Autonomous Automobile Behavior through Context-based Reasoning

Fernando G. Gonzalez

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816 USA
(407)823-3987

fgonzale@pegasus.cc.ucf.edu

Grejs Patric

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816 USA

Avelino J. Gonzalez

Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816 USA
(407)823-5027

ajg@ece.engr.ucf.edu

Abstract

Today's driving simulators are used in vehicle research and design as well as in training. However, most simulators are not convincing because the degree of realism is not adequate. To achieve greater realism, a simulator must include autonomous vehicles in the environment.

An essential feedback component to the driver of a car is the vehicular traffic sharing the same road environment. A car simulator must provide information about all elements of an environment that affects the driving task, such as the road layout, traffic control devices, and any autonomous vehicles or objects that exhibit reactive behavior. The simulator must also include logical information, such as information about passing lanes, lane direction, and position of one object relative to another object.

The intent of this research is to develop a system that produces an interactive traffic model that behaves autonomously and intelligently. Furthermore, this model is to be effective, computationally efficient and developed with relative ease.

Introduction

Today's increasing traffic volume requires efficient solutions. In this context, traffic modeling and simulation techniques are of crucial importance to investigate and evaluate the alternative controlling strategies before using them in "real" traffic.

The most essential feedback component to the driver of a car is the vehicular traffic sharing the same road environment. Context-Based Reasoning (CxBR) is an excellent choice for modeling autonomous entities in a simulated world. CxBR was introduced by Gonzalez and Ahlers [Gonzalez & Ahlers, 1993] to represent and reason with human behavior. CxBR uses different contexts to represent intelligent entity behavior, where the meaning of contexts is the situation, environment or surroundings faced by the different entities in the simulated world. The representation paradigm uses three different types of contexts, mission-context, major-context, and sub-context. The mission-context contains the overall goal for a certain scenario. The major-context is a tactical operation that assists in the mission-context to achieve its goal. The sub-context is a lower level tactical procedure that assists the major-context to achieve its goal. See Gonzalez [Gonzalez 1998] for an in-depth description of CxBR.

Until now the CxBR paradigm has only been tested on military applications. In a military scenario there exists many different missions, but in a driving simulation there only exists one, which is to drive safely from point A to B. Thus in a driving scenario, a mission-context is not needed. The prototype developed will consist of three different contexts, major-context, sub-context, and sub-sub-context. The major-context will be responsible for the overall scenario. Events that occurs within a major-context will be controlled by sub-contexts (i.e., passing vehicles, following vehicles, etc). The lowest level of contexts will be the sub-sub-contexts. These contexts assist the sub-context to reach its goal. For example, the sub-context FOLLOWING needs three sub-sub-contexts (accelerate, decelerate, and stop) to successfully perform its task.

The intent is to develop a system that produces an interactive traffic model, which simulates cars that drive autonomously and intelligently. The model will represent road networks in detail and will simulate vehicle movement using the context-based reasoning paradigm. The model will also represent lane changes, traffic signal logic, passing vehicles, intersection logic, following logic, and evasive maneuvers.

Literature Survey

State Operator and Result (SOAR) is a cognitive architecture technique that is used to implement intelligent agents. SOAR is one of the most promising candidates for developing agents that behave like humans [Jones, Tambe, Laird, et al., 1993]. One of the strength of Soar is it's flexibility and adaptive behavior. Furthermore, SOAR allows the smooth integration of planning and reaction in decision making [Pearson et al., 1993].

However, during any execution cycle, all rules that include relevant information about the current situation are considered. When one of these rules is fired, it may cause that other rules will also fire, thus requiring multiple subgoals to be evaluated before they can be retracted. This can cause the memory problems. Another weakness with SOAR is that it requires more time to learn than other programming languages and systems.

The Rational Behavior Model (RBM) developed by [Byrnes 1993], is a multi-paradigm, multi level intelligent control architecture. It is composed of three levels: i.e., Strategic, Tactical, and Execution levels. The strategic level performs the behavioral control. The Strategic level governs the operation of the tactical level. The Tactical level embodies behavior by maintaining behavioral attributes for the system, which includes system memory and world-and-local memory models. The Tactical level also forms a representation of internal behaviors for the strategic level. Finally, the actual behavioral interface to other behaviors is located in the Execution Level. Therefore, RBM explicitly supports the three components required for implementing behaviors. This architecture uses an AND/OR goal tree is very efficient when the problem domain is small. However, when the size of the goal tree increases, the usefulness of these types of system rapidly decreases.

