2010

Smart Events and Primed Agents

Catherine Stocker
University of Pennsylvania, cstocker@seas.upenn.edu

Pengfei Huang
University of Pennsylvania, pengfei@seas.upenn.edu

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Recommended Citation

Stocker, C., Huang, P., & Badler, N. I. (2010). Smart Events and Primed Agents. *Lecture Notes in Computer Science: Intelligent Virtual Agents, 6356* 15-27. http://dx.doi.org/10.1007/978-3-642-15892-6_2

10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010.

Smart Events and Primed Agents

Abstract
We describe a new organization for virtual human responses to dynamically occurring events. In our approach behavioral responses are enumerated in the representation of the event itself. These Smart Events inform an agent of plausible actions to undertake. We additionally introduce the notion of agent priming, which is based on psychological concepts and further restricts and simplifies action choice. Priming facilitates multi-dimensional agents and in combination with Smart Events results in reasonable, contextual action selection without requiring complex reasoning engines or decision trees. This scheme burdens events with possible behavioral outcomes, reducing agent computation to evaluation of a case expression and (possibly) a probabilistic choice. We demonstrate this approach in a small group scenario of agents reacting to a fire emergency.

Keywords
smart events, primed agents, agent-based simulation

Disciplines
Artificial Intelligence and Robotics | Computer Sciences | Databases and Information Systems | Engineering | Graphics and Human Computer Interfaces | Software Engineering

Comments
10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010.

Smart Events and Primed Agents

Catherine Stocker[1], Libo Sun[2], Pengfei Huang[1], Wenhu Qin[3], Jan M. Allbeck[4], Norman I. Badler[1]

[1] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389
{cstocker, pengfei, badler}@seas.upenn.edu
[2] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
{irisgrace}@tju.edu.cn
[3] Department of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
{qinwenhu}@seu.edu.cn
[4] Department of Computer Science, George Mason University, Fairfax, VA 22030
{jallbeck}@gmu.edu

Abstract. We describe a new organization for virtual human responses to dynamically occurring events. In our approach behavioral responses are enumerated in the representation of the event itself. These *Smart Events* inform an agent of plausible actions to undertake. We additionally introduce the notion of agent *priming,* which is based on psychological concepts and further restricts and simplifies action choice. Priming facilitates multi-dimensional agents and in combination with Smart Events results in reasonable, contextual action selection without requiring complex reasoning engines or decision trees. This scheme burdens events with possible behavioral outcomes, reducing agent computation to evaluation of a case expression and (possibly) a probabilistic choice. We demonstrate this approach in a small group scenario of agents reacting to a fire emergency.

Keywords: Smart events, primed agents, agent-based simulation

1 Introduction

Real-time virtual human simulation has attracted considerable attention in recent years due to its applications in entertainment, education, architecture, training, urban engineering and virtual heritage. Often, spaces are populated with large groups of mostly homogeneous (though possibly visually differing) characters. Ideally, they would act as purposeful, functional individuals who enrich an environment, but it is difficult to keep the computational cost of intelligent agents low enough to simulate large populations. Most present simulations address scalability at the expense of expressivity by just animating walking pedestrians.

To keep the cost of agents low, simulations generally focus on emergent behaviors during collision avoidance. Alternatively, some methods center on agents with "heavy" reasoning, planning, or decision-making processers, and are too computationally intensive to be scalable to large context-dependent groups. Our aim is to simulate groups of differentiated, functional agents with context-dependent behaviors at a low computational cost.

In this paper, we propose a new organization for virtual human responses to dynamically occurring events that embeds agent behavior options into a "Smart Event-Primed Agent" model. This model supports a simple but powerful mechanism for behavior selection. The main cost of this approach is borne in the user authoring of an event's representation. Events are then stored in a database for easy re-use in varying scenarios. The cost to each agent at run-time is simple expression evaluation. We demonstrate this approach in a small fire scenario and show that it can not only produce realistic simulation of group behaviors, but is also scalable.

The paper is organized as follows: in the next section, we briefly review related work. Section 3 focuses on the details of the Smart Event model. In Section 4, we describe the Primed Agent model. Section 5 illustrates a fire scenario where a Smart Event influences a number of Primed Agents. We discuss our conclusions and future work in Section 6.

2 Background

In order to produce behaviorally interesting agents, simulations often take one of two approaches: navigation-based motion controllers or agent-based cognitive systems. Navigation-based motion controller approaches aim at achieving real-time simulation for very large crowds, thus the behavior of each individual is not as important as long as the overall crowd movement produces realistic emergent behavior. The focus is on locomotion and collision avoidance while maintaining appropriate velocities, motions and directions. Classically [1] this was done with social force models [2] , cellular automata models [3] , or rule-based models [4] . More recently a real-time, hybrid approach was proposed [5] with a dual representation for simulating agents as both a discrete and single continuous system.

