



## The ATON Project

Center for Research in Electronic Art Technology  
University of California, Santa Barbara  
Computer Vision and Robotics Research Laboratory  
University of California, San Diego  
CalTrans Test-Bed Center For Interoperability



# ATON Report 2001.06.8: Notes on the Integration of the Paramics Simulation Software with the ATON System

Andreas Engberg and Stephen Travis Pope  
{andreas, stp}@create.ucsb.edu  
CREATE, UCSB, October, 2001

## Introduction

As part of the ATON project, we have used a variety of tools and sensors simulate and interact with real-world events (such as vehicular traffic) in a virtual environment (VE). One of tools we planned to use was Paramics, which is a sophisticated simulator of traffic flow; it is widely used within Caltrans. Our plan was the retrieve information from Paramics about the different positions of vehicles in the simulation, to transmit it over a network (using CORBA messages), and finally to translate and attach the positions to objects in a virtual world. This report explains the use of the Paramics SW package, including the problems we encountered and their solutions.

## Traffic Simulation in the ATON VE

When we initially started working with Paramics, we did it without the software development kit (SDK). Another group in the Dept. of Geography at UCSB had used Paramics to build a model of several roads around our campus. Our first test showed that we were able to retrieve information about vehicles simply by storing the simulation's output data to a text file as it ran. This file could later be read and played back and thus "faking" a Paramics session.

This approach turned out to be a dead-end, however, since the format used for storing information did not provide the absolute position of vehicles (even though the manual implied that this was the case). When we later (in early 2001) talked to a Paramics consultant, we got the information that it was possible to achieve what we wanted using the SDK.

Because we were committed to delivering an integrated system demonstration on schedule, we decided to create something similar to Paramics by ourselves while we were waiting for the SDK

to arrive. We call this program the SplineServer. It reads a data file with “roads” and generates a stream of many CORBA messages that describe the motion of “vehicles” along these trajectories. It implements the CORBA interface that we had designed for use with the Paramics SDK. The most difficult task we faced was how to make cars follow a lane, where the lane could be of arbitrary form and length. Also, we wanted to be able to customize the speed (e.g. the distance travelled on the lane at any given point in time) of simulated vehicles.

We chose to use spline curves to describe the traffic lanes. This has several advantages: (1) They are very flexible and can be of any length; (2) they are very simple to describe (one BSpline uses 4 parameters and one interpolator); and (3) the spline can be shared by several users at the same time. There are, however, some disadvantages with BSplines such as continuity problems (when connecting BSplines) and unevenly distributed velocity at large and complex BSplines. The difficulty with velocities does not effect us since our splines are not complex or large, and the continuity problem was solved by using the simplest of continuities (G0). Higher-order continuity can be achieved by a careful and more accurate choice of splines.

The final version of the SplineServer program works quite well. For our VRML model of the UCSB campus, we have created five different lanes to represent Campus Road. In the model, each lane consists of three or four BSplines, and each is accessed via a CORBA service (connected to the spline engine or SplineServer) that allows us to connect virtual objects to a lane. Each object (“car”) knows how far on the lane it has travelled. This length is passed to the SplineServer and converted into a position and orientation. The most difficult task was then to map (determine the coordinates) the splines onto our already existing virtual world. This step was done by hand.

## Paramics

Once we got the Paramics SDK (after the official end of the ATON project), we found out that it was rather difficult to get the positions of vehicles. As it turns out, the SDK only provides the distance a vehicle has travelled on a lane, which is useless unless one knows exactly how the lane is defined. Therefore, we focused on understanding the description of a simulation. There are four files in a Paramics simulation; categories, centers, links, and nodes. The node file contains information about the “world,” e.g., the size of the world and the different nodes (defined as the absolute location of a junction). Straight roads are then interpolated between nodes while curves are described in the centers file. A curve is defined as center point and a radius, both in real-world coordinates. The properties of the roads are defined in the categories file, and includes the width of the road, the number of lanes, the maximum speed, and the type of the road. Finally, the most important file is the links file. It describes where and how the different roads are connected.

We have created a basic parser that can read all the Paramics data files, but it lacks several key features, such as lanes, curbs and other important traffic-related items. It simply sees all roads as one-lane at a constant width. The program creates an internal database consisting of the different files. Once the files have been parsed and stored internally, a user can retrieve

the position of a vehicle by cross-referencing its current link, the next exit-link and the current location on the current link.

## Conclusions

Integrating Paramics with the outside world was much more difficult than we had anticipated, and we were given unclear or false information at several stages about what the features of the program or the SDK's API were. We also did not receive the actual SDK software until after the end of the ATON project. Since it was our mission to deliver a "professional" integrated simulation, we chose to implement the SplineServer, which provided similar functionality to the Paramics-driven simulation.

In trying to integrate our infrastructure with the Paramics simulation, we could not find any reasonable way to retrieve the position of vehicles. The only solution we were able to design was to create a parser that scanned the Paramics files and created an internal database. The database could later be used to calculate a vehicle's position. The parser worked well for simple worlds, where the roads only used one lane and had constant width. As the worlds get more complex, however, the parser could not accurately calculate the positions of all vehicles. After posting several questions to the Paramics newsgroups (see below), they realized how difficult this was and said that it would be available in a future release of Paramics.

Finally, the SplineServer we wrote worked quite adequately for our simple traffic simulations. Even though it was a simple tool, it significantly enhanced the realism of the virtual world. It also served as a good test platform for CORBA message throughput, since it allowed us to see how much load our system could handle without degrading in overall (user-perceived) performance. The source code for the SplineServer is included in the UCSB ATON release in the directory `release/inputservers/pc/splineserver`.

Given the new features of the (yet-to-be-released) updated version of the Paramics SDK, we can envision a scenario where we are indeed able to integrate it with the ATON virtual environment system, but this is outside of both the budget and the time-frame of the ATON project.

## Appendix: News Group Messages

*From:* Andreas Engberg <andreas@create.ucsb.edu>  
*Date:* 7/19/01 11:04 AM  
*Subject:* Worldposition

Hi!

I'm working as a researcher at U.C. Santa Barbara.

We are currently using Paramics for traffic simulation and our goal is to connect Paramics to a 3D renderer, e.g. connect the position of the vehicles to objects in our 3Dworld.

Ok, the questions is: Is it possible to get world coordinates for all vehicles at the different timesteps?

As far as I know, there is no support for it in the API (only to get the distance a vehicle has travelled).

Any ideas?

Thanks,

Andreas

*From:* "Ewan Speirs" <paramics-info@quadstone.com>  
*Date:* 9/5/01 3:03 AM  
*Subject:* Re: Worldposition

Andreas,

We have added this feature for you in the next update to the Programmer API.

This should be ready for distribution to selected parties (U.C. Santa Barbara included) in the next 2 weeks. We will contact you directly when this update is available.

Regards,

Ewan Speirs

Paramics Technical Development Manager