May 2003

# Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects

Jason B. Cornwell
*University of Pennsylvania*

Kevin O'Brien
*University of Pennsylvania*

Barry G. Silverman
*University of Pennsylvania*, basil@seas.upenn.edu

Jozsef A. Toth
*Institute for Defense Analyses*

# Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects

**Abstract**
This paper describes an effort to revise the PMFserv agent architecture in order to implement J.J. Gibson's Affordance Theory. The theoretical justification for this revision is outlined along with the engineering constraints that inspired it. We describe the resulting architectural changes and the impact of those changes on the flexibility, ease of rapid scenario creation, and ability to reuse previous investments in knowledge engineering offered by our architecture. The level of effort required to build a new scenario within PMFserv both with and without the revisions suggested by Affordance Theory is compared. We conclude that Affordance Theory is an elegant solution to the problem of providing both rapid scenario development and the simulation of individual differences in perception, culture, and emotionality within the same agent architecture.

**Keywords**
affordance theory, software agents, culture, stress, emotion, crowd models, emergence

# Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects

*Jason B. Cornwell, Kevin O'Brien, Barry G. Silverman*
Ackoff Center for the Advancement of Systems Approaches (ACASA)
University of Pennsylvania, Towne 229c
Philadelphia, PA 19104-6315
barryg@seas.upenn.edu

*Jozsef A. Toth*
Institute for Defense Analyses
4850 Mark Center Drive, Room 3605
Alexandria, VA 22311-1882
703-578-2867, FAX 703-931-7792
jtoth@ida.org

Keywords:
Affordance Theory, software agents, culture, stress, emotion, crowd models, emergence

**ABSTRACT:** *This paper describes an effort to revise the PMFserv agent architecture in order to implement J. J. Gibson's Affordance Theory. The theoretical justification for this revision is outlined along with the engineering constraints that inspired it. We describe the resulting architectural changes and the impact of those changes on the flexibility, ease of rapid scenario creation, and ability to reuse previous investments in knowledge engineering offered by our architecture. The level of effort required to build a new scenario within PMFserv both with and without the revisions suggested by Affordance Theory is compared. We conclude that Affordance Theory is an elegant solution to the problem of providing both rapid scenario development and the simulation of individual differences in perception, culture, and emotionality within the same agent architecture.*

## 1. Affordance Theory

Affordance Theory (AT) stems from origins quite distinct from those of cognitive science, and its modern interpretation is oftentimes the subject of spirited debate. James J. Gibson [1] coined the term "affordance"—one of a few cornerstones central to his theory of ecological psychology and direct perception. At the time, although many sub-disciplines in cognitive science were proceeding well as individual areas of research, attempts to integrate and utilize them in working systems—particularly in time- and resource-limited environments—met with unsatisfactory or incoherent results. Increasingly complex mathematical approaches, including non-monotonic reasoning, required lengthy principled proofs that could not match the performance or capabilities observed in humans. In a typical model, perception and attention were assumed to have occurred and the emphasis was on the cognitive or mathematic manipulation of symbols. In conventional AI, anything considered relevant in the environment also had to be represented and maintained in the agent's mind in order to keep the truth maintenance essential to AI knowledge representations coherent. Whatever existed in the external world had to be replicated in the internal "world model" of the mind. By overlooking the processes of perception and attention, the modeled agent acquired all the information in the world, whether it was relevant or not, in order to sustain truth maintenance, and was riddled with problems (e.g., [2], [3], [4]).

In a complementary way, perception is all that Gibson seemed to want to write about—he strongly opposed what he referred to as the "computer bandwagon". Although supported by behavioral evidence, until the advent of situated action, ecological psychology was without a

computational component (although cybernetics certainly promoted ideas consistent with it).

Gibson's notion of the affordance is based on a more unified relationship between agent and environment—form and function—more in keeping with the Gestalt, or "transactional" view [5]. Thus, salient, functional features present in the environment lead to direct sensation, perception, and action by the agent. Any notion of information is functional or situational, involves both agent *and* environment, and the affordance-based relationship between the two. According to Gibson, we perceive objects in terms of the possibilities for action they offer, or *afford*, us. The contour and shape of a coffee mug handle *affords* grasping–lifting–drinking; a sidewalk *affords* locomotion in a general direction; the size, shape, placement, and features of the Windows OK and CANCEL buttons *afford* accepting or denying the current option, and so on.

