ISSUES RELATED TO THE COMMONALITY AND COMPARABILITY OF DRIVING SIMULATION SCENARIOS

Omar Ahmad
Research & Development Project Leader

Abstract

The primary motivation for using driving simulators for research is the ability to safely expose participants to traffic situations in a controlled and repeatable manner. The process by which a simulation is programmed to expose participants to consistent situations is typically referred to as *scenario authoring*. The ability to re-use, and in turn, replicate scenario events is critical in comparing results across studies and/or different simulators. Unfortunately, scenario event replication is often a time consuming and error prone process. This paper describes some fundamental issues involved in scenario replication and presents a technique that allows accurate replication of scenario events with minimal human effort.

Introduction

The typical lifecycle of a driving simulation research project involves a researcher with a particular problem in mind working with a simulation engineer to define the scenes and scenarios that will be used in the research study. Data is collected from multiple research participants during several simulator drives. Before analysis, the data must often be reduced into meaningful variables which are then used to prove or disprove the original hypothesis. During a typical year of operation of the National Advanced Driving Simulator (NADS), it is not uncommon to run over 30 data collection efforts using hundreds of scenarios and involving over a thousand simulator hours and thousands of research participants. To support that level of effort, graphical interactive tools are used to help author scenarios [].

The ISAT system utilizes triggers and traffic sources in conjunction with autonomous microscopic traffic to create repeatable scenarios. A trigger monitors a set of conditions and when the conditions are true, performs actions that cause a scenario to take place. Traffic sources generate vehicles according to strict timing and geometrical constraints. Autonomous traffic navigates the virtual road network following rules of the road, unless forced by a trigger to act in a deterministic manner.

For the purpose of our discussion, the term *scenario* encapsulates a number of events that take place during one run of the simulation. A scenario typically ranges in time from a few minutes to over an hour. An *event* refers to one particular area or part of a scenario such as an intersection crossing or the series of events that simulate being cut-off by a vehicle. As implied by the previous definition, an event may consist of a sequence of sub-events. A typical event lasts a few seconds.

The ISAT uses a referential approach to building events and scenarios. This means that scenario elements are placed relative to the road network, in effect creating a dependency between a scenario and a specific road network. Information on existing scenario authoring approaches is not detailed enough to assess the nature of other scenario authoring systems, however, several systems for which information is available also use a referential model of specifying scenarios.

Motivation

It is very common to have situations where the same event is re-used several times, within the same scenario, study or even across studies. For example, a researcher may have read about a prior study utilizing a particular event and ask to use that same event on the new project. Or the design of a study may necessitate programming the same event several times during a drive.

Unfortunately, almost all cases involving re-use of events necessitate the re-development of these events from scratch, even though the final outcomes often resemble each other. The reasons for this paradox are many:

- a researcher may want to perform some minor change, such as the type of vehicle involved in the event
- the replicated event must be placed in a different part of the virtual environment
- the ambient characteristics of the road network are different (i.e., number of lanes, or intersection topology).

Presented at the IMAGE 2005 Conference
Scottsdale, Arizona July 2005.

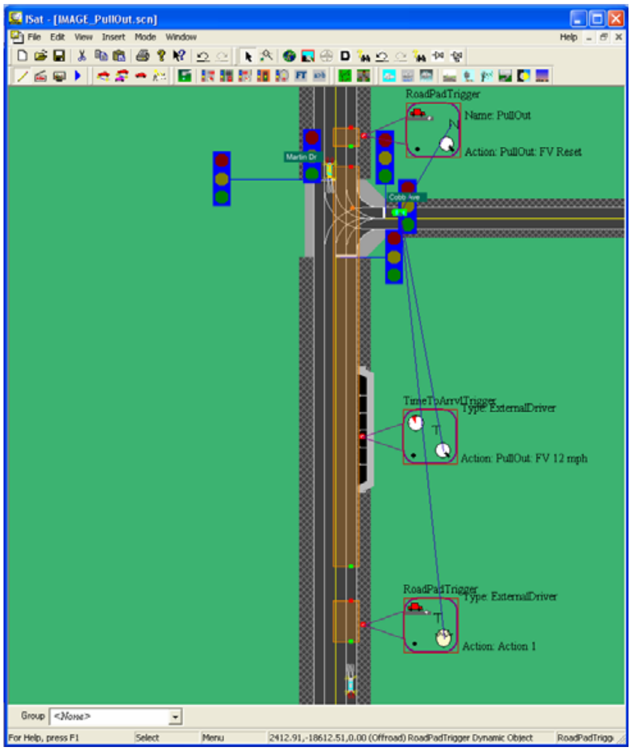

Fig. 1 A pull-out event.

