This paper appears in Proceedings of DSC '97 (Driving Simulation Conference), Lyon, France, September 8-9, 1997, pages 75–84

Embedding Scenarios in Ambient Traffic

Olivier Alloyer
Esmail Bonakdarian
James Cremer
Joseph Kearney
Peter Willemsen

Computer Science Department
University of Iowa
Iowa City, Iowa 52242 USA

Abstract

In this paper we present an approach to embedding scenarios in ambient traffic through the use of on-line casting, wherein vehicles are dynamically assigned scenario roles during simulation. We present a model of a scene as an abstraction that encapsulates the actions of principal role players, the background activity, and the setting in which it is to take place. We explain how we stage scenes using traffic managers that dynamically place and remove vehicles, and directors that assign roles and coordinate the action. We demonstrate the approach with scenes involving crash threats at intersections.

Resumé

Dans ce document, nous présentons une méthode pour intégrer des scénarios dans un trafic ambiant, par l'intermédiaire de castings qui distribuent dynamiquement les rôles aux véhicules pendant la simulation. Nous présentons un modèle de scène considérée en tant qu'abstraction dans laquelle sont encapsulées les actions des acteurs principaux, les tâches de fond, la mise en scène. Puis nous expliquons comment nous développons les scènes en utilisant des gestionnaires de trafic pour placer et retirer les véhicules, et des directeurs pour désigner les rôles et coordonner les actions. Enfin nous mettons en oeuvre cette méthode avec des scènes comprenant des actions dangereuses aux intersections.

1 Introduction

For many applications, driving simulators require that specific scenario events take place in a backdrop of ambient traffic. Using a movie-making metaphor, we can think of the ambient traffic as "extras" in the scene that are there to provide a sense of plausibility and context. Typically, the experimenter's script involves periods of free driving to accustom the driver to the environment interspersed with test events that occur at critical junctures. The challenge for scenario modeling is to stage these critical events in which actions are tightly controlled without prematurely alerting the subject to the prescribed activity.

An important component of the staging process is scene preparation: making sure that vehicles are in position to play the essential roles required by the scene. It can be very difficult to produce realistic ambient traffic, with appropriate density and flow, while concurrently ensuring that there are vehicles in the right places at the right times to produce scenario events. It is particularly difficult to choreograph scenarios embedded in ambient traffic when the assignment of roles is fixed and predetermined, i.e. when particular vehicles are selected, before the start of the simulation, to perform specific actions at specific places. Such strict scripting requires that individual vehicles be maneuvered, over possibly extended periods of time, to arrive at the right places at the right times to take assigned actions. By delaying the assignment of roles until run-time, we can select the vehicles most conveniently located to play roles shortly before the action is to take place. We call this dynamic assignment of roles "on-line casting."

On-line casting simplifies staging by requiring only that appropriate vehicles be positioned to take the essential roles of a scenario instead of a specific set of pre-selected vehicles. Role players can be selected from the ambient traffic. To stage a scenario event, we must specify the requirements for scenario role players and ensure that the ambient traffic will (with high probability) provide opportunities for selecting vehicles to perform the central actions.

In this paper we report on our progress in staging scenarios through on-line casting. After a brief summary of related research, we outline the primary features of Hank, an open research-oriented driving simulator we developed to support research on scenario and behavior modeling. Next, the Scenario Modeling section presents our concept of a scene as an abstraction that encapsulates a single situation with identifiable requirements for place and background activity (a set), roles (a cast), and action (a script). The next section describes the mechanisms we use to implement and execute scenes in Hank. It is followed by a section that illustrates on-line casting through an example scenario consisting of two rural intersection scenes. In the final section, we identify plans for future research.

2 Related work

Several simulators provide scripting languages for programming driving scenarios. Wolffelaar and Van Winsum[14] developed a simple and elegant language for programming scripted scenarios called the Scenario Specification Language. Sophisticated scripting facilities are also provided in the SCANeR simulator[8].

Generation of ambient, autonomous traffic for driving simulation has been widely addressed [1, 12, 13]. Work in the SmartPATH project [5, 6] encompasses both macroscopic traffic management and microscopic-level vehicle modeling, simulation, and control in order to support evaluation of Intelligent Vehicle Highway Systems (IVHS).

