Functional Crowds

Jan M. Allbeck
George Mason University
jallbeck@gmu.edu
http://cs.gmu.edu/˜jallbeck

Abstract

Most crowd simulation research either focuses on navigating characters through an environment while avoiding collisions or on simulating very large crowds. Our work focuses on creating populations that *inhabit* a space as opposed to passing through it. Characters exhibit behaviors that are typical for their setting. We term these populations *functional crowds*. A key element of this work is ensuring that the simulations are easy to create and modify. We use roles and groups to help specify behaviors, we use a parameterized representation to add the semantics of actions and objects, and we implemented four types of actions (i.e. scheduled, reactive, opportunistic, and aleatoric) to ensure rich, emergent behaviors.

Keywords: crowd simulation, intelligent behavior, autonomous agents

1 Introduction

Our research in virtual crowds focuses on *functional crowds*. These populations inhabit spaces as opposed to merely passing through them. Characters with roles are portrayed going about their daily activities, creating a tapestry of human activity for games, training simulations, architectural visualizations, or even the precursor for evacuation simulations. Through the CAROSA (Crowds with Aleatoric, Reactive, Opportunistic, and Scheduled, Actions) framework, we facilitate the authoring of simulations that have functional, heterogeneous crowds that are appropriate to time and place and have semantically meaningful interactions with the environment and other agents. To do so, we:

Figure 1: Screen shot of the university scene.

- Specify the characteristics (e.g. roles, goals, constraints) of individuals or groups including their behaviors and how they might differ from other individuals.
- Establish the temporal (e.g., daily) activities of such individuals or groups according to their occupations or roles.
- Access a library of parameterized animated behaviors that can be selected contextually, varied statistically, applied to agents, and executed in a simulation environment.
- Give the agents enough attention and perception to react to the environment and people around them.
- Link the framework to Commercial-Off-The-Self (COTS) software used for scheduling, enabling non-programmers to create simulations.

We incorporate four different broad action types into CAROSA: scheduled, reactive, opportunistic, and aleatoric. Scheduled activities arise from specified roles for individuals or groups; reactive actions are triggered by contextual events or environmental constraints; opportunistic actions arise from explicit goals and priorities; aleatoric actions are random but structured by choices, distributions, or parametric variations. The CAROSA architecture enables the specification and control of actions for more realistic populations in virtual worlds, links human characteristics and high level behaviors to animated graphical depictions, and relieves some of the burden in creating and animating heterogeneous 3D animated human populations. Figure 1 depicts a floor of a university that we created. Character roles in this sample environment include, professors, students, researchers, administrators, housekeepers, and maintainers.

2 Background

Most crowd simulation research either focuses on navigating characters through an environment while avoiding collisions or on simulating very large a populations. There has been some work on creating richer behaviors including agent interactions [1, 2] though they do not focus on characters inhabiting a space. In [3], Yu and Terzopoulos used to decision networks to select actions resulting in more complex behaviors, but they required crafting the prior probabilities of each action in context.

Our work combines ideas from different disciplines to create a framework that enables non-programmers to easily create functional crowds. From more traditional artificial intelligence research, we have action and object representations that provide semantics and levels of detail. From embodied autonomous agents research, we include individual differences such as roles. From computer graphics and animation, we are extending crowd simulation research to create heterogeneous crowds that are responsive to context.

The CAROSA framework has been created on top of a high-density crowd simulator called HiDAC [4]. HiDAC provides functionality common to many crowd simulators as well as, among other things, extensions for high-density, dynamic scenarios, communicating navigation information between characters, and behaviors such as line formations and pushing.

3 Parameterized Action Representation

The Parameterized Action Representation, PAR, is an ontology for simple and complex physical behaviors [5]. It was designed as a natural language and animation intermediary and actually contains representations for both actions and objects. They are stored in two hierarchies in a MySQL database called the *Actionary* and come in uninstantiated and instantiated forms. Uninstantiated actions and objects contain general parameter information that will tend to be true independent of any specific scenario. For objects in particular, these uPARs define types, such as furniture or weapons or even rooms. Instantiated actions and objects include specific information, such as which object is to be *picked up* or the current position of a specific chair.