The ZAROFF system presented in [Moore, Geib, & Reich 1995] implements the behavior of human figures in a virtual environment. The ZAROFF system is a controller for the players in a game of hide and seek. The system features visually realistic human figure animation, including realistic human locomotion. ZAROFF includes the following behaviors: attraction, avoidance, field-of-view, path following, and chasing. To control the agent's behaviors, a set of finite state machines, which run in parallel with the simulation, is used. These machines are responsible for behavior scheduling and act as a high level interface among the behaviors. A disadvantage with the ZAROFF system is that it is quite complex and becomes very difficult to implement. Furthermore, to control the behavior of the agents, the system uses FSMs and this leads to a lower degree of autonomy.

Virtual Roadway Environment Database (VRED) was developed to provide information about the road environment in the Iowa Driving Simulator (IDS). VRED models many driving related features in a virtual environment and supports efficiently queries that provide information directly relevant to the driving task [Papelis, Bahauddin, 1996]. VRED is also responsible for maintaining information about the physical state of an arbitrary number of moving entities in relation to the road network, and the use of a road coordinate system that greatly simplifies the determination of spatial relationships of entities that travel on the road network. One of the advantages with VRED is its execution rate. The execution rate for the physical model is typically 30 to 60 Hz and 10 to 20 for the behavior model, which is satisfactory. The disadvantage with VRED is its complexity. VRED is one of the components currently under development in the Iowa Driving Simulator (IDS).

Hierarchical Concurrent State Machine (HCSM) is a model for autonomous driving behavior useful for creating ambient traffic as well as experiment specific scenarios for driving simulation [Creamer, Kearney, Papelis, 1995]. The model follows roadways, obeying the rules of the road and it reacts to nearby vehicles and traffic lights. HCSM also supports a range of behaviors including passing, lane changes, and safe navigation through intersections. One advantage with HCSM is that its state machines avoid many of the problems of traditional finite state automata (retain the degree of autonomy) while retaining easy-to understand execution semantics. Since this system includes many different operations it may have a tendency to be slow. The development of the HCSM is strongly motivated by the needs of the Iowa Driving Simulator [Kuhl et al., 1995].

Approach

CxBR represents the behavioral model for a vehicle with a combination of objects, pattern matching rules, and script-like structures. This knowledge is then used to create an intelligent agent, which acts in the same way as a real driver. The agents are controlled by their overall mission. These missions are declared as instances of object classes representing plans, which must be carried out. The behavior of a vehicle is defined in Major-contexts, sub-contexts, and sub-sub-contexts. These contexts are structured in a hierarchical manner. For example, a Major-context has one or more sub-contexts associated with it and a sub-context has one or more sub-sub-contexts associated with it.

The Driver Behavior Model has four major-contexts

- Rural-Road,
- T-Intersection,
- Traffic-Light, and
- 4-Way-Intersection.

The details of these contexts are presented in [Grejs 1997]. The ***Rural-Road*** major context is presented here as a representative example of the others.

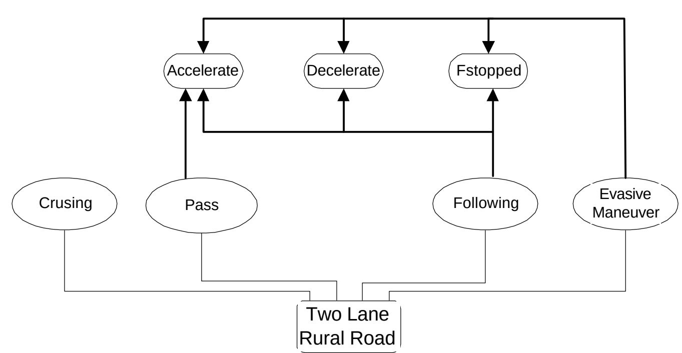

Figure 1: Rural-Road context

The RURAL-ROAD context (Figure 1) is active everywhere in our exercise except when the vehicle is in an intersection. This major-context has seven sub-contexts *cruising*, *passing1*, *passing2*, *passing3*, *following*, *evasive1, and evasive2*. When a vehicle has the sub-context *cruising* it cruises along the road at maximum speed without any interference. This sub-context has no sub-sub-contexts. The *following* sub-context has three sub-sub-contexts *accelerate*, *decelerate*, and *fstopped*. The system has two types of stop (stopped and fstopped) to avoid a conflict between a stop at an intersection and a stop on the rural road.