On-line direction of behaviors by scenario managers is used in the Iowa Driving Simulation[11]. Renault Research Center's TRaCs project [10] has directable vehicles that respond to commands to follow an itinerary and to low-level directives to modify basic driving behavior. Scenario control is supported both programmatically and interactively through instructor input at control stations.

Scenario management is also an important component of research in interactive drama, storytelling, and computer gaming. Research in improvisational behavior and on-line character direction[2, 7, 9] is especially relevant to our work.

2.1 Hank, the Simulator

In order to provide an open and flexible software system that enables wider participation in driving simulation research and fosters collaboration among groups working on different aspects of driving simulation, The

Figure 1: An image produced by the Hank simulator from the scenario described later in the paper.

University of Iowa Computer Science Department, with support and collaboration from Ford Motor Company, IRISA, the University of Rennes, the University of Valencia, and Renault Research, has developed the Hank driving simulator.

Hank provides the essential elements for experimentation in real-time driving simulation. Hank includes database modeling of complex scenes involving detailed terrain and cultural features, vehicles, bicyclists, pedestrians, and other objects. (After all, there's more to life than cars.) It provides high-quality visual display on Silicon Graphics and Evans & Sutherland platforms, as well as simpler two-dimensional visuals for debugging and testing on lower-end workstations. Hank has facilities for modeling the dynamics of scene entities, and a powerful and flexible mechanism for modeling higher level reactive and intentional behaviors through Hierarchical Concurrent State Machines (HCSM)[3].

A primary goal of the Hank project is to develop a driving simulation platform capable of supporting experimental research in scenario and behavior modeling. We believe scenario modeling will be the focus of growing attention as real-time driving simulation and other virtual environment applications (military, entertainment, etc.) expand.

HCSM is particularly well-suited for experimentation with novel approaches to scenario control. HCSM state machines include control panels through which entities can receive asynchronous communication giving directions that guide their behaviors. The result is an expansion of the reactive behavior modeling provided by traditional automata to a more flexible notion of directable behavior modeling. Director HCSMs can manage and coordinate behaviors of scene entities by sending messages to HCSMs associated with individual entities. Furthermore, director HCSMs can choreograph higher-level scenario activity by directing the activities of other director HCSMs.

Figure 1 shows an image produced by Hank from the scenario described later in the paper. More information about Hank can be found on the Hank World Wide Web homepage: http://www.cs.uiowa.edu/ hank.

3 Scenario Modeling

Adapting theatrical terminology, we break a multipart scenario into a sequence of scenes. A scene represents a coherent set of actions aimed to satisfy a single purpose. The decomposition of a scenario into more elementary scenes is important in promoting reuse of scenario code. Our goal is to encapsulate scenes so that they can be rearranged and combined in different ways to build complex scenarios. We intend to create a library of parameterized scenes from which an experimenter could compose a scenario appropriate to her needs.

We first explain what a scene is and how we specify it. In the next section, we explain how we currently implement scenes. A goal of our ongoing work is to automatically generate code implementing scenes and scenarios from specifications.

3.1 Scene Specification

The scene specification describes who is to be present, what is to happen, and when it will happen in a scene. The specification includes a description of the area in which the scene is to take place, the route the subject is expected to take, and a timeline of scene events. We divide the specification into three parts: the set, the cast of principal actors, and the script of actions:

Set: A set describes the context for a scene. It specifies the physical surroundings in which the scene will take place. In addition, the set defines the dynamic milieu in which the scene will unfold. It specifies the number, arrangement, and kinds of actors that will be present in the scene (i.e. the type and distribution of traffic, pedestrians, bicycles, traffic lights, etc.). Many of these actors will play largely undistinguished roles as extras providing a background of ambient activity. The principal actors will be opportunistically selected from this chorus of extras at run-time. Lastly, the setting identifies marks in the set-named locations where actors must be positioned or where events will take place at critical points in the action.

Cast: The cast identifies the principal actors in the scene. Roles are described by characteristics and capabilities the role actors must have. For example, a scene might require a truck to take the role of encroaching into the subject's lane. Actors will be cast into these roles at run-time from the extras.

