Agent-Based Systems Engineering and Intelligent Vehicles and Road Systems

Valentino Crespi and George Cybenko

Institute for Security Technology Studies
Thayer School of Engineering
Dartmouth College, NH 03755 USA
{valentino.crespi,gvc}@dartmouth.edu

Abstract

This report summarizes our preliminary efforts at applying the methodology and techniques of *agent-based systems engineering* (ABSE) to the problem of designing and operating Intelligent Roads and Vehicles Systems (IRVS). First, we briefly summarize the key elements of both agent-based systems engineering and IRVS. We then show how the taxonomy of ABSE applies to the IRVS concept and how some performance metrics can be introduced. The report concludes with plans for continued work in this direction.

1 Introduction

One of the main goals of agent-based architectures is to enable the design and operation of future decentralized and distributed information systems, especially those requiring large-scale control and coordination. Autonomous agents, operating in a dynamically changing, heterogeneous environment, have their own goals and their own means towards achieving those goals. Desirable global behaviors are hopefully emergent from individual agents' behaviors which are all that can be programmed a priori.

Classical centralized control systems are typically based on centralized knowledge of system state and dynamics. As a result, powerful mathematical tools can be applied to study the performance of such systems in various dimensions. By contrast, performance characterizations of agent-based systems are largely ad hoc and empirical today.

Our research program in agent-based systems engineering strives to develop a mathematical framework for autonomous software agent systems that will allow quantitative analysis of system performance and behavior. This paper describes our early analysis of an IRVS architecture.

Most of the optimism surrounding software agents is based on the analogy between autonomous agents and "social systems" in the biological, real world (human societies or natural ecologies for example). By allowing individuals to follow their own aspirations, and by assuming that each individual is able to behave "intelligently" (see [15] for a possible definition of intelligent agent) a multi-agent system can operate efficiently and stably.

For example, a widely held heuristic about interacting intelligent agents posits that if the agents can:

- *react* promptly to changes in the surrounding environment in order to reevaluate and update current personal goals;
- *pro-actively* focus on significant and viable goals in the long run and avoid eternal and inconclusive indecisiveness;
- *interact*, or better still, *negotiate* with other agents to improve their knowledge of the outside world and to derive new options that might lead to new intentions and commitments;

then it is likely that the overall multi-agent system will find some sort of dynamic equilibrium and show desirable "emergent" macro-behaviors [15].

In other words, global effective behaviors should be the logical consequence of "reasonable attitudes" taken by the individual agents in negotiating intelligently with each other and establishing an acceptable trade-off between "what they would like to do" and "what they can do." This requires that the "rules" of the game are clear for everybody, are accepted by most of the components, and are designed to guarantee autonomy and behavioral flexibility. Moreover, such systems should be capable of tolerating some lack of cooperation and even maliciousness.

In order to make the transition from heuristic wishful thinking to scientific theory and ultimately solid engineering practice, intelligent software agent systems need to be quantified in a framework amenable to modeling and formal analysis. Possible mathematical tools that could be applicable then would be some combination of control theory, game theory, statistical mechanics, diffusion theory, for example.

This cross-fertilization between systems/control theory and agent-based computing systems constitutes the main part of our DARPA TASK project (see [5]). One of the project's hypotheses is that agent systems can be decomposed, by analogy to what happens in classical systems theory, into three ingredients: **observation, state estimation and control** with the ultimate goal of being able to derive reliable performance predictions and guarantees in multi-agent systems [11]. Our agent taxonomy actually includes a fourth component that has no counterpart in control theory, so-called **brokering agents**, whose role is to match agents with resources in an equitable way.

Unlike the classical monolithic agent model, the architecture based on our taxonomy is more flexible. It allows a further decomposition of complex agents into simpler sub-agents. For example we can view a BDI agent [15] as being composed of 4 types of sub-agents, one for each of the internal functions. In particular we define the following agent types:

- **information agents** that gather information about the environment and allow dissemination of it in a distributed manner (these would be "sensors" in classical systems theory for example);
- **modeling agents** that collect data from many information agents and update internal knowledge, i.e., produce new estimates of "real world" state (these would be state estimators, like Kalman filters, in classical systems theory);
- **planning agents** that use the current world state estimates, the viable action or control options and the current goals or intentions to plan new actions to carry out. These agents may need additional information for their planning operation, and so they may task **brokering** agents to report on available resources such as additional state and action information.