The passing mechanism includes three sub-contexts *passing1*, *passing2*, and *passing3*. *Passing1* is in force when the vehicle is changing lane, *passing2* when the vehicle is in the passing lane and *passing3* when the vehicle returns to the lane after a successful passing. The first sub-context of the evasive maneuver mechanism directs the vehicle towards the edge of the road and the second moves it back to the lane.

One or more sub-context(s) are associated with each major-context. These sub-contexts contain the behavior for a certain situation. For example, the sub-context ***passing1*** is associated with the major-context ***rural-road***. The sub-contexts can also be associated with more than one major-context. This is the case for the ***following*** sub-context. One example of this might be when a vehicle is following another vehicle and changes major-context, then the sub-contexts will not change. Again the details of the sub-contexts are presented in [Grejs 1997]. Only the ***following*** and the ***evasive*** sub-contexts are presented here.

When a vehicle is approaching another vehicle, its sub-context changes to ***following***. The following mechanism has three sub-sub-contexts, these are ***accelerate***, ***decelerate***, and ***fstopped***. The vehicle that is following another vehicle constantly checks the distance to the vehicle in front. If the distance becomes too large, the vehicle changes sub-sub-context to ***accelerate*** and if the distance becomes to short, the vehicle changes sub-sub-context to ***decelerate***. When more than one vehicle is following a leader every vehicle checks its distance to the nearest vehicle in front.

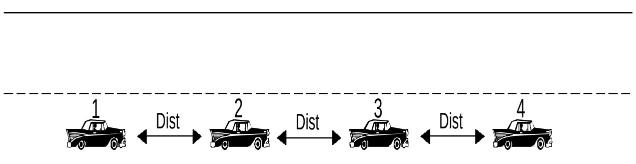

Figure 2: Following vehicles

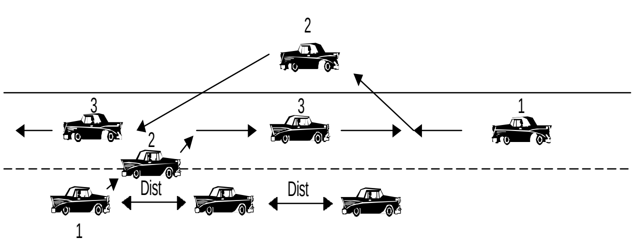

Figure 3: Evasive maneuver

The vehicle constantly checks the lane for other traffic. When a vehicle going the wrong direction occupies its lane, it performs an evasive maneuver to prevent a collision. When the vehicle undertakes an evasive maneuver, it changes sub-context to evasiv1 and sub-sub-context to decelerate, which will turn the vehicle towards the side of the road and slow it down. When the vehicle has reached the side of the road it checks to see if the lane is clear. If the lane is clear, it changes sub-context to evasive2 and returns to the lane, otherwise it remains at the side of the road until the lane is clear.

Test Bed and Evaluation

Two types of evaluation where conducted on the Driver Behavior Model (DBM); a qualitative test and a quantitative test.

The Test Program

The testbed road section is approximately 3 miles in length, and it includes straight sections, curves, and intersections See Figure 4. The road is defined in a Cartesian coordinate system, which needs three points and a radius to represent a curve and two points and an angle to represent a straight segment. The angle for the straight segment is used to calculate the next coordinates for the vehicle. The three points that represent the curve are where the curve starts, where the curve ends, and where the two straight segments. The straight segment points are the starting point and the end point of the curve. For a vehicle to follow the road it has to calculate the next coordinates in the direction it is going. Whether the vehicle is on a straight segment or a curve, the coordinates are calculated according to trigonometric formulas. The testbed also include three intersections: a T-intersection, a traffic light and a 4-way-stop-intersection. The intersections are represented with lines in the testbed. The traffic light is represented with dots that changes color depending on the current status of the traffic light.