Script: The script provides a detailed description of the actions to be performed and the relationships between actors' behaviors. At present, actions are strictly ordered. However, we are investigating the use of partial orders and nonlinear scenes in which subject responses influence the choice of subsequent actions. Dependencies on subject behavior are identified in the script. This includes synchronization of events such as the timing of traffic lights and the initiation of actions. The script may also specify compliances between actor behavior and subject performance. For example, the script may require that for some period of time a simulated vehicle precedes the subject vehicle with a specified separation distance. The script may also include conditions to be met in the scenario such as the absence of vehicles in an area around the subject at a particular moment in the scene.

Script specifications identify actors from the cast to perform actions using the marks in the set to place the actions. It is very helpful to have sketches showing snapshots from a typical instance of the scene, much as storyboarding is used in the film industry.

There is explicit and implicit interplay between the set, cast, and script. There are explicit requirements that the actors in the script be members of the cast and have the capabilities to perform the actions asked of them, and that the marks appear in the set. It is reasonably simple to test whether or not these explicit requirements are met in an implementation. The script also places more subtle implicit constraints on the set to provide ample opportunities to perform on-line casting. That is, the chorus of extras must, with high probability, include actors at the right places at the right times to take the principal roles.

Sometimes the requirements for a role are so specific that it is most effective to pre-select a specific actor to play the part. For example, consider a scene in which a single car located on the shoulder of the road is to pull out in front of the driver. Where there is no advantage to delaying the binding between actor and part, we permit scene specifications to identify cast members with pre-assigned roles.

4 Scenario Implementation and Execution

We implement scenes in the simulator by strategically placing semi-autonomous actors in the environment and directing their behaviors, on-line, to perform the actions demanded by the script, contingent on the behavior of the subject. Our stable of actors includes vehicles, pedestrians, bicyclists, and traffic lights [4].

We've found it very useful to divide the task of scene creation into two sub-tasks: stage management and directing. Stage management is the task of placing and removing actors from the set. Direction includes both on-line casting and the fine coordination of actors to accomplish actions laid out in the script.

4.1 Stage Management

Stage managers are responsible for creating instances of actors, placing them in the set, and taking them off the set. This is accomplished by placing actor sources (producers of objects) and actor sinks (consumers of objects) at strategic locations on the set, out of view of the subject. It is important to delete objects that are no longer involved in the scene to avoid the cost of simulating their behaviors and to prevent unnecessary clutter that may interfere with future scenes.

Sources produce actors at appropriate locations in the set. They must avoid creating objects that occupy the same physical space. Sources can be tailored to produce actors at fixed or variable intervals, to create homogeneous or heterogeneous types selected randomly or in fixed sequence, and to place actors at specific locations or at conditionally determined places. For example, a source could be asked to produce a random distribution of trucks, buses, and passenger vehicles at a location 500 meters in front of the subject. The timing, location, and production method can also be conditioned on the situation.

Similarly, sinks can be conditionally situated, conditionally enabled, and can be tailored to consume actors with only certain properties. Sources can be used to thin the population of actors by consuming actors in proportion to observed densities.

4.2 Directing

Directors take responsibility for casting actors into the principal roles of a scene and guiding their actions according to the script. As a scene progresses, the director must look for "extras" with the characteristics and capabilities required by roles that have yet to be cast. For on-line casting to be successful, the set must be crafted to create rich opportunities for casting of unfilled roles. Stage managers typically accomplish this by producing continuous streams of new vehicles heading to the sites of actions.

The second role of the director is to orchestrate the actions of the principal actors as dictated by the script. We program actor behaviors with directability in mind. Each type of actor (vehicle, pedestrian, bicyclist, and traffic light) responds to external directives that influence its behavior. Our goal is to make actors that are highly versatile so that they can fill many different roles. Consequently, the director need not look for actors with specialized talents; any actor of the right type in the right place is able perform a role.

Directors monitor the progress of the scene and send directives to actors assigned principal roles. To give a sense of these interactions we describe directives accepted by traffic lights and vehicles.

4.2.1 Traffic Light Directives

Experiments frequently call for traffic lights to be synchronized with the actions of a scene. For instance, we might want a light to turn red just as a driver enters an intersection. It is important that the subject perceive the traffic light to be operating normally - the timing of state changes should appear to be happenstance.

