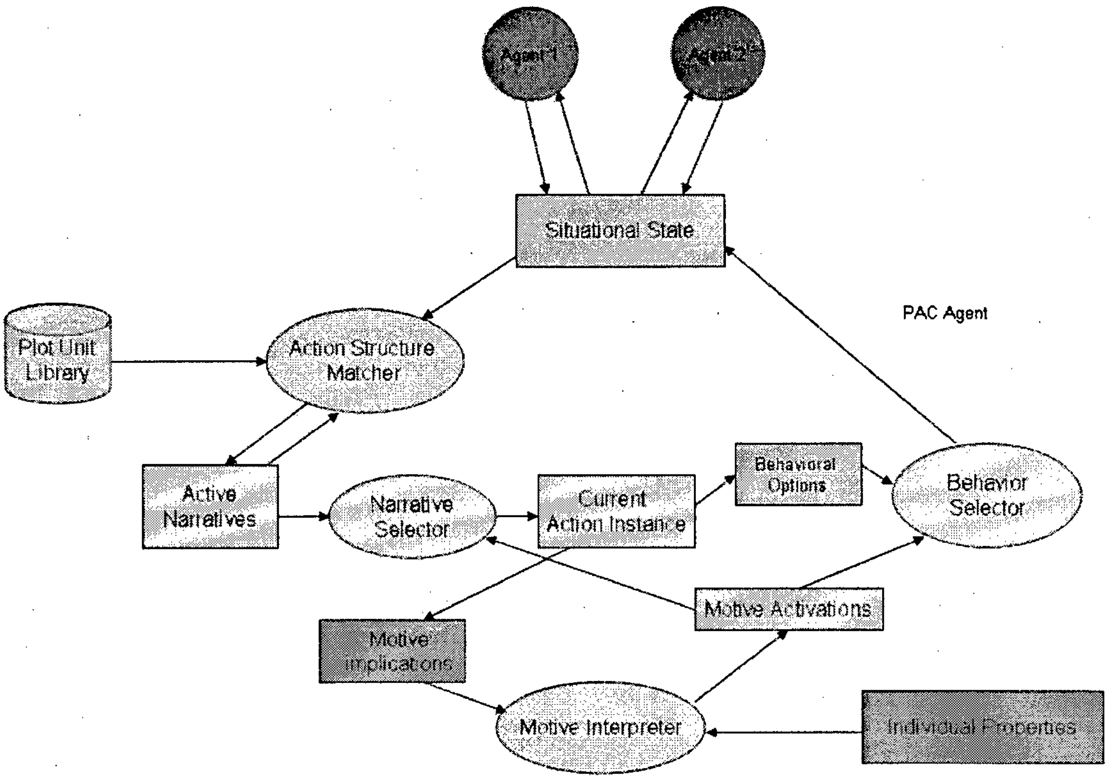


Figure 8. Spiral 1 PAC Architecture

An execution cycle is initiated when the PAC agent is presented with the occurrence of a new situational state. At the first cycle, the Action Structure Matcher attempts to relate the situational state to one or more plot units from the plot unit library. Each plot unit is represented as a succession of action structures. To account for a variety of situations, however, a plot unit is not simply a linear sequence of an action structure, but rather a graph. Each node in the tree branch indicates a choice of different actions and different expectations in return to an action. Each action structure contains a pattern that can be applied against the properties describing a situational state. To match against a plot unit, the properties of a state must match the pattern of its first action structure. This action structure refers to a set of possible subsequent action structures, and a set of possible behavior options. Figure 9 shows an example of a plot unit as a graph of action structures. In this example, two different entry points are possible for this plot unit.

If the pattern of a first action structure matches the current situational state, a new Plot Unit instance, called narrative, is created and stored in the active narrative storage.

A narrative contains a sequence of action instances, which are instantiation of the actions structures that could be matched to situational states so far. The current action instance of a narrative is the one that was last instantiated. At the creation of the narrative, during this first cycle, the narrative contains only a single action instance, resulting from the instantiation of the first action structure of the plot unit.

As more than one narrative might be created at the same time, the narrative selector selects the one that is the most relevant to the current motive activations. Each action structure can be tagged with motive implications, indicating how they might support one or more particular motives. These motive implications are used both for the selection of the current narrative, and for the calculation of the motive activations by the motive interpreter.

Once a narrative has been selected, its current action instance becomes the current action instance depicted in the blue box of the architecture schema. The behavior options associated to the corresponding action structure become the set of possible behavior options. Finally, the behavior selector selects the behavior that is the most relevant to the current motive activations. The situational state referenced by the selected behavior option becomes the current situational state of the other agent.

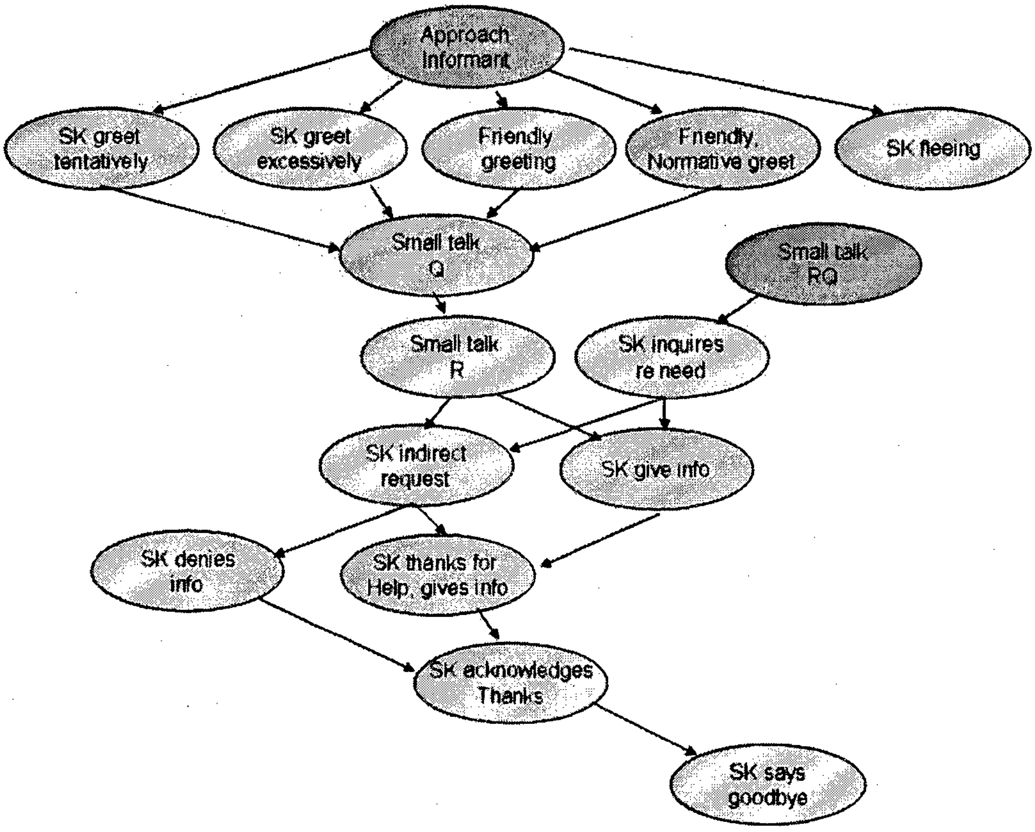


Figure 9. Example of plot unit as graph of action structure

For the following execution cycle, the agent will try to follow the plot unit already initiated. For this purpose, it will try to match again the new situational state, all the action structures that followed the current action structures in its plot unit tree. If successful, a new action instance will be added to the current narrative and one of the corresponding behavioral option will be chosen and yield a new situational state. If none of the following action structures can be matched, then the narrative will be removed from the set of current narratives.

3.3.2.2 Plot Units and Narratives

In this section, we review the different aspects of plot units and narratives in more details.

- A plot unit is composed of a collection of action structures organized as a graph, as depicted in the example in Figure 9. One or more actions structures are entry points for the plot unit. Only the entry points action structure are attempted to be matched against the current situational state, before the plot unit becomes instantiated as a narrative.

- A situational state is represented as collection of properties, where each property has a name and a value (numerical or symbolic). It also has an id, and a comment for debugging and tracing purpose.
- An Action structure is composed of the following elements:
    - a name, for debugging and execution tracing purpose,
    - a list of property patterns,
    - a list of possible behavioral options,
    - a set of motive implications,
    - a list of possible subsequent action structures,
    - a flag indicating if the action structure is an entry point for the plot unit.
- A property pattern is composed of a name and a predicate. The name is used to identify against which property of the situational state this property pattern is intended to be matched. The predicate is evaluated on the value of the property of the situational state. The evaluation of the predicates returns true or false.
- A behavioral option is composed of the following elements:
    - a name, for debugging and execution tracing purpose,
    - a reference to a situational state (its id),
    - a set of motive implications.
- A motive implication contains the name of a motive and numerical value between 0 and 1.
- A Narrative is an instantiation of a plot unit and is composed of a list of action instances. A single action instance corresponding to the entry point action structure of the plot unit is initially inserted. More action instances are subsequently added to the list as more action structures are being matched against situational states.
- An action instance contains a reference to an action structure.

When a new situation state is presented to the agent, the action structure matcher attempts to match against its properties:

- the property patterns of all the action structure entry points of all the plot units in the plot unit library,
- the property patterns of all the subsequent action structures of the action structures associated to the current actions instances of the current narratives.

When an entry point action structure is matched against the situational state, a new narrative is created, and an action instance of the action structure is inserted in its list of action instances. The narrative is then stored in the active narrative repository. Note that several narratives may be active at any time from different plot units.

A narrative is removed from the active narrative repository if none of the subsequent action structures of the action structures associated to its current actions instance can be matched against the situational state.

An action structure is considered matched against a situational state, if all its property patterns can be matched against the properties of the situational state. A property pattern is considered matched against situational state property if they have the same name, and if predicate evaluated with the value of the property returns true.

The narrative selector selects the narrative whose current actions instance is the best match for the motive activations. The best action instance is the one whose motive implications maximize a scalar product between the motive implications and the motive activations. The scalar product calculated using only the 2 highest motive values. The formula to calculate the scalar product is as follows:

$$R = \sum_i a_i b_i$$

Where R is the result of the scalar product,

i is an index iterator on all the types of motive,

a is a motive implication of an action structure,

b is a motive activation level as provided by the motive interpreter.

Motivational goals, or motives, are defined by the underlying psychological theory and the specific model to be executed. As an example of possible motives, here is a list of motives used in one of the example models discussed in 4.1 below:

- Status
- Control
- Explore
- Social Stimulation
- Gain
- Knowledge
- Caring
- Protect
- Dependable
- Support
- Avoid Separation
- Avoid Disrespect
- Avoid Control
- Avoid Harm

It was more flexible to let the PAC modeler specify his or her own motives in this first version of the PAC system. This list may be used as default motives in future versions.

Motives are separated in two different categories: related to avoidance goals or not related. In the example above, the last four motives are avoidance goals. These motives use different parameters for the calculation of their level of activation.

The motive interpreter calculates the level of activation for all the motives. We refer to these values as motive activations. Figure 10 shows the organization of the motive interpreter.

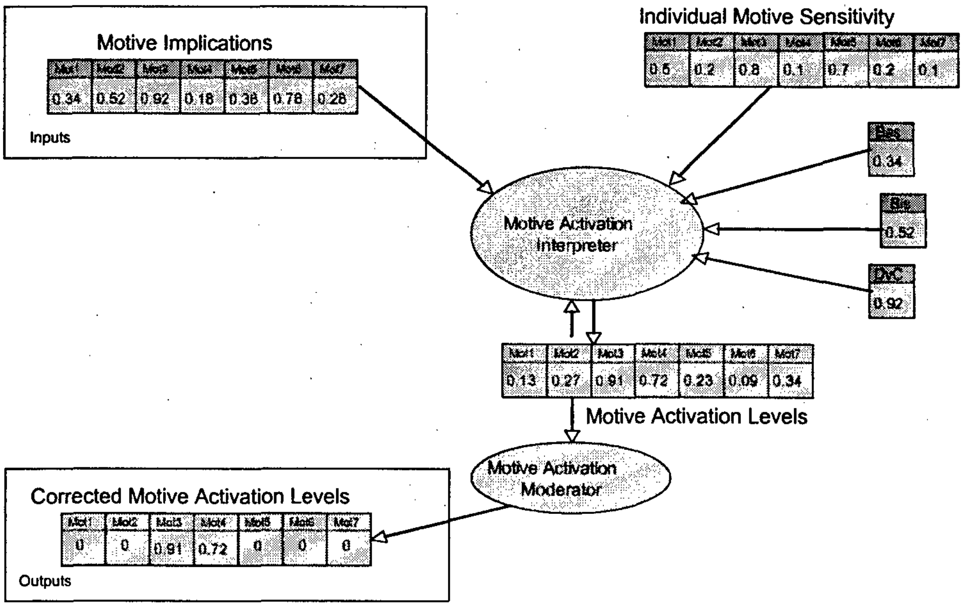

Motive Implications
Mot1 Mot2 Mot3 Mot4 Mot5 Mot6 Mot7
0.34 0.52 0.92 0.18 0.38 0.78 0.28
Inputs
Individual Motive Sensitivity
Mot1 Mot2 Mot3 Mot4 Mot5 Mot6 Mot7
0.5 0.2 0.8 0.1 0.7 0.2 0.1
Bas
0.34
Bis
0.52
DvC
0.92
Motive Activation Interpreter
Mot1 Mot2 Mot3 Mot4 Mot5 Mot6 Mot7
0.13 0.27 0.91 0.72 0.23 0.09 0.34
Motive Activation Levels
Motive Activation Moderator
Corrected Motive Activation Levels
Mot1 Mot2 Mot3 Mot4 Mot5 Mot6 Mot7
0 0 0.91 0.72 0 0 0
Outputs

Figure 10. Motive Interpreter

To calculate the motive activations, the motive interpreter receives three types of data:

- a set of motive implications from the current action structure,
- a set of predefined individual motive sensitivity,
- three individual specific parameters: Bis, Bas and DvC.

The motive implications have been presented in the previous section. The individual motive sensitivity are defined for the particular individual being simulated and can be changed to modify his or her personality. The demonstration of this Spiral 1 architecture consisted in showing how the modification of individual sensitivity was affecting the selection of behavior of the agent.

The other three individual specific parameters are described as follows:

- BAS: A general neurobiological system that affects the gain or sensitivity to stimulation of the approach related goals that fall into that system. Generally believed to be implemented in terms of widespread dopamine transmission to the left prefrontal cortex.
- BIS: a general neurobiological system that affects the gain or sensitivity to stimulation of any avoid related or punished related goals. Not clear how it is

neurobiologically implemented, although it is believed to reside in the right prefrontal cortex.

- DvC: A general inhibitory field that serves to dampen down or inhibit the activation of all goals. One effect of this is that higher Constraint leads to stronger goal focus and less disorganized behavior.

For each motive, the motive interpreter calculates its level of activation by using one of two formulas. The first formula is used when a motive implication for this motive is provided:

$$(1) \quad R = 1 - \frac{1}{1 + \gamma[I + S - DvC]_+}$$

Where:

$[x]_+ = x$ if $x > 0$ and $[x]_+ = 0$ if $x <= 0$;

R is the resulting level of activation of the motive,

I is the motive implication as provided by the current action structure,

S is the individual sensitivity for this motive,

γ is either the BIS or BAS depending on the type of motive. If the motive is of type avoidance, then γ = BIS; for all other motives γ = BAS.

The other formula is used when no motive implication is available for a particular motive. It implements a decay mechanism that progressively returns the activation level to the individual sensitivity level:

(2) $R_n = k(S - R_{n-1}) + R_{n-1}$

Where:

R_n is the resulting level of activation of the motive for the n cycle and, R_{n-1} the activation level at the previous cycle.

k is the decaying factor,

S is the individual sensitivity for this motive.