In this fashion, cognitive science and ecological psychology (including AT) proceeded in these forms largely in isolation from each other [6] until the mid-1980s, when the two finally met in the setting of AI planning, robotics, and game playing. This union was referred to as "situated action". As a challenge, Agre and Chapman [7] deliberately chose the video game environment to stress the limited resources, quick decisions, and perceptual processing required for successful performance in that domain. In a sense, these early pioneers were on the outside of cognitive science and ecological psychology, and were able to recognize what one had to offer the other. With the sudden rearrangement of the roles of agent, environment, perception, and cognition, and a slight redefinition of "causality", "information", and "context", agents could now reason in arbitrarily complex environments.

In 1993, Vera and Simon [8] published a major critique of situated action, claiming the approach was simply a reification of conventional symbolic cognitive science. Their argument had merit, but was from the Newtonian-interactional perspective. In physics, this would be tantamount to a theorist attempting a Newtonian argument in Quantum terms, or vice versa. What Vera and Simon failed to explain is how Gibson's notion of the "stimulus flux", for example, is transformed into symbols, the stuff of cognition. Nor did they understand the transactional, bi-directional, acausal nature of the various perceptual and cognitive modules in situated action architectures versus the unidirectional causal flow of control and data in conventional symbolic architectures and the several instances thereof. They also failed to point out *why* situated action was required in the first place—i.e., the computational intractability intrinsic to symbolic architectures. Although Simon's notion of bounded rationality is still a powerful edifice in cognitive science and economics, instances of the theory realized by his students and others usually required *unbounded* resources when approached from the interactional perspective.

Since the formative years of situated action, the theory of affordances has broadened within cognitive science, which itself has evolved into an alternate approach we refer to as "Situated Cognitive Science" [9]. Given the requirements of an agent-based architecture, there are those aspects to which affordances pertain—namely, perceptual aspects—and others to which they do not—namely, symbolic manipulation. The current generation of agent architectures appears to favor the hybrid approach, combining when necessary performance moderator functions, pattern-matching subsystems, rule-based sub-systems, path-planning sub-systems, and so forth (see also Hawkins and Chattam, in these proceedings). For the purposes of modeling and simulation the application of affordance theory

is more of a solution to an engineering problem rather than a scientific problem.

In addition to the architecture advanced in this paper, the current generation of modeling and simulation systems and other environments such as *The Sims* (2002) and *AI-Implant* (Toth et al., Van Lent et al., in Hawkins and Chatam, in these proceedings) makes use of affordances, particularly sensory-perceptual affordances, and appear to be best suited for meeting the subtle demands of the environment, in the context of its immediate surroundings. A *Sims* modeler, who can even be a child (no Ph.D. required!), programs the environment, not the agents. The characters' behaviors are the result of the agents' affordances or attunements to the environment, not descriptions of what goes on in their minds (see Clancey 1997, Part I, for a discussion). The Soar cognitive architecture, on the other hand, which is a symbolic processing system, is better suited for short-, medium-, and longer-term reasoning in keeping with its longstanding tradition of manipulating post-perceptual information.

This new trend in modeling, simulation, and gaming, is very promising. The following sections will describe how AT, standard symbolic cognition, economic utility theory, culture, and emotions can be combined in new and innovative ways to take advantage of the best features from all of these approaches.

## 2. Why affordance theory?

Our interest in the use of Affordance Theory in multi-agent systems arose from engineering constraints rather than a theoretical predisposition. Agents created in earlier revisions of our architecture each contained a functional representation of every other object in its environment and made its decisions by consulting this internal schema. State information was passed between agents and the environment, but every decision made by an agent was grounded in its own internal data structures.

This is a common design for intelligent agents. It reflects
our understanding of human cognition as the manipulation of mental models [10]. If people each carry within themselves a functional representation of the objects in their environment – a robust mental model of the world maintained by perceptual information – shouldn't AI programs intended to traverse and manipulate a virtual landscape possess a similar structure? In a traditional strategic simulation involving perfectly rational actors, such a capacity is overkill. In a simulation that takes social dynamics into account, however, where individual points of view are significantly different and agents act with less than perfect knowledge of the world, such a capacity is a basic requirement. In order to begin to capture the subtleties of social interaction or simulate human emotionality, agents must act based on their own unique socio-cultural background and personal experience.