Agents must navigate in order to get to places where they are needed or should perform actions. Navigation-based models focus on fast navigation but sacrifice individuality for scalability. Agent-based approaches, on the other hand, focus on the realism of individual behavior by simulating *choice* through cognitive functions such as perception, memory, planning and emotion in every agent. The most developed of these, the SOAR (State, Operator And Result) architecture [6] , attempts to construct general intelligence systems by implementing a variety of cognitive functions, specifically memory, behavioral and learning systems. CML (Cognitive Modeling Language) [7] specifies domain knowledge and requires characters individually determine how to fulfill goals by searching a situation tree for a set of appropriate actions. PMFServ [8] aims to create culturally valid agents by using performance moderator functions (PMFs) that span the functionality of perception, biology, personality, social interactions, decision making and expression. The goal of these simulations is to cause agents to react to events in specific and individual ways that indicate internal psychological processes. The drawback is that they are generally not scalable to large groups of agents.

The agent-based systems that simulate human cognition by imposing a heavy computational load on their agents are based on the assumption that humans are logical creatures who make thoughtful and rational evaluations before acting. This assumption is often incorrect. Emotions, instincts and phobias are all well known aspects of human personality that override rationality [9] [10] [11] . A less well known phenomena, but more pervasive in the automaticity of everyday interactions, is *priming*. Priming refers to the activation of conceptual knowledge structures by the current situational context [12] . This effect is considered a result of spreading neural activation, is an automatic, unconscious process, and affects both thought and behavior. Studies have demonstrated that priming can influence a wide array of behaviors from aggressiveness [13] to walking speed [12] to test performance [14] .

There already exists a small body of work combining cognition- and navigation-based systems. Shao and Terzopolous [15] proposed a model of autonomous pedestrians, each with their own perceptual, behavioral and cognitive system. The cognitive system is needs-based and relies on each agent evaluating a potentially large set of internal variables. Although this model works for a small number of events, in our opinion it is not scalable to a complex environment with many events, because each event could require multiple new needs be added to each agent's cognitive set. Each new need would then continuously need to be monitored by every agent.

Extending the work of [15] Yu and Terzopolous [16] introduced a decision network framework. Based on a combination of probabilities of internal traits and external observations of the world, the system uses a hierarchy of decision networks to reason and choose actions. Without a centralized point of information, all agents must reason about the ambiguous world and attempt to answer questions such as "is someone else seeking help?" Instead of requiring agents to individually keep track of other agents, we centralize the heaviest cognition into the Smart Event. We could also simulate the ambiguity of incomplete information, by assigning ambiguous behaviors without the burden of requiring multiple agents to reason about them. We believe that using probabilities to assign actions at the trait group level will be as realistic as assigning them using complicated decision trees in each agent. Our justification for this belief is that a group of individuals with appropriate and plausible *collective* behaviors will appear functionally realistic.

CAROSA (Crowds with Aleatoric, Reactive, Opportunistic and Scheduled Actions) [1] is a framework for creating and simulating functional, heterogeneous populations. Its aim is to allow a user to easily create simulations that contain virtual humans with assigned roles and appropriate, contextual behaviors. CAROSA was built on top of HiDAC [1] which provides navigation and motion control. As the name suggests, CAROSA includes a variety of actions that together result in behavior rich simulations. What it does not have is an event representation. We implement Smart Events and Primed Agents on top of CAROSA-based agents.

The contributions of our framework are: a Smart Event model that acts as a resource manager, assigning agent interactions and monitoring agent participation; a Primed Agents model based on human cognition that quickly selects the behaviors provided by the Smart Event without intensive reasoning; and a virtual human simulation model that strikes a balance between individualism and scalability while simplifying scenario authoring by allowing actions to be authored for sets of agents dependent on their traits, rather than individually coded for each agent.

3 Smart Events

Our approach centers the behavioral responses of agents in the representation of the event itself, analogous to the way Smart Objects [17] inform an agent of the actions needed to accomplish manipulations on itself. Smart Objects contain interaction information of various kinds: intrinsic properties, information on how to interact with them, functionality and expected agent behaviors. Similarly, Maim [18] proposed a spatial navigation graph annotated with semantic tags that trigger specific actions of virtual characters that cross that spot, such as looking into windows or entering a shop and subsequently leaving with bread. These features inspire the Smart Event model.