Figures 11, 12, 13 and 14 provide an indication on how R varies as a function of I+S for various values of DvC and γ. In these figures, y = R, teta = DvC and Vm = I+S

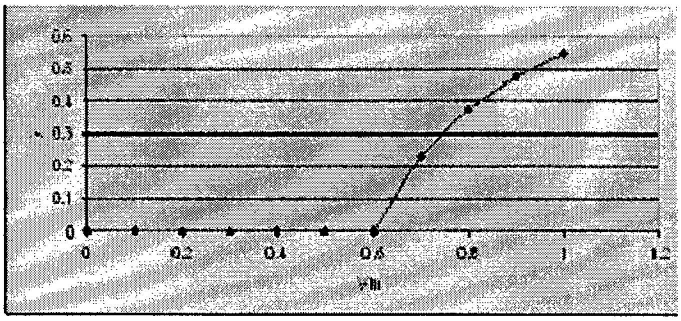


Figure 11: equation (1) for DvC = 0.6 and γ = 3

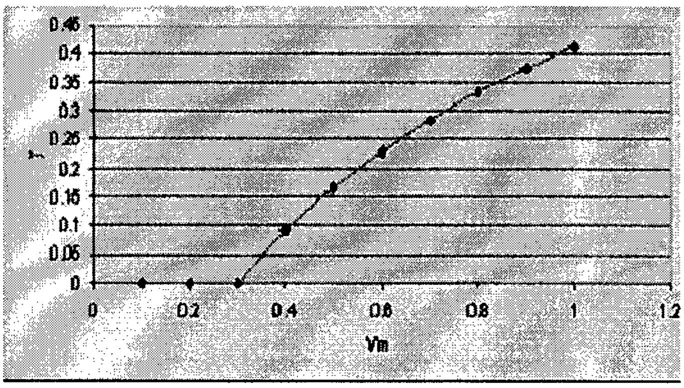


Figure 12: equation (1) for DvC = 0.3 and γ = 1

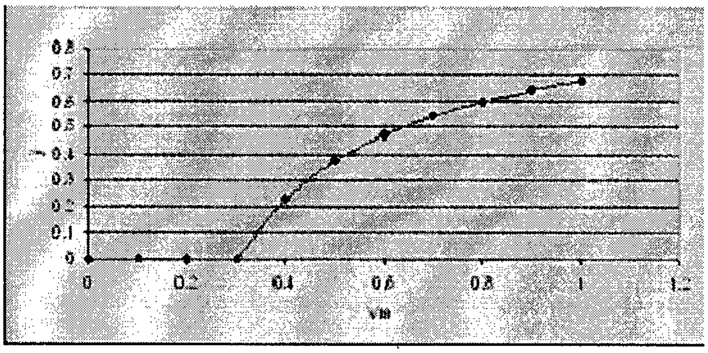


Figure 13: equation (1) for DvC = 0.3 and γ = 3

Figure 14 provides an example for the decay equation (2) where k= 0.5 and S = 0.4:

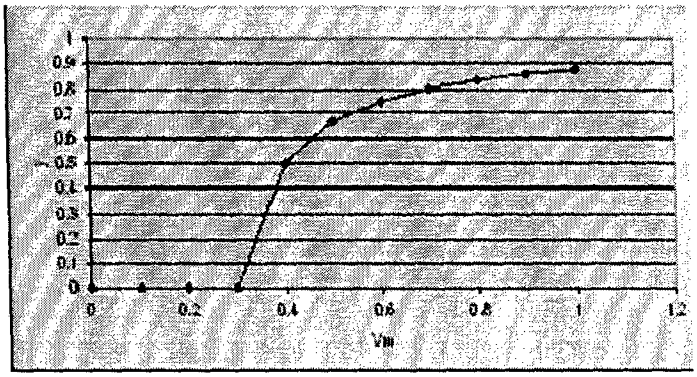


Figure 14: Decay function for k= 0.33 and γ = 10

Before submitting the result for the level activation, a moderating effect is applied. To take into account that at most only 2 motives can really be attended to at any time, only the two motives with the highest values are retained. All the other motives are given a value of 0. Note that the non-moderated value is used for the calculation of the decay. The moderated value is used for the selection of the current narrative, and the selection of the behavior option.

Behavioral options are directly associated to some action structures. When an action structure becomes the current one, the behavioral options associated to it are stored in the behavioral option repository.

A behavioral option contains the following information:

- a name for debugging and execution tracing purpose,
- a reference to a situational state
- a set of motive implication.

If the current action structure has no behavioral option, the agent does not change the current situational state. If the current action structure has a single behavioral option, the situational state it refers to becomes the current situational state for other agents to act upon. If the current action structure is associated to more than one behavioral option, the behavior selector selects the one that is the closest match to the current motive activations. The situational state becomes then the one referred by the selected behavioral option.

Like the narrative selector, the behavior selector selects the behavioral option that maximizes a scalar product between its motive implications and the motive activations.

The scalar product only calculated using only the 2 highest motive values, as all other motives are assigned an activation level of 0. The formula to calculate the scalar product is as follow:

(3) $R = \sum_i a_i b_i$

Where:

R is the result of the scalar product,

i is an index iterator on all the types of motive,

a is a motive implication of a behavioral option,

b is a motive activation level as provided by the motive interpreter.

3.3.3 PAC Software Implementation

The PAC software was implemented as an integrated development environment or (IDE) which include the PAC model-execution software, along with a model editing component, a model execution interface, and an environment framework. Thus, PAC was implemented in a way that provided the PAC model-builder with a set of tools for performing various development tasks on PAC models, ranging from model authoring to model execution. The relationship between these various tools and the PAC execution software is shown in Figure 15, below.

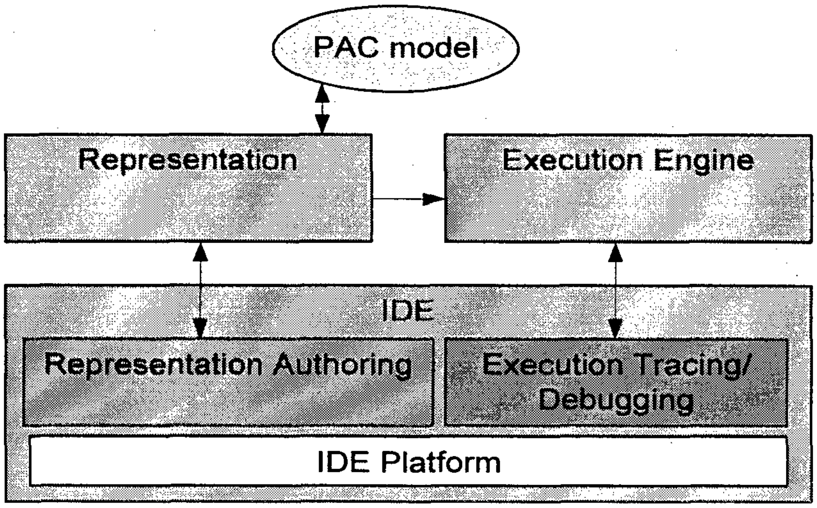


Figure 15. PAC Software components

The implementation closely followed the spiral development model described in 2.2.2 above. During the reoccurring software design cycles, the following considerations have been given great importance:

- *Usability.* An important aspect of every development environment, usability is often deferred for the last stages of product development. The usability of a GUI is tightly related to early design decisions, making such interventions difficult and expensive. In the case of PAC, experiments with the development environment have extensively taken place since Spiral 1, thus, adding to the importance of the IDE's usability. The PAC development focus has been on the creation of simple concepts; uncluttered user interfaces; well integrated subcomponents and taking advantage of CHI Systems' experience with development of work-centered systems.
- *Graphical representations.* A significant part of the PAC development tools is the use of graphical representational components. The architecture has been designed as a modeling framework with emphases on authoring and execution. Therefore, graphical metaphors play an important role towards the goal of identifying better representation techniques for authoring and tracing the execution. The Spiral 1 software includes graphical action structure tracing, however, the bulk of the authoring is done through conventional interfaces. Spiral 2 and 3 are gradually introducing alternative graphical views.
- *Refactoring.* Modern development environments employ refactoring as a tool for modifying the representational model while always remaining in an executable and consistent state. Environments without refactoring support are prone to introducing more mistakes and unnecessary maintenance code ("plumbing").
- *Flexibility.* With frequent changes to the design requirements and experimentation with various modeling structures and representations, the PAC development environment should be flexible enough to handle various changes without a significant development cost. Well known programming techniques such as modularity, various levels of abstraction, use of design

patterns, externalization of specialization settings are being used to provide code flexibility.

- *Code reuse, open standards, low-maintenance.* The IDE platform (Figure16) is the least architecture-dependent component. It provides project management functionality, project navigation and windowing. The IDE platform serves as host for model specific rendering modules through a set of defined APIs. The IDE platform has been based on a set of open technologies such as FlexDock (https://flexdock.dev.java.net/), XStream (http://xstream.codehaus.org/) and the Apache project's XML tools (http://xml.apache.org)

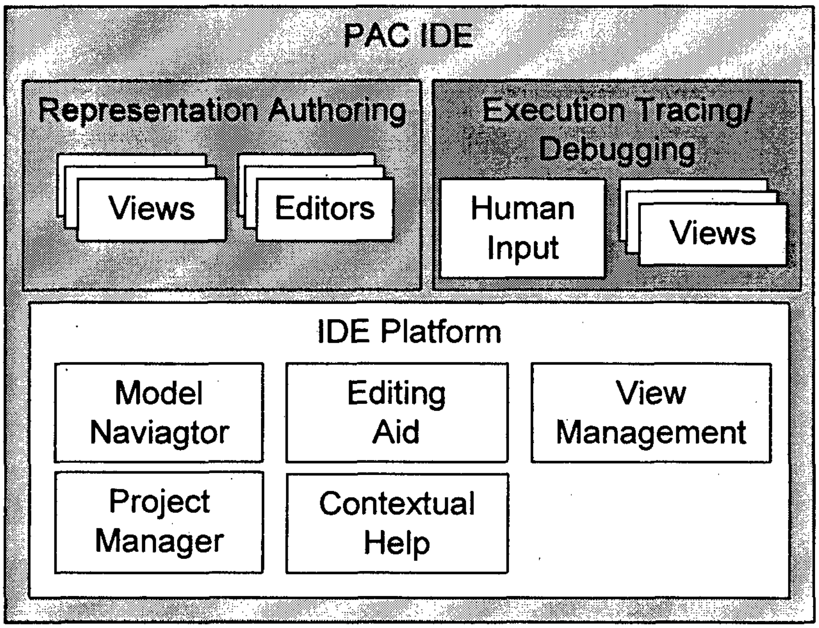


Figure 16. PAC IDE sub-components

The representation authoring component connects a set of editing components to the IDE platform. Most representation entities have at least one conventional user interface in addition to alternative graphical views. This component also defines contextual help and editing aid for the representation entities.

The execution tracing/debugging module provides a user interface allowing a human-controlled individual input in addition to tracing the execution of every individual. The graphical representation of action structures has been implemented using JUNG (http://jung.sourceforge.net/)/.

3.4 User Interface to the PAC IDE

The PAC software architecture is complemented by a set of development tools. These tools, existing in the form of an Integrated Development Environment (IDE), were designed to facilitate the production, execution, and analysis of experimental PAC models.

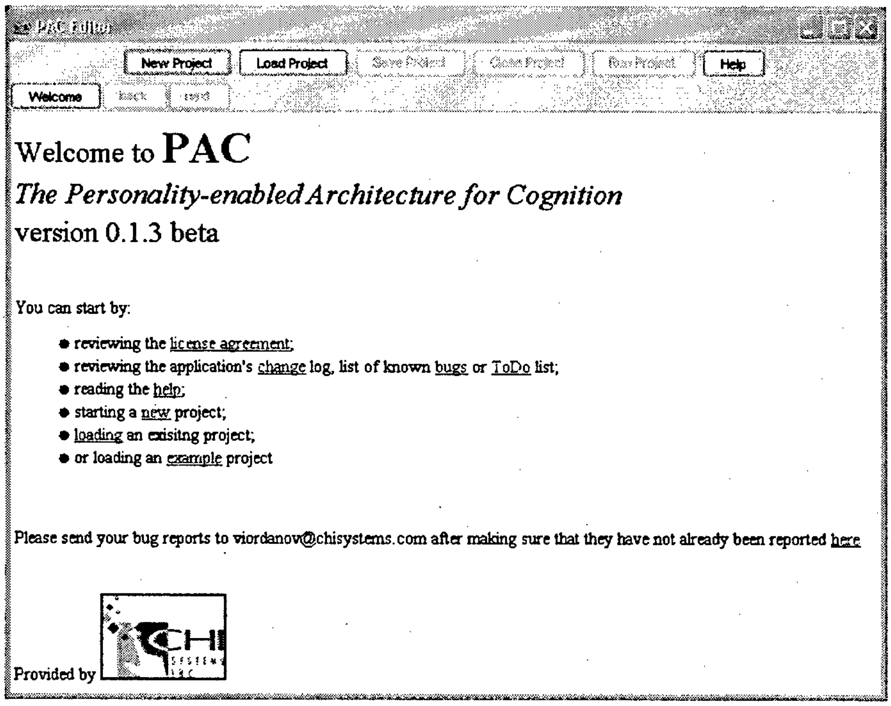
3.4.1 PAC IDE Overview

The primary objective for the Spiral 1 IDE was to provide a rudimentary yet effective means of testing and evaluating the various components of the PAC architecture during their inception and continuing evolution. A secondary objective was to enable us to assess the usability and feasibility involved in model creation. To these ends, a utilitarian approach was taken to the design of this version of the IDE. Efforts were concentrated on creating an unadorned, straightforward interface to allow us to investigate the elements that were the key focus of this spiral, as well as the psychological and personality theory at the core of our initiative. This approach yielded a simple but powerful testbed that allowed us to both create and iteratively improve the first spiral of PAC.

The PAC IDE (Figure 17) is comprised of two distinct components. Each component represents an independent graphical user interface (GUI) for interacting with different aspects of the PAC architecture. The first of these components, the editor, allows the user to design and maintain models, agents, and their constituent structures. Once a model has been accurately assembled, the second, runtime component, both executes the model and provides a practical toolset for analyzing the results of this execution. The details of each of these components and their use are outlined in subsections 3.4.1 and 3.4.2 respectively.

3.4.2 Editor Component

The editor component of the IDE serves as the primary means for authoring models in our architecture. Within the IDE, these models are maintained in the form of project files. When launched, it is the editor interface which initially appears, displaying the 'welcome' screen depicted in Figure 17. This screen is actually a hyperlink-enabled page allowing the user to, among other things, create new projects, load preexisting projects, browse the provided documentation, as well as examine the basic example model discussed later in section 4.1.1. Similarly, buttons are provided along the top of the interface to create, load, and save the user's projects.



Welcome to **PAC**

The Personality-enabledArchitecture for Cognition

version 0.1.3 beta

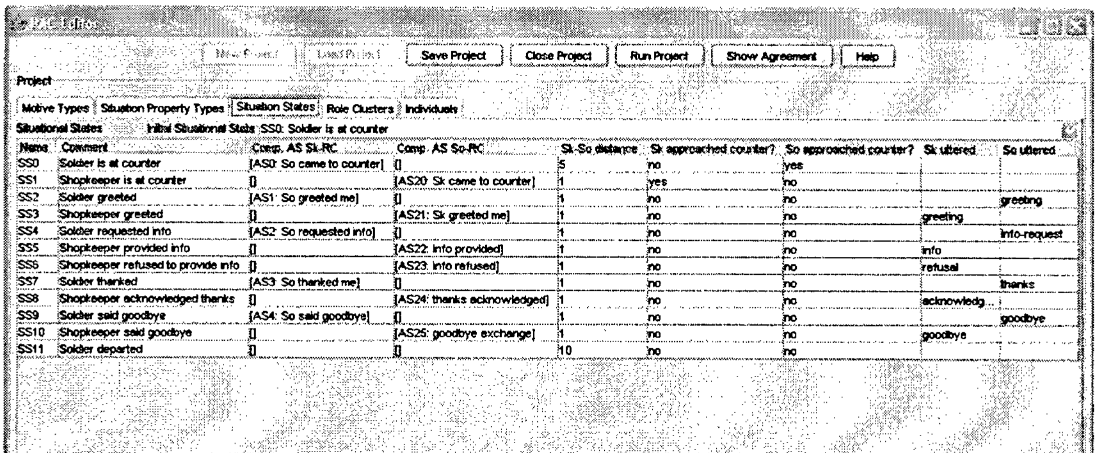
You can start by:

- reviewing the license agreement;
- reviewing the application's change log, list of known bugs or ToDo list;
- reading the help;
- starting a new project;
- loading an exisitng project;
- or loading an example project

Please send your bug reports to viordanov@chisystems.com after making sure that they have not already been reported here

Provided by

Figure 17. PAC Integrated Development Environment



Situational States Initial Situational State SS0: Soldier is at counter

Name	Comment	Comp. AS Sk-RC	Comp. AS So-RC	Sk-So distance	Sk approached counter?	So approached counter?	Sk uttered	So uttered
SS0	Soldier is at counter	[AS0: So came to counter]	[]	5	no	yes		
SS1	Shopkeeper is at counter	[]	[AS20: Sk came to counter]	1	yes	no		
SS2	Soldier greeted	[AS1: So greeted me]	[]	1	no	no		greeting
SS3	Shopkeeper greeted	[]	[AS21: Sk greeted me]	1	no	no	greeting	
SS4	Soldier requested info	[AS2: So requested info]	[]	1	no	no		info-request
SS5	Shopkeeper provided info	[]	[AS22: info provided]	1	no	no	info	
SS6	Shopkeeper refused to provide info	[]	[AS23: info refused]	1	no	no	refusal	
SS7	Soldier thanked	[AS3: So thanked me]	[]	1	no	no		thanks
SS8	Shopkeeper acknowledged thanks	[]	[AS24: thanks acknowledged]	1	no	no	acknowledg...	
SS9	Soldier said goodbye	[AS4: So said goodbye]	[]	1	no	no		goodbye
SS10	Shopkeeper said goodbye	[]	[AS25: goodbye exchange]	1	no	no	goodbye	
SS11	Soldier departed	[]	[]	10	no	no		

Figure 18. Situational State Editor

The intent in the current design of the editor component was to provide direct, transparent access to the entire suite of representations comprising the PAC architecture. Accordingly, the editor's interface centers around sets of related tables (Figure 18 shows an example of one such table). Each of these tables provides a means to specify and manipulate the necessary elements of a PAC model:

- *Motive Types*

  This table enables the user to create and name the types of motives that will be of concern in the model. Additionally, a column is provided to specify whether each of these motives belongs to the Behavioral Inhibition System or the Behavioral Approach System.