Take as an example an American helicopter flying over Mogadishu in Somalia in 1992. For some Somalis, the helicopter was a threatening menace that blew women's skirts above their knees and ripped babies from their arms. For Mohamed Fararah Addid's clansmen the helicopter contained their enemies and persecutors. For the US Rangers and Special Forces stationed in Somalia the helicopter was a means of transport and a source of support. As a result of these distinct representations a successful attack on the helicopter would result in distinct emotions for the different groups mentioned above. Onlookers might feel fear, joy, or anger as a result of the event, and would subsequently select their actions on the basis of those feelings. Each would tell a different story if asked to describe the event, because each would perceive the event through the lens of their own unique personal and social history.

Representing these different points of view is especially important in a military simulation, as without differing points of view there would be no conflict at all! For a simulation grounded in emotion simulating differences in point of view is an absolute requirement.

Unfortunately, representing those distinct points of view is a nightmare for developers, as the time and effort required to create an individual knowledge representation for each agent in such a system grows exponentially as additional agents and objects are added. Each agent contains a unique semantic markup of the world describing every perceived object in terms of the agent's own cultural and emotional history. To add a new object to that world, each and every agent would need to be revised to include this object and the actions available as a result of its presence into their individual semantic markup. With a simulation containing more than just a few agents, such a solution is untenable.

Affordance Theory offers an elegant solution to this problem. If the semantic markup of the objects in the environment is contained within and broadcast by the objects themselves rather than the agents perceiving them, then agents and objects can be added independently. A simulation developer adding a new agent type to the system need not worry about what agents or objects are already instantiated. The objects in the simulation will broadcast their affordances, or the actions that they afford to the agent in combination with some measure of the anticipated results of those actions, to any new agent, allowing it to manipulate them with no a priori knowledge of that object whatsoever.

Affordances cannot be uniform for all agents. Each agent must still have a unique view of the objects in its environment. The affordance approach offers two possibilities for introducing individual differences in perception. The first is to have multiple perceptual types for each object, accompanied by perception rules that determine which type is active for any given agent. For example, the helicopter object might contain a rule set that tests the allegiance of the agent perceiving it. If the perceiving agent is an American, it will be perceived as a friendly helicopter. If the perceiving agent is a member of Addid's clan, it will be perceived as an enemy helicopter. These perceptual types will provide different actions and anticipated effects. For an American the helicopter might offer extraction from a dangerous situation, whereas for the Somali it might offer a ready target. The second possibility is to provide some mechanism in each agent that will automatically modify or interpret the affordance according to some property internal to the agent. For example, an agent system might be devised that categorized actions in terms of certain central goals. If a helicopter object affords the action "attack," this action might be defined in terms of its respective success and failure on the opposing goals "Kill Americans" and "Protect Americans." The American agent will prefer actions that succeed at "Protect Americans" and fail at "Kill Americans" whereas the Somali agent will prefer actions with the opposite pattern of results. Despite a uniform affordance, the agents interpret the actions available to them differently based on internal data.

We feel that the best approach is a union of these two possibilities. Each perceivable object (including agents) should contain a variety of perceptual types representing fundamentally different perceptions of that object. Concurrently, each agent should contain a system for interpreting the actions afforded by each object according to its own properties.

## 3. Methodology

The knowledge engineering necessitated by this approach is by no means trivial, though it is far, far less time-intensive than would be required for an emotional agent system without affordances. Every agent and object must be

wrapped in a semantic markup that fully specifies the available perceptual types, the perception rules, and the set of affordances belonging to each perceptual type. Furthermore, agents must be populated with parameters or properties through which they can create individualized responses to common afforded options. Much of the knowledge engineering required to build an emotional agent system without using affordances is still required for an affordance-based approach. The benefits come when a simulation developer needs to add a new agent, add or remove an object, or change the behavior of an agent. Furthermore, an affordance-based system will benefit from the creation of libraries of pre-constructed agents and objects. Using such a library it should be possible to build a new scenario extremely quickly. By facilitating the reuse of agents and objects, the creation of a new scenario can be accomplished with a minimum of additional programming effort or knowledge engineering.