For trajectory planning and a basic agent model, we extend the CAROSA system [1] to include Primed Agents reacting to Smart Events based on a primed trait. The semantics of actions and objects in CAROSA are represented in PAR (Parameterized Action Representation) [19] . A PAR may specify either single, multiple or hierarchic actions, thus {"put out the fire"} may consist of {"obtain a hose", "walk to the fire", "spray fire"}. The action types available in CAROSA are:

- **Aleatoric actions** – Random but structured by choices, distributions, or parametric variations.
- **Reactive actions –** Triggered by context.
- **Opportunistic actions** – Response to agent needs and automatically scheduled based on priorities and context.
- **Scheduled actions** – Assigned by a user and triggered by the passing of time.

We now define the Smart Event by specifying its representation, evolution and communication with Primed Agents.

3.1 Event Definition and Representation

We define an event as any scheduled or external (by environmental factors or agents other than self) assertion (fact) inserted into or deleted from the world model. A Smart Event is an event represented by the following parameters:

- **Type -** The type of event, such as emergency, social, work, etc.
- **Position** - Map coordinates of the event
- **Location** - Object (such as room) that contains the event
- **Start time** - When the event begins
- **End time** - When the event ends (may be undefined if unknown)
- **Evolution** - A Finite State Machine that alters event state variables over time and in response to internal or external triggers
- **Influence region** - The region (physical or communication) affected
- **Participants** - Lists of which agents are involved
- **Event emergency level (eEL)** - Severity of the event
- **Corresponding actions** - Set of possible actions for agents to select from

3.2 Communication Between Events and Agents

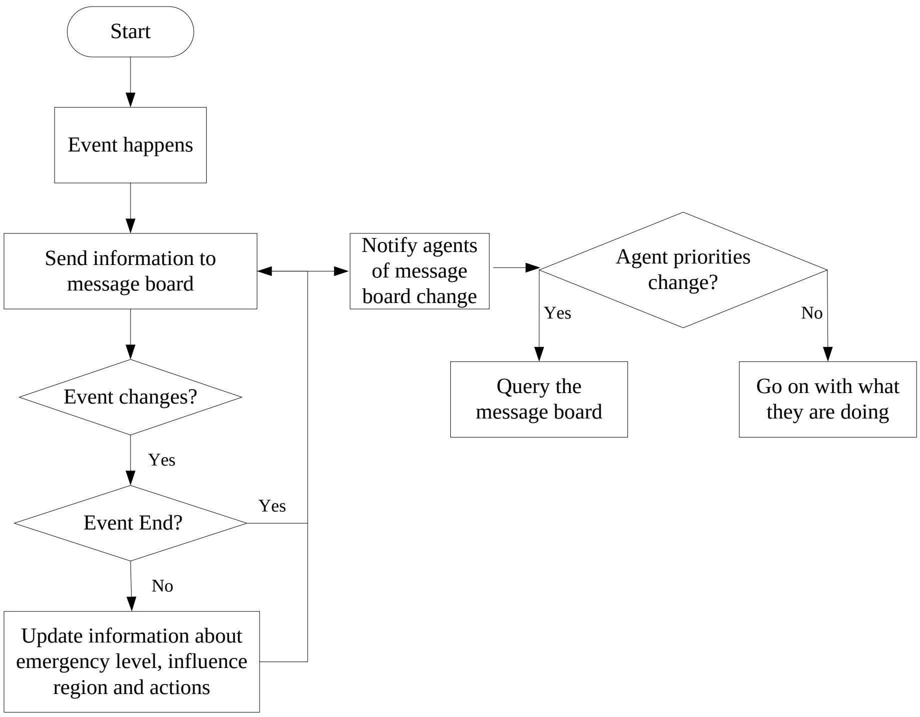

Fig. 1. Communication between the event, message board, and agent.

The communication between Smart Events and Primed Agents is moderated by a Message Board. The Message Board is responsible for broadcasting and updating relevant information for an evolving event. The process (Fig. 1) is as follows: when the event begins, its details are posted to the relevant Message Boards according to the event's *influence region;* the *influence region* determines the location-based Message Board to post to (e.g., if there was a fire in a school building, the event would be posted to the school building's Message Board) as well as any relevant communication-based Message Boards to post to (e.g., for the same school fire, the event would be posted to the associated firehouse's Message Board, among others).