To coordinate the light cycle with the motion of the driver's vehicle, our traffic light model responds to a directive to be at a specific point in its cycle at a future target time. The traffic light cycle is represented on a normalized clock. That is, state transition times are expressed as fractions of the total green-yellow-red cycle duration. A light synchronization directive consists of a normalized cycle time and a target event time. For example, the light could be directed to be halfway through its green phase at the time the subject vehicle is expected to arrive at the intersection extrapolating from its current position and speed. The traffic light will adjust the cycle duration, uniformly compressing or expanding intervals between the current time and the target time, so that the light will be at the appropriate point in its cycle at the target time. This process maintains the normal progression of states and makes the smallest possible change to the normal cycle.

4.2.2 Vehicle Directives

Vehicles are programmed using the hierarchical concurrent state machines of HCSM. They have the ability to autonomously drive along roads, to navigate through intersections, to follow leading vehicles, to change lanes, and to pass other vehicles. Vehicles observe posted speed limits and obey traffic control devices. A detailed description of our vehicle behavior models is given in [4]

At present, vehicle models include interfaces for five directives:

AdjustSpeed: This directive influences the vehicle's proclivity to follow the posted speed limit. Positive settings tend to increase normal driving speeds and negative settings tend to decrease normal driving speeds.

IgnoreLight: This directive instructs the state machine to disregard traffic lights.

Halt: This directive tells the vehicle to rapidly decelerate. Repeated requests will cause the vehicle to stop.

Change-Lane: This directive tells the vehicle to change lanes.

Turn: This directive influences the vehicle's disposition to turn at the next intersection. On multi-lane roads, the directive must indicate both the direction of the turn and the destination lane.

4.2.3 Design Issues: Persistence, Competence, and Conflict

An important issue in the design of directable behaviors is the length of time that directives hold sway over actors. Many directives cause a single immediate change in the behavior of an actor influencing performance modes, set points, or decision variables. For example a director might instruct a vehicle to change lanes or change preferred driving speed. Sometimes it is important to maintain this state for a period of time. For example, we may want to keep a car in a certain lane for a period of time in preparation for an event. We are experimenting with a variety of ways to influence the persistence of behaviors. Sometimes the directors enforce persistence by repeatedly sending a directive to the actor. Some directives include persistence parameters that tell the actor to observe the directive for a period of time.

Some situations require that actors couple their behaviors to the subject's vehicle for a period of time. For example, we may wish to have a vehicle traveling in front of the subject's vehicle with some separation interval. Such compliant motions could be accomplished by building the behavior into the vehicle and providing a directive to initiate the behavior. Alternatively, a director could maneuver a vehicle into position and maintain the desired separation through a sequence of directives to change lane, speed up, or slow down. We are investigating the tradeoffs in placing competence in the director versus in the actor.

Lastly, we often find it convenient to have multiple directors operating simultaneously, each responsible for some aspect of a scene. This creates the possibility that directors will compete for the control of actors. At present, actors respond to the most recently received directive. This requires careful planning of concurrent directors to avoid conflicts. We are investigating methods to help anticipate and avoid these conflicts.

5 An Example Scenario

In this section, we present two example scenes. The scenes were developed independently and then combined to form a scenario with the two scenes in succession.

5.1 The Run-Red-Light Scene

The Run-Red-Light Scene The first scene is illustrated in Figure 2. It involves a subject's response to a vehicle driving through a red light into the path of the oncoming subject vehicle. The scene takes place on a rural two-lane road at an intersection with another road. The intersection is controlled by a four-way traffic light. Ambient traffic is to be present on the crossing road traveling in both directions.

The scene calls for the light to be halfway through its green cycle when the subject reaches the intersection. As the subject approaches the intersection, the first vehicle stopped at the light on the road to the subject's left will quickly accelerate and cross the intersection in violation of the red light.

A scene director is responsible for synchronizing the sequencing of the traffic light to the approach of the subject. The director estimates the time at which the subject will arrive at the intersection and directs the traffic light to be halfway through its green cycle at that time.

Lastly, the scene director monitors the distance between the subject vehicle and the upcoming intersection. As the subject approaches the intersection, the first vehicle on the crossing road is selected to perform the violation. At a moment near the time when the subject vehicle reaches the intersection, the violator is sent a directive to ignore the traffic light and quickly accelerate into the intersection.