- *Situation Property Types*

  In this table, the user creates the list of the properties that will constitute and distinguish the different situational states of the model. Each property type defined by the user here will automatically appear as a column in the *Situation States* table.

- *Situation States*

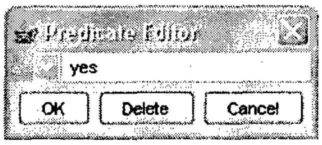
  Here the user is able to enumerate every possible state that can occur in the model. One column is provided for each of the property types defined in the previous table, allowing the user to specify an exact value for each and, thus, fully define a situational state. For purposes of convenience, additional columns are provided to automatically indicate to the user, once these property values have been specified, exactly which actions structures from the respective role clusters include patterns that will now match this situational state.

- *Role Clusters*

  This table is used to create and name new role clusters, each representing a new set of action structures and behavioral options which may be later assigned to particular individuals of the model. Hence, each single role cluster created in this table causes a new pair of action structure / behavioral option tables to be provided for the user.

- *Action Structures*

  Beneath each newly defined role cluster, a table is provided for the user to specify its required set of action structures. The columns in this table fall into three categories. First, a set of columns exist which are equivalent to the set of properties defined in the *Situation Property Types* table. These are presented to enable the user to specify a property pattern for those properties of the situational state deemed relevant to this action structure. When the user clicks underneath one of these columns, a dialog box interface is offered for specifying this pattern. This interface, called the *Predicate Editor*, is shown below in Figure 19. Again for convenience purposes, an additional column will show the user which situational states will ultimately match the action structure as each of these patterns, or predicates, are defined. Next, a second set of columns is provided equivalent to the set of defined motive types. Under each of these columns, the user may enter a numerical value representing the motive implication for the respective motive type. A column also exists to allow the user to specify the list of behavioral options associated with this action structure. The behavioral options the user specifies here are references to elements in the *Behavioral Options* table.



yes

Figure 19. Predicate Editor

Lastly, since it is the sequencing of action structures that characterizes particular plot units in our architecture, this table provides additional columns for specifying these interconnections. Under one column, the user indicates which other action structures, from this role cluster, may potentially follow (i.e. are 'subsequent') to the current action structure. Under another column, the user indicates whether the current action structure represents the head of a plot unit, and thus the beginning of a potential narrative.

- *Behavioral Options*

Here the user identifies the possible actions that may be carried out by the individual which will be assigned to this role cluster. Two important pieces of information must be specified for each behavioral option. First, the situational state this option, or action, will result in if chosen by the individual. Therefore, a column is available for selecting a situational state, from the *Situational States* table, for each option. Second, the motive implications for the behavioral option must be defined. As in the *Action Structures* table, one column exists for each motive type created, thus allowing the user to specify a value for the respective motive implication of each.

- *Individuals*

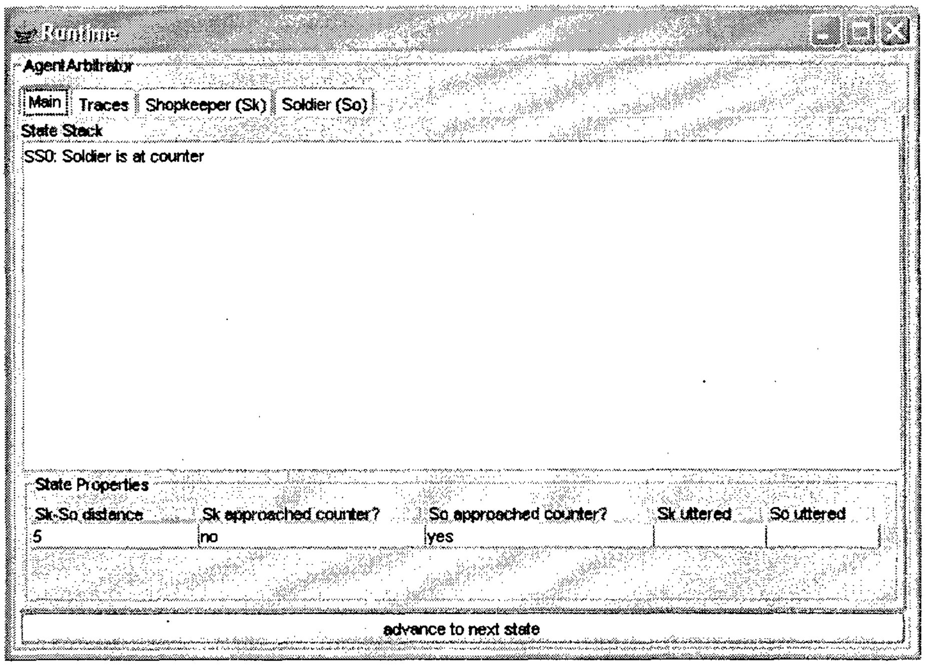
  Finally, in this table the user ties together the many pieces of information detailed in all of the preceding tables. For each individual created, the user assigns to it a name, as well as the appropriate predefined role cluster. The remaining columns exist to enable the user to provide the required set of individual properties. These properties will characterize the individual's particular sensitivity to each of the defined motives. Accordingly, there exists a column for each motive type under which the user must provide a specific value. In the remaining columns, the user must provide activation values for this individual's Behavioral Inhibition, Behavioral Approach, and Disinhibition/Constraint Systems.

In order to assist the creation, execution, and analysis of these models from within the IDE, several supplemental pieces of information have been added to those outlined above. For example, the user may assign a descriptive comment to each situational state, action structure, and behavioral option. These comments are carried over into the runtime component of the IDE for convenient reference. Also, the user is provided with the ability to selectively disable certain motive types, action structures, and individuals that have been defined, causing them to be excluded from the model during its execution.

3.4.3 Runtime Component

This component of the IDE represents the interface to the runtime mechanisms of the PAC architecture. Once the user has, via the editor interface, fully defined all necessary aspects of a project, the *Run Project* button launches the runtime component, thereby executing the model. Referred to as the *Agent Arbitrator* (Figure

20), this interface has several views. Under all views, the *advance to next state* button at the bottom of the interface provides the user with the means to execute their model in a step-by-step manner. As the participating agents take turns perceiving and acting, their corresponding tabs in the interface (visible in Figures 20, 21, and 23) change colors appropriately.



State Stack

SS0: Soldier is at counter

State Properties

Sk-So distance	Sk approached counter?	So approached counter?	Sk uttered	So uttered
5	no	yes		

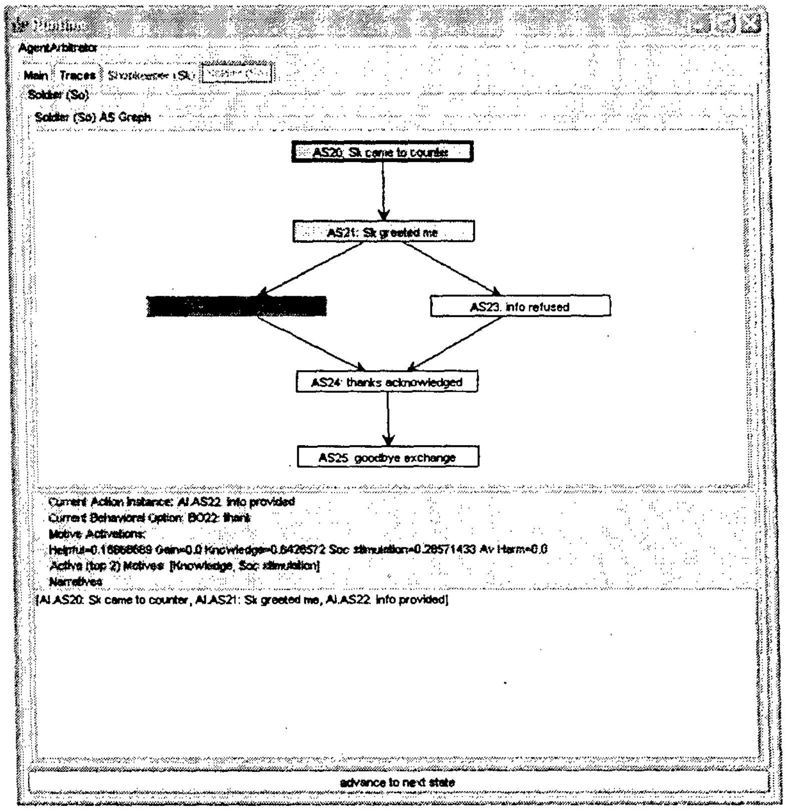
Figure 20. Agent Arbitrator (Main View)

3.4.2.1 Main View

The Agent Arbitrator's main view, shown above in Figure 20, has the *State Stack* as its central feature. This provides the user with a clear overview of the transitions occurring from state to state during model execution. As execution proceeds, the current situational state is always displayed on the stack's top. As mentioned previously, any descriptive comment assigned to a situational state is displayed for the user's convenience. Below the State Stack, a single row of information presents the exact values for each property of the current situational state.

3.4.2.2 Agent Views

The Agent Arbitrator also offers agent-oriented views, one per agent participating in the model. Shown in Figure 21, these views focus on the particular agent's internal structures and the manner in which they change over time. The view's upper frame depicts the action structures of the agent. Their chronological sequence, shown through the connecting arrows, is dependent on the subsequent action structures designated by the user within the editor interface. Consequently, this view also serves to provide the user with a method of verifying correct connection of the agent's plot units. With each advance of model execution, the action structure which has been deemed appropriate by the agent for the current situational state is highlighted in red, preceding action structures highlighted in yellow. The view's lower frame reveals the set of potential narratives, or instantiated plot units, maintained by the agent. Each line consists of the list of action instances comprising the narrative. As execution progresses, this set will grow and shrink as the agent posits new possible narratives and eliminates others. Between the two frames, further details regarding the respective agent's internal state are provided to the user: the current action instance, the most recently chosen behavioral option, the current activation level of each motive, and finally the two highest, or 'active' motives for the agent.



```
Current Action Instance: AI.AS22: info provided
Current Behavioral Option: BO22: thank
Motive Activations:
Helpful=0.16666669 Gain=0.0 Knowledge=0.6428572 Soc stimulation=0.28571433 Av Harm=0.0
Active (top 2) Motives: [Knowledge, Soc stimulation]
Narratives
[AI.AS20: Sk came to counter, AI.AS21: Sk greeted me, AI.AS22: info provided]
```

Figure 21. Agent Arbitrator (Agent View)

Given the particular significance applied to motivation in this spiral, this view of the Agent Arbitrator was supplemented with a standalone charting display for motive activation. This display, depicted in Figure 22, is also provided for each agent of the model. The chart illustrates for the user the agent's activation level of each motive type as they change over the course of the model's execution. By hovering the mouse over a particular bar in the activation chart, the user is able to examine the exact value of a particular motive across the history of the execution. Activation bars displayed in red indicate to the user that the respective motive was one of the active motives for the agent at that particular time. The motives that are active at the current execution step have their names highlighted in red.

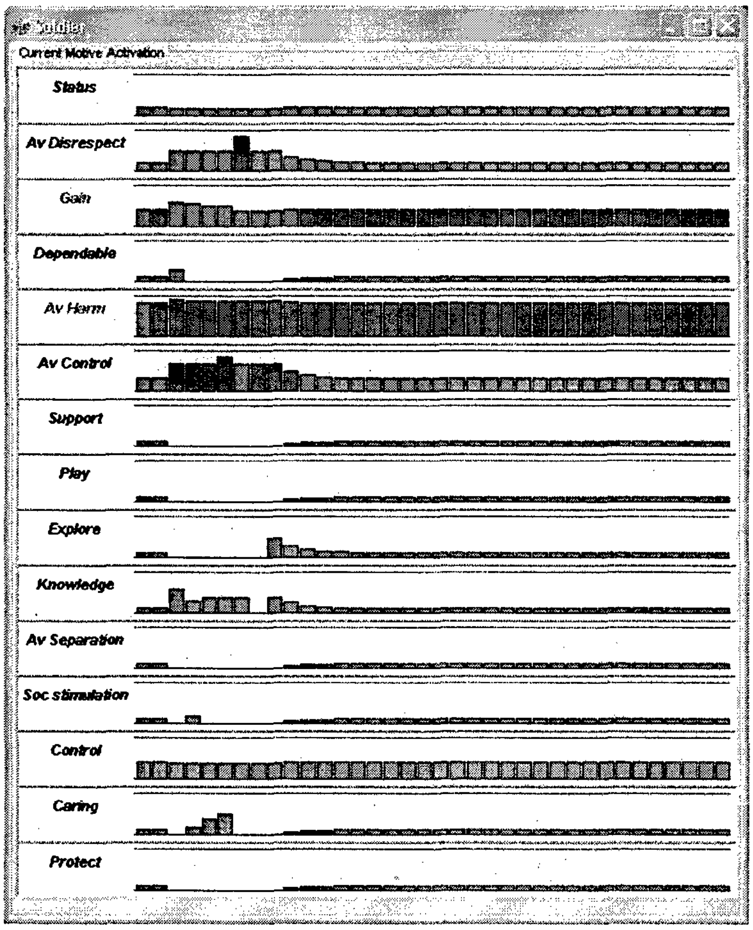
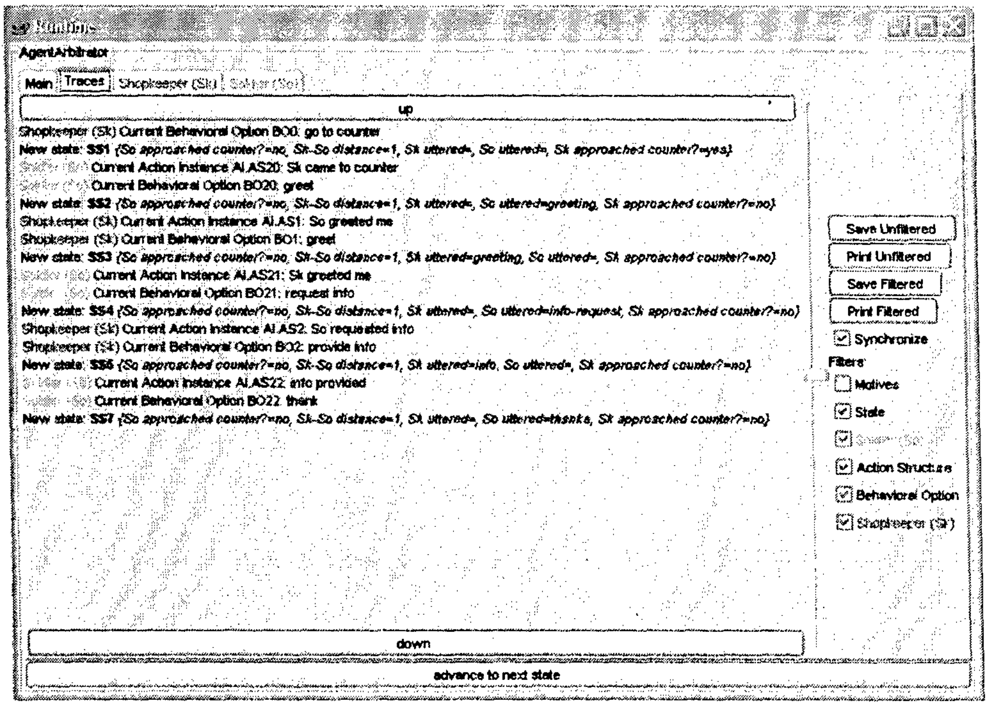


Figure 22. Agent Arbitrator (Motive Activation Display)

3.4.2.3 Tracing View

This final view is considered most apt for debugging and complete analysis of PAC models. Shown in Figure 23, it is capable of providing the user with a full trace of execution. Information displayed includes details of each situational state which has occurred, each action instance chosen by the participating agents, the behavioral options taken by those agents, and the agents' current motives at time of selection. The filter control seen along the right-hand side of Figure 23 allows the user to view only certain aspects of this information at any given time. It may be configured to reveal trace information regarding solely or any combination of motives, states, action structures, or behavioral options. Additionally, information relating to only particular sets of agents may be shown. To aid in the analysis and comparison of multiple models over time, facilities are also provided to save such execution traces. Stored in their color-tagged HTML format, the user can choose to save either the full execution trace or that subject to the filter's configuration.



```
Shopkeeper (Sk) Current Behavioral Option BO0: go to counter
New state: SS1 {So approached counter?=no, Sk-So distance=1, Sk uttered=, So uttered=, Sk approached counter?=yes}
Seller (So) Current Action Instance AI.AS20: Sk came to counter
Seller (So) Current Behavioral Option BO20: greet
New state: SS2 {So approached counter?=no, Sk-So distance=1, Sk uttered=, So uttered=greeting, Sk approached counter?=no}
Shopkeeper (Sk) Current Action Instance AI.AS1: So greeted me
Shopkeeper (Sk) Current Behavioral Option BO1: greet
New state: SS3 {So approached counter?=no, Sk-So distance=1, Sk uttered=greeting, So uttered=, Sk approached counter?=no}
Seller (So) Current Action Instance AI.AS21: Sk greeted me
Seller (So) Current Behavioral Option BO21: request info
New state: SS4 {So approached counter?=no, Sk-So distance=1, Sk uttered=, So uttered=info-request, Sk approached counter?=no}
Shopkeeper (Sk) Current Action Instance AI.AS2: So requested info
Shopkeeper (Sk) Current Behavioral Option BO2: provide info
New state: SS6 {So approached counter?=no, Sk-So distance=1, Sk uttered=info, So uttered=, Sk approached counter?=no}
Seller (So) Current Action Instance AI.AS22: info provided
Seller (So) Current Behavioral Option BO22: thank
New state: SS7 {So approached counter?=no, Sk-So distance=1, Sk uttered=, So uttered=thanks, Sk approached counter?=no}
```

Figure 23. Agent Arbitrator (Tracing View)

4. DISCUSSION

4.1 Example PAC Models

Examples are needed to examine how the PAC model operates in general and how the model operates differently in social interaction when parameters for individual differences are differentially set. In the current work, we developed example models based on a VECTOR scenario involving an exchange between a soldier and a shopkeeper. We refer here to two types of PAC models of this scenario. The first, the simplest 'base-case' model was developed for the purpose of verifying the basics of the architecture. And a second model was developed to test higher-level nuances and complexities of the underlying psychological modeling and to create numerous different personalities (e.g., approach and avoid motivation settings, disinhibition/constraint settings and motive settings) for both soldiers and shopkeepers.