It should be noted that our architecture describes decision-making only. Once a decision is made, action execution is simulation-specific. This approach should therefore be theoretically applicable to any simulation in which human or agent decision-making plays a role. The granularity of the decisions is of no consequence, assuming that the actions under consideration can be wrapped according to our specifications. In fact, multiple levels of granularity could co-exist in the same system. For example, a commander agent might make strategic decisions about troop placement while those under his command decide how to act given his orders.

### 3.1 Scenario Design

The following section will outline a basic methodology for designing a scenario from scratch in an affordance-based agent architecture. This methodology should apply irrespective of the simulation environment in which it is deployed and the level of granularity at which decisions are being made by the agents. In keeping with our Mogadishu examples above, we will use an example mini-scenario loosely drawn from Mark Bowden's book Black Hawk Down. In this scenario, a helicopter has crashed in Mogadishu leaving one surviving pilot. A crowd, comprised of armed militia and civilians, is advancing on the crash site.

**Step1:** Create Agent/Object Inventory -- Generating a complete list of agents and objects is crucial, as an agent will only be able to perform an action if that action is made available to it by an object in the scenario. The inventory should therefore contain not only physical objects but also composite (e.g. crowds, squads, etc) and conceptual (e.g. orders that can be followed, etc) objects as called for by the specific demands of the scenario. If objects can be in different physical states over the course of the scenario, and those different states afford completely different opportunities for action, then they might be represented by multiple objects that replace each other when the state changes. For example, a dead body should be a different object than its living counterpart.

As our example scenario is simple and only loosely based on actual events, the inventory of objects and agents is quite short. We will need at least the following objects:

`Crashed_Helicopter`, `Pilot`, `Dead_Pilot`, `Militia_Member`, `Dead_Militia_Member`, `Civilian_Man`, `Dead_Civilian_Man`, `Civilian_Woman`, `Dead_Civilian_Woman`, `Building`, `Rubble`, `Rifle`, `Pistol`, and `Grenade`.

Our example is simple enough to avoid the need for conceptual objects, but we might have added

an orders object for the `Militia_Member` agents that specified that the pilot should be captured alive and held as a hostage.

**Step 2:** Identify Perceptual Type for Each Object/Agent--Each simulated object should contain a set of perceptual types that describe the variety of perceptions of that object available to other agents. The list of perceptual types reflects every distinct manner in which an object can be perceived, so it is quite important to specify a full list.

For our example scenario, we will examine just two objects: `Pilot` and `Civilian_Woman`. The pilot object could be perceived by a sympathetic Somali as a `friendly_american_soldier`, by other Somalis as a `hostile_foreign_soldier`, by militia members as an `enemy_soldier`, by some militia members as a `potential_hostage`, or by a fellow American as a `compatriot`. Dependent on the status of `Pilot`, he might also be perceived as a `combatant_enemy`, or as an `unarmed_american_soldier`, `etc.`

One of the issues that American soldiers faced in Somalia was the frequent use of women and children as human shields by Somali combatants. The `Civilian_Woman` object might therefore include a set of perceptual types that represent this behavior in the simulation. An American soldier might view `Civilian_Woman` as an `innocent_bystander`, a `human_shield`, or an `enemy_combatant` depending on her current activity. Other agents might perceive her as a `friend`, a `woman`, a `potential_human_shield`, or an `active_human_shield`.

**Step 3**: Codify Afforded Actions -- Each perceptual type for any given object should offer a set of possible actions and the results of those actions anticipated by the agent perceiving that object. These anticipated results need to be presented in a format that can be readily understood and interpreted by all agents in the simulation. For example, results might be presented in terms of the degree to which they satisfy a predetermined set of needs common to all agents (though not necessarily weighted equally by all agents). For the purposes of this general explanation we will assume that our agents understand the results of actions in terms of a series of goals: safety, self-esteem, and glory. Each of the anticipated results should be described in terms of these three goals.