Once an event is posted, a Message Board notifies a specified number of subscribers, according to its capacity limit, since not all agents are interested in or useful for that event. In this case, when the number of notified agents reaches the capacity threshold, the relevant message about that event is marked full. As a result, the rest of the agents do not need to see or respond to the event. Agents can be either *static* or *dynamic subscribers*. *Dynamic subscribers* subscribe when they enter the area overseen by the Message Board, e.g., they subscribe to the school Message Board when they enter the school or *physical radius*. *Static subscribers* are subscribed to a Message Board regardless of their location, e.g., a firefighter is always subscribed to the firehouse Message Board and thus event notifications on the firehouse's Message Board will be pushed out to him even if he is not at the firehouse: he is always within the *communication radius*. Using Message Boards, only relevant agents, according to *physical* and *communication radius*, are notified of events.

Agents can choose to respond, based on a very simple attention model: a comparison of their current *action's eEL* and the event's *eEL*. If agents find their current *action* to be less important than the event, they will acknowledge the event and query the Message Board for appropriate *actions* to perform. If they are "busy" (their *eEL* is > the event's *eEL*), they will ignore the event and continue what they were doing before being interrupted. When agents either run out of assigned *actions* becoming "idle", or becomes "bored" (i.e., their *eEL* falls below a threshold), or if a specified amount of time has elapsed, they will check the Message Boards they subscribe to in order to find new events. Until then, when "idle", they will perform a default *action*, as specified by their CAROSA defined roles. We see this communication system as analogous to an email/text message/voicemail system, in which people are notified of events but may not have the ability to learn about them or attend to them until they are free from obligation and can check their messages.

3.3 Event Evolution

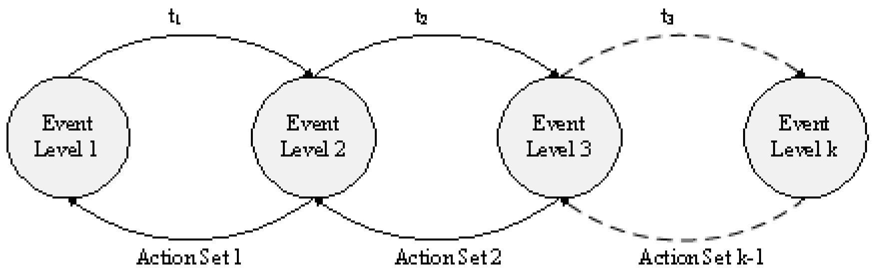

Fig. 2. The relationship between event evolution, passing *time* (*t*) and performed sets of *actions*

Smart Events have the ability to change and evolve as time passes. The corresponding agent *actions* should also change to reflect their awareness and understanding of the evolving event. As shown in Fig. 2, we can use a finite state machine to model and modify event evolution based on *time* and *actions* performed by agents. In every timestep eEL is computed as a function of time and the actions of the agents involved.

As the event evolves, it will notify the Message Board of updates to its state, such as changes in the *eEL*, the *influence region* or the corresponding agent *actions*; finally when the event is over, it will notify the Message Board, which will push the information out to relevant agents before removing the event from itself. The frequency of updating information on the Message Board is a function of the evolution of the event, specifically the Δt specified in Fig. 2.

In the next section we discuss the final efficiency gain of this Smart Event-Primed Agent architecture: agent priming.

4 Primed Agents

The psychology literature clearly documents that people can be "primed" to activate one self-concept over others, and this priming can affect their resultant behavior. The dynamic constructivist view of culture [20] claims that this *frame switching* occurs when discrete constructs (categories, theories, stereotypes, schemas, etc.) of cultural self-concepts are brought to the forefront of an individual's mind in response to cues such as language and context. Individuals may have multiple networks of constructs and even contain conflicting constructs, as long as only one is activated or primed at a time.

One important way these constructs are formed as part of the self-concept is based on inclusion in a social group. Deaux et al. [21] performed a cluster analysis on trait property ratings to identify six distinct categories of social identity: relationships, vocation/avocation, political affiliation, stigma and ethnicity/religion. We base our trait model on these 6 groups, but make slight modifications. Based on the numerous studies of priming on gender and ethnicity we believed those two traits deserved their own group – thus we pulled gender out of the relationship group and split ethnicity/religion into ethnicity and religion. In addition, we felt age was an important category to include in a simulation that may require strenuous movement – thus we pulled it out into its own group also. We extend the theory of dynamic constructivism to simulating realistic individual *actions* based on the priming of one of these traits at a time. These factors are demonstrated with representative values in Table 1.

