Development of a Distributed Road Traffic Simulation System

T. Keawmanee, K. Seubpradit, P. Tandayya
Department of Computer Engineering, Faculty of Engineering,
Prince of Songkla University,
Kohong, Hat Yai, Songkhla 90112, Thailand
Tel. +66-74-287352, Fax.: +66-74-212895
e-mail : pichaya@coe.psu.ac.th, tot_k@hotmail.com

Abstract: This paper concerns the development of distributed interactive simulation of a road traffic system. The simulation involved models of multiple and distributed road vehicles running on different locations. The simulation system represented a traffic system as if the distributed simulation models were in the same environment. The development of the distributed road traffic simulation was based on High Level Architecture (HLA), a state-of-the-art IEEE standard for the distributed and real-time simulation. Other work concerned modeling and simulating the road vehicles and building the map database for the virtual distributed shared environment.

The information used in the simulation system was only in X-axis and Y-axis as the insignificant data in the Z-axis was omitted to simplify the simulation. However, the traffic system has visualized a 3-D coordinate system. The road vehicle models were able to avoid collision. The next direction of a vehicle can be chosen from the provided choices of further paths.

1. Introduction

This paper presents a road traffic simulation system applying the state-of-the-art distributed simulation framework, High Level Architecture (HLA) [1,2,3]. Traditional traffic simulation models were standalone. The outstanding features of the HLA are interoperability and reusability. HLA provides new flexibility and connectivity that are demanded on current network-oriented computer systems. HLA combines computer simulations into a larger (single) simulation and also allows additional simulations in the future.

In HLA, a combined simulaion system created from the dirstributed simulations is called a *federation*. Each simulation that is combined to form a federation is called a *federate*.

A federation contains federates and Runtime Infrastructure (RTI), a supporting software that has services to create and manage the execution of the federation, i.e. federation setup and dismantle, transmitting data amongst federates, synchronization. The specification of the interface between federates and the RTI is specified in the *Interface Specification*. Figure 1. illustrates the structure of a federation in HLA.

Generally, the development of a simulation system involved at least two branches of science, that are computer science and the natural science of the simulated system. In this road traffic simulation, simple mathematical models of the road vehicles were applied. The model of a road vehicle also contained instructions for generating behaviors of multiple vehicles under different circumstances. It also concerned how to control the traffic of multiple and distributed vehicle models to avoid collision. Two basic