The proposed architecture based on this taxonomy encourages the development of agent systems based on small, simple but very reactive agents that have well defined functionality individually, amenable to quantification and modeling. Collectively, with individual agents quantitatively modeled, it is hoped that mathematical analysis of the overall system will be possible, leading to performance predictions and guarantees.

2 Intelligent Roadway Vehicle System Specification

Intelligent Roadway and Vehicle Systems (IRVS) have been a subject of study for several years. The goal of such a system is to make heavily traveled highways much more efficient and safe than the present highway and vehicle system allows. The key idea behind IRVS

is to replace human operators and relatively "dumb" traffic flow controllers, such as traffic lights and access ramp regulators, with smart vehicular and traffic controllers as well as an assortment of roadway and vehicle sensors, all communicating in some manner.

Horowitz and Varaya [14], for example, provide a description of the control architecture of an Automated Highway System (AHS) that has been developed over the past ten years at the University of California Partners for Advanced Transit and Highways (PATH) program, in cooperation with the State of California Department of Transportation (Caltrans) and the United States Federal Highway Administration (FHWA).

The architecture in question is characterized by a hierarchy of 5 layers: *physical, regulation, coordination, link and network*. The functions of the first three ones are realized by on board controllers that together constitute a hybrid control system (see [9, 10]) whereas those of the last two ones are competence of the roadway infrastructure. In particular there is one network controller that has global visibility over the whole system and performs global optimizations. Then, there is one link layer controller for each *link*, i.e., each segment of highway of length 0.5 to 5 km. This controller monitors the traffic flow along its link. It receives from the network controller demands on the inlet flow and constraints on the outlet flow (global optima) and consequently issues activity plans to be broadcast to all the coordination layer controllers aboard the vehicles transiting within the link.

The significant increase of the capacity of the highway system is obtained by organizing the traffic in groups of up to 20 tightly spaced ($1 - 2$ meters) vehicles called *platoons*. Under such conditions full automation becomes necessary due to the limited human reaction time.

It has been observed that even with an inter-platoon distance of 60 meters the mean inter-vehicle distance would drop down enough to allow a capacity of about $8,000$ vehicles per hour per lane (against the $2,000$ in today's highways). Furthermore, such a small inter-vehicle distance would reduce the relative impact velocity in case of collision as well as the aerodynamic drag. This implies that the platooning tactics increases safety and optimizes fuel consumption at the same time.

The layer structure is outlined in the following table:

Layer	Functions, Controllers and Models
Network	Control of the entering traffic and of the route traffic flow within the AHS. At this level the associated controller computes, for each link, **demands** on the inlet flow and **constraints** on the outlet flow to achieve a **global optimum**. **Model**: Capacitated graph.
Link	Monitoring of the traffic flow along a specific link and computation of activity plans. There is a controller associated with each link of highway. It receives data from the network controller concerning the inlet and outlet flow and issues **activity plans** to be broadcast to all the vehicles on the link. It does not have visibility of single vehicles. Those **plans** are meant for types of vehicle, i.e., private cars, emergency vehicles, etc., in a specific lane of the link and with a role (leaders or followers). Example: "all private cars in lane 1 that are platoon leader have permit to split their platoon and change lane". **Model**: Fluid flow with distributed control.

Coordination Selection of an activity, i.e., a **maneuver**, in coordination with peers (link layer controllers of the neighboring vehicles) and in compliance with an activity plan. There is an associated on board controller that receives the activity plan from the link controller and checks with other peers whether it can start one of the maneuvers in it. Once the coordination is achieved and the activity selected the regulation layer takes care of performing the corresponding maneuver. **Model**: Finite State Machines.

Regulation Execution of single maneuvers commanded by the coordination layer. The associated controller has to deal with the physical layer to perform the given maneuver safely. In case of failure it notifies back the anomaly to the coordination layer that aborts the maneuver. Maneuvers, called laws at this level, are all grouped in tasks that can be, in turn, of three different types: lateral, longitudinal or special.
Task 1 (longitudinal). *Follower law.* Vehicle followers (non leaders) must keep the platoon formation.
Task 2 (longitudinal). Platoon leaders or free agents can perform four longitudinal maneuvers:
a) *Leader law.* Regulate the platoon velocity at a desired value;
b) *Join law.* Join the preceding platoon;
c) *Split law.* Split the platoon;
d) *Split-to-change-lane law.* Split from a platoon in order to change lane.
Task 3 (lateral). *Keep lane.* Platoon leaders or free agents are to keep the assigned lane.
Task 4 (lateral). *Change lane.* Platoon leaders or free agents must move to an adjacent lane.
Task 5 (special). Group of maneuvers for Platoon leaders or free agents that must leave the link through an exit ramp.
Task 6 (special). Group of maneuvers necessary for a newcomer to enter safely the link through an entrance ramp.
Model: Feedback laws based on linear differential equations.

