Distributed Simulation of an Automated Highway System with Intelligent Vehicles [*]

R. DONATH[+], APRIL DEAN, D. GIRARDI AND S. RAMASWAMY

Software Automation and Intelligence Laboratory, Department of Computer Science and Center for Manufacturing Research

Tennessee Technological University, Cookeville TN 38505. Phone: (931)-372-3691 Email: srini@acm.org / srini@ieee.org

Abstract: This paper presents the simulation of an automated highway system with intelligent vehicles. The major objective of this work is to simulate intelligent communication and coordination between vehicles that operate on the automated highway. The simulation is being designed to study communication paradigms, traffic concepts, and road design. The simulation is visualized in a three-dimensional space and implemented as a client server distributed system. Intelligent vehicles communicate through a 6-way dynamic socket mesh (DSM) with other vehicles on the highway. The DSM is implemented as a six-way socket linked graph. Every client has access to the DSM, which contains information on current traffic situations in the simulation. DSM traversal is through socket connections in a peer to peer fashion for increased real-time coordination. Nodes in the DSM contain all relevant information about any vehicle that needs to be controlled.

I. Introduction

Automatic Highway Systems (AHS) were first introduced by General Motors Corporation (GM) in the 1939 World Fair [1]. A five-layered architecture consisting of a network, link, planning, regulation and physical layers is proposed in [2], for the development of AHS. In this architecture, each layer is assigned specific responsibilities as stated below: *(i)* The network layer is responsible for controlling the admission of vehicles and updates aggregate traffic status within the system. *(ii)* The link layer is responsible for assigning a path balancing traffic for all the lanes and provides this information to all the vehicles along with a target speed for travel on the particular highway section. *(iii)* The Planning and regulation layers develop a detailed plan for the vehicle's movement and execute the vehicle's trajectory to conform to the plan. *(iv)* The physical layer is responsible for providing the necessary sensor data, while responding to the actuator signals.

The biggest disadvantage of the AHS architecture described above is that it inherently assumes vehicles will have limited intelligence, which is manifested within the physical layer in collecting and providing the necessary sensory data for decision making. While such a system may perform admirably in a test environment, we believe that individual cars themselves need to be bestowed with greater intelligence and autonomy for complete automation of highway systems. Moreover, coordination within the components of the AHS is highly centralized. This can lead to potential bottleneck situations in large traffic situations. Thus, further development of the physical layer and the intelligence contained within the components of this layer, i.e., the vehicles, is critical for the efficient realization and operation of such an automated system. To bring such a system to fruition the following need to be addressed. *(i)* Embedding intelligence within the specific layers such that control information is passed to the higher layers above only if a particular layer is incapable of performing the necessary functions efficiently. *(ii)* Embedding basic intelligence and information sharing capabilities with the control units of individual vehicles so that they are capable of exhibiting intelligent behaviors.

Several key concepts in this area have emanated from the University of California, Berkeley's PATH laboratory, which is part of a nine-member consortium funded by Federal Highway Authority (FHWA) for designing, evaluating, and demonstrating a prototype AHS [3]. One such technique is platooning which is based on the observation of human driving behaviors; people tend to drive slower as the spacing between cars on the highway becomes narrower. Platooning aims at increasing the capacity of AHS [4] by addressing the proper utilization of available highway space. In this technique, cars are grouped together to form larger groups that move at predefined

---

[*] This work was carried out at the Software Automation and Intelligence Laboratory in the Department of Computer Science at Tennessee Technological University as part of our research on intelligent coordinating entities, or ICE. Dr. S. Ramaswamy is an Associate Professor and Chair of the Computer Science Department at Tennessee Technological University. Bob Donath, April Dean and David Girardi are undergraduate students in Computer Science at Tennessee Tech. Phone: (931)-372-3691. Email: srini@acm.org / srini@ieee.org..

[+] The author wishes to personally thank M. Thelan, an undergraduate student in Computer Science for stimulating conversations on several issues concerning the project. The authors additionally wish to thank the following people for their timely help in various stages of the implementation. James Lafever, Rocky Casteel, Chad Matthews and Hansraj Chudasama - undergraduate students in Computer Science; and, J. Gadiyaram, A. Karri - Graduate students in Mechanical Engineering.