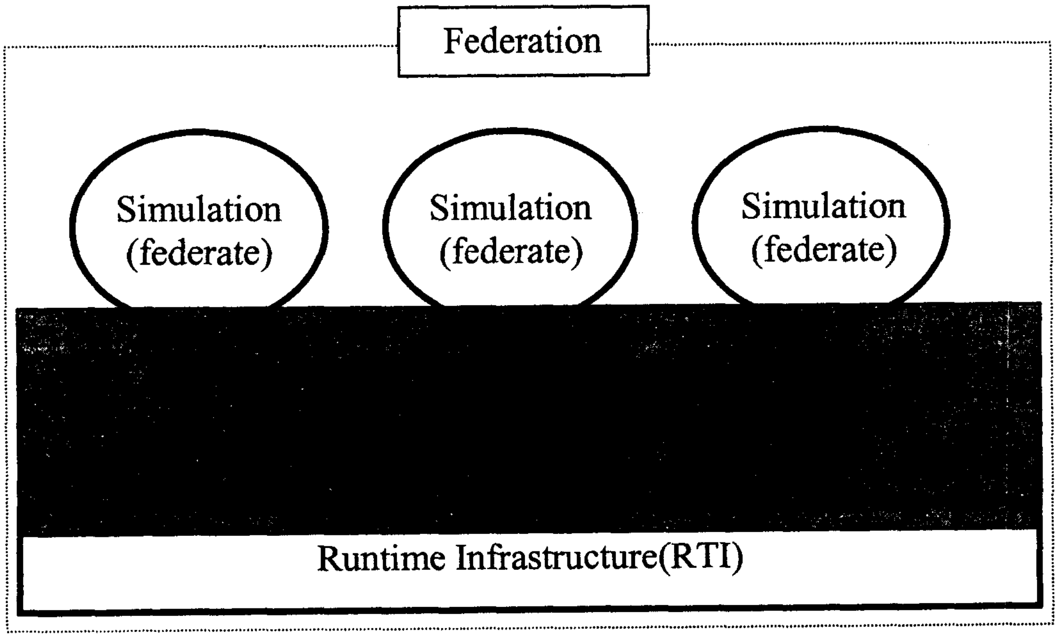

Figure 1. Structure of a federation in HLA

components in the road traffic system were models of the environment (a road map database and other traffic or landmark objects), and models of road vehicles. The 2-D map database contained lists of bi-directional paths. The visualization of this traffic simulation system was represented with 3-D graphical models on a 3-D environment. However, the movement in Z-axis was omitted.

When multiple vehicles run on the same path, a following vehicle needs to adjust its speed to the speed of the vehicle in front in order to avoid collision. The method of Bounding Spheres with a slightly bigger radius was used to help in avoiding collision.

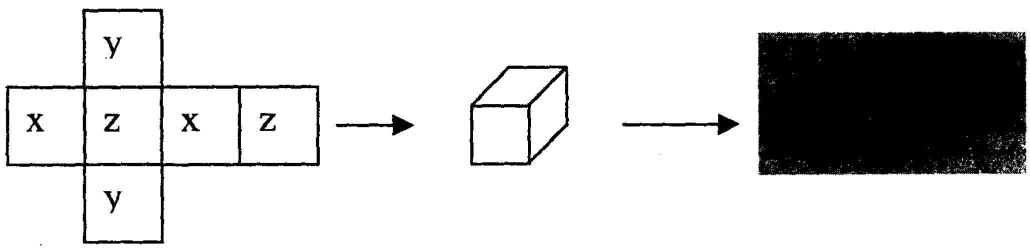

Figure 2. Model Space

RoadID	Starting & End Points	Width	No of Lanes	Equation	Connected Paths

Figure 3. Structure of a road

2. Visualization and Animation with DirectX

Using DirectX [4], an animation tool, for a 3-D visualization and animation, the coordinate system used was a left-handed coordinate system, of which its origin was (0, 0, 0) in the X-Y-Z axis. For the graphical model of a simulated road vehicle, a model space was used to wrap a 3-D coordinate position inside. A model space has a region of a 6-side squared box. The 6 sides were to compose a graphical model that appeared like a vehicle model (Figure 2.).

Applying the same idea, a world space is a 3-D world box or sky box that contains model space boxes inside. After model space boxes are located inside the world space, their origin is the same as the orrigin of the world space.

For the visualization, the user can change the position of the viewpoint arbitrarily, i.e. inside or outside a car, top view, and free look camera view.

3. Road Vehicle Simulation

The new location of vehicles was calculated from the following simple and well known equation; s = v t + ½ (a t). The additional features of the simulation model of a road vehicle involved the following methods.

.1 Collision Detection

he collision detection method used in this work was checking the vehicles' bounding spheres [5]. This method detects whether two objects are collided with each other by using circular balls that have the origin at the location of each vehicle. If R1 and R2 are the radius of the considered objects and D is the distance between the objects measured from the origin of the objects, R1+R2 <= D when the two objects collide with each other.

2 Vehicle Following

4. Structure of the Roads

Applying the designed map database, there can be several road maps involving in a traffic simulation. An additional map can be added on the fly. Each map composed of many paths (roads and lanes). Each path was specified by an ID number and had the equations of the road shape as shown in Figure 3. A path also had bi-directional information of starting and end points and further paths. The path structure was specified in a class that contained description to facilitate the movement of the vehicles on the road.

5. Dead Reckoning

In simulating a synthetic environment collecting simulations running on distributed locations, data updates from distributed simulations must be transmitted to inform one another. Simulations or federates do not communicate to each other directly but through the RTI. The approach allows the synchronisation of federates in the entire federation but causes twice the number of traffic usage.

In order to reduce the number of transmitting and receiving updates, a dead reckoning algorithm [6] is used to extrapolate next data until a threshold is reached or a new update arrives. In this case, update frequency can be reduced. This method can, however, cause disjointed displays of simulation. Additionally, this method requires more workload on each federate in order to compute the dead reckoning models of other federates and to have a hash table to contain data from all participating federates. Figure 4. Shows how the dead reckoning models reduce the number of updates.