Physical Decoupling lateral and longitudinal vehicle guidance control. There are several associated controllers including the on-board sensors and actuators. The function of this layer is to approximately linearize the physical layer dynamics. **Model**: High order nonlinear differential equations.

The above summary shows that the PATH AHS design is based on a trade-off between centralization and autonomy in the control. The link layer issues activity plans destined to "types" of vehicles, trying to meet the global demands and constraints. Nonetheless, the single vehicles transiting within the link react with a certain degree of autonomy. In fact they first need to coordinate with their neighbors in order to select one of the maneuvers consistent with the broadcast activity plan.

The idea of redesigning the AHS using agent-based architectures prompts for a modification of this balance in favor of a greater decentralization of the control. We expect, in this way, to increase significantly the flexibility of the whole design and the reusability of its constituent parts.

Frazier and Newman ([7], section 2), for example, have observed, among the others, that the PATH AHS centralized design requires the link layer to have full knowledge of all the vehicles' physical specifications. As a consequence, with the present approach it is possible

to manage only vehicles populations with very limited heterogeneity. In addition they have pointed out that the full authority of the link and network layer over the sources of traffic would lead to a network with limited-access highways.

With this and other nontrivial motivations they have advocated a distributed approach ([7], section 3). In their REF they suggested that an IRVS agent architecture should consist of agents associated with autonomous vehicles, infrastructure entities (traffic signals, entrances/exits, bridges and tollbooths) as well as specialized control and observation units that could influence traffic behavior through dissemination of information. In this context, individual vehicles would be given a set of pro-active goals like reach a given destination, optimize travel time, safety, comfort and economy. Analogously, infrastructure agents would receive well-defined goals, e.g., maximization of intersection throughput and/or optimization of fairness by a traffic signal agent, and so on. It is also assumed that the individual vehicles would typically have conflicting goals and different priorities assigned to various goals. Furthermore, additional control agents can be accommodated as long as they act autonomously and respect, in turn, the autonomy of the vehicles and other entities. With this setting we are supposed to expect collective satisfaction to be *emergent* from individual behaviors.

In the following section we will provide a brief review of our agents taxonomy as a general methodology to approach the design of agent-based architectures and to quantify its *systems* and *functional* performance. By systems performance we mean the resources that are required by an agent-based system – network bandwidth, memory, computing time and others – whereas by functional performance we mean how well the adopted architecture solves the initial problem.

3 Our Taxonomy

The purpose of defining a taxonomic categorization of the agents according to their role within the system being modeled is twofold [5].

On the one hand we wish to provide a scientific approach to agent-based computing using concepts and paradigms from classical system engineering. On the other hand we want to develop a fully general and flexible method that could be naturally and easily applied to model real systems allowing quantifiable performance analysis.

In this respect we consider the IRVS problem an interesting case study that will help understand the features and the advantages of our methodology.

Our taxonomy consists of four categories of agents that we are going to describe through examples of applications and results:

Information agents: Obtain observations of the real world system state.

Consider, for example, the problem of monitoring a set of information sources $\{S_i\}$ that we cannot observe simultaneously in time due to bandwidth limitations. Suppose that those sources are modeled by random variables $s_i(t)$: the true state of source i at time t. Since, for tractability reasons, we do not expect the current state of the sources at a specific time t to be determined with absolute certainty, still we would like to develop a quantifiable notion of observability. To that purpose we defined the novel concept of (α, β)–currency. We say that our information agent is (α, β)–current if the probability that a randomly chosen source S_i is β–current – i.e. it has not changed within the last β units of time since the last observation – is at least α. Useful and concrete applications of these ideas to the web can be found in Brewington and Cybenko [3].

Modeling agents: Collect data from different Information agents to produce an estimate of the real world state based upon some abstract model.

Common examples of modeling agents are multi–object tracking systems or data fusion engines. Results on quantifiable models of such modeling agents, in the form of highly performing algorithms, can be found in Alberola and Cybenko [2].

Planning agents: Take a current estimate of the world state and produce a plan or course of actions. Those agents can task other agents to search for additional information or data or to execute some specific operations.