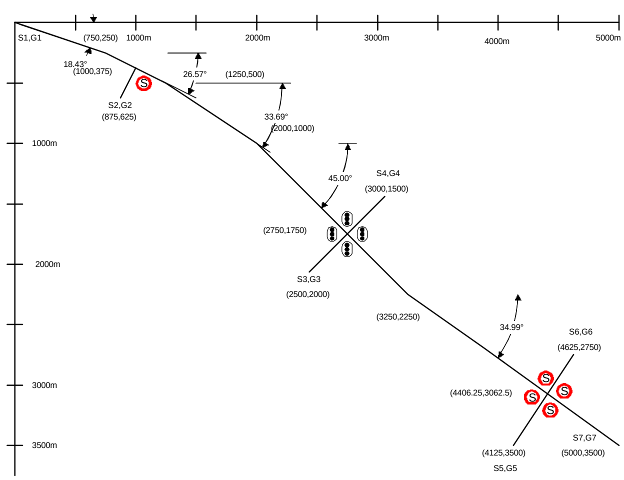

Figure 4: Road representation

Qualitative test

47 test cases where executed in order to evaluate the behavioral validity of the Driver Behavior Model (DBM). Each test case evaluates a certain operation of one or more vehicles. The test cases allow all vehicles to transition between all contexts, sub-contexts, and sub-sub-contexts in order to complete their mission, which is to drive safely on the road. The driver behavioral model was able to generate realistic behavior for all scenarios (i.e., following, passing, evasive-maneuver, t-intersection, traffic-light, and 4-way-stop-intersection).

Quantitative test

The purpose of quantitatively testing is to investigate how fast the Context-based Reasoning approach executes compared with other approaches.

One important issue to consider when comparing an AI approach with a non-AI approach is the speed of each system. In this case, the comparison is between CxBR and a car-following algorithm. The following section will discuss the time performance of the Driver Behavioral Model (DBM). First, where the bottleneck in the DBM is located and second a reality test to determine when the DBM reaches its limit compared with reality. To perform these tests the performance analyzer, Borland Turbo Profiler was used.

When DBM runs with the CxBR part, the time for each car is 22.2ms. In this case the execution rate for DBM is 11Hz. When DBM runs without the AI part the time for each vehicle is 11ms.

These times differs depending on the number of vehicles in the simulation at a certain time, because all vehicles divide the system overhead.

This test case uses the most time consuming context in the system, which is the FOLLOWING context. This context is most demanding because it continuously checks the distance to the vehicle in front.

This test conclude that approximately half the time spent by the Driver Behavioral Model Prototype is due to system functions and the other half to the CLIPS part.

Execution rate

The execution rate differs depending on the number of vehicles that are in the simulated world at one time. The rate also changes depending on which behavior (contexts) is performed at the time. The following two graphs shows the execution rate (Hz) depending on the number of vehicles and their contexts. The Following context is the most demanding, while the Cruising context is the least so.

The first graph

Figure 5 shows the execution rate when all vehicles are driving down the road without any interference (i.e., sub-context ***cruising***). For two vehicles DBMs execution rate is 38 Hz, but when more vehicles are added the execution rate degenerates rapidly.

This graph Figure 6 shows the execution rate when all vehicles are driving down the road following each other (i.e., sub-context ***following***). This is the most demanding context in the traffic model, because all vehicles are constantly calculating the distance to the nearest vehicle in front.

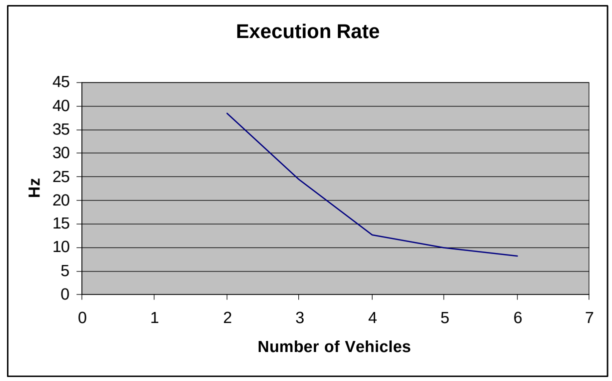

Figure 5: Execution rate for Cruising

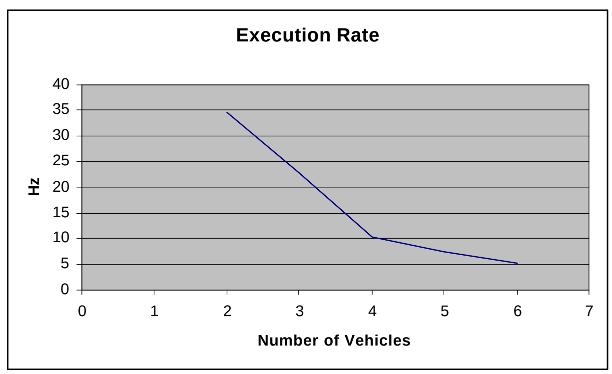

Figure 6: Execution rate for Following

Both graphs have the same decline, which is a proof that DBM is consistent in executing contexts. The difference between the two graphs is approximately 4 HZ, which is the time spent by the following mechanism in the prototype.