4.1.1 Basic Model

One of the first necessities in the development of the initial spiral was to construct an example model that represented the simplest case possible, while still allowing us to prove the correct operation of the basic elements of the PAC architecture. This trivial model served as a base-case for design and development that was, and will continue to be, a readily available tool to revisit as different mechanisms of the architecture are fine-tuned.

The elements of the architecture that were focused on most, and hence fine-tuned most frequently, included narrative and behavior selection, and motive interpretation and updating. The extended example models discussed in the following section contain complex embedded storylines and interactions that produce very attractive patterns of emergent behavior and yet, for the same reason, make their use for regular debugging purposes difficult. The specification for this model was, therefore, a clearly straightforward storyline that contained, as its minimum requirement, a single branch point of behavior for the agents involved.

Borrowed and scaled down from several VECTOR scenarios, this model is comprised of two agents only, a shopkeeper and a soldier. As in the original scenario, the setting is the shopkeeper's store. The storyline (Figure 24a) is, for the reasons discussed, simple. The soldier approaches the counter, his main intent to obtain a piece of information. The shopkeeper meets him at the counter, where they proceed to engage in small talk. In this example, small talk is restricted to a single exchange of greetings. Once this pleasantry has been accomplished, the soldier proceeds to request the information of interest, the content of which, for our current purpose, is inconsequential. At this time, based on his particular level of the relevant motives, the shopkeeper will make a choice: he will decide to provide the requested information or deny the soldier's request. This behavioral alternative represents the single point of divergence in the story that was the requirement for the model. In either case, the story resumes with closing courtesies similar to the initial greetings. The soldier thanks the shopkeeper, regardless of whether the information was provided. After the shopkeeper acknowledges thanks and the two exchange goodbyes, the soldier exits the shop, thereby ending the story. It is worth noting that the model could be easily elaborated to provide a branch in the behavior of not merely one, but both agents involved. For instance, based on the shopkeeper's decision to honor or deny the request for information, the soldier could have two resulting behavioral options, 'thank you' or 'thanks anyway' respectively. However, for the intent of this model, a branch point for a single agent sufficed.

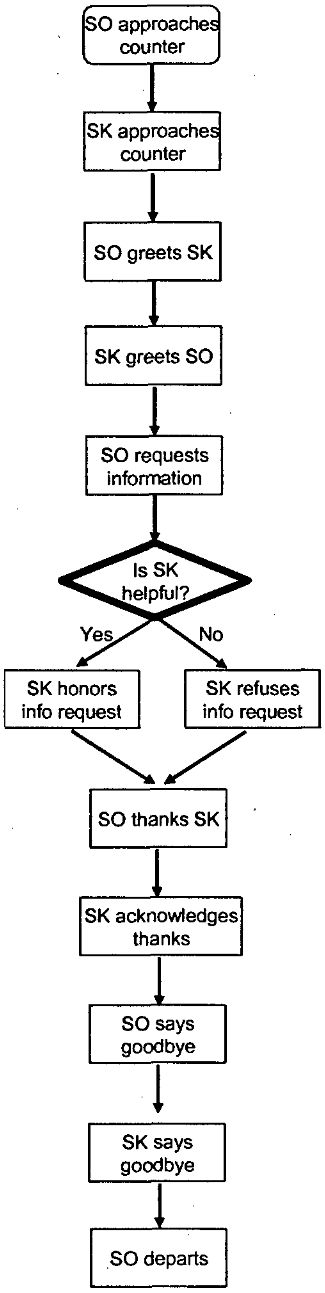


Figure 24a. Basic Model Storyline

The extended models described in the following section contain a rich set of motivations drawn from a broad range of research in personality theory and psychology. The basic model used a very simplified subset of these motives, shown in Table 1. This subset was considered the bare requirement to meet the model's objective, to create a single, motivationally-influenced behavioral choice.

Table 1. Basic Model Motive Structure

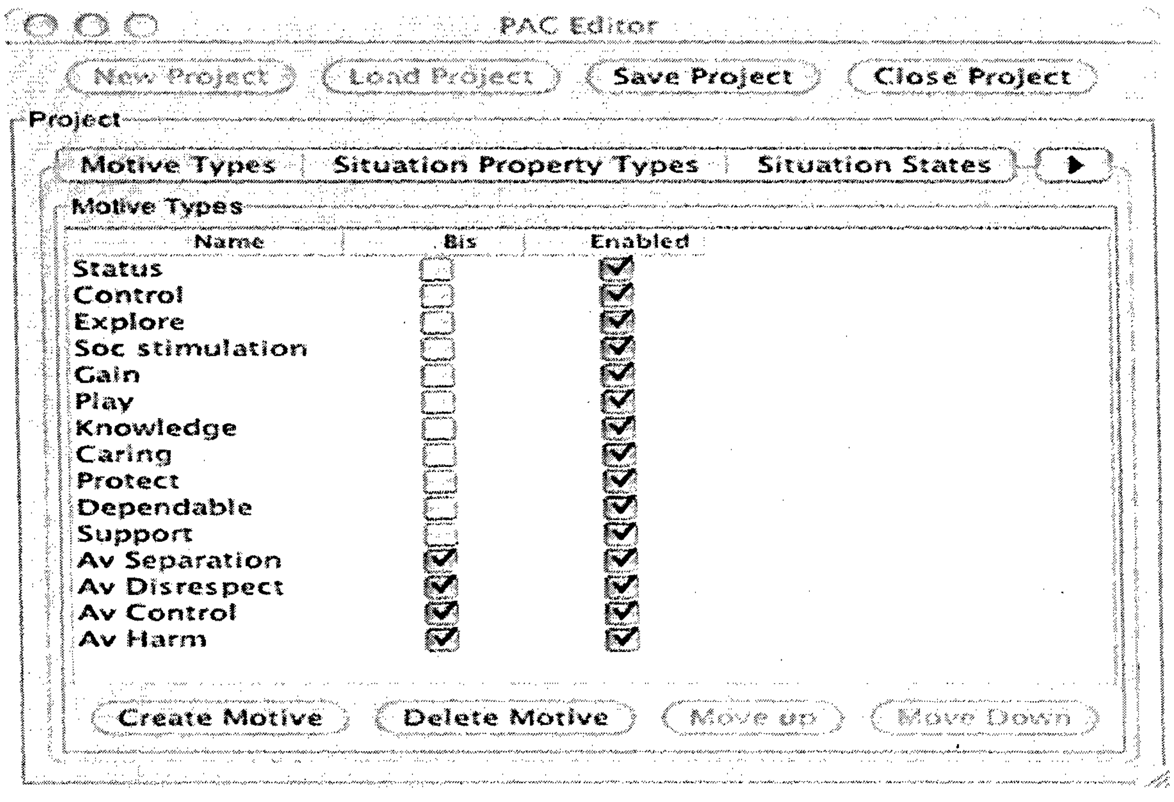
Motive Type	BIS or BAS
Avoid Harm	BIS
Gain	BAS
Helpful	BAS
Knowledge	BAS
Soc stimulation	BAS

The majority of the motive types involved are those regarded as belonging to the Behavioral Approach System. Only the motive *Avoid Harm* is designated as part of the Behavioral Inhibition System. The most noteworthy motives involved are *Helpful* and *Knowledge. Helpful* was used to characterize the shopkeeper's desire to accommodate the soldier by providing the information he in turn desires. Conversely, *Knowledge* was used to represent the soldier's desire to gain this information. The remaining motive types were added essentially for completeness. In fact, for the model to serve accurately as a true base-case for evaluating the architecture, it was important that it included other types of motivations aside from the aforementioned central ones. As discussed in previous sections, the current implementation designates the two motives with the highest level of activation as 'active' motives, and it is only these that are permitted to affect narrative and behavioral selection. Therefore, it was crucial the model contained other motives that were allowed to compete with the *Helpful* and *Knowledge* motives.

As the table indicates, *Helpful* is the motive type of key significance to this model. It is the shopkeeper's activation level of this motive that determines whether he chooses to provide or deny the information the soldier has requested. As a result, altering his initial value of the *Helpful* motive causes the simple split in possible behavior we desired. The model as it exists currently has this starting value set to 0.5. Lowering this value results in the shopkeeper denying the soldier's request; raising the value results in the shopkeeper honoring the soldier's request. Since this motive is marked as part of the Behavioral Approach System, merely increasing the shopkeeper's BAS activation level will also increase his propensity to provide the information. Furthermore, as the other competing motives of the model are marked as part of the Behavioral Inhibition System, one can also increase his BIS activation level, causing him to be more likely to deny information. The reverse holds true as well for decreasing both the BIS and BAS levels. The degree to which these values must be changed to cause the respective behavioral change in the shopkeeper depends on the specifics of the model's configuration. It is these simple effects that have allowed us to constantly evaluate the core mechanisms of the architecture as we iteratively improved certain aspects.

4.1.2 Extended Model

As indicated in Figure 24b in the PAC editor window for motive types, we have specified a variety of motives that are consistent with those suggested by our recent cluster analyses of personality traits (Reference 32).

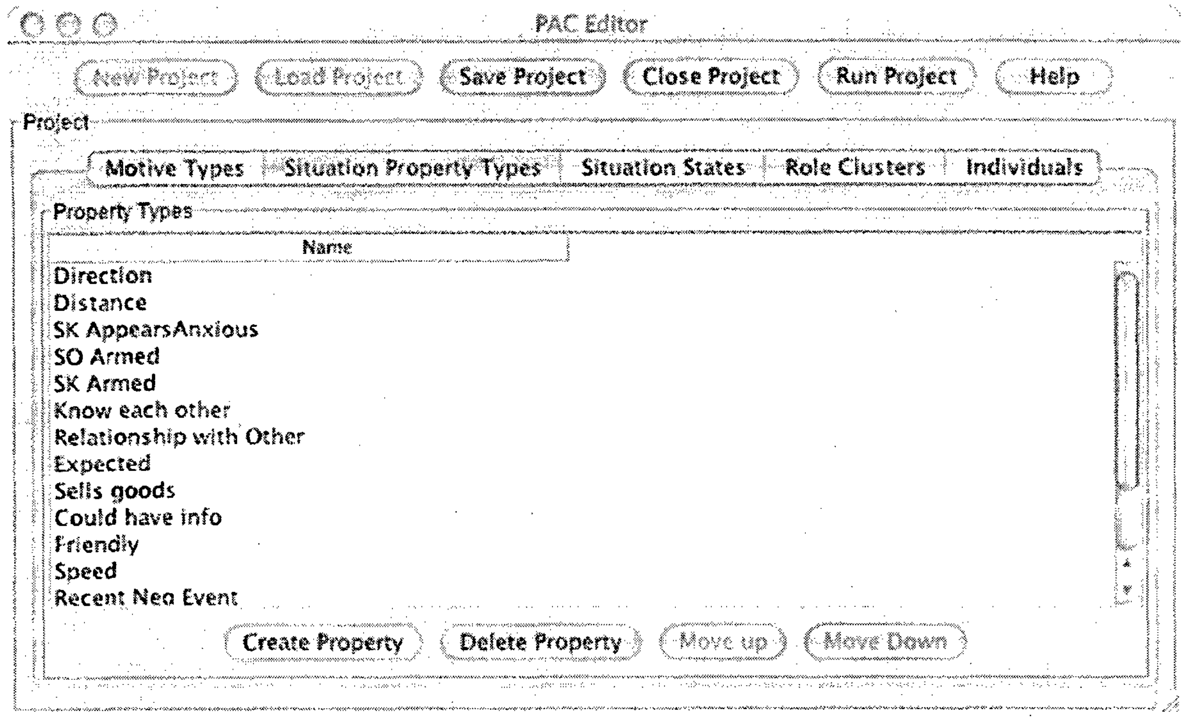


Name	Bis	Enabled
Status	☐	☑
Control	☐	☑
Explore	☐	☑
Soc stimulation	☐	☑
Gain	☐	☑
Play	☐	☑
Knowledge	☐	☑
Caring	☐	☑
Protect	☐	☑
Dependable	☐	☑
Support	☐	☑
Av Separation	☑	☑
Av Disrespect	☑	☑
Av Control	☑	☑
Av Harm	☑	☑

Figure 24b. Motive Types Window

These Motives include: desiring status, desiring control, seeking to explore, seeking social stimulation, seeking material gain and resources, seeking opportunities to play, seeking knowledge, caring, protecting others, being dependable, and supporting others. All of these seem to be approach oriented (BAS) motives. On the avoid side (BIS) are motives such as avoiding separation, avoiding disrespect, avoiding loss of control, and avoiding harm.

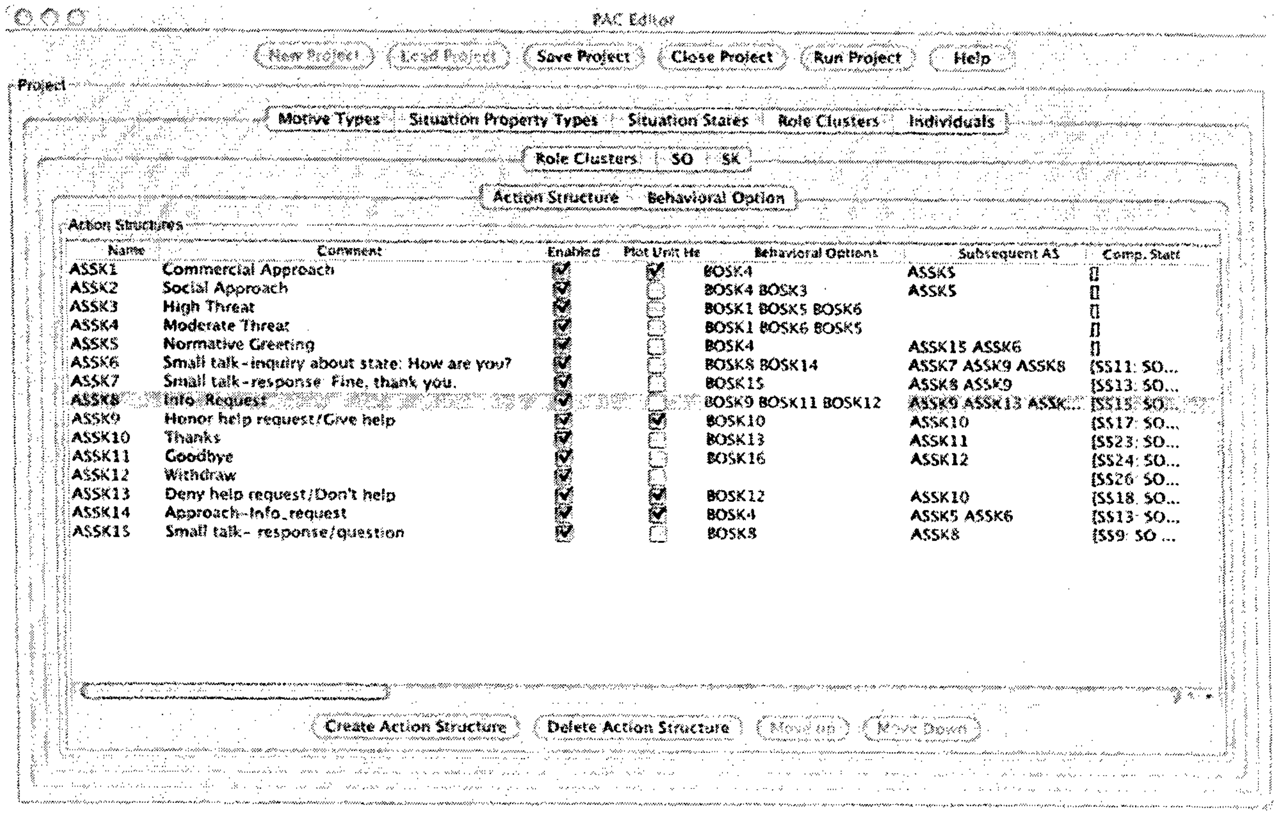
As indicated in Figure 25, in this example a number of situation property types have been defined in order to specify each of the situational states that are specified in the PAC editor for this example.