An example of planning agent problem is the "Traveling Agent Problem" (TAP) that can be stated as follows. One or more mobile agents looking for some information, distributed over a network, must plan a search itinerary that should result into the smallest expected time for completion or in the highest success probability within a given deadline. Results and solutions to the TAP can be found in Moizumi and Cybenko [12].

Brokering agents: This type of agent has no counterpart in classical system engineering. They are tasked to locate other types of agents dynamically, create a possibly unique system realization and facilitate interoperability between the different constituent agents. In classical control and systems engineering, everything is "hardwired" so that resource discovery and interoperability are not an issue.

We can mention two examples of performance modeling for brokering. The first one is based on measuring the success rate of locating desired resources given their presence within the network. The second one is based on measuring how reliably a broker matches keywords and other forms of requests with the correct agent resource [1, 4, 6]. In this case the problem is to verify that the matching between the functionalities of the requesting agent with those of the resource agent is semantically correct. This validation problem can be formulated as a machine learning process.

In the next section we will try to embed the IRVS problem into our framework.

4 Modeling the IRVS: a first proposal

We shall now approach the issue of introducing agents according to our taxonomy and defining how they interrelate. Unlike in the AHS PATH design, previously described, we would like to relax the rigid layered hierarchy to provide more elasticity to the whole architecture. Since we now suppress the central role of the link layer, each single vehicle will be asked to strive for the achievement of strategic goals as well as to solve tactical issues. In fact, as also pointed out by Frazier, Newman and Roszman [8], we can distinguish three levels of vehicle behaviors:

Tactical. At this level, our vehicle is an object, endowed with sensors and actuators and capable of some basic maneuvers like join, split, leader, follower and change-lane. It needs to take decisions, possibly in coordination with the neighboring vehicles, to achieve short–term objectives, without broadcast activity plans. These objectives include, for example, to become part of a convenient platoon with respect to the vehicle's maneuvering capabilities and the conditions of the "small" environment – the state of the current link. Examples of the state of the environment are: high or low vehicle density or presence of obstacles.

Strategic. At this level, our vehicle is a point moving along a circuit that has to plan a route in order to reach its final destination with minimum travel time and maximum safety, given the conditions of the "big" environment – the state of the roadway connections.

Examples of the state of the environment are: high or low congestion or unavailability of links, ramps or bridges.

Operational. The various selected maneuvers must be translated into control operations in order to be executed. The functionalities of this level include monitoring and control of the sensors and actuators to realize specific physical dynamics.

With respect to the three levels of behaviors, our taxonomy is transverse in that we can define information, modeling, planning or brokering agents realizing functionalities that belong to both the tactical and the strategic level. The agents that directly control the actuators at the operational level will be called *Actuator Agents* (AA).

Consistently with the idea of using concepts and paradigms from classical system engineering we will flexibly refer to the PATH 5 layers of functionalities as a starting point. So, among the others, there will be on-board system agents that embody the functionalities of the physical, coordination and regulation layer and roadway system agents that embody those of the link and network layer. On top of that there will be additional agents that serve the purpose of facilitating the non hierarchical interrelations between the layers.

Let us consider first the *on-board vehicle system.* We need to define two **information agents** (IA). The first one to monitor the sensors of the physical layer and the second one to be associated with the coordination layer and so in charge of exchanging data with the neighboring vehicles and with the roadway system. One **modeling** agent (MA) will collect data from the two information agents and produce an estimate of the state of the vehicle in the small and big environment (in this sense it serves both tactical and strategic purposes). One **planning agent** (PA) associated with the regulation layer will have to deliberate an action to be executed by an actuator agent. In case the maneuver must be aborted it will notify the coordination layer.

Now let us discuss the *roadway system.* For each link – piece of road of length 0.5 to 5 km – we define two groups of **information agents**. The first one associated with the roadway physical layer (roadsensor traffic) and the second one with the link layer. The link layer IA's must exchange data with the coordination layer IA aboard the vehicles and in addition they receive data from the on board MA about the state of the specific vehicle transiting within the link. One **modeling** agent and one **planning** agent, both associated with the Network Layer, will have to take care of the strategic issues. Those include, for example, the production of sets of itineraries that maximize the global roadway flow capacity. The situation is depicted in fig.1

4.1 Agents Overview

The following table summarizes the agent structure of the IRVS, according to our taxonomy.