Since the following mechanism is the most time consuming context, it is safe to say that this graph Figure 6 presents the minimum hertz for a certain number of vehicles at one time. For example, the minimum execution rate of the prototype with three vehicles is approximately 23 Hz, no matter which contexts the vehicles are in.

The Driver Behavioral Model (DBM) was compared with two other systems, VRED [Papelis & Bahauddin, 1996] and a car-following algorithm [Ellis, 1997]. When comparing the size of the program (i.e., lines of code), Ellis` model has approximately 4000 lines while the CxBR prototype has nearly 4500 lines of code. However, Ellis` car-following model only includes two behaviors (following and passing), while the CxBR prototype represents six behaviors (following, passing, evasive-maneuver, t-intersection, traffic-light, and 4-way-stop-intersection). This concludes that the CxBR approach represents knowledge in a more compact (i.e., fewer lines of code) manner than the car-following model. Compared with the VRED system the DBM had similar execution rate. The execution rate for VRED's behavior model is 10 – 20 Hz and the DBM's execution rate is a comparable 5 – 20 Hz depending on the number of vehicles in the simulation at one time.

Conclusion

The intent of this research was to investigate how well the CxBR representational approach performs comparing with other traffic generating methods, especially car-following algorithms. The Driver Behavioral Model was tested in a small test program.

The CxBR paradigm presents an alternative approach to modeling autonomous traffic. The tactical knowledge is very important because it captures the information needed to react in any situation. The CxBR paradigm captures this knowledge in a concise and effective way as shown in previous research [Gonzalez, 1994; Brown, 1994].

One advantage with CxBR is that expanding the system would be very easy, for example if a new behavior were to be included. This new behavior could very easily be included in the system just by adding new contexts (i.e., classes) with associated rules and procedures.

Many systems that generate autonomous traffic today rely heavily on Finite State Machines (FMS's) which limit the degree of autonomy. One could argue that CxBR also has the basic structure of FSM. The difference is that FSMs are a very general data structure, which is not specific for what we want to do. CxBR, on the other hand is a very flexible and intuitive approach for representing intelligent behavior.

References

Byrnes, R., "An Experimental Comparison of Hierarchical and Subsumption Software Architecture for Control of an Autonomous Underwater Vehicle." IEEE, Washington DC, 1993.

Creamer, J., Kearney, J., Papelis, Y., "HCSM: A Framework for Behavior and Scenario Control in Virtual Environments", ACM Transactions on Modeling and Computer Simulation, Vol. 5, No. 3, July 1995, pp. 242-267.

Ellis, R. B., "Traffic Generation for the UCF Driving Simulator" Master's Thesis, Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, November 1997.

Gonzalez, A. J., Ahlers, R. H., "A Context-based Representation of Tactical Knowledge for Use in Simulator-based Autonomous Intelligent Platforms" Proceedings of the 15th Annual Interservice/Industry Training Systems and Education Conference, Orlando, FL, November 1993, pp. 543-552.

Gonzalez, A. J., Ahlers, R. H., "Context-Based Representation of Intelligent Behavior In Training Simulations" Transactions of the Society for Computer Simulation International, 1998, Vol. 15, No. 4, pp. 153-166.

Grejs, P. "Autonomous Automobile Behavior through Context-based Reasoning " Master's Thesis, Dept. of Electrical and Computer Engineering, University of Central Florida, 1997.

Jones, R., Tambe, M., Laird, J., Rosenbloom, P., "Intelligent Automated Agents for Flight Training Simulators." Proceedings of the Third Conference on Comuter Generated Forces and Behavior Representation, Orlando, Florida, May 17-19 1993, pp. 33-42.

Kuhl, J., Evans, D., Papelis, Y., Romano,R., Watson, G., "The Iowa Driving Simulator- an immersive research environment.", IEEE Computer. July 1995, pp. 35-41.

Laird, J.E., Jones, R. M., Nielsen, P. E., Coordinated Behavior of Computer Generated Forces in TacAir-Soar. University of Michigan, 1996.

Moore, M. B., Geib, C., Reich, B. D., "Planning for Reactive Behaviors in Hide and seek." University of Pennsylvania, 1995.

Papelis, Y. E., Bahauddin, S., "Logical Modeling of Roadways to Support Real-Time Simulation of Autonomous Traffic", Center for computer aided design, The University of Iowa, Iowa city Iowa, 1996.

Pearson, D., Huffman, S., Willis, M., Laird, J., Jones, R., "Intelligent multi-level control in a highly reactive domain." Proceedings of the International Conference on Intelligent Autonomous Systems, 1993.