To better understand the scenario development process and why re-use often necessitates re-authoring, consider the example shown in **Fig. 1**, which is an actual implementation of a simplified pull-out event. This event involves the participant vehicle along with 2 other vehicles and 3 triggers. The participant approaches the intersection traveling Northbound on Martin Dr. The intersection traffic light timing has been set so that the participant has a green light and the pull-out vehicle has a red light. The pull-out vehicle gets created on the side street (Cobb Ave.) as the participant vehicle runs over the first road-pad trigger. The pull-out vehicle is created with an initial forced velocity of 0 mph. A static vehicle has also been parked on the oncoming lane to provide filler. The pull-out vehicle is forced to accelerate to 18 mph and turn right from Cobb Ave. onto Martin Dr. as the as the participant gets within 5 seconds of the intersecting point. This action is initiated by the participant running over the 2nd trigger from the bottom (the time-to-arrival trigger). The forced velocity action is necessary to override the vehicle's inherent desire to obey the red light and stop. Finally, as the pull-out vehicle turns onto Martin Dr. in front of the participant and clears the intersection, it runs over the 3rd road-pad trigger and resets the forced velocity action to accelerate to its normal cruising velocity.

This event has been designed to work well on this particular intersection. Based on the specifications from the experimenter, the numbers were carefully chosen by the scenario engineer to provide the desired event severity. The value of 5 seconds was chosen specifically as that allows the pull-out vehicle enough time to accelerate from a stopped velocity to pull-out in front of the participant's vehicle. This takes into account the distance that the pull-out vehicle has to travel from its stopped position to get to the intersecting point. As the scenario engineer looks at replicating the event from one intersection to another, it becomes clear that this distance changes between different types of intersections (compare the distance from the solid white stopping line between **Fig. 1** and **Fig. 2** or **Fig. 3**.

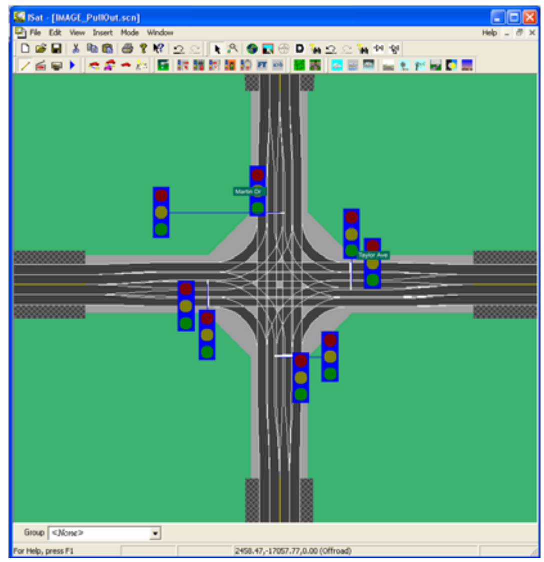

Fig. 2 Four-way intersection for the pull out event.

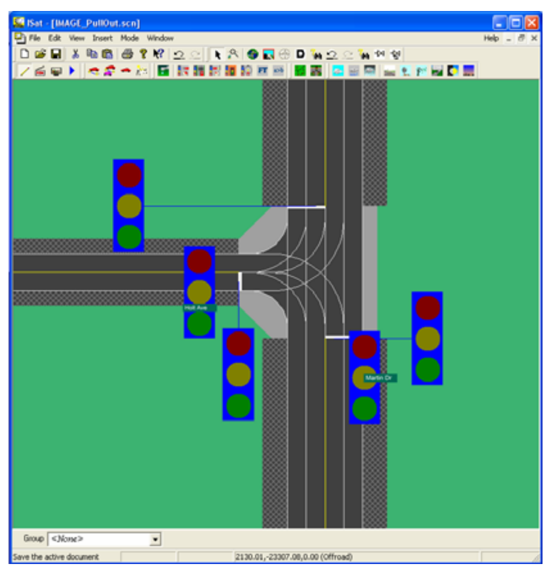

Fig. 3 Three way intersection for the pull out event.

The need to re-author an event and in turn the scenario is due to the fact that references to the road network must be re-assigned to the new network, even if the road topology is identical.