Table 1: Traits and their assigned values

Trait	Possible Values
Age	Child, adult, elder
Gender	Male, female
Ethnicity	American, European, Asian, African, Australian, Hispanic
Religion	Christian, Jewish, Muslim, Hindu, Buddhist
Vocation	Firefighter, policeman, teacher, student
Relational	Mother/father, daughter/son, husband/wife, friend/stranger
Stigma	Smoker, homeless, deaf
Political Affiliation	Democrat, Republican

Every agent may be assigned values for any set of the *traits* in the table. Agents may also possess multiples of any *trait*, such as having $relational_1$ = *mother* and $relational_2$ = *wife*. In addition, some of the *traits* contain secondary tags with additional information. For $relational_1$ = *mother* there may be two secondary tags that specifically identify the agent's daughter and son.

Any of the *traits* can be brought to the forefront (i.e., primed). For simplicity in the examples that follow, priming is restricted to occurring only upon entering a location or interacting with another agent. It can, of course, be extended to any sort of interaction that a system may allow: reading, viewing, hearing – anything that may cause a trait to come to the forefront of the agent's self-concept. Additionally, priming is restricted to activating only one self concept at a time, paralleling the way it is activated according to the psychology literature. Determining a priority algorithm for priming of two or more traits is left for future work.

We give an example of priming in pseudocode below. This example is restricted to defining priming situations that are used in the Fire Event example in the next section:

```
function enter( location )
  if( location==myWorkplace )
         prime( myVocation )
  else if( location==myHome )
         prime( myMainRelationalStatus )

function interact( otherAgent )
  if( otherAgent==myParent )
         prime( myChildRelationalStatus )
  else if( otherAgent==myChild )
         prime( myParentalRelationalStatus )
  else if( otherAgent==myCoworker )
         prime( myVocation )
  else if( otherAgent.prime==age )
         prime( myAge )
```

Thus, if an agent has an *age trait*=*child* and they begin talking to someone who is primed as an *adult*, their childishness will come forward. Or, if an agent's *vocation trait* is firefighter, with a secondary tag *myWorkplace* = *firehouse*, he will be primed as a firefighter when he enters a firehouse. If his *vocation* is not firefighter, there is simply no priming when walking into the firehouse.

After an agent has been notified of a new event, or has run out of "interesting" actions to perform (based on becoming idle or bored as explained in 3.2), he will query the Message Board(s) to obtain possible actions to execute. Action choices over the set of possibilities are made based on the trait with which he is currently primed. We will demonstrate this further in the Fire Event example.

5 Fire Event Example

To illustrate the architecture we will construct a scenario of a Fire Event (FE) occurring inside a school. We represent the FE using a subset of possible agent *traits* because not all *traits* make sense as influences on behavior during every event. The *traits* we utilize are *age*, *vocation* and *relational*. The instantiated values for these traits are: child, adult, elder; firefighter, policeman, teacher, student; mother, father, daughter and son, respectively. Table 2 shows all functions of priming as well as specified *actions* during the evolving FE. Fig. 3 shows the FE as a finite state machine.

Recall that priming can occur when agents interact with each other and is based on either a secondary *relational* tag (e.g. myChild) or a primed *trait* in the other agent (e.g. age). It can also occur when agents enter a location or event radius that matches a secondary location tag (e.g. myWorkplace).

Table 2. Priming and *actions* chosen during the evolving Fire Event (FE) example.
*Note, neither firefighters nor policemen would have been called during the small and medium states of this event, but if they are within the *location* radius, they should still respond appropriately.

Trait	**Value**	**Priming**	**Action selected during Fire Event:** Small	Medium	Large
Age	Child	interact(age)	stare	calmFollow	panickedFollow
	Adult	interact(age)	pourWater smother	extinguisher	leadAway
	Elder	interact(age)	pourWater smother	callPolice leadAway	calmFollow
Vocation	Firefighter	enter(firehouse) enter(emergency)	smother*	extinguisher*	fightFire
	Policeman	enter(station) enter(emergency)	smother*	extinguisher*	manageCrowd
	Teacher	enter(classroom) interact(student)	pourWater smother	extinguisher	leadAway
	Student	enter(classroom) interact(teacher)	stare	calmFollow	calmFollow
Relational	Mother/ Father	enter(home) interact(myChild)	pourWater smother	leadAway	leadAway
	Daughter/ Son	enter(home) interact(myParent)	stare	calmFollow	panickedFollow
Default			pourWater smother	extinguish	leadAway

The beginning state of the world is:

- Agent 1 - walked into classroom, primed as teacher
- Agent 2 - walked into classroom, primed as student
- Agent 3 - bringing son to school, primed as mother
- Agent 4 - coming to school with his parent, primed as son
- Agent 5 - at firehouse, primed as firefighter
- Agent 6 - at police station, primed as policeman