Direction
Distance
SK AppearsAnxious
SO Armed
SK Armed
Know each other
Relationship with Other
Expected
Sells goods
Could have info
Friendly
Speed
Recent Neo Event

Figure 25. Situation Property Types

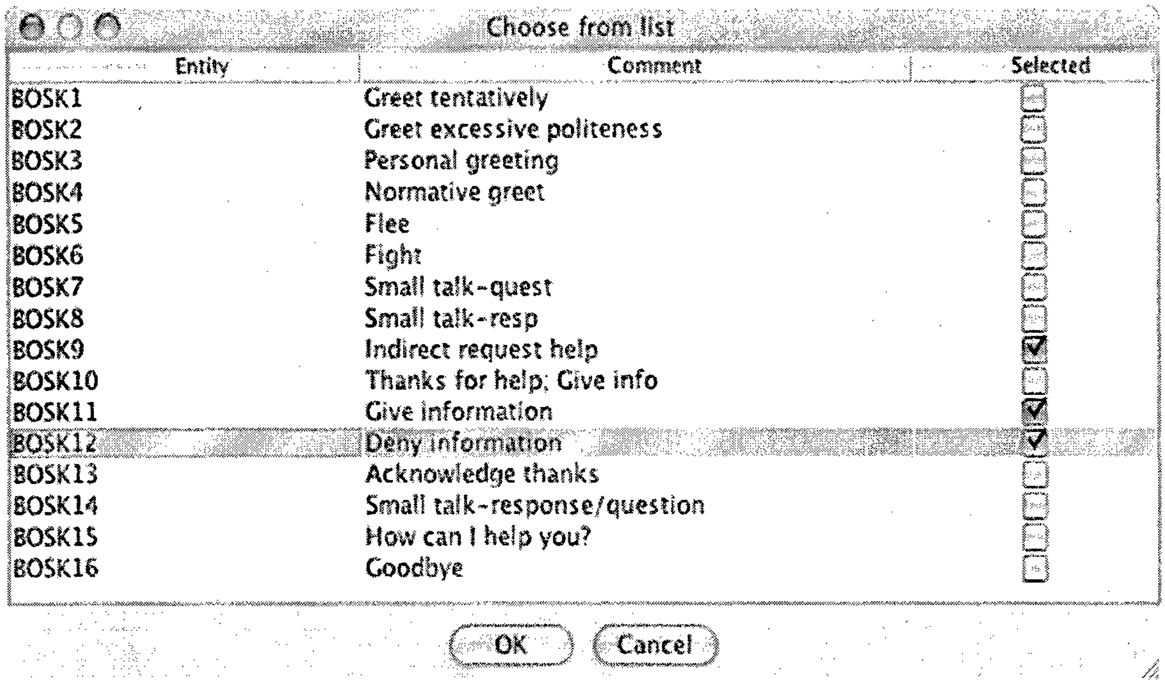
For example, for the situational state SO1, the soldier enters the shop. This is specified in terms of the direction (towards the shopkeeper) and the distance (2). But, additional parameters are specified that could capture the way in which the shopkeeper enters the shop that would change the meaning of the situation. For example, if the SK was armed, or if the relationship was extremely negative, or if the soldier was expected or not, moving rapidly (running) versus moving normally, and so forth. The shopkeeper and the soldier each have their own Role Clusters associated with matching an action structure against these situational states. The shopkeeper's are specified in Figure 26. Separately the role clusters for the shopkeepers and the soldiers specify which action structures might be the heads of plot units. Each of these action structures is associated with motive implications for each of the motives.



Name	Comment	Enabled	Plot Unit He	Behavioral Options	Subsequent AS	Comp. State
ASSK1	Commercial Approach	☑	☑	BOSK4	ASSK5	[]
ASSK2	Social Approach	☑	□	BOSK4 BOSK3	ASSK5	[]
ASSK3	High Threat	☑	□	BOSK1 BOSK5 BOSK6		[]
ASSK4	Moderate Threat	☑	□	BOSK1 BOSK6 BOSK5		[]
ASSK5	Normative Greeting	☑	□	BOSK4	ASSK15 ASSK6	[]
ASSK6	Small talk-inquiry about state: How are you?	☑	□	BOSK8 BOSK14	ASSK7 ASSK9 ASSK8	{SS11: SO...
ASSK7	Small talk-response: Fine, thank you.	☑	□	BOSK15	ASSK8 ASSK9	{SS13: SO...
ASSK8	Info Request	☑	□	BOSK9 BOSK11 BOSK12	ASSK9 ASSK13 ASSK...	{SS15: SO...
ASSK9	Honor help request/Give help	☑	☑	BOSK10	ASSK10	{SS17: SO...
ASSK10	Thanks	☑	□	BOSK13	ASSK11	{SS23: SO...
ASSK11	Goodbye	☑	□	BOSK16	ASSK12	{SS24: SO...
ASSK12	Withdraw	☑	□			{SS26: SO...
ASSK13	Deny help request/Don't help	☑	☑	BOSK12	ASSK10	{SS18: SO...
ASSK14	Approach-Info_request	☑	☑	BOSK4	ASSK5 ASSK6	{SS13: SO...
ASSK15	Small talk- response/question	☑	□	BOSK8	ASSK8	{SS9: SO ...

Figure 26. Role Clusters (Shopkeeper) Action Structured

In addition, the behavioral options are specified for each action structure in Figure 26 and defined in Figure 27.



Entity	Comment	Selected
BOSK1	Greet tentatively	□
BOSK2	Greet excessive politeness	□
BOSK3	Personal greeting	□
BOSK4	Normative greet	□
BOSK5	Flee	□
BOSK6	Fight	□
BOSK7	Small talk-quest	□
BOSK8	Small talk-resp	□
BOSK9	Indirect request help	☑
BOSK10	Thanks for help; Give info	□
BOSK11	Give information	☑
BOSK12	Deny information	☑
BOSK13	Acknowledge thanks	□
BOSK14	Small talk-response/question	□
BOSK15	How can I help you?	□
BOSK16	Goodbye	□

Figure 27. Behavioral Options Window

There are essentially three alternative narratives that are possible in this example at present (see Figure 28). The first of these (Narrative #1) might be defined as an Implicit Trade, Positive Reciprocity, or Tit for Tat (Positive) in which the shopkeeper gets what he wants from the Soldier and the Soldier gets what he wants from the shopkeeper; the second (Narrative #2) as Negative Reciprocity in which the shopkeeper is denied what he wants by the Soldier and the Soldier is denied what he wants by the shopkeeper. A third narrative might be viewed as an "Americanized" or one-sided trade (I want this (information) and expect you to give it to me without anything in return). The same narrative might also operate as a kind of "Give soldiers what they want (if it isn't too costly) to move them on their way quickly and avoid harm when one is feeling threatened by them."

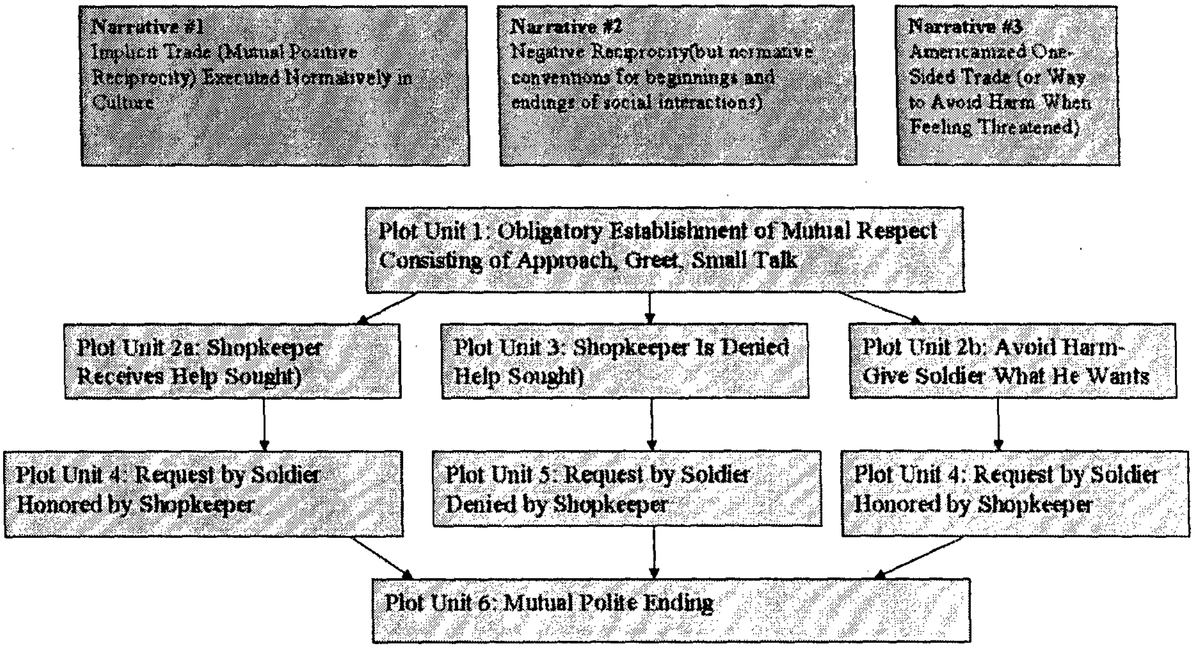

Narrative #1
Implicit Trade (Mutual Positive Reciprocity) Executed Normatively in Culture
Narrative #2
Negative Reciprocity(but normative conventions for beginnings and endings of social interactions)
Narrative #3
Americanized One-Sided Trade (or Way to Avoid Harm When Feeling Threatened)
Plot Unit 1: Obligatory Establishment of Mutual Respect Consisting of Approach, Greet, Small Talk
Plot Unit 2a: Shopkeeper Receives Help Sought)
Plot Unit 3: Shopkeeper Is Denied Help Sought)
Plot Unit 2b: Avoid Harm-Give Soldier What He Wants
Plot Unit 4: Request by Soldier Honored by Shopkeeper
Plot Unit 5: Request by Soldier Denied by Shopkeeper
Plot Unit 4: Request by Soldier Honored by Shopkeeper
Plot Unit 6: Mutual Polite Ending

Figure 28. Alternative Narratives

These narratives are described via a set of action structures we have clustered here, for ease of presentation, as distinct plot units, some of which are combined with others to produce narratives with very different meaning and with different implications for behavior of the soldiers and shopkeepers. To understand these plot units in greater detail, we have specified the action structures involved for the soldier and the shopkeeper and the responses associated with these for each sequence for each plot unit (see Figure 29 through Figure 35).

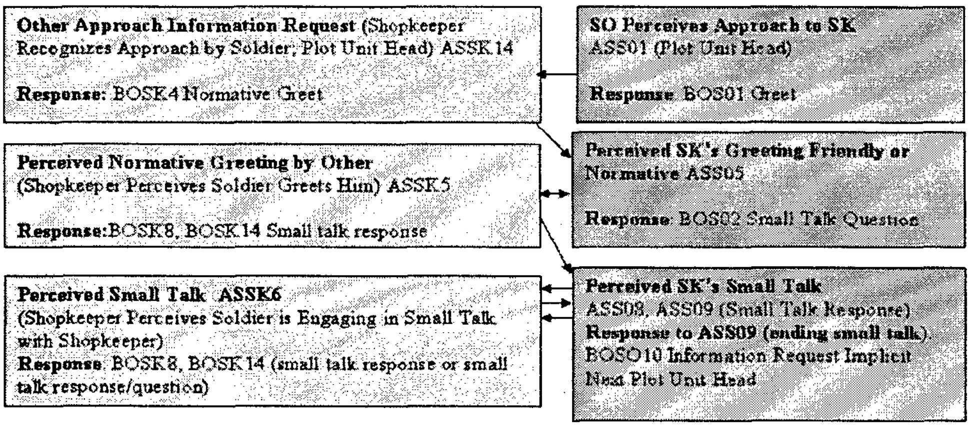

Other Approach Information Request (Shopkeeper Recognizes Approach by Soldier; Plot Unit Head) ASSK14
Response: BOSK4 Normative Greet
SO Perceives Approach to SK ASS01 (Plot Unit Head)
Response BOS01 Greet
Perceived Normative Greeting by Other (Shopkeeper Perceives Soldier Greets Him) ASSK5
Response:BOSK8, BOSK14 Small talk response
Perceived SK's Greeting Friendly or Normative ASS05
Response BOS02 Small Talk Question
Perceived Small Talk ASSK6 (Shopkeeper Perceives Soldier is Engaging in Small Talk with Shopkeeper)
Response BOSK8, BOSK14 (small talk response or small talk response/question)
Perceived SK's Small Talk ASS08, ASS09 (Small Talk Response)
Response to ASS09 (ending small talk). BOSO10 Information Request Implicit Next Plot Unit Head

Figure 29. Plot Unit 1: Obligatory Establishment of Mutual Respect Consisting of Approach, Greet, Small Talk

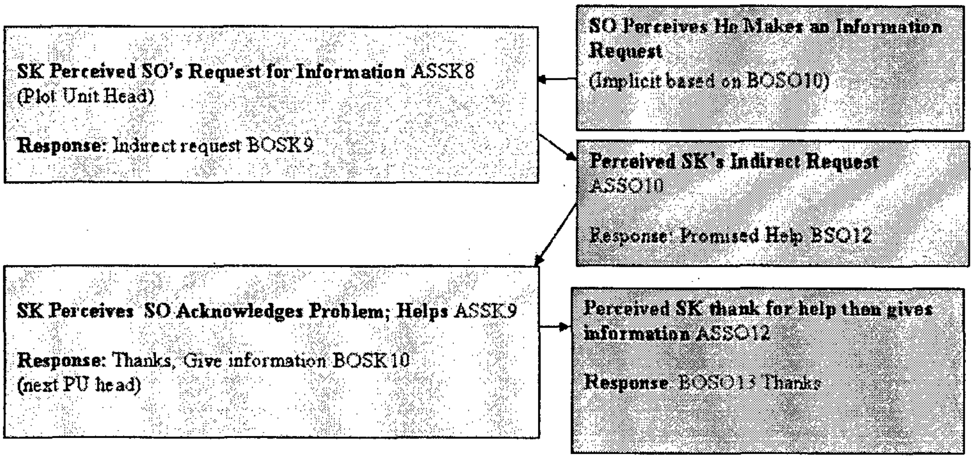

SK Perceived SO's Request for Information ASSK8 (Plot Unit Head)
Response: Indirect request BOSK9
SO Perceives He Makes an Information Request (Implicit based on BOSO10)
Perceived SK's Indirect Request ASSO10
Response: Promised Help BSO12
SK Perceives SO Acknowledges Problem; Helps ASSK9
Response: Thanks, Give information BOSK10 (next PU head)
Perceived SK thank for help then gives information ASSO12
Response BOSO13 Thanks

Figure 30. Plot Unit 2a. Shopkeeper Receives Help Sought (as Embedded Step in Trade with Shopkeeper Cued by SO's request)