Continuing with our earlier example, the `Pilot` object, we can define the actions available for the `hostile_foreign_soldier`, `enemy_soldier`, and `unarmed_hostile_foreign_soldier` perceptual types. The `hostile_foreign_soldier` perceptual type represents the perception of those that oppose the American presence but are not actively fighting against it. The available actions could be `Pilot.hostile_foreign_soldier.attack()`, which might fail considerably on safety and succeed a small amount on glory, and `Pilot.hostile_foreign_soldier.flee_from()`, which might succeed on safety and fail on glory and self esteem. The `unarmed_hostile_foreign_soldier` actions would look similar: `Pilot.unarmed_hostile_foreign_soldier .attack()` might fail only a small amount on safety but succeed considerably on self esteem, while `Pilot.unarmed_hostile_foreign_soldier.taunt()` might only succeed a little bit on esteem. `Pilot.enemy_soldier.attack()` might succeed on glory and self esteem but fail on safety, while `Pilot.enemy _soldier.flee_from()` might succeed on safety but fail considerably on glory and self esteem.

An agent who sees the `Pilot` as an `enemy_soldier` and an agent who sees the `Pilot` as an `hostile_foreign_soldier` would therefore both have the option to flee from the `Pilot`, though they would attribute a different pattern of goal success and failure to that action. Even agents who saw the `Pilot` as the same perceptual type might have different responses to the same pattern of goal successes and failures.

The choices made in the process of knowledge-engineering an affordance-based agent system should reflect as accurately and completely as possible the various perceptions of each object that real people would experience - and the variety of choices that real people would encounter - were they in the scenario themselves. We recommend extensive literature review and the consultation of SMEs (subject matter experts) as a starting point for developing the semantic markup of the agents and objects in the world. The final list, though hopefully inspired by valid data, will invariably be dictated in large measure by common sense and by the perceptual types necessary to support specific behaviors that scenario developers may want to simulate in their systems.

## 4. Case Study – PMFserv Scenarios Before AT and After AT

PMFserv was conceived as a software product that would expose a large library of well established and data-grounded Performance Moderator Functions (PMFs) and Human Behavior Representations (HBRs) for use by cognitive architectures deployed in a variety of simulation environments. Its principal feature has been and continues to be a model of decision-making based on emotional subjective utility constrained by stress and physiology [13]. PMFserv quickly grew to become an agent architecture in its own right – with the flexibility to either act as a meta-level emotional arbitrator for others' cognitive architectures or provide a fully functional stand-alone system to simulate human decision making.

### 4.1 PMFserv Architecture

PMFserv is built around a blackboard data structure loosely corresponding to a short-term or working memory system. Modular PMF subsystems manipulate data contained both in the blackboard and in a long-term memory store. Information is layered on the blackboard such that each layer is dependent on the layers below it for a given decision cycle of the agent (see Figure 1).

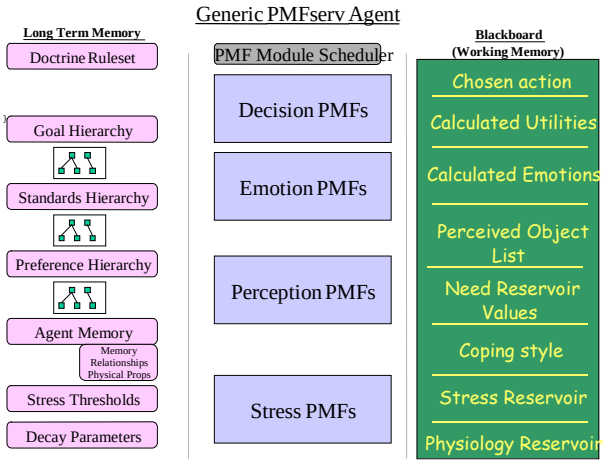

Figure 1 – PMFserv Overview

Walking up the blackboard from the bottom up reveals the decision cycle of a single agent. Physiological data across a range of measures (including arousal, fatigue, hunger, thirst, injury, etc) are combined to set the levels of a series of stress reservoirs. The stress reservoirs then determine the agent's coping style (a measure of the agent's current awareness and capacity for rational thought) for the current decision cycle. Need reservoirs corresponding to the degree to which the agent has satisfied the needs outlined by Maslow [11] are set based on any action that might have occurred in between decision cycles.