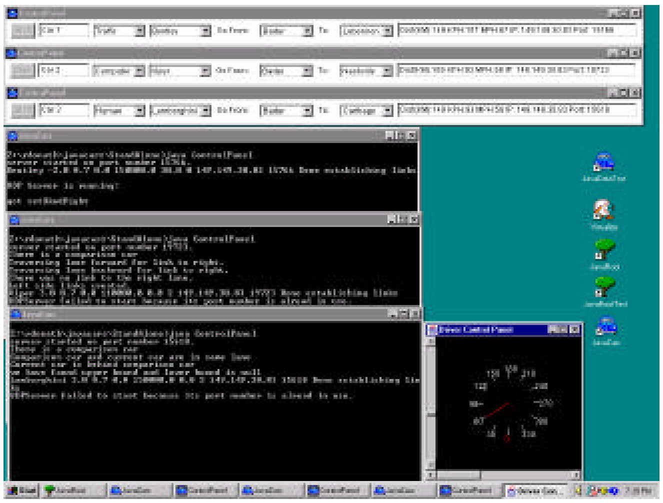

Figure 1. Three Types of Cars in the Simulation

speeds. Cars within a platoon are spaced at equal intra-platoon distances, with the lead car being called the leader, and the rest, followers. Experiments on platoons have demonstrated that this concept increases the highway capacity drastically, since the cars are now spaced at small, equal intra-platoon distances, moving at constant speed. Such small inter-car spacing calls for an automatic driver [5]. For such automated cars, the major maneuvers will include three critical operations: i.e., entry, exit and lane change. These maneuvers require intelligent coordination between the vehicles [6].

This paper is organized as follows: Section II details some of the design issues faced and Section III provides different scenarios from the implementation with respect to performance issues. Section IV concludes the paper.

II. Design Issues

In our research, we are concerned about designing and implementing better ways to improve the coordination between such intelligent vehicles. For vehicles to demonstrate intelligence in their operational behaviors, they first need to have the ability to communicate and coordinate effectively. In this paper, we present the design architecture for such intelligent vehicles and develop a simulation environment as the test-bed for evaluating performance. The following communication paradigms between the vehicles are currently being studied: *(i)* Autonomous resolution: Autonomous resolution means that the decision making by individual vehicles based on local observation. For example, in this scenario, drivers of these vehicles do not have the capability to request another driver to "let them into" their lane. *(ii)* Master/slave based resolution: Unlike many other distributed applications, in this simulation we do not use the master/slave paradigm for a physical communication between the vehicles. However, when vehicles choose to be in a platoon, then a master controller for the platoon is responsible for a majority of decisions concerning the platoon. Thus the platoon controller is assumed to have an omniscient perspective giving instructions to members of its platoon. For example, this situation is similar to a police officer directing traffic at a busy intersection and the first car in some given lane pulling the rest of the cars along with it; upon appropriate directions from the officer. The AHS is a more centralized master-slave approach in that the planning and regulation layers perform planning for all the vehicles. *(iii)* Mutual resolution by negotiation: In this situation, vehicles in close proximity have the ability to independently communicate with each other and resolve their needs such as lane changes, exit lane transfers, etc. without a central controller. Intuitively, this is the best desirable situation for an automated highway and will pave the way for completely automated highways with intelligent vehicles. Application of a particular communication paradigm depends on the current environment in the simulation.

As shown in Figure 1, three types of vehicles exist in the simulation; these include vehicles controlled by a human driver, a computer, or traffic vehicles. Car 1 is a traffic vehicle and is persistent throughout the simulation; i.e. it does not need to be reinitialized within the simulation. Car 2 is a computer-controlled car and incorporates a decision making logic for navigational purposes. Car 3 is a human controlled vehicle and is controlled entirely by an operator at the terminal. The odometer panel is for the human controlled vehicle in Figure 1.

III. Implementation

The simulation initially used VRML for the visualization (virtual reality modeling language); interpreted by a VRML browser. To manipulate objects the external authoring interface, or EAI, a set of Java classes that allows manipulation of objects in the VRML world, was used. While a three dimensional visualization of the data in real-time allows a better interpretation of the simulation data, rendering a three dimensional environment is processor intensive and competes with the decision logic mechanism. Also while creating and controlling simple objects in VRML is straight-forward; and complicated objects can be created and exported to VRML format through three dimensional design tools like True Space, scalability was a bottleneck issue with VRML. Thus the intelligent decision-making for the individual cars and the visualization is implemented using JAVA. The use of Java also enhances the portability and the distributed networking capability required for the implementation.