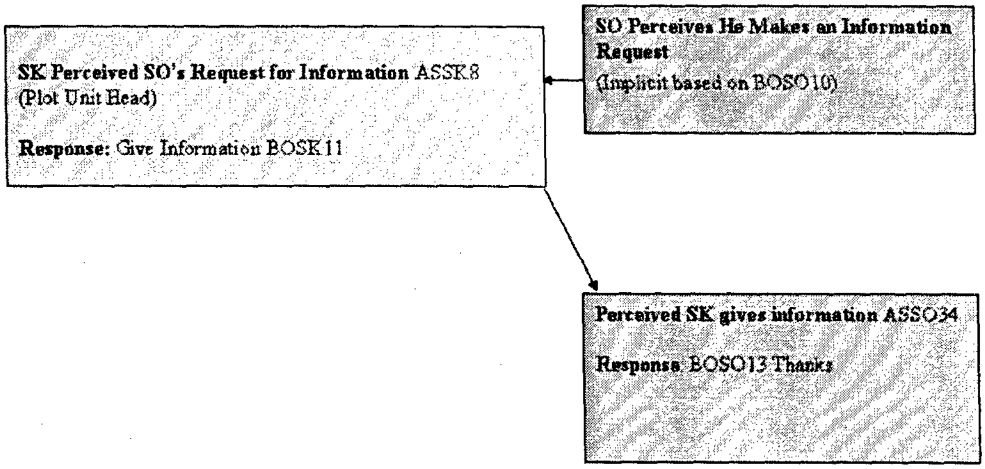

SK Perceived SO's Request for Information ASSK8
(Plot Unit Head)
Response: Give Information BOSK11
SO Perceives He Makes an Information Request
(Implicit based on BOSO10)
Perceived SK gives information ASSO34
Response BOSO13 Thanks

Figure 31. Plot Unit 2b: Threatened Shopkeeper Tries to Minimize Interaction (and Avoid Harm) by Quickly Giving Soldier What He Wants

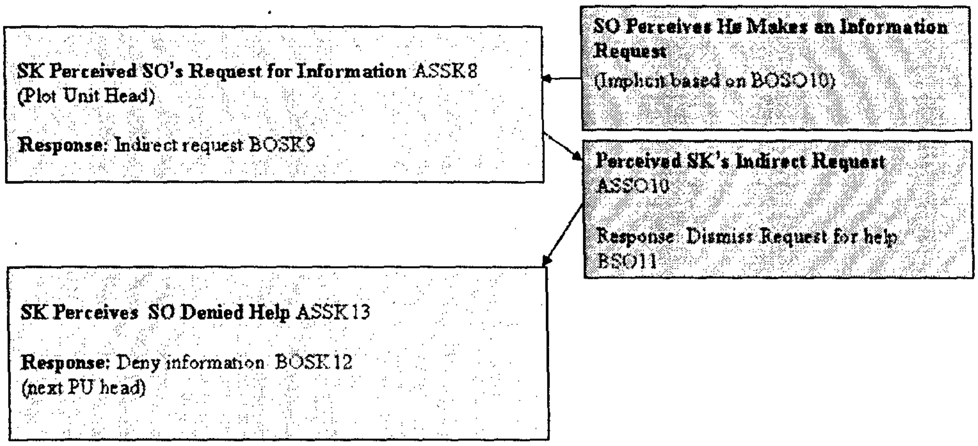

SK Perceived SO's Request for Information ASSK8
(Plot Unit Head)
Response: Indirect request BOSK9
SO Perceives He Makes an Information Request
(Implicit based on BOSO10)
Perceived SK's Indirect Request ASSO10
Response: Dismiss Request for help BSO11
SK Perceives SO Denied Help ASSK13
Response: Deny information BOSK12
(next PU head)

Figure 32. Plot Unit 3: Shopkeeper Is Denied Help He Sought (as Embedded Step in Trade with Shopkeeper Cued by SO's request)

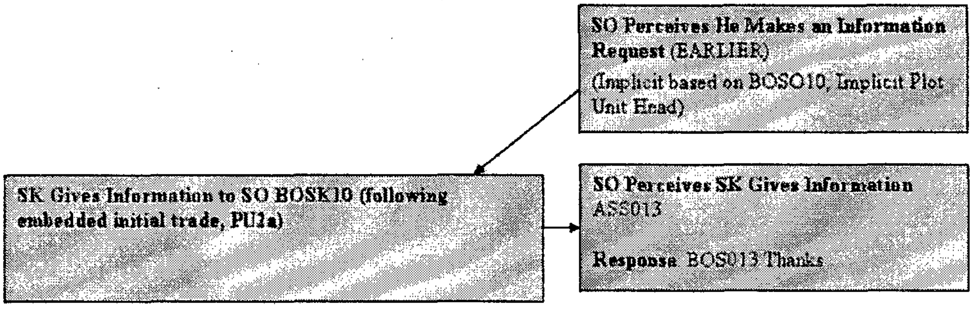

SO Perceives He Makes an Information Request (EARLIER)
(Implicit based on BOSO10, Implicit Plot Unit Head)
SK Gives Information to SO BOSK10 (following embedded initial trade, PU1a)
SO Perceives SK Gives Information ASS013
Response BOS013 Thanks

Figure 33. Plot Unit 4: Request by Solider Honored by Shopkeeper

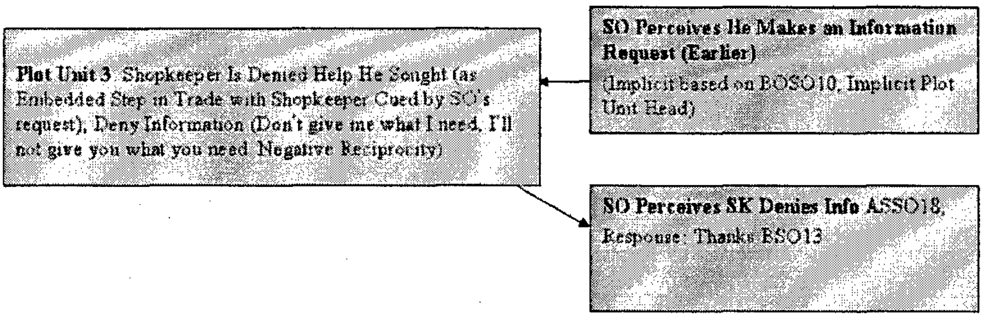

Plot Unit 3 Shopkeeper Is Denied Help He Sought (as Embedded Step in Trade with Shopkeeper Cued by SO's request), Deny Information (Don't give me what I need, I'll not give you what you need Negative Reciprocity)
SO Perceives He Makes an Information Request (Earlier)
(Implicit based on BOSO10, Implicit Plot Unit Head)
SO Perceives SK Denies Info ASSO18,
Response: Thanks BSO13

Figure 34. Plot Unit 5: Request for Information by Soldier Denied by Shopkeeper

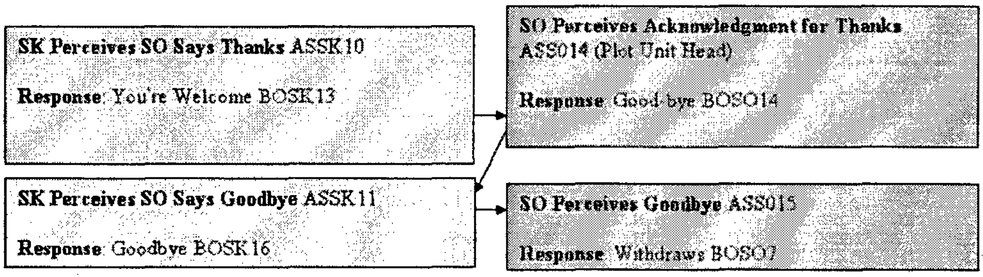
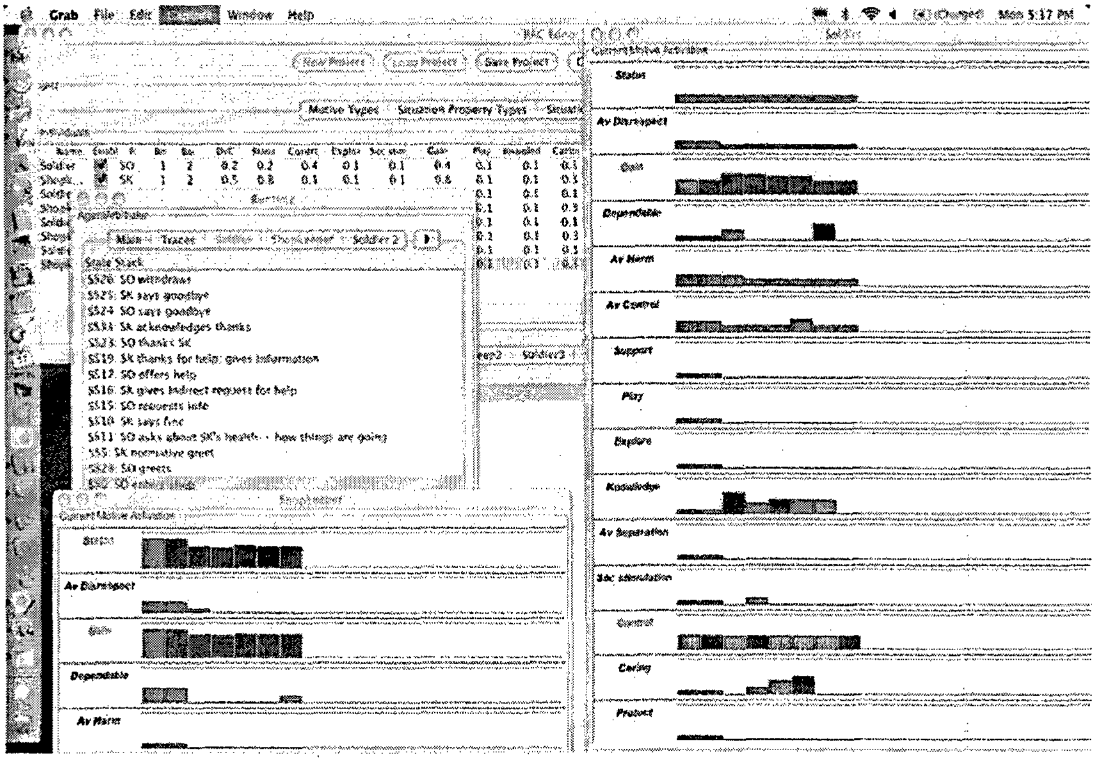

SK Perceives SO Says Thanks ASSK10
Response: You're Welcome BOSK13
SK Perceives SO Says Goodbye ASSK11
Response: Goodbye BOSK16
SO Perceives Acknowledgment for Thanks ASS014 (Plot Unit Head)
Response Good-bye BOSO14
SO Perceives Goodbye ASS015
Response Withdraws BOSO7

Figure 35. Plot Unit 6: Mutual Polite Ending

4.1.3 Individual Differences

When will each alternative narrative be manifest? This depends upon the "personalities" of the soldier and shopkeeper, sometimes in interaction with one another. Figure 36 illustrates the portion of the PAC architecture that is concerned with individual differences in BIS, BAS, DvC, and Motives. As illustrated in Figure 36, here is the run of a soldier and shopkeeper with the BAS set a little higher than the BIS for both, and higher values of inhibitory control set for the DvC for the shopkeeper (SK1) than for the soldier (SO1). The gain motive for SK1 is set relatively high (.8) compared to the soldier (.4), as is the need for status (SK1 .8; SO1 .2). On the other hand, parameters such as the need for control are set higher for SO1 (.4) than for SK1 (.1). These chronic motivations (along with settings for BIS, BAS, and DvC), plus the motives that are activated in the sequence, impact the current level of motivations driving behavioral choice as soldiers and shopkeepers interpret current situations (via action structures) and respond to them. Soldier 1 and Shopkeeper 1 together produce a sequence that maps onto Narrative 1, an implicit trade or positive reciprocity (see Figure 28). What if we merely manipulated the personality parameters in the system, and left the rest of the PAC system alone: Would this be enough to alter which scenarios emerged when different personalities interacted? Would some soldiers (or shopkeepers) always have the same outcomes, regardless of with whom they interacted? Or, would different soldiers interact with different shopkeepers to produce some of the different narratives?



State Stack

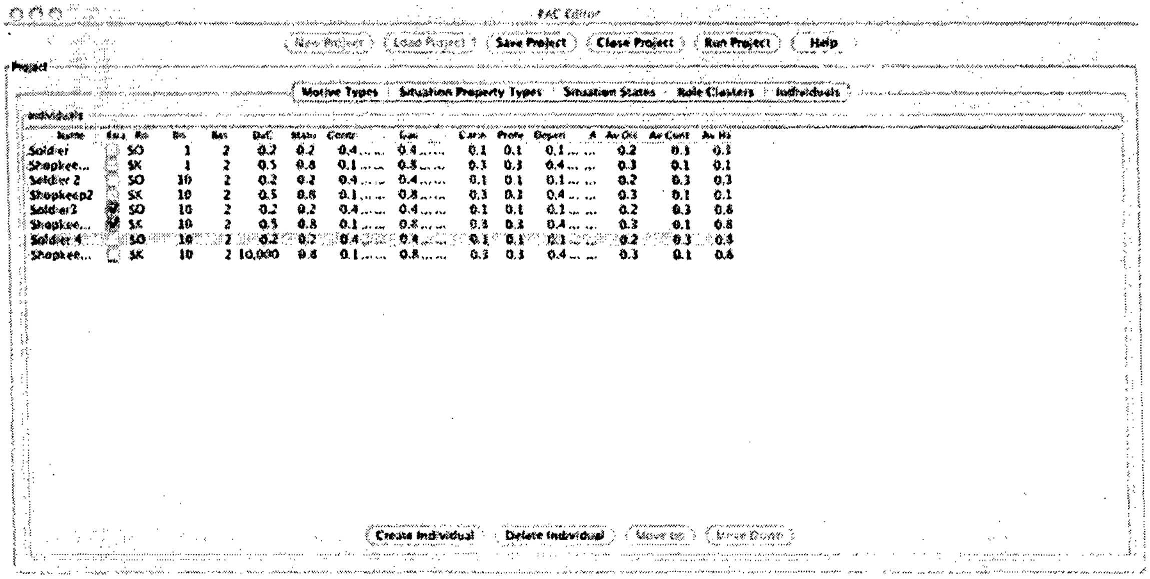
SS26: SO withdraws
SS25: SK says goodbye
SS24: SO says goodbye
SS33: SK acknowledges thanks
SS23: SO thanks SK
SS19: SK thanks for help; gives information
SS17: SO offers help
SS16: SK gives indirect request for help
SS15: SO requests info
SS10: SK says fine
SS11: SO asks about SK's health - how things are going
SS5: SK normative greet
SS28: SO greets
SS2: SO enters shop

Figure 36. Run of an Interaction for a particular Soldier 1 and Shopkeeper 1 (Each With Different Salient Individual Motives and BIS/BAS and DvC Activation; Illustrating the Un-Folding Sequence and Activation of Goals During the "Run"

4.2 Analysis of PAC Extended Model

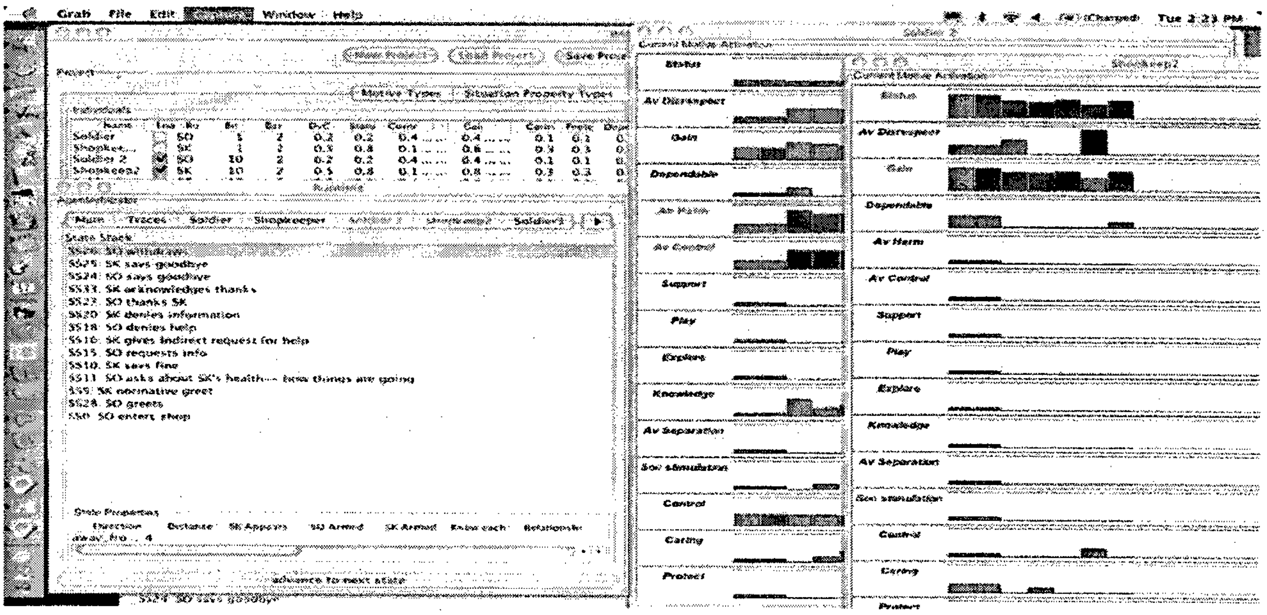
To address these last questions, we developed a number of different soldiers and shopkeepers. For soldiers and shopkeepers 2 through 4, the BIS is activated compared to SO1 and SK1, while the DvC is held in the same proportions for soldiers (.2) versus shopkeepers (.5), except for SK4, where this value was extremely elevated (10,000) (see Figure 37). The only other differences across soldiers and shopkeepers can be found for soldiers and shopkeepers 3 and 4, where avoid harm is elevated to .8 for both shopkeepers and soldiers whereas for soldiers one and two this is kept at the same proportion (.3 for soldiers and .1 for shopkeepers). Just these changes were sufficient to generate the three alternative narratives. Thus, the second narrative involving negative reciprocity can be achieved by simply elevating the BIS of both the soldier and shopkeeper as illustrated by Figure 38. If in addition, the motive "avoid harm" is also elevated (soldier 3 and shopkeeper 3), then the effect is to produce

Narrative 3, where the shopkeeper has become so fearful that he just quickly gives the information to the soldier and ends the interaction normatively (Figure 39). If the shopkeeper's DvC (disinhibition control) is elevated (Figure 40), as is the case for SK4, without changes to any other parameter compared to SK3, SK4 now responds by denying the soldier the information, as SK2 does in his interaction with SO2.