At this point Affordance Theory begins to play a prominent role in the decision cycle. Based on the agent's coping style, physiology, and any

memory elements that might have been created prior to the current cycle, each object in the system executes its perception rules to determine which objects are currently perceivable by the agent and generates a list of the perceptual types and corresponding affordances currently available to the agent.

These affordances are represented in terms of the agent's emotion model. Our emotion model, abbreviated here as OCC in homage to the trio of psychologists Ortony, Clure, and Collins [12] who proposed it, is described in great detail in numerous other papers (available at http://www.seas.upenn.edu/~barryg) so our discussion here will be brief. The general idea is that an agent posses three hierarchical concern trees that describe the agent's *Goals for Action*, *Standards for Behavior*, and *Preferences for People, Objects, and Situations*, respectively. An action can be represented by a series of successes and failures on the sub-nodes of these three trees. Each sub-goal is given a weight that describes how much it contributes to its parent node. To determine the emotional utility of an action, the OCC model multiplies the degree of success and failure of each node up the trees. From the top nodes on each tree, 11 pairs of oppositely valenced emotions are generated. By summing those emotions we arrive at a utility value for the action under consideration. This process is completed for every afforded action available to the agent. The action with the highest utility value is chosen and executed.

The OCC model allows for a common set of *Goals*, *Standards*, and *Preference* trees whose structure is shared by all agents. The weights, however, are unique for each agent. We are therefore able to give each agent a robust and individual worldview. We have tried to construct the trees to allow for the representation of as wide a variety as possible of cultural, ideological, and personal backgrounds as possible.

### 4.2 Scenarios Built Before Affordance Theory

Before the current set of revisions, our architecture looked significantly different. A full description of the original architecture is available in [13] and at our website. The basic functionality of the OCC model itself was not fundamentally different from its current incarnation, although each agent was given its own unique set of trees. An agent's decisions, however, were driven by internal Markov chains that represented every possible state of the world as far as the agent was concerned. Each state was tied to a set of successes or failures on its OCC tree nodes that provided a means to evaluate which states were preferable to others. As in the current system, the agent's physiology contributed to a stress value that determined the agents coping style. Rather than influence the agents' perception, however, coping style served to constrain how far into possible future states the agent would look in order to calculate emotional utility. Whenever one agent could affect the state of another agent, a forced state change would occur. Our agents had no capacity for anticipating such a forced change. They acted as though they were the lone actor in a static world.

A number of demonstration scenarios were built using this architecture. The earliest was an ambush at a school bus inspired by the fourth dream depicted in General Paul Gorman's *In Defense of Fombler's Ford*. The second simulated the target selection procedures of a Terrorist group. Shortly thereafter a number of scenarios were built that depicted crowd scenes. Each featured a crowd that had gathered to protest a social injustice. In one series of scenarios this injustice was a roadblock that kept people from going to work. In several others it was a protest outside of a prison. All of these scenarios featured similar characters: a group of protesters comprised of both females and males, employed and unemployed, a police

presence, a group of onlookers, and an instigator or two trying to rouse the crowd.

It was during the development of these scenarios that the limitations of our approach became apparent. Despite the fact that the characters involved in the scenes were virtually identical from scenario to scenario, they had to be built from scratch each time to accommodate new agents, objects, or behaviors. To produce meaningful, scenario-specific behavior the agents' Markov chains needed to include sets of states that could represent the specific, complex social dynamics present in each scenario. Furthermore, the OCC trees needed to be highly customized and tuned to produce believable behavior for any given Markov chain. While we were producing stand-alone, unique scenarios this was not a problem. As soon as we started to build related scenarios with shared characters, however, we realized that there was no way our architecture could support rapid scenario development or facilitate the editing of scenarios. Agents built with static Markov chains could not be easily edited. The addition of a new object or agent necessitated a revised Markov chain and OCC tree set for every existing agent in the scenario.

We came to the conclusion that the only way to design a rapidly composable system of social, emotional agents was to decouple all knowledge of the world from the agents' internal data structures and somehow generate that knowledge automatically at runtime. Affordance Theory was the obvious choice for doing so, as it allowed complete encapsulation of the knowledge necessary to manipulate objects in the world within those objects themselves. In essence, we are using Affordance Theory as a software engineering shortcut. Rather than build mental models on a per-agent basis we allow those models to be generated at runtime based on the objects present in the environment at the time. This allows us to build agents that can respond to very complex situations without having to painstakingly design those complexities into their Markov chains from the start.