III. A. *Scenario I – Accuracy of DSM Link Maintenance*

This scenario is designed to illustrate the accuracy of the link maintenance algorithm within the visualized section of the highway. As stated earlier, a traffic and a human controlled car have to do all of the same work that a computer controlled car has to do except for running the decision making logic for navigation. Figure 2 shows three cars traveling on the highway and their use of the DSM based communication graph that allows efficient decentralized communication between the cars on the highway. The red cars (two) are under human control while the green car is a traffic car. The first data display corresponds to the first red car (a human controlled Dodge Viper), the second corresponds to the second human controlled car (a Ferrari), and the third data display corresponds to the traffic car (a Bentley). Note that the directions displayed in the data display windows for these cars are based on the perspective of the individual car. For example, the traffic (green) car, represented by the third data display window (from the left), has a forward right link to the Ferrari, while the Ferrari is linked from the viper as a backward link. Similarly, the Ferrari also has appropriate links to the other two cars. The green car is traveling faster than the two red cars. In Figure 3, the state of the graph as the green car passes the first red car is displayed. The data display windows show the 'before' and 'after' links for each car ***Figure 4*** shows the modified graph as the green car passes both the red cars. The use of better encapsulation and data manipulation mechanisms provided by an object-oriented programming language has a significant impact on code clarity and the efficient maintenance of the graph. ***Figure 5*** represents the situation where the speeds of the two human cars are increased so that they pass the traffic car. All of the data display windows for each respective car shows the updated final links, which are the same as the original links now that the cars are back in their initial configuration.