Name	Exa	Ro	Bo	Bas	DvC	Stabi	Contr	Gai	Carin	Profe	Depen	A	Av Obs	Av Cont	Av Hi
Soldier	☐	SO	1	2	0.2	0.2	0.4......	0.4......	0.1	0.1	0.1... ...		0.2	0.3	0.3
Shopkee...	☐	SK	1	2	0.5	0.8	0.1......	0.5......	0.3	0.3	0.4... ...		0.3	0.1	0.1
Soldier 2	☐	SO	10	2	0.2	0.2	0.4......	0.4......	0.1	0.1	0.1... ...		0.2	0.3	0.3
Shopkeep2	☐	SK	10	2	0.5	0.8	0.1......	0.8......	0.3	0.3	0.4... ...		0.3	0.1	0.1
Soldier3	☑	SO	10	2	0.2	0.2	0.4......	0.4......	0.1	0.1	0.1... ...		0.2	0.3	0.6
Shopkee...	☑	SK	10	2	0.5	0.8	0.1......	0.8......	0.3	0.3	0.4... ...		0.3	0.1	0.8
Soldier 4	☐	SO	10	2	0.2	0.2	0.4......	0.4......	0.1	0.1	0.1... ...		0.2	0.3	0.3
Shopkee...	☐	SK	10	2	10,000	0.8	0.1......	0.8......	0.3	0.3	0.4... ...		0.3	0.1	0.6

Figure 37. Examples of Individual Differences in the PAC Architecture



Name	Ena	Ro	Bc	Bar	DvC	Stats	Conv	Gai	Cons	Topic	Depe
Soldier	○	SO	5	2	0.2	0.2	0.4 ...	0.4 ...	0.1	0.1	0.
Shopkee...	○	SK	1	2	0.3	0.8	0.1 ...	0.6 ...	0.3	0.3	0.
Soldier 2	☑	SO	10	2	0.2	0.2	0.4 ...	0.4 ...	0.1	0.1	0.
Shopkeep2	☑	SK	10	2	0.5	0.8	0.1 ...	0.8 ...	0.3	0.3	0.

State Stack

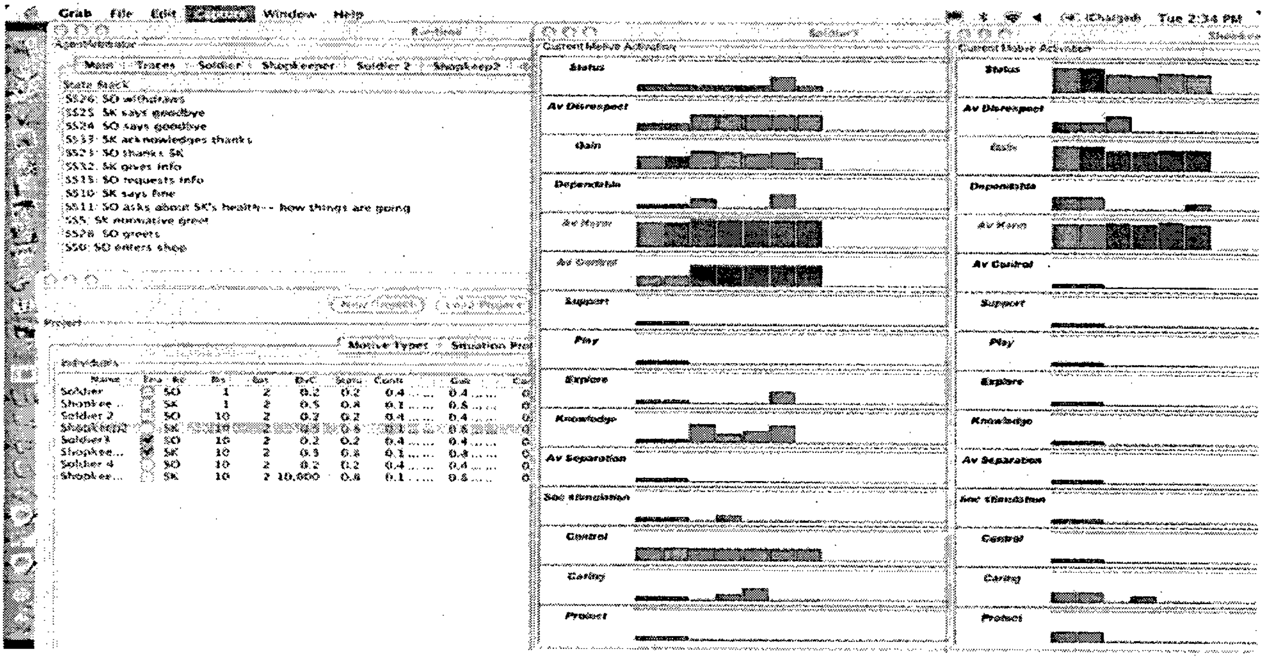
SS26: SO withdraws
SS25: SK says goodbye
SS24: SO says goodbye
SS33: SK acknowledges thanks
SS23: SO thanks SK
SS20: SK denies information
SS18: SO denies help
SS16: SK gives indirect request for help
SS15: SO requests info
SS10: SK says fine
SS11: SO asks about SK's health-- how things are going
SS5: SK normative greet
SS28: SO greets
SS0: SO enters shop

State Properties

Direction	Distance	SK Appears	SO Armed	SK Armed	Know each	Relationship
away_fro...	4					

SS24: SO says goodbye

Figure 38. Soldier 2 and Shopkeeper 2 Run Where BIS (avoid system) has been more heavily activated

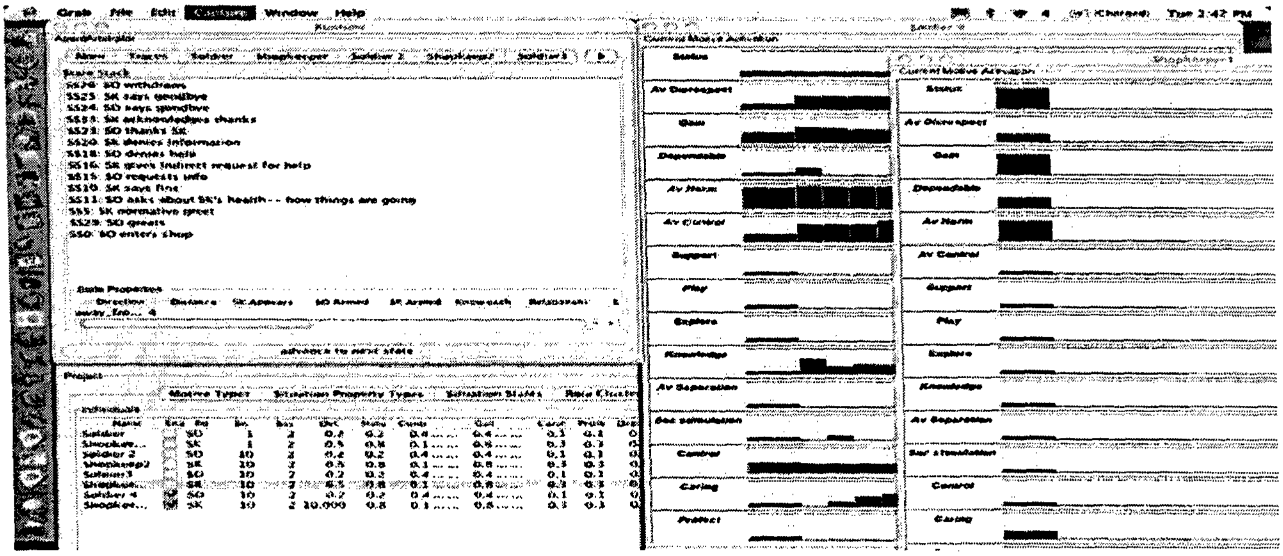


State Stack

SS26: SO withdraws
SS25: SK says goodbye
SS24: SO says goodbye
SS33: SK acknowledges thanks
SS23: SO thanks SK
SS32: SK gives info
SS15: SO requests info
SS10: SK says fine
SS11: SO asks about SK's health-- how things are going
SS5: SK normative greet
SS28: SO greets
SS0: SO enters shop

Name	Ena	Ro	Bs	Bas	DvC	Stats	Conts	Gai	Ca
Soldier	○	SO	1	2	0.2	0.2	0.4 ...	0.4 ...	0
Shopkee...	○	SK	1	2	0.5	0.8	0.1 ...	0.8 ...	0
Soldier 2	○	SO	10	2	0.2	0.2	0.4 ...	0.4 ...	0
Shopkeep2	○	SK	10	2	0.5	0.8	0.1 ...	0.8 ...	0
Soldier3	☑	SO	10	2	0.2	0.2	0.4 ...	0.4 ...	0
Shopkee...	☑	SK	10	2	0.3	0.8	0.1 ...	0.8 ...	0
Soldier 4	○	SO	10	2	0.2	0.2	0.4 ...	0.4 ...	0
Shopkee...	○	SK	10	2	10.000	0.8	0.1 ...	0.8 ...	0

Figure 39. Soldier 3 and Shopkeeper 3 with elevated Motive "Avoid Harm" compared to SO2 and SK2



State Stack
SS26: SO withdraws
SS25: SK says goodbye
SS24: SO says goodbye
SS22: SK acknowledges thanks
SS21: SO thanks SK
SS20: SK denies information
SS18: SO denies help
SS16: SK gives indirect request for help
SS15: SO requests info
SS10: SK says fine
SS11: SO asks about SK's health - - how things are going
SS5: SK normative greet
SS29: SO greets
SS0: SO enters shop

Figure 40. Shopkeeper 4 and Soldier 4 differ from SK3 and SO3 by elevating SK's DvC Only

What if the different soldiers and shopkeepers were paired with one another, what narratives would emerge based on the "main effects" of particular soldiers or shopkeepers or on the interaction between the two? We systematically explored this and found the following pattern as illustrated in Table 2.

Table 2.

Shopkeeper	Soldier			
	SO1	SO2	SO3	SO4
SK1	Trade-Pos. Reciprocity	Negative Reciprocity	Negative Reciprocity	Negative Reciprocity
SK2	Trade-Pos. Reciprocity	Negative Reciprocity	Negative Reciprocity	Negative Reciprocity
K3	Give Info; Avoid Harm	Give Info; Avoid Harm	Give Info; Avoid Harm	Give Info, Avoid Harm
SK4	Trade-Pos. Reciprocity	Negative Reciprocity	Negative Reciprocity	Negative Reciprocity

Essentially, as suggested above SK3 has the same impact on all interactions with Soldiers. Except for this extreme shopkeeper, however, soldier 1 generally has a positive impact on the interaction with the shopkeeper, whereas soldiers 2 through 4 generally have a negative impact on the outcome of the interaction with the shopkeeper. We are currently analyzing the limits of the parameters that can be set and their impact on the behavior of pairs of shopkeepers and soldiers.

4.3 Utility of The PAC IDE

The simple, economical approach described here proved invaluable. During the design and development of the first spiral, many aspects of the architecture were often altered. These aspects included, but were not limited to, the core mechanisms of narrative and behavior selection, and motive interpretation and updating. As we constantly experimented with such elements, transparent access to all representational structures as well as the analytic tools offered by the runtime component permitted us to regularly assess the effects of this experimentation and our overall progress. Moreover, this preliminary design was indispensable in revealing to us where we wish to go in the future in regards to development tools for the PAC architecture.

Although there have been many noteworthy opportunities for improvement realized, those considered most essential fall under the rubric of visualization. Possible visualization enhancements exist in two forms.

The first is in the domain of model authoring. While the direct structural access spoken of here was vital to PAC's initial spiral, the model creator's difficulty in maintaining comprehension of the relations between the separate tables soon became evident. As the underlying elements of the architecture reach maturity in future spirals, it is our hope that such tables will ultimately be completely eliminated from the editor component. Conceptions for potential interfaces to the editor component more closely resemble something akin to the action structure graphs of the runtime component depicted in Figure 28. The user would potentially have the ability to design a PAC model merely by creating and manipulating similar graphical structures. Such graphical structures would be available not solely for action structures, but rather the spectrum of PAC structures (e.g. situational states, behavioral options, individuals). Furthermore, the relationships between these structures would perhaps be more easily perceptible if represented in such a graphical form.

The second domain for visualization enhancement involves model execution. Once a model has been designed, it is equally important that the runtime component expose all relevant aspects of its execution in the most illustrative manner possible. Again, this may entail extending the set of graphical representations in the interface to depict the remainder of the runtime structures and how they evolve over the course of execution, beyond the already present action structure and motive activation graphs.

Combined, these possibilities hold the promise of yielding more natural model creation as well as more analyzable model execution.

4.4 Significance

The PAC system will leverage many of the features of existing cognitive architectures (e.g., COGNET/iGEN®, ACT-R, and SOAR) such as their implementations of plans and planning structures, the ability to represent the impact of resource availability (and other conditions) on plan enactment, and beliefs, knowledge and norms that could act to constrain or afford behavior. However, novel to the PAC system is the hierarchically organized, biologically based motivational system described above. This motivational system provides a means by which personality differences could dynamically and realistically operate to prioritize which goals would guide the specific implementation of which tasks and behavior.

We believe that the PAC system should also be able to capture many aspects of culture and emotion, by manipulating both the underlying motivational structures of agents and the knowledge structures used to interpret the meaning of events. To the degree that personality and affect are culturally constrained (and afforded), the PAC framework will be able to generate HBRs that exhibit culture-specific personality and affective variability.

The current PAC captures two important aspects of how personality is represented: (1) underlying motivational systems, and (2) how actions and events are interpreted in motive relevant ways. A third important aspect of personality is the experience and expression of emotion. People differ in their readiness to experience different emotions and in how they express emotions. Because the expression of emotion is an important part of social interaction, to build realistic and effective agents we need to build agent's that can realistically display individual differences in the expression of human emotions.

It is also worth noting that dealing with the emotions of others is an important part of dealing with a social interaction. The emotion of others can both dramatically influence one's own emotional state, as well as affecting what kinds of things the individual needs to do. It is obviously quite different to have to deal with someone who is enraged versus dealing with someone who is sad. For example, if someone is enraged we need to quickly assess whether there is a threat to our physical safety and we need to decide what to do. On the other hand, if someone is sad, our necessary response is quite different. Thus, building agents that can display individual differences in emotion is an important part of building an effective training environment.

The current consensus in theories of emotion is that emotions are largely driven by what happens to the organism's goals and motives. Since a central part of PAC involves tracking the current state of the agent's motives, it seemed relatively natural to extend the architecture to the expression of emotion.

Our extension of the PAC to capture emotions and emotional expression is based on the extensive literature on emotion appraisal theories (Reference 72, 73, 74, 75, 76, 77). The central idea in these theories is that emotions are a response to the individual's appraisal of what happened to her goals or motives: were they achieved or were they blocked?, as well as to the appraisal of that outcome along several further dimensions, such as agency for the outcome, likelihood of the outcome, efficacy in changing the outcome, etc. The specific emotional response is the result of the specific appraisal.

In extending the architecture, several issues became obvious. First, it became clear that emotions differed in how directly they were activated, that is the extensiveness of the appraisal that was necessary. Second, and somewhat related to this point, it seemed that emotions differed in the extent to which they were tied to the activation of a particular motive or could occur in response to what happened to a wide range of motives.

The emotions of Fear and Anger provide examples of both these points. First, Fear frequently seems to be directly activated by the relevant situational cues, such as a large, looming object, a sudden, loud noise, a gun; whereas anger frequently requires a more prolonged appraisal process that requires the identification of the party responsible for the blockage of the goal. Second, Fear seems to be largely a response to the activation of one particular motive: to Avoid Physical Harm; whereas Anger can be a response to the blockage of a wide range of different goals. That is, Anger can be a response to the blockage of essentially any goal, including the goal of Avoiding Physical Harm; whereas Fear is primarily tied to the desire to Avoid Physical Harm, (although it might also be strongly tied to the desire to avoid Social Rejection).