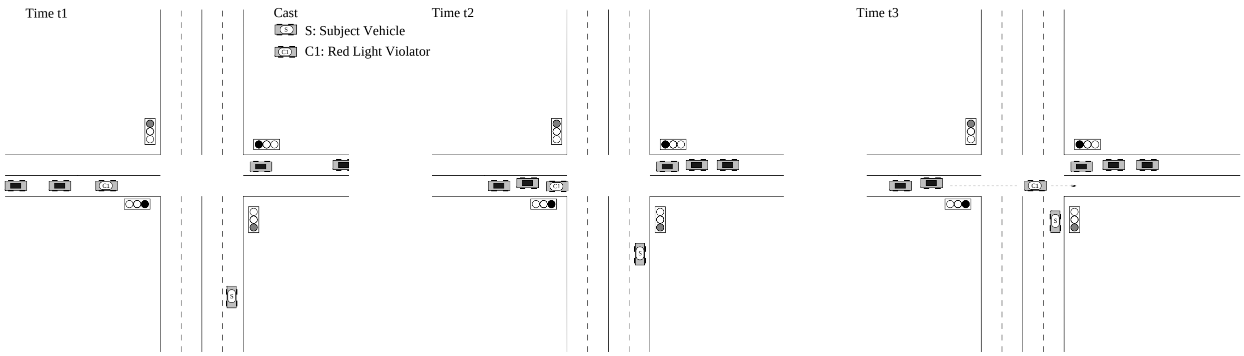

Fig 2. Run-Red-Light Scene

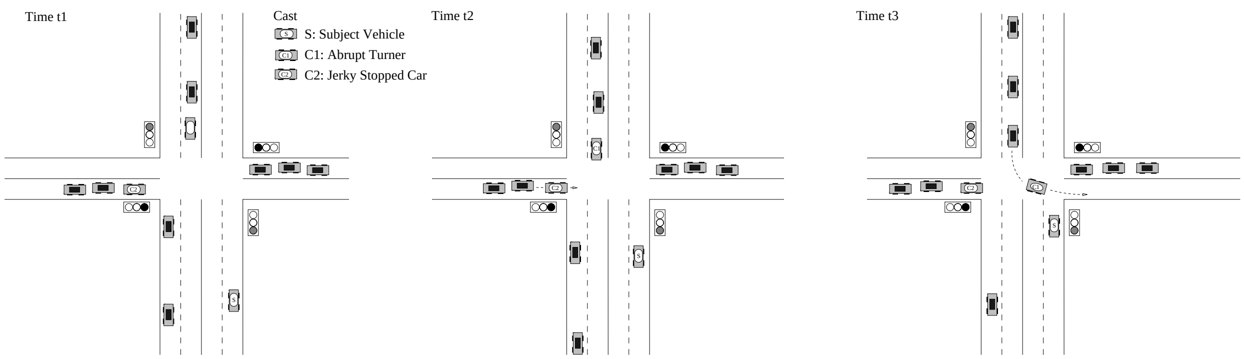

Fig 3. Divided Attention Scene

5.2 The Divided Attention Scene

The second scene, illustrated in Figure 3, has two principal roles. As in the first scene, the second scene takes place at an intersection on a two-lane highway controlled by a four-way traffic light. Ambient traffic is to appear in the oncoming lane and in both lanes of the crossing road. The light facing the subject is to be green as the subject approaches the intersection. The first role is to be played by a vehicle approaching the opposite side of the intersection in the oncoming lane. This vehicle is to reach the intersection shortly before the subject and pose a collision threat by abruptly turning in front of the subject.

The second role is to be played by a vehicle stopped at the red light on the crossing road to the left of the subject. This purpose of this role is to distract the subject by presenting a potential threat to violate the light, as happened in the first scene. The role is accentuated by having the vehicle lurch forward, feinting entry into the intersection, just before the first vehicle turns in front of the driver.

As before, the scene director is responsible for synchronizing the sequencing of the traffic light to the approach of the subject vehicle. As the subject approaches the intersection, the scene director selects an appropriate vehicle from the stream of vehicles approaching the side of the intersection opposite the subject. The selected car is dynamically cast into the “sudden left turn” role. The director chooses the vehicle that is projected to arrive at the intersection just before the subject.