The fire starts on the floor of the building where Agents 1-4 are located (note, we only use six agents here for simplicity, but any or all of these agents could be thought of as a set of agents primed with the specified trait). Because they are within the *influence region*, they are all notified of the FE by the school building Message Board. All acknowledge the FE because the *eEL* is higher than the *action* they were performing, and all query the Message Board to obtain appropriate behaviors. Agents 1 and 3 are assigned an *action* that is an aleatoric choice between pouring water on the fire and smothering the fire (with a .5 probability for each). Agent 1's choice is to pour water, and the action specifies the preparatory *action* of going to the nearest bathroom to obtain water (note that a feature of CAROSA and PARs are that actions can be composed of sub-actions, including preparatory actions; *thus these actions need not be explicitly requested by the event*). Agent 3's choice is to smother the fire, which has a preparatory *action* of obtaining a towel or blanket. Agent 2 and 4, primed as a student and son respectively, are transfixed by the small fire as they are assigned the *action* of staring. Agents 5 and 6 are not within the physical or communication *influence region* at this stage of the event because the fire is small, and thus are not notified.

If Agents 1 or 3 were able to complete either of their *actions* in time, the fire will change its state to *out*, but let's assume they took longer than the specified *time* to do this because the preparatory actions took a long time to complete. The state is updated to *medium* on the Message Board after 1 minute. This is reflected in the Message Board by increasing the *eEL*, altering the specified *actions* of agents and, because an alarm is set off, now includes firefighters and police as part of the communication *influence region*. Agent 1 is notified by the Message Board of a change and because the *eEL* has risen, aborts the *action* of obtaining water in order to perform the newly specified *action* for a primed teacher: obtain a fire extinguisher. Agent 3, for the same reasons and primed as a parent, is assigned the *action* of leading people away. Agents 2 and 4 modify their *actions* to perform a calm following of the nearest leader, in this case, Agent 3.

If Agent 1 was able to complete the action in time, the FE would evolve back to a *small* fire that could be smothered, but let's assume he took longer and it has grown to a *large* fire. Let's also assume Agents 2-4 have not yet moved out of the *influence region* and that Agents 5 and 6 have just arrived. Agent 1 and 3 now share responsibility for leading people away. Agent 2, primed as an obedient student, continues to follow calmly, but Agent 3, primed as a child, panics and begins to perform panicked following *actions* such as crying and pushing. Agent 5, primed as a firefighter, begins to fight the fire with a fire hose, while the primed policeman, Agent 6, manages the crowd by helping agents find the exit and then blocking anyone from entering. The fire will eventually end, either because the actions are performed repeatedly, or because the FE has lasted too long and will destroy the building.

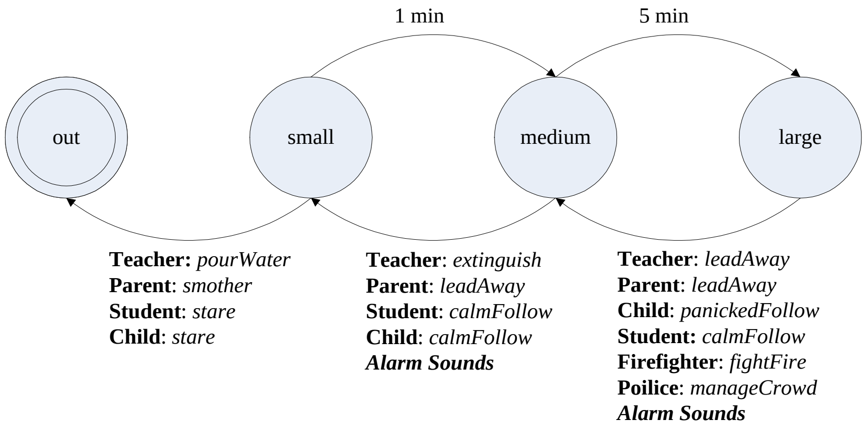

Fig. 3. The evolution of the Fire Event (FE)