Take, for example, the simple Mogadishu scenario that we began to sketch in our new architecture in Section 3. Before we moved to our current affordance-based approach, developing a similar scenario would have involved the following effort:

**Step 1:** Generate a list of all agents, objects, and events involved in the scenario. -- The level of effort here was equivalent to the level of effort required using the affordance-based approach, assuming that there are no pre-existing agents or objects that we could pull from a library or otherwise reuse.

**Step 2:** For each agent type, develop a complete Markov chain describing the agent's possible states and valid state transitions -- Because events or actions were described in terms of the actor rather than the recipient of the action, there was a greater tendency to oversimplify scenarios and omit critical events, which necessitated lengthy revisions. As a result the process of crafting Markov chains was iterative and tedious. It was also quite limiting. Events could only occur in certain sequences pre-specified by the Markov chain. If that event did not unfold as planned, a revision was necessary for every agent whose states were affected by the event.

In our new system, we do not need to create a static state-space for each agent, as the agents' possible actions are determined at run-time by the objects in the environment. The available actions are coded into each object only once rather than repeatedly describing all actions available to each agent. Very little needs to be "pre-scripted" so the emergence of a far greater range of behaviors and outcomes is possible. Because the development cycle of an agent ceases to be an iterative, trial-and-error process,

and because actions are coded on a per-object rather than a per-agent basis, the time and effort required for this step is vastly reduced in our new architecture.

**Step 3:** Design unique concern trees that correspond to each agent's Markov chain --In the old system a new tree needed to be constructed for every agent. Each agent then required its weights to be set and validated. This process was repeated every time a significant change was made to the Markov chain to accommodate new possibilities for action.

In the current system the trees themselves are static. We developed a common concern set that all agents possess. Only the weights need to be set on a per-agent basis. Though the resulting concern trees are significantly larger than their predecessors, they provide a systematic framework for describing individual differences in culture and belief systems between agents that maintains the ability to produce unique agents present in our earlier architecture. As a result, each agent or agent type developed under our new architecture requires merely a new set of weights, not an entirely new tree. Because the tree structure itself requires no revision or tweaking during scenario development the effort required for this step is significantly reduced in our new architecture.

**Step 4**: Implement the execution of all events in code along with forced state transitions -- The effort required here is equivalent. The simulation environment, not PMFserv, handles the execution of actions. In our earlier demos we built our own simple simulation environment for analysis and testing, but we are now working to integrate our agent architecture with external simulation systems that handle action execution themselves.

In summary, then, the effort required to develop a scenario has been drastically reduced by the implementation of affordance theory within our architecture. A certain level of subject matter research and knowledge engineering will always be required to develop valid, complex agent scenarios. However, our new approach eliminates redundant knowledge structures and greatly facilitates the reuse of previously implemented and validated agents and objects.

### 4.3 Subsequent Work

To help us better understand the knowledge engineering demands imposed by our new system, we enlisted the help of a group of Systems Engineering students who were looking for a project that would fulfill their senior design requirement. These students spent a semester designing a scenario that could be implemented in our system, though the system itself is still under development. This included an extensive literature review of the events depicted in the book "Black Hawk Down." The students were asked to extract several key agent types, derive weights for the standards, goals, and preference trees along with documented justification for those weights, and construct a limited ontology of objects in their scenario along with their perceptual types and affordances. By observing their effort we gained some insight into the level of effort that will be required to build up a large library of validated agents and object presets.

The very fact that we were able to "outsource" the development of a scenario to a student team demonstrates a major advantage of Affordance Theory. In our previous architecture the development cycle was iterative and confusing as a result of the interdependencies between agents, objects, and their environment. Students were not able to create scenarios in this system without quite a bit of hand-holding. Because the objects now contain their own data, however, scenario components can be

developed in relative isolation from each other without having an impact on the overall scenario development.