These two issues have obvious implications for how the architecture needs to be modified to successfully capture emotions. Implementing emotions such as Fear, which are directly activated and tied to a limited set of motives, would seem to require minimal modifications. Essentially, Fear would occur any time the motive to Avoid Physical Harm is strongly activated. And the intensity of the Fear would be related to the strength of activation of the motive.

However, other emotions would seem to depend on a much more extensive appraisal process. To capture this more extensive appraisal process, we will need to modify the PAC in several ways. First, we will need to explicitly track whether a motive is achieved or not, or whether pursuit of a motive is blocked. In the current architecture, we only track the current activation of motives, but there is no way to tell, for instance, whether a drop in motive activation means that motive has been successfully achieved or whether the individual has simply moved to another situation that no longer supports or activates the motive. Second, we need to be able to make a variety of different appraisals, such as the agent of an outcome and judgments of one's ability to overcome any motive blockage.

In the preceding discussion, we suggested that implementing Fear required fewer modifications than implementing emotions such as Anger. However, it is almost certainly the case that if we wanted to capture the full range of Fear responses, things are more complicated. For instance, the intensity and duration of Fear will probably depend, at least partially, on judgments of one's efficacy to overcome the threat and on the likelihood that the threat will lead to actual harm. So while the initial Fear response may be fairly quick and direct, its ultimate intensity and duration will depend on whether the individual thinks they can handle the threat. For example, an individual with high self-efficacy for handling the threat or one who thinks the threat is highly unlikely to cause harm, will have less intense and shorter Fear responses. This argues that implementing even direct emotions, such as Fear, will require the appraisal mechanism that we are constructing.

REFERENCES

1. Pew, R. & Mavor, A. (Eds.) (1998). *Modeling Human and Organizational Behavior: Application to Military Simulations.* Washington, DC: National Academy Press.

2. Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R., Gobet, F., & Baxter, G. D. (2000). Techniques for modeling human performance in synthetic environments: *A supplementary review. (State-of-the-art-report). Wright-Patterson Air Force Base*, OH: Human Systems Information Analysis Center.

3. Vaubel, K. P., & Gettys, C. F. (1990). Inferring User Expertise for Adaptive Interfaces, *Human Computer Interaction, Vol. 5*, pp. 95-117, Lawrence Erlbaum.

4. Zachary, W., Cannon-Bowers, J., Bilazarian, P., Krecker, D., Lardieri, P., & Burns, J. (1999). The Advanced Embedded Training System (AETS): An intelligent embedded tutoring system for tactical team training. *International Journal of Artificial Intelligence in Education, 10*, 257-277.

5. Gluck, K. and Pew, R. (Eds.) (forthcoming). *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation.* Mahwah, NJ: Erlbaum.

6. Jones, R., Laird, J., Nielsen, P., Coulter, K., Kenny, P., & Koss, F. (1999). Automated Intelligent Pilots for Combat Flight Simulation. *AI Magazine.* Spring, 1999.

7. Weiland, M.Z., Campbell, G., Zachary, W.W., and Cannon-Bowers, J. (1998). Application of Embedded Cognitive Models in a Combat Information Center. In *Proceedings of the 1998 Command and Control Research and Technology Symposium,* Monterey, CA.

8. Damasio, Antonio (1994). *Descartes' Error: Emotion, Reason, and the Human Brain.* NY: Avon.

9. Saucier, G., & Goldberg, L. R. (1996a). Evidence for the Big Five in analyses of familiar English personality adjectives. *European Journal of Personality*, 10, 61-77.

10. Tellegen, A., & Waller, N. G. (1997). Exploring personality through test construction: Development of the Multidimensional Personality Questionnaire. In S. R.

Briggs, J. M. Cheek, & E. M. Donohue (Eds.), *Handbook of adult personality inventories*. New York: Plenum Press.

11. Costa, P. T., Jr., & McCrae, R. R. (1992). Revised NEO Personality Inventory (Neo-PI-R) and NEO Five-Factor Inventory (NEO-FFI): *Professional manual.* Odessa, FL: Psychological Assessment Resources.

12. Wiggins, J. S., & Trapnell, P. D. (1996). A dyadic-interactional perspective on the five-factor model. In J. S. Wiggins (Ed.), *The five-factor model of personality: Theoretical perspectives* (pp. 88-162). New York: Guilford Press.

13. McCrae, R. R., & John, O. P. (1992). An introduction to the Five-Factor Model and its applications. *Journal of Personality, 60,* 175-215.

14. Read, S. J., Jones, D. K., & Miller, L. C. (1990). Traits as goal-based categories: The role of goals in the coherence of dispositional categories. Jrnl of Personality and Social Psych., 58, 1048-1061.

15. Read, S. J., & Miller, L. C. (1998). On the dynamic construction of meaning: An interactive activation and competition model of social perception. In S. J. Read & L. C. Miller (Eds.), *Connectionist models of social reasoning and behavior.* Mahwah, NJ: Erlbaum.

16. Clark, L. A., & Watson, D. (1999). Temperament: A New Paradigm for Trait Psychology. In L. Pervin & O. John (Eds.), *Hndbk. of Personality: Theory and Research* (2nd ed.). NY: Guilford Press, pp.399-423.

17. Depue, R. A. (1996). A neurobiological framework for the structure of personality and emotion: Implications for personality disorders. In J. Clarkin & M. Lenzenweger (Eds.), *Major theories of personality disorders* (pp. 347-390). New York: Guilford Press.

18. Pickering, A. D., & Gray, J. A. (1999). The Neuroscience of Personality. In L. Pervin & O. John (Eds.). *Hndbk. of Personality: Theory and Research* (2nd ed.). NY:Guilford Press, pp.277-299.

19. Rothbart, M. K., & Bates, J. E. (1998). Temperament. In W. Damon (Series Ed.) & N. Eisenberg (Vol. Ed.). *Handbook of Child Psychology*: Vol. 3. Social, emotional, and personality development. (5th ed.) (pp. 105-176). New York: Wiley.

20. Panksepp, J. (1998). *Affective Neuroscience: The Foundations of Human and Animal Emotions*. New York: Oxford University Press.

21. Panksepp, J. (2000). Emotions as natural kinds in the mammalian brain. In M. Lewis & J. M. Haviland-Jones (Ed.), *Handbook of Emotions (2nd edition)*. (pp. 137-156). New York: Guilford Press.

22. Bugental, D. B. (2000). Acquisition of the algorithms of social life: A domain-based approach. *Psychological Bulletin*, 126, 187-219.

23. Fiske, A. P. (1992). The four elementary forms of sociality: Framework for a unified theory of social relations. *Psychological Review*, 99, 689-723.

24. Kenrick, D. T., & Trost, M. R. (1997). Evolutionary approaches to relationships. In S. Duck (Ed.), *Handbook of personal relationships: Theory, research, and interventions* (pp. 151-177). Chichester, England: Wiley.

25. Miller, L. C., Pedersen, W., Putcha-Bagavatula, A., & Read, (2003). Attachment fertility theory: Interwoven design in an embodied evolutionary dynamics. *Unpublished manuscript, University of Southern Califomia.*

26. Gray, J. A. (1987). *The psychology of fear and stress*. (2nd Ed). New York: Cambridge University Press.

27. Depue, R.A. & Morrone-Strupinsky, J. V. (in press). A neurobehavioral model of affiliative bonding: implications for conceptualizing a human trait of affiliation. *Behavioral and Brain Sciences*

28. Cacioppo, J. T., Gardner, W. L., & Berntson, G. G. (1999). The affect system has parallel and integrative processing components: Form follows function. *Jml of Personality and Social Psych.*, 76, 839-855.

29. Lang, P. J., Bradley, M. M., & Cuthbert, B. N. (1990). Emotion, attention, and the startle reflex. *Psychological Review*, 97, 377-395.

30. Watson, D., Wiese, D., Vaidya, J., & Tellegen, A. (1999). The two general activation systems of affect: Structural findings, evolutionary considerations, and psychobiological evidence. *Joumal of Personality and Social Psychology*, 76, 820-838.

31. Davidson, R. J., Jackson, D. C., & Kalin, N. H. (2000). Emotion, Plasticity, Context, and Regulation: Perspectives from Affective Neuroscience. *Psychological Bulletin*, 126, 890-909.

32. Davidson, R. J., & Irwin, W. (1999). The functional neuroanatomy of emotion and affective style. *Trends in Cognitive Sciences*, 3, 11-21.

33. Monroe, B., Read, S. J., & Miller, L. C. (2004). Mapping Personality Space with Cluster Analysis: Uncovering the Structure of Personality Dynamics. *University of Southem Califomia, Unpublished manuscript.*

34. Tellegen, A. (1985). Structures of mood and personality and their relevance to assessing anxiety, with an emphasis on self-report. In A. H. Tuma & J. D. Maser (Eds.), *Anxiety and the anxiety disorders.* (pp. 681-706). Hillsdale, NJ: Erlbaum.

35. Watson, D., & Clark, L. A. (1993). Behavioral disinhibition versus constraint: A dispositional perspective. In D. M. Wegner & J. W. Pennebaker (Eds.), *Handbook of mental control.* (pp. 506-527). Upper Saddle River, NJ: Prentice Hall.

36. Nigg, J. T. (2000). On inhibition/disinhibition in developmental psychopathology: Views from cognitive and personality psychology and a working inhibition taxonomy. *Psych. Bulletin*, 126, 220-246.

37. White, R. W. (1959). Motivation reconsidered: The concept of competence. *Psychological Review*, *66*, 297-333.

38. Hofstee, W. K. B., de Raad, B., & Goldberg, L. R. (1992). Integration of the Big-Five and circumplex approaches to trait structure. *Joumal of Personality and Social Psychology*, 63, 146-163.

39. Benedict, R. (1959). *Pattems of culture.* Boston: Houghton Mifflin.

40. D'Andrade, R. (1992). Schemas and motivation. In R. G. D'Andrade and C. Strauss (Eds.), *Human motives and cultural models* (pp. 23-44). Cambridge: Cambridge University Press.

41. Schwartz, S. H., & Boehnk, K. (2004). Evaluating the structure of human values with confirmatory factor analysis. *Joumal of Research in Personality, 3 8,* 230–255

42. Shiraev, E. & Levy, L. (2004). *Cross-cultural psychology: Critical thinking and contemporary applications*. New York: Allyn and Bacon.

43. Smith, P. B., & Schwartz, S. H. (1997). Values. In J. W. Berry, M. H. Segall & C. Kagitcibasi (Eds.), *Handbook of Cross-cultural psychology: Social behavior and applications* (Vol. 3, pp. 77-118). Needham Heights, MA: Allyn-Bacon.

44. Hofstede, G. (1980). *Culture's consequences*. Beverly Hills, CA: Sage.

45. Bakan, D. (1966). *The duality of human existence*: An essay on psychology and religion: Rand McNally.

46. Triandis, H.C. (1995). *Individualism and collectivism*. Boulder: Westview.

47. Triandis, H.C. (2002). Cultural influences on personality. *Annual Review of Psychology*.

48. Triandis, H. C., & Suh, E. M. (2002). Cultural Influences on Personality. *Annual Review of Psychology, 53*, 133–60.

49. Berry, J. W., Segall, M. H., & Kagitçibasi, C. (1997). (Eds.). *Handbook of Cross-cultural psychology: Social behavior and applications* (Vol. 3). Needham Heights, MA: Allyn-Bacon.

50. Lehman, D. R., Chiu, C., & Schaller, M. (2004). Psychology and Culture. *Annual Review of Psychology, 55*, 689–714.

51. Markus, H. R., Kitayama, S., & Heiman, R. J. (1996). Culture and Basic Psychological Principles. In E. T. Higgins & A. W. Kruglanski (Eds.), *Social Psychology: Handbook of Basic Principles.* (pp. 857-913). New York: Guilford Press.

52. McCrae, R. R., Costa, P. T., Jr., Ostendorf, F., Angleitner, A., Hrebickova, M., Avia, M. D., Sanz, J., Sanchez-Bernardos, M. L., Kusdil, M. E., Woodfield, R., Saunders, P. R., & Smith, P. B. (2000). Nature over nurture: Temperament, personality, and life span development. *Joumal of Personality and Social Psychology*, 78, 173-186.

53. Van Ijzendoorn, M. H., & Sagi, A. (1999). Cross-cultural patterns of attachment: Universal and contextual dimensions. In J. Cassidy & P. R. Shaver (Eds.), *Handbook of Attachment: Theory, Research, and Clinical Applications* (pp. 713-734). New York: Guilford Press.

54. Costa, P. T., Jr., & McCrae, R. R. (1994). Set like plaster? Evidence for the stability of adult personality. In T. F. Heatherton & J. L. Weinberger (Eds.), *Can Personality Change?* (pp. 21-40). Washington, DC: American Psychological Association.

55. Goldberg, L. R. (1981). Language and individual differences: The search for universals in personality lexicons. In L. Wheeler (Ed.), *Review of personality and social psychology: Vol. 2* (pp. 141-165). Beverly Hills, CA: Sage.

56. Depue, R.A., & Collins, P. F. (1999). Neurobiology of the Structure of Personality: Dopamine, Facilitation of Incentive Motivation, and Extraversion. *Behavioral and Brain Sciences, 22,* 491-569.

57. Read, S. J., & Miller, L. C. (2002). Virtual Personalities: A neural network model of personality. *Personality and Social Psychology Review*, 6, 357-369.

58. Boehm, B. (1988). *A Spiral Model of Software Development and Enhancement,* IEEE Computer, pp. 61-72, May 1988

59. Boehm, B. (2002). *Get Ready for Agile Methods with Care,* IEEE Computer, pp. 64-69, January 2002.

60. Beck, Kent (1999). *Extreme Programming Explained: Embrace Change.* Addison-Wesley, Boston. 1999.

61. Jacobson, I., (1987). Object oriented development in an industrial environment. OOPSLA '87: Object-Oriented Programming Systems, Languages and Applications. ACM SIGPLAN, pp. 183-191.

62. Cockburn, A., (2002). Use Cases, Ten Years Later. *Software Testing and Quality Engineering Magazine (STQE),* Vol. 4, No. 2, March/Apr 2002, pp. 37-40.

63. Shaver, P., Schwartz, J., Kirson, D., and O'Connor, C. (1987) Emotion Knowledge: Further Exploration of a Prototype Approach. *Journal of Personality and Social Psychology, Vol. 52,* No. 6, pp. 1061-1086.

64. Lehnert, W. G. (1981). Plot units and summarization. *Cognitive Science, 4*, 293-331.

65. Lehnert, W. G., & Loiselle, C. L. (1989). An introduction to plot units. In D. L. Waltz (Ed.). *Semantic structures: Advances in natural language processing.* Hillsdale, NJ: Erlbaum.

66. Schank, R. C., Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Hillsdale, NJ: Erlbaum.

67. Schank, R. C., & Abelson, R. P. (1995). Knowledge and memory: The real story. In R. S. Wyer, Jr. (Ed.), *Knowledge and memory: The real story. Advances in social cognition, vol. 8 (pp. 1-85).* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

68. Miller, L. C., & Read, S. J. (1991). On the coherence of mental models of persons and relationships: A knowledge structure approach. In G. J. O. Fletcher & F. Fincham (Eds.), *Cognition in Close Relationships.* (pp. 69-99). Hillsdale, NJ: Erlbaum.

69. Read, S. J. (1987). Constructing causal scenarios: A knowledge structure approach to causal reasoning. *Journal of Personality and Social Psychology*, 52, 288-302.

70. Read, S. J., & Miller, L. C. (1989). Interpersonalism: Toward a goal-based theory of persons in relationships. In L. Pervin (Ed.), *Goal concepts in personality and social psychology.* Hillsdale, NJ: Erlbaum.

71. Lazarus, R. S. (2001). Relational meaning and discrete emotions. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal processes in emotion: Theory, methods, research (pp. 37-67).* Oxford: Oxford University Press.

72. Ortony, A., Clore, G. L., & Collins, A. (1988). *The Cognitive Structure of Emotions.* New York: Cambridge University Press.

73. Roseman, J. J. (2001). A model of appraisal in the emotion system: Integrating theory, research, and applications. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal processes in emotion: Theory, methods, research (pp. 68-91).* Oxford: Oxford University Press.

74. Scherer, K. R. (2001). Appraisal considered as a process of multilevel sequential checking. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal processes in emotion: Theory, methods, research (pp. 92-120).* Oxford: Oxford University Press.

75. Smith, C. A., & Kirby, L. D. (2001). Toward delivering on the promise of appraisal theory. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal*

processes in emotion: Theory, methods, research (pp. 121-138). Oxford: Oxford University Press.

76. Gratch, J. & Marsella, S. (in press). A Domain-independent Framework for Modeling Emotion. *Journal of Cognitive Systems Research.*

77. Stein, N. L., Trabasso, T., & Liwag, M. D. (2000). A goal appraisal theory of emotional understanding: Implications for development and learning. In M. I. Lewis & J. M. Haviland-Jones (Eds.), *Handbook of Emotions* (2nd edition) (pp. 436-457). New York: Guilford Press.