6 Discussion and Conclusions

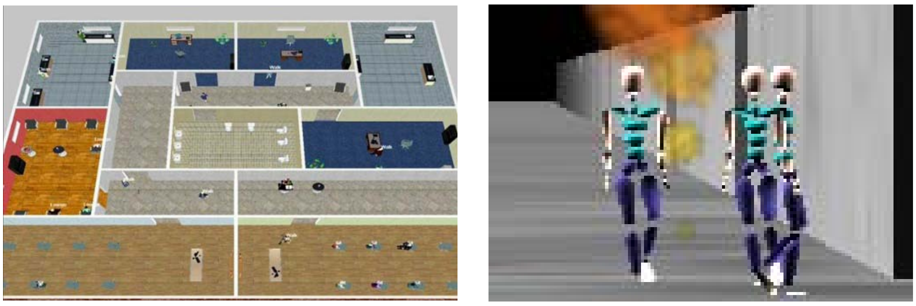

Fig. 4. A school building environment in the CAROSA framework (left) and agents responding to the fire event scenario (right)

We have proposed a "Smart Event - Primed Agents" model. The Smart Event model embeds agent behaviors into the event in order to simplify the process of action selection. It does this by avoiding deep reasoning, while allowing for realistic and diverse agent behavior that is modified by differences in agent traits. To avoid the use of large "heavy" decision methods to simulate a variation in individual differences, we introduce a Primed Agent model that selects appropriate *actions* based solely on the most recently activated *trait*, analogous to human priming. The limited reasoning performed by each agent at run-time makes our method scalable and suitable for the simulation of large groups of differentiated agents with context-dependent behaviors.

To evaluate the complexity of our system, we compare it first to a needs-based model [15] that must make *nm* evaluations every *t* timesteps, where *m* is the number of agents, *n* is the number of needs each has, and *t* is the number of timesteps agents wait between re-evaluating their needs. Although, based on the CAROSA architecture, our agents also make a small, constant number of biological need evaluations regularly (hunger, thirst, tiredness, i.e., *3m*), for external events we believe it is more natural and less computationally intensive for the incoming event to trigger most state updates. Thus, our agents perform very few monitoring actions regularly. They must only make internal state evaluations when they update their primed trait, are notified of an event, or decide to act on an event.

In addition, the state evaluation that occurs during priming is not dependent on the number of events that occur in the world, but instead on a constant number of psychologically-based human traits and meaningful associations (i.e., myWorkplace, myChild, etc.) and should thus remain relatively small in the face of a large number of events, unlike a needs-based approach which has a need assigned for every type of event (i.e., need to buy a ticket, watch a performance, etc.). Thus a large number of evaluations are moved to occurring only every $t_1 >> t$, where t_1 is the average time between priming and is much larger than *t*.

When notified of an event, we represent attention as a function with one simple evaluation: a comparison of the agent's current *eEL* to the event's *eEL*. Thus an agent must only make one comparison to decide whether or not to switch his *action*. In addition, because of the Message Board system and *influence region* specifications, we anticipate event notifications in a realistic simulation occurring no more frequently than every *t* timesteps, thus we believe that each agent may have to make 4 evaluations (3 biological and 1 *eEL* comparision) at most every *t* timesteps, which should be an improvement over *n*. Finally, upon becoming involved in an event, agents must only make another small evaluation because we have explicitly constrained the decision process to a simple evaluation of a few important, psychologically-based traits.

In decision network systems [16] a cognitive model aims to make deliberative agents that can exploit knowledge, reason about the world, and conceive and execute plans based on uncertainty. Decision networks require crafting the prior probabilities of each action in context and thus authoring behaviors has the potential to become very difficult in scenarios with a lot of actions and contexts. Our methods aim to remove the need for agents to deliberate over uncertainty.

Many agent-based systems [6] focus on reasoning, planning and goal decomposition, using preferences to make decisions between actions. Our system hypothesizes that this level of reasoning is much deeper than is necessary or realistic for an agent in everyday life situations. We use direct matching of event parameters with primed agent traits to facilitate a quick selection of individualistic behaviors from a plausible set.

The contributions of our framework are: a Smart Event model that acts as a resource manager, assigning agent interactions and monitoring agent participation; a Primed Agents model based on human cognition that quickly selects the behaviors provided by the Smart Event without intensive reasoning; and a virtual human simulation model that strikes a balance between individualism and scalability while simplifying scenario authoring by allowing actions to be authored for sets of agents dependent on their traits, rather than individually coded for each agent.

In the future we intend to extend our scenarios to include additional events. These events could be completely new, non-emergency events such a coffee machine that needs to be cleaned or they could be sub-events of the FE such as a person getting injured or a fire extinguisher becoming empty. We also intend to expand the system of agent priming. There are many psychological theories on priming and cultural frame switching that could be used to refine our models, and explicit data resources that can be used to create a specific population and load appropriate action sets for specific events. Finally, we hope to test our theory that our simple Smart Event and Primed Agent model produces agent behaviors that are no less reasonable than those generated by more computationally intensive decision-theoretic and planning approaches. To this end we take advantage of the extensive action representation framework extant in PAR and CAROSA to offload deliberative, aleatoric and preparatory actions. Animated scenarios are in development; agent graphics models will require robust action animations but this architecture will tell them what they should be doing when, where and how.