The details of PAR and the system that processes them are beyond the scope of this paper. For details, we refer the reader to [5]. We will highlight a couple of the features particularly useful for authoring and simulating functional crowds. First, PAR actions contain a field called *preparatory specifications* that provide a simple form of backward-chaining. Preparatory specifications are a list of condition, action pairs. The conditions must be true to perform the action, and the paired action can be performed to establish the condition. Many actions require the character to be co-located with an object participant. For example, picking up a cup requires the character to be near the cup. The preparatory specifications for *Pickup* include a condition for checking to see if the character is near the object to be picked up and if the condition is false, a locomotion action is performed to get the character close to the object. This means that a scenario author can concentrate on higher level behaviors and need not worry about these type of planning details. It also helps ensure that actions are performed in the proper context. Generally PARs capture the semantics of actions and objects, abstracting away details for scenario authors and allowing them to concentrate on the goals of their simulations.

The data driven nature of the PAR system also makes it more robust. The PARs themselves are designed to be general and reusable, but also the *intelligence* is encoded as data in parameters instead of complex, highly interconnected networks or rules that are expressed in code. This makes creating and modifying scenarios much easier and also facilitates integration with other software systems.

4 Action Types

To create richer, more realistic simulations in the CAROSA framework, we extended PAR to included different action types.

Scheduled actions are the type of actions one would put on their calendar (e.g. meetings, appointments, classes). These actions add structure to simulations and also help define character roles. In CAROSA, they can be assigned to individuals or groups of characters and can be primitive or complex actions. Authoring these actions simply requires specifying *who*, *what*, and *when*. *Where* can also be specified, but it is not required. A *Resource Manager* automatically creates and stores resources based on locations and object types as the environment file, which describes the geometry of the scene, is read. It is then used to allocate locations or objects to actions that require such parameters but have not had them specified by the scenario author. This helps ensure that actions are performed in a correct context while not taxing the author.

One of the goals of CAROSA was to enable non-programmers to author these functional crowds. Due to its wide use and presumed ease of use, we chose Microsoft Outlook© as a sample interface to CAROSA. Groups and roles can be defined through Outlook's *Contacts* and *Tasks* panels. The *Calendars* panel allows authors to schedule actions for virtual groups and individuals as easily as they would their real counterparts (See Figure 2). Clicking a button on a tool bar then stores all of the information in a MySQL database where it is accessed, processed, and simulated by CAROSA.

Reactive actions add life to simulations. They promote emergent behaviors that take away the

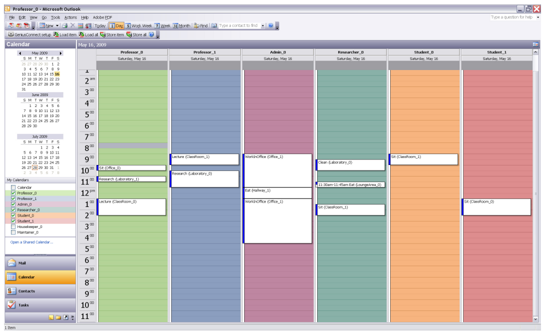

Figure 2: Using Microsoft Outlook© to author a simulation.

robotic feel that a simulation would have with scheduled actions alone. In CAROSA, reactions can be created for individuals, groups, or an entire population and can have stimuli that are individuals, groups, objects instances, object types, or properties. Work is in progress to add reactions to actions and events. Reactions are triggered through the collision detection mechanism of HiDAC and can suspend or preempt any actions the character may have been performing when the stimulus was seen.

Opportunistic actions add a structured unpredictability. These actions are based on needs. Needs are defined with decay rates. Fulfillments are defined as PAR actions with PAR object types (e.g. *Eat food*) and growth indicates. The priorities of opportunistic actions increase over time as the need becomes greater. Characters attempt to dynamically schedule these actions based on their existing scheduled actions and the distance it would take to get to a needed object from a path to the scheduled location. As an example, if a character is getting hungry and has a meeting scheduled soon, she will leave for the meeting a little early and stop by a location with food that is near to the path she travels to get to the meeting location. If a character has a need that becomes great enough, fulfilling it will preempt any other actions it is performing.