6. Conclusion and Future Work

In this paper described work done a distributed road traffic simulation system applying the HLA framework. An

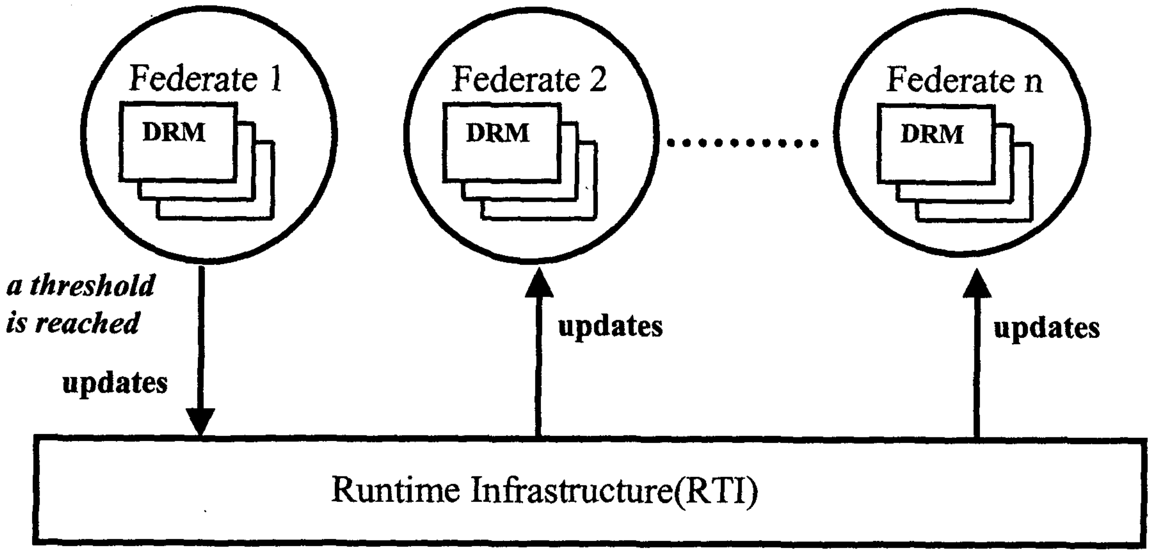

Figure 4. Dead Reckoning Models (DRMs) reduce the number of updates

Figure 5. Road traffic simulation

example of how the animation looks like is demonstrated in Figure 5. Mathematical and graphical models of road vehicles were used to simulate a road traffic simulation. A map database was also built for the virtual distributed shared environment.

The information used in the simulation system was based on 2-D information and the insignificant data in the Z-axis was omitted to simplify the simulation. However, the visualization was done in a 3-D coordinate system. The simulation models applied a method to avoid collision and do car following. The next direction of a vehicle was chosen from the choices of further paths in the map database.

For further development, the Z-axis can be added so that a simulated road vehicle can move upward and downward in a full 3D graphical system. Moreover, the new features of the vehicle model can include lane changing, overtaking and traffic lights and traffic rule handling.

References

[1] Defense Modeling and Simulation Office (DMSO*), The High Level Architecture Homepage*, 1998. Online available at http://hla.dmso.mil.

[2] R. Fujimoto, *Parallel and Distributed Simulation Systems*, John Wiley & Sons, New York 2000.

[3] F. Kuhl, R. Weatherly and J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice Hall, New Jersey 2000.

[4] Peeraphat Sawangpean, *DirectX: Beginning to 3D world*, Cwinapp homepage. Online available at http://www.cwinapp.com/tutorials/048.asp.

[5] W. Wisutimethagoon, *Collision Detection and Dynamic Impact Simulation of Mechanisms*, PhD Thesis, University of Wisconsin-Madison, 1998

[6] K.-C. Lin and D. E. Schab, "The Performance Assessment of the Dead Reckoning Algorithms in DIS,' *Simulation*, Vol. 63, no. 5, pp. 318-315, 1994.