Even assuming that we can address the referencing issue by using a re-map technique that replaces references from the source location to the destination location, there are additional problems. If the re-mapping is done manually, there is a high likelihood that clerical errors will change the events in ways that affects the performance, yet such changes are not apparent. For example, while replicating the scenario mentioned above to a different intersection, maybe the one shown in **Fig. 3**, if the trigger pad is not placed at the exact same distance from the incursion, the event changes. Even in cases where range-rate is used as a triggering mechanism, inconsistent placement of a scenario may change the overall dynamic of the situation in ways that are not immediately apparent.

Another problem in the lack of consistency between key reference points in the source and destination location. The best example of this problem is when looking at the intersections in **Fig. 2** and **Fig. 3**. If the scenario elements are placed relative to the thick white line marking the stop point for the intersection, the distance a vehicle has to travel before creating an incursion changes. Once this changes, the whole event is changed and any performance data obtained by the two scenarios cannot be compared. In this particular example, a better reference point for placing the incurring vehicle traveling west-bound may have been the center of the northbound lane, as this would provide a consistent time-to-collision event to all participants, independent of the road geometry and intersection topology.

The ability to automate event replication depends on a deterministic specification of the unique aspects of an event. Ultimately, what matters is how the data will be used to calculate performance measures, and thus it is up to the scenario author to first identify what constitutes the critical dependencies among the virtual road network and all the scenario elements comprising an event. Our approach depends on having access to this information, and provides facilities for ensuring that these dependencies remain intact when replicating a scenario on a different location, even if the topology is not identical.

Replication Approach

We now present our approach to addressing the automatic replication of event. The basic theory resembles templates as used in C++. In effect, events are specified relative to a prototypical virtual environment, in effect creating a template that is not dependent on any specific road network. The actual mapping to the road network takes place during instancing, at which point specific features of the event are mapped to the actual environment in which the event is replicated. Interactions that are geographically linked to each other can be defined with respect to anchor points that are individually mapped between the prototype and actual road network.

In particular, there are four steps involved in creating an event in a way that can be replicated: (1) specify a generic topology, (2) add anchor points, (3) add scenario elements relative to anchor points, and (4) apply the generically specified event to the real environment by mapping the anchor points. The following discusses these steps in detail.

Specify Generic Topology

To specify an event such that it doesn't get tied down to a specific geometry, the event must be initially defined on a generic topology. The user should choose the type of roadway or intersection needed for the event from a list of generic intersections and roads with varying numbers of lanes. The intersection should be chosen so that it has the minimum number of incoming and outgoing lanes to implement the event. Similarly, the road should be chosen so that it has the minimum number of lanes to implement the desired event.

The event shown in **Fig. 1** needs at least a 3-way intersection and roads connected to this intersection need to have a minimum of one lane going in each direction. Therefore, the generic topology shown in **Fig. 4** can be used to implement this event.

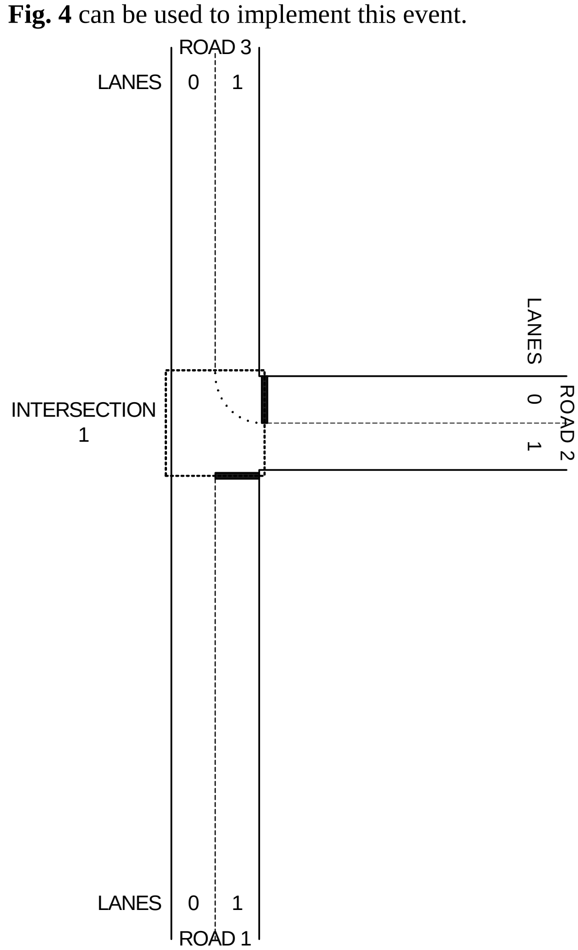

Fig. 4 Generic topology for the pull-out event.