## 5. Conclusion

We are currently working to provide a standards-based interface that would allow PMFserv to drive the behavior of agents situated in a variety of simulation and game environments. We envision PMFserv as a multipurpose toolkit from which simulation developers will be able to either drag-and-drop agent minds onto the agent bodies in their simulations or use specific PMF components as needed to moderate the behavior of their own cognitive sub-systems.

The initial test bed for this effort is a joint project with a group at ICT. We are developing a hybrid architecture that uses PMFserv to moderate decisions made by SOAR agents within the Unreal game engine. AI-Implant is being used to manage art resources and provide low-level implementations of actions that can be triggered by SOAR or PMFserv directly (e.g. navigation, movement, physical actions, etc.). By exploring ways of tying these systems together, we will increase our understanding of the requirements for integration significantly.

This demonstration will set the stage for future integration efforts with real-world simulation systems and provide valuable insight into the requirements for behavioral interchange standards that we will be able to share with others attempting similar efforts.

Revising our agent architecture to support AT was by no means a trivial effort. A fundamental redesign was required to include AT at every level. The work was well worth it, however. Without AT scenarios were more difficult to design and debug. We struggled and failed to find an elegant solution to provide rapid scenario development capabilities and the construction of libraries to facilitate agent re-use. As a result of our redesign, these capabilities have become a cornerstone of our architecture and our agents themselves are able to exhibit far more complex, emergent behaviors than was previously possible.

Affordance Theory is by no means a panacea for simulation developers, however. As a theory of cognition it is far from being accepted. Architectures that are designed to be unified theories of cognition unto themselves or that model expert performance (SOAR, ACT-R, etc) would be better off employing a perception model with a stronger physiological grounding. Conversely, AT would probably be overkill for most cellular automata or other artificial life simulations. For multi-agent systems that simulate the cognition and/or emotionality of individual agents, and that aspire to a high degree of reuse and rapid composability, however, AT is practically a requirement.

## Acknowledgement

The authors would like to acknowledge the financial support of The Defense Modeling and Simulation Office (DMSO).

## References

[1] J. J. Gibson: *The ecological approach to visual perception.* Boston: Houghton Mifflin, 1979.

[2] R. Brooks: “Intelligence without representation.” *Artificial Intelligence, 47,* 139-159, 1991.

[3] D. J. Chalmers, R. M. French, D. R. Hofstadter: “High-level Perception, Representation, and Analogy: A Critique of Artificial Intelligence Methodology.” *Journal of Experimental and Theoretical Artificial Intelligence, 4*(3), 185-211, 1992.

[4] J. A. Toth: “Book review. Kenneth M. and Patrick J. Hayes, eds., Reasoning Agents in

a Dynamic World : The Frame Problem." *Artificial Intelligence, 73*, 323–369, 1995.

[5] I. Altman, B. Rogoff: "World views in psychology and environmental psychology: Trait, interactional, organismic, and transactional perspectives," in I. Altman and D. Stokols (Eds), *Handbook of Environmental Psychology*. New York: Wiley, 1987.

[6] W. J. Clancey: *Situated Cognition: on Human Knowledge and Computer Representations.* Cambridge University Press, Cambridge, UK, 1997.

[7] P. E. Agre, D. Chapman: "Pengi: An implementation of a theory of activity." *Proceedings of the Sixth National Conference on Artificial Intelligence,* Seattle, 1987 : 196-201.

[8] A. H. Vera, H. A. Simon: Situated action: A symbolic interpretation. *Cognitive Science, 17*, 7–48, 1993.

[9] W. J. Clancey: *Conceptual coordination: How the mind orders experience in time*. Erlbaum, 1999.

[10] K. Craik: *The Nature of Explanation*. Cambridge: Cambridge University Press, . 1943.

[11] A. H. Maslow: *Motivation and Personality*, New York: Harper & Row, 1954.

[12] A. Ortony, G.L. Clore, A. Collins: The Cognitive Structure of Emotions, Cambridge University Press, Cambridge 1988.

[13] B. G. Silverman, J. B. Cornwell, K. O'Brien: "Progress to Date on the Human Performance Moderator Function Server (PMFserv) for Rapidly Generating Reusable Agents, Forces, and Crowds" Technical Report, Philadelphia, PA, Univ. of Penn/ACASA Technical Report. 2003. Available at http://www.acasa.upenn.edu/hbr.htm