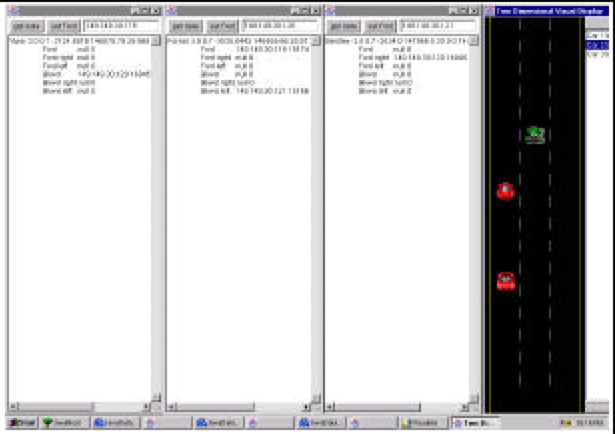

Figure 2. Scenario I – Shot I

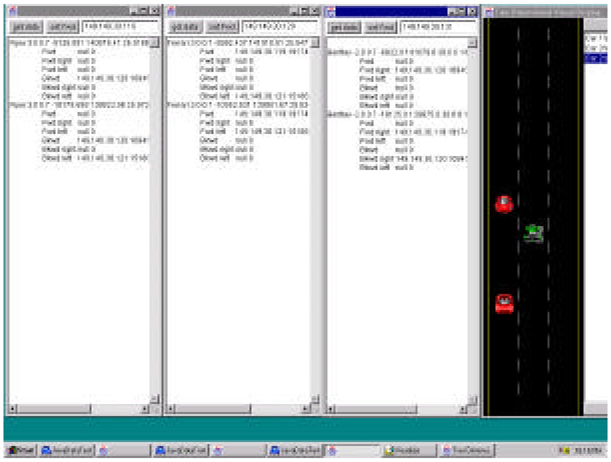

Figure 3. Scenario I – Shot II

III. B. *Scenario II – Computational Overheads:*

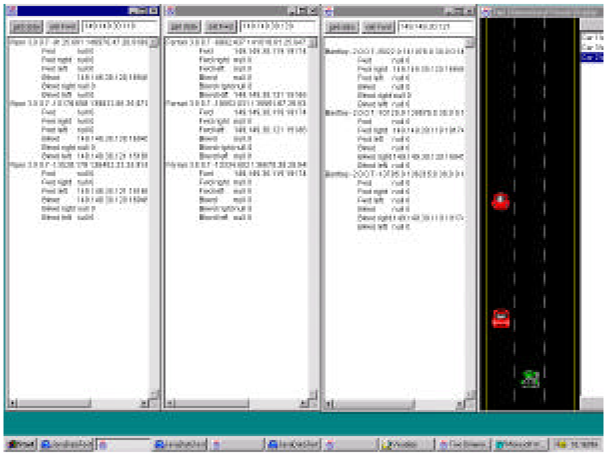

Figure 4. Scenario I – Shot III

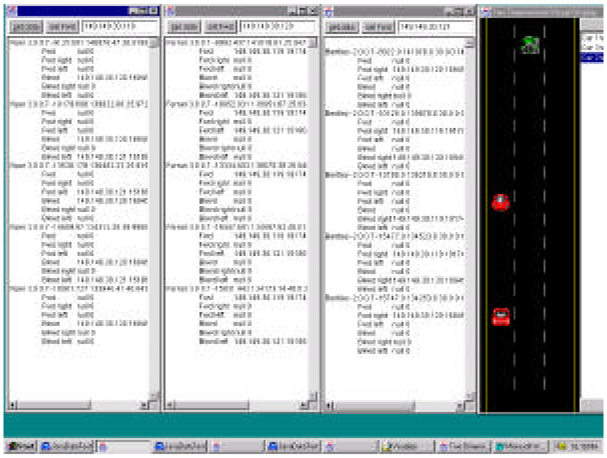

Figure 5. Scenario I – Shot IV

III.B.1. Scenario IIa: Overheard of Simulating Traffic Cars:

This scenario is designed to illustrate the minimal overheads involved in simulating traffic cars on a single machine, with and without "hooks" onto the visualization module. Traffic and human controlled cars perform the same tasks that a computer-controlled car does except that neither the traffic nor the human controlled cars execute the decision-making logic for navigation. Figure 6 showing ten traffic cars running on a single machine, demonstrates that the overhead of communications graph is negligible, while Figure 7 illustrates the same shot with a connection to the visualization module. The processor usage displayed for Figure 7 shows an additional 10% (14% vs. 4% in Figure 6) usage when running visualizations along with the cars on a single machine. However, the total processor usage is still minimal. These figures also demonstrate the potential capacity for the massive amounts of cars that could be part of the simulation as well as the excellent scalability of the simulation. The remote visualization, compared to a local VRML based visualization, requires only a small load on the machine that the cars are executing on. Figure 6 is showing the maximum processor usage for the ten traffic cars; that is when all of the cars on that computer are being visualized.

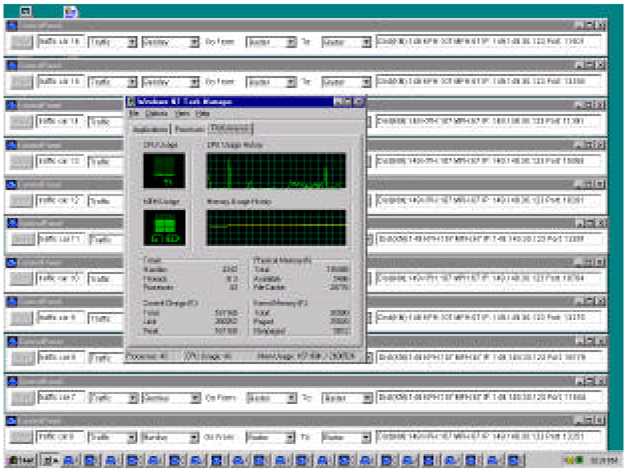

Figure 6. Scenario IIa - Case I

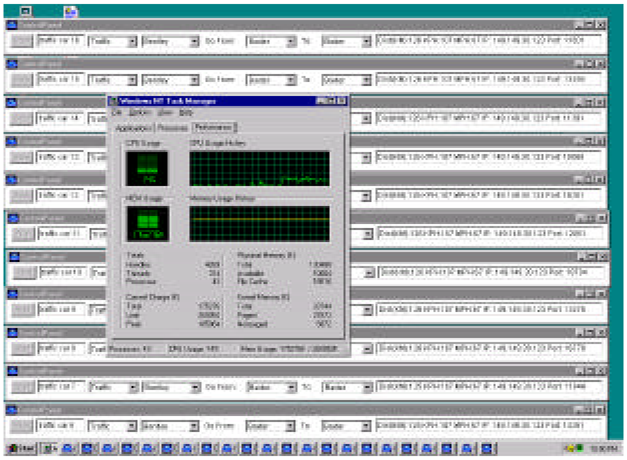

Figure 7. Scenario IIa - Case II

III.B.2. Scenario IIb: Overhead of Simulating Computer and Human Controlled Cars:

Figure 8 demonstrates the simulation of computer controlled car along with the added load from visualizing this car. Computer controlled cars have the overhead of setting up and maintaining the DSM as well as perform navigational logic. Currently, the navigational logic is based on their stated goal, which is to travel to the fastest lane in order to get to their destination in the shortest amount of time. . The logic is the same that was used in the original single processor design. Figure 8 demonstrates that while the computer-controlled car is executing the decision logic, the processor usage is only nine percent.