Specify Anchor Points

An anchor point is a logical element that may be placed on a road or an intersection. It is the point from which all other elements are offset. In our generic topology, any point on the road can be identified using the following information:

- road name
- lane id
- distance along the lane

These 3 pieces of information uniquely identify any point on the generic topology road network. Distances are measured from the start of one end of the road to the other end.

The user may place any number of anchor points. However, the greater the number of anchor points, the more work the user will have to do when actually placing this event on the real environment.

To continue our example, two anchor points, A1 and A2, are chosen as shown in **Fig. 5**. Their locations are:

A1: <Road3, 0, 1.0>

A2: <Road3, 1, 2.0>

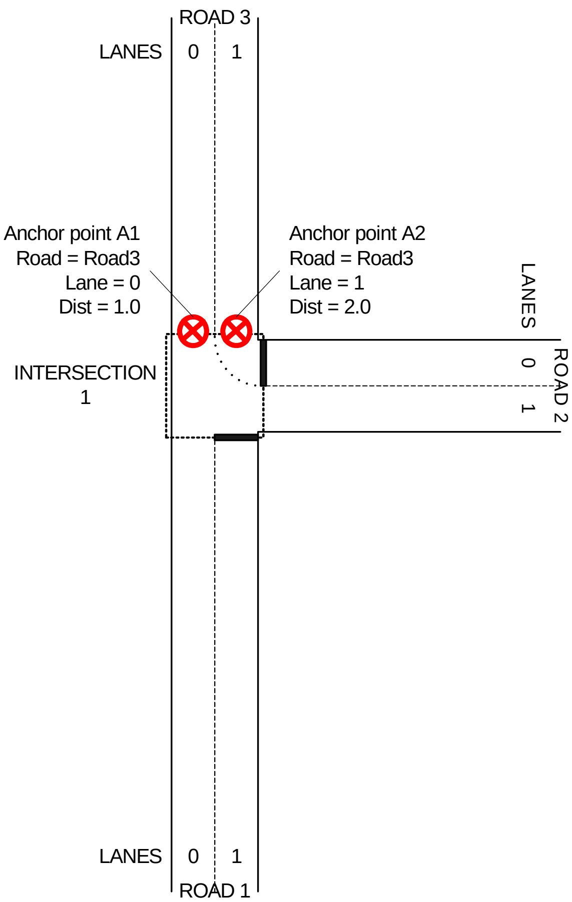

Fig. 5 Specifying anchor points on the generic topology.

Add Scenario Elements

Once the anchor point(s) have been placed on the generic topology, the scenario elements may be added relative to the anchor points. A point in the generic topology can be identified relative to an anchor point with the following information:

- anchor point
- road name
- distance along the road network from anchor point along
- lateral distance from center of lane

These 4 pieces of information allow placement of scenario elements relative to anchor points. Distances are measured from the anchor points along the road network. If the anchor point and scenario element are on the same road then a negative distances implies that the scenario element is placed after the anchor point along the road's direction of travel. Similarly, a positive distance implies that the scenario element is located before the anchor point.

As shown in **Fig. 6**, anchor point A1 is used to offset the vehicle in the oncoming lane. Therefore, vehicle 2 is located at:

Vehicle 2: <A1, Road3, 9.0, 0.0>

Anchor point A2 is used to offset everything else as follows:

Vehicle 1: <A2, Road2, 76.0, 0.0>

CreationTrigger: <A2, Road1, 500.0, 0.0>

TimeToArrivalTrigger: <A2, Road1, 220.0, 0.0>

ResetForcedVelocityTrigger: <A2, Road3, -7.0, 0.0>

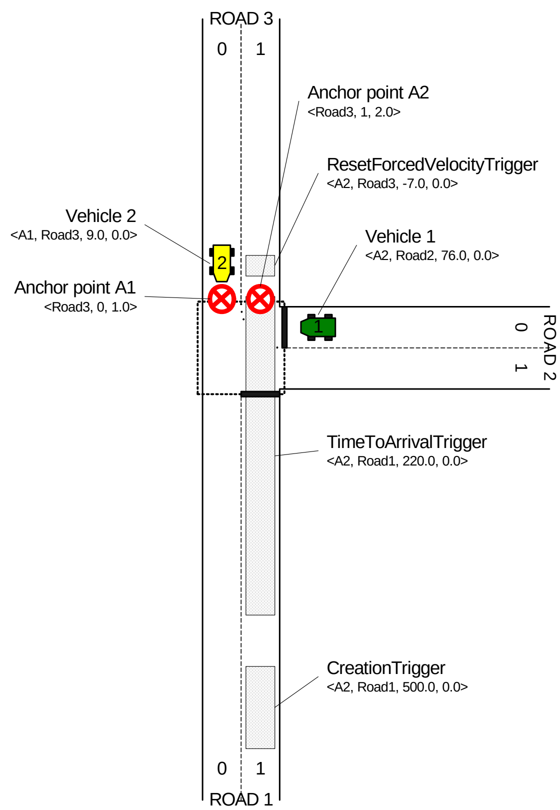

Fig. 6 Adding scenario elements for the pull-out event on the generic topology.

Apply Event to Real Environment

Once the user has specified the event on the generic topology, he/she can then place it on the real environment. The set of anchor points determine candidate locations for where the event may be placed. Once the user places the anchor points on the real environment, the element(s) associated with that anchor point can be automatically placed on the real environments relative to the anchor points.

Unresolved Issues

There are several issues that need to be addressed, beyond what is covered in this paper, before such a technique can be widely used in a production environment. In particular, the approach described here focuses on authoring events without explicit ties to the road network thus allowing the user to easily replicate an event within or across a scenario. We would like to see this notion being expanded to incorporate building a series of events or entire scenarios on generic topologies that are not tied to a specific road network.

An event's severity is determined by several factors including the relative placement of scenario elements, weather conditions, and visibility due to buildings or scenery. The process of replicating an event should, therefore, involve tracking visibility from the participant's viewpoint. For an event to be accurately replicated from one location in the road network to another location, the replication process must warn the user of impact of changes in visibility. For example, the severity of the event shown in **Fig. 1** can vary with the placement, or lack, of buildings around the intersection as that may determine at what point the pull-out vehicle will become visible to the participant.

As mentioned in previous sections, researchers often want to replicate events by changing a type or geometry of vehicle being used. We would thus like to expand the scope of replicating an event to allow the user to introduce such changes while keeping the overall event the same.

Finally, we would like to see support added for ambient traffic and features. A traffic manager creates and deletes ambient traffic in a scenario. Event replication should take into account the density of ambient traffic as an event is replicated in different areas of the road network.

Conclusion

This approach allows experimenters to specify events generically without having to get caught up in the details of specific geometry. This more closely mirrors the thought processes involved in designing scenarios.

References

Papelis, Y., Ahmad, O. & Watson, G. (2003). Developing Scenarios to Determine Effects of Driver Performance Techniques for Authoring and Lessons Learned. Driving Simulation Conference North America.

Papelis, Y., Ahmad, O. & Schikore, M (2001). Scenario Definition and Control for the National Advanced Driving Simulator. ESV: 17th International Technical Conference on the Enhanced Safety of Vehicles, Report No. 346. Amsterdam, Netherlands.

References to scenario control papers and URLs for the actual papers when I was able to find them.

Allen, R.W., Rosenthal, T. J., et. al. (2003). Scenarios Produced by Procedural Methods for Driving Research, Assesment and Training Applications. DSC 2003, Detroit, Michigan.
http://www.systemstech.com/Papers/STI-P-621dsc-NA%202003.pdf

Allen, R.W., Rosenthal, T. J., et. al. (2001). A Scenario Definition Language for Developing Driving Simulator Courses. DSC 2001, Sofia Antipolis (Nice), France.

Allen, R.W., Park, G., et. al. (2004). A Process for Developing Scenarios for Driving Simulations. IMAGE 2004 Conference, Scottsdale, Arizona.
http://www.systemstech.com/Papers/STI-P-632Image_txt.pdf

Wolffelaar, P. V., Bayarri, S., and Coma, I. Script-based Definition of Complex Driving Simulator Scenarios. Proceedings of the Driving Simulator Conference DSC'99, Paris, France.
http://www.inrets.fr/services/manif/manif99/dsc99/abstract/a4p4e.htm

Wright, S., Fernando, T., Ward, N. J., and Cohn, A. G. (1998). A Framework for Supporting Intelligent Traffic Within the Leeds Driving Simulator. Workshop on Intelligent Virtual Environments, ECAI 98.
http://portal.acm.org/citation.cfm?id=375735.376310