Concurrently, the scene director casts the head of the queue of vehicles stopped for the red light to the subject’s left. As the subject driver nears the intersection, the director tells the head vehicle to suddenly lurch forward a bit, as if it were about to violate the red light.

6 Discussion

In this paper, we presented an approach to embedding scenarios in ambient traffic through the use of on-line casting. We present a model of a scene as an abstract unit of a scenario and explain how we create scenes using traffic managers to place and remove dynamic objects, and directors to assign roles and coordinate the action. Additional research in scenario generation is needed in a wide variety of areas including the design of directable behaviors, directing methodologies, automatic synthesis of scene code from specifications, and techniques to combine large numbers of scenes while maintaining consistency and continuity.

Software and Further Information

The HCSM software that constitutes the core of scenario modeling in Hank is available on the Hank World Wide Web homepage at URL: `http://www.cs.uiowa.edu/~hank`. We expect to release additional components of the Hank system in late 1997.

Acknowledgements

This research was supported in part by National Science Foundation grant IRI-9506624 and by Ford Motor Company.

References

[1] Bruno Arnaldi, Rémi Cozot, and Stéphane Donikian. Automated Cars in a Virtual Urban Environment. In Martin Göbel, editor, *Virtual Environments '95*, pages 171–184. Springer Verlag, January 1995.

[2] Bruce Blumberg and Tinsley Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Computer Graphics (SIGGRAPH 95)*. ACM, August 1995.

[3] James Cremer, Joseph Kearney, and Yiannis Papelis. HCSM: A Framework for Behavior and Scenario Control in Virtual Environments. *ACM Transactions on Modeling and Computer Simulation*, 5(3):242–267, July 1995.

[4] James Cremer, Joseph Kearney, and Peter Willemsen. Directable behavior models for virtual driving scenarios. *Transactions of The Society for Computer Simulation*, 14(2):87–96, June 1997.

[5] F. H. Eskafi. *Modeling and Simulation of Automated Highway System.* PhD thesis, University of California, Berkeley, 1996.

[6] Aleks Göllü, Akash Deshpande, Praveen Hingorani, and Pravin Varaiya. Smartdb: An object-oriented simulation framework for intelligent vehicles and highway systems. In *Proceedings of the 1994 Conference on AI, Simulation, and Planning in High Autonomy Systems*, pages 244–250, Gainesville, December 1994. IEEE Computer Society Press.

[7] B. Hayes-Roth and R. van Gent. Story-making with improvisational puppets. In *Proceedings of the First International Conference on Autonomous Agents*, pages 1–7, Marina del Rey, CA, February 1997.

[8] J-M. Kelada and J.-M. Piroird. Traffic generation and scenario control in the scaner simulator. In *Proceedings of DSC '95 (Driving Simulation Conference)*, pages 224–234, Sophia Antipolis, France, September 1995.

[9] Margaret Kelso, Peter Weyhrauch, and Joseph Bates. Dramatic presence. *PRESENCE: Teleoperators and Virtual Environments*, 2(1):1–15, 1993.

[10] M. Kergall, S. Regnier, and A. Kemeny. New enhancements in driving simulation at renault. In *Workshop on Scenario and Traffic Generation in Driving Simulation*, Orlando, December 1996. (proceedings available through The University of Iowa Computer Science Department).

[11] Yiannis Papelis. Graphical authoring of complex scenarios using high level coordinators. In *Workshop on Scenario and Traffic Generation in Driving Simulation*, Orlando, December 1996. (proceedings available through The University of Iowa Computer Science Department).

[12] Douglas A. Reece. *Selective Perception for Robot Driving*. PhD thesis, Carnegie Mellon University, May 1992.

[13] Rahul Sukthankar. *Situational Awareness for Tactical Driving*. PhD thesis, Robotics Institute. Carnegie Mellon University, January 1997.

[14] P. van Wolffelaar and W. van Winsum. Traffic modelling and driving simulation - an integrated approach. In *Proceedings of DSC '95 (Driving Simulation Conference)*, pages 235–244, Sophia Antipolis, France, September 1995.