Acknowledgments. Partial support for this effort is gratefully acknowledged from the U.S. Army "SUBTLE" MURI W911NF-07-1-0216 and the Lockheed-Martin Corporation through a Strategic Technology Thread Grant. We also appreciate technology support from Autodesk, nVidia, and the Susquehanna International Group.

References

[1] Pelechano, N., Allbeck, J. Badler, N.: Virtual Crowds: Methods, Simulation, and Control. Morgan Claypool Publishers, San Rafael (2008).

[2] Helbing D., Farkas I., Vicsek T.: Simulating Dynamical Features of Escape Panic. Letters to Nature, 487-490 (2000)

[3] Chenney, S.: Flow Tiles. In: ACM SIGGRAPH/ Eurographics Proceedings of Symposium on Computer Animation, pp. 233-242 (2004)

[4] Reynolds, C.W.: Flocks, Herds and Schools: A Distributed Behavioral Model. ACM Computer Graphics, 21(4), 25-34 (1987)

[5] Narain, R., Golas, A., Curtis, S., Lin, M. C.: Aggregate Dynamics for Dense Crowd Simulation. In: ACM SIGGRAPH Asia 2009 Papers, pp. 1-8 (2009)

[6] Laird, J.E., Newell, A., Rosenbloom, P.: SOAR: An Architecture for General Intelligence. Artificial Intelligence, 33, 1-64 (1987)

[7] Funge, J., Tu, X., Terzopoulos, D.: Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. 29-38 (1999)

[8] Silverman, B.G., Johns, M., Cornwell, J., O'Brien, K.: Human Behavior Models for Agents in Simulators and Games. Presence: Teleoperators and Virtual Environments, 15(2), 139-162 (2006)

[9] Langer, E. J.: Rethinking the Role of Thought in Social Interaction. In: Harvey, J.H., Ickes, W., Kidd, R.F. (eds.) New Directions in Attribution Research, vol. 2, pp. 35-58. Erlbaum, Hillsdale (1978)

[10] Simon, H.: Models of Bounded Rationality. MIT Press, Cambridge (1982)

[11] Bargh, J.A, Chartrand, T.L.: The Unbearable Automaticity of Being. American Psychologist, 54(7), 462-479 (1999)

[12] Bargh, J.A., Chen, M., Burrows, L.: Automaticity of Social Behavior: Direct Effects of Trait Construct and Stereotype Activation on Action. Journal of Personality and Social Psychology, 71(2), 230-244 (1996)

[13] Carver, C. S., Ganellen, R. J., Froming, W. J., Chambers, W.: Modeling: An Analysis in Terms of Category Accessibility. Journal of Experimental Social Psychology, 19, 403-421 (1983)

[14] Shih, M., Pittinsky T.L., Ambady N.: Stereotype Susceptibility: Identity Salience and Shifts in Quantitative Performance. Psychological Science, 10, 80–83 (1999)

[15] Shao, W., Terzopoulos, D.: Autonomous pedestrians. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05) pp. 19-28 (2005)

[16] Yu, Q., Terzopoulos, D. : A Decision Network Framework for the Behavioral Animation of Virtual Humans. In: Proceedings of the 2007ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 119-128 (2007)

[17] Kallmann, M., Thalmann, D.: Direct 3D interaction with smart objects. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 124-130 (1999)

[18] Maïm, J., Haegler, S., Yersin, B., Mueller, P., Thalmann, D. and Van Gool, L.: Populating Ancient Pompeii with Crowds of Virtual Romans. In: Proceedings of the 8th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST’07), pp. 26-30 (2007)

[19] Badler, N.I., Allbeck J.M., Zhao L., Byun, M.: Representing and Parameterizing Agent Behaviors. In: Computer Animation, pp. 133-143 (2002)

[20] Hong, Y.-Y., Morris, M.W., Chiu, C.-Y., Benet-Martinez, V.: Multicultural Minds: A Dynamic Constructivist Approach to Culture and Cognition. American Psychologist. 55(7), 709-720 (2000)

[21] Deaux, K., Reid, A., Mizrahi, K., Ethier, K.A.: Parameters of Social Identity. Journal of Personality and Social Psychology, 68, 280-291 (1995).