Figure 9 demonstrates one human controlled car with the added load from visualizing this car. Human controlled cars have to work to maintain the communications graph when passing other cars and changing lanes. This figure demonstrates that like the traffic cars, human controlled cars require phenomenally low processor usage approaching zero percent. Human controlled cars require the computer-controlled cars to logically handle unpredictable real life scenarios in order to prevent collisions.

IV. Conclusions

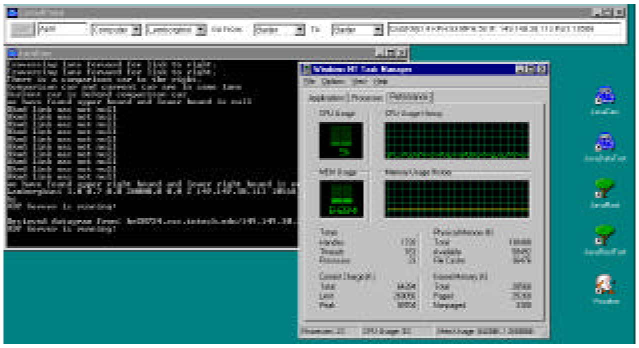

Figure 8. Scenario IIb - Case I

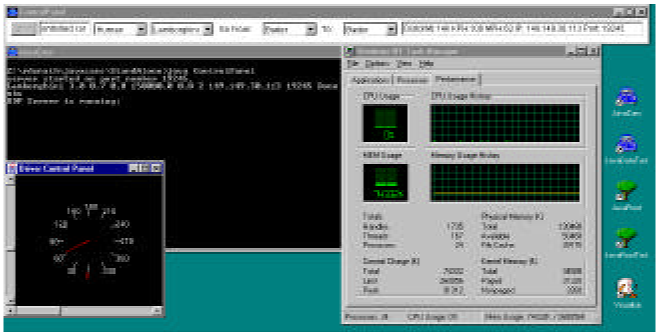

Figure 9. Scenario IIb - Case II

This paper presented a distributed simulation of an automated highway system with three different types of cars; viz. vehicles controlled by a human driver, a computer, or traffic vehicles. Such a simulation involving these three different kinds of vehicles captures the most plausible scenario for any futuristic automated highway system. While intelligent cars possess the capability to intelligently navigate the highway, traffic and human controlled cars do not possess such a capability. The design of the simulation however, is modularized sufficiently with the expectation that future technologies will allow intelligent automation modules to be installed on some human controlled / traffic cars. Due to the distributed nature of the entire simulation, the use of Java, an interpreted language and hence slower than other types of computing languages, does not overly restrict the simulation. In fact, it provides very good structure for maintenance and future adaptation as well multi-platform compatibility that improves portability.

V. References

1. R. E. Fenton and R. J. Mayhan, "Automated Highway Studies at Ohio State University- An Overview," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 100-113, Feb. 1991.
2. C. Unsal, P. Kachroo and J. S. Bay, "Multiple Stochastic Learning Automata for Vehicle Path Control in an Automated Highway System," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 29, no.1, pp. 120-127, Jan. 1999.
3. A. Deshpande and P. Varaiya, "Design and Evaluation Tools for Automated Highway Systems," *http://www.path.berkeley.edu/~akash/akash.html*, as of May 21st, 2000.
4. J. G. Bender, " An overview of system studies of automated highway systems," *IEEE Transactions on Vehicular Technology*, vol. 40, no.1, pp. 82-99, Feb 1991.
5. P. Varaiya, "Automated Highway Systems: A Technology for the 21st Century," *http://www.path.berkeley.edu/~varaiya/papers_ps.dir/cdc.pdf*, as of Jul. 28th, 2000.
6. P. Varaiya, "Smart Cars on Smart Roads: Problems of Control," *IEEE Transactions on Automatic control*, vol. AC-38, no. 2, pp. 195-207, Feb 1993.