Aleatoric actions are essentially stochastic actions. They are composed of sub-actions with probabilities. Aleatoric actions and their sub-actions are both PARs. When an aleatoric action is executed, sub-actions are chosen according to their probabilities to fill the duration specified as a parameter of the PAR. The overall result is reasonable, dynamic behavior over a pe-

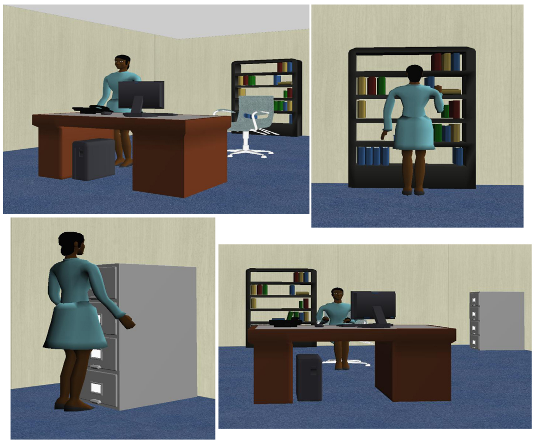

Figure 3: Aleatoric action of working in an office.

riod of time. For example, *WorkInOffice* is an aleatoric action composed of typing, talking on the phone, reading, filing papers, and watering plants (See Figure 3). These are actions one might see an individual working in an office perform throughout a day. The exact order and duration of each rarely matters, so aleatoric actions are a way of generating plausible behavior without taxing the simulation author with tediously specifying actions for every minute of the day.

5 Future Work

Additional details about CAROSA can be found in [6]. Extensions to CAROSA are on going. In the near future, we would like to add more individual differences, including personality [7] and culture. To gain better visuals and animations, we have nearly completed a port to the open source game engine, Ogre. We would like to have even richer, more intricate object interactions through inverse kinematics and spatial prepositions and richer, more intricate character interactions such as coordination, cooperation, and competition through team representations. We are also working on a multi-university project to instruct characters in real-time using a robust natural language interface. Finally, we are constructing additional scenarios including outdoor scenes and would like to run larger user studies to validate the ease of authoring simulation with CAROSA.

Acknowledgements

Partial support for this effort is gratefully acknowledged from the U.S. Army SUBTLE MURI W911NF-07-1-0216 , We also appreciate donations from Autodesk and nVidia.

References

[1] H. Yeh, S. Curtis, S. Patil, J. vanDenBerg, D. Manocha, and M. Lin. Composite agents. In *Proceedings of Symposium on Computer Animation*, pages 39–47. Eurographics Association, 2008.

[2] A. Lerner, E. Fitusi, Y. Chrysanthou, and D. Cohen-Or. Fitting behaviors to pedestrian simulations. In *Proceedings of Symposium on Computer Animation*, pages 199–208. ACM, 2009.

[3] Q. Yu and D. Terzopoulos. A decision network framework for the behavioral animation of virtual humans. In *Proceedings of Symposium on Computer Animation*, pages 119–128. Eurographics Association, 2007.

[4] N. Pelechano, J.M. Allbeck, and N.I. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of Symposium on Computer Animation*, pages 99–108. ACM Press, 2007.

[5] R. Bindiganavale, W. Schuler, J. Allbeck, N. Badler, A. Joshi, and M. Palmer. Dynamically altering agent behaviors using natural language instructions. In *Proceedings of Autonomous Agents*, pages 293–300. AAAI, 2000.

[6] N. Pelechano, J. Allbeck, and N. Badler. *Virtual Crowds: Methods, Simulation, and Control*. Synthesis Lectures on Computer Graphics and Animation. Morgan and Claypool Publishers, San Rafael, CA, 2008.

[7] F. Durupinar, J. Allbeck, N. Pelechano, and N. Badler. Creating crowd variation with the ocean personality model. In *Proceedings of Autonomous Agents and Multi-Agents Systems*, pages 1217–1220, 2008.