IRVS Components	Our taxonomy
On-board Vehicle System	**Physical Layer IA**. It monitors the sensors aboard. **Coordination Layer IA**. It communicates with other fellow Coordination Layer IA's and with the Link Layer IA. It receives notifications also from the Regulation Layer PA. **On-board MA**. It gathers data from all the Info Agents and computes an estimate of the state of the vehicle. It provides a representation of the current state to the Regulation Layer Planning Agent and to the Roadway System. The Physical Layer IA provides only tactical information whereas the Coordination Layer IA provides both tactical and strategic information. **Regulation Layer PA**. It receives an estimate of the state of the vehicle from the on-board MA and deliberates a new **action** to be communicated to the Actuator Agent.
Local Roadway System: 0.5–5km	**Physical Layer IA**. It monitors the traffic through road-sensors along the link. Information sources at the physical layer include also data about the conditions of ramps, bridges, tunnels and tollbooths. **Link Layer IA**. This agent does not issue directives but provides strategic and tactical information to the on-board Coordination Layer IA. Strategic information include for example road directions, whereas tactical information include data about the global density conditions of the link, the inlet and outlet flow, the occurrence of exceptional conditions within the link and off the on-board sensors' range.
Global Roadway System	**Network Layer MA**. It collects data from all the Information Agents associated with the links and produces an estimate of the global state of the Roadway Network system. **Network Layer PA**. It receives data from the Network Layer MA and computes global strategic plans. For example it computes sets of viable **itineraries** ranked according to global optimization criteria. Its main goal is to maximize the overall network flow capacity. It returns the results of its computation back to the Link Layer IA's.

5 Future Work

We need first to decide more precisely which degree of autonomy should be granted to the single vehicles, now that the centrality of the link layer has been abolished.

Here is a list of steps that we believe need some attention.

- Definition of the mathematical models for the PA, MA and IA agents in order to quantify the architecture performance.
- Decision of how the various vehicles should negotiate with each other and with the roadway system in order to fulfill their individual goals.
- Finally, development or employment of a simulator to provide empirical evidence of the properties of the overall system: safety and high capacity. Verification of the effectiveness of our method through a comparison between the theoretical models and the experimental results.

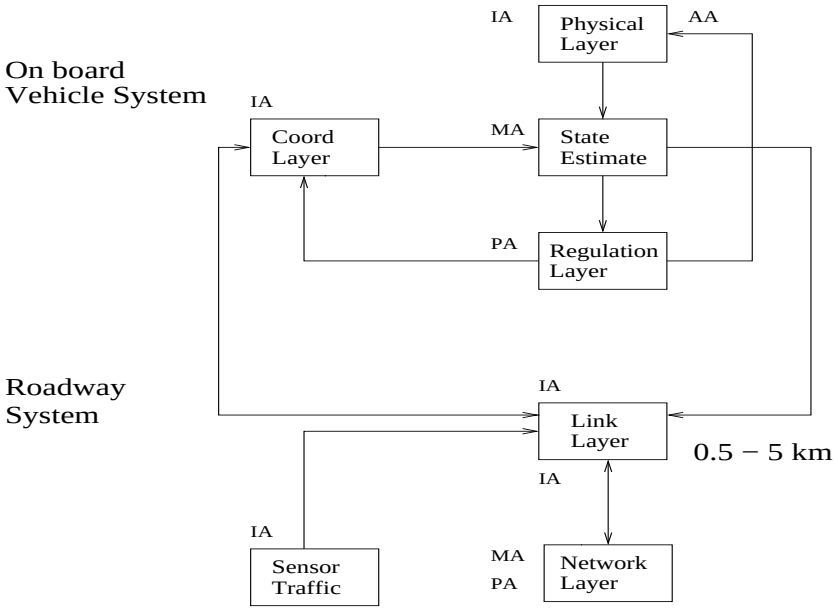

Fig. 1. AA = "Actuator Ag.", PA = "Planning Agent", MA = "Modeling Agent", IA = "Information Agent".

References

1. Ashley, K., Aleven, V.: Reasoning symbolically about partially matched cases. International Joint Conference on Artificial Intelligence, IJCAI–1997. San Francisco:Morgan Jauffmann, 335–41 (1997)
2. Alberola, C., Cybenko, G.: Tracking with Text-based Messages. IEEE Intelligent Systems, 14, (1999)
3. Brewington, B., Cybenko, G.: Who fast is the web changing. To appear in IEEE Computer (2000)
4. Clansey, W., J.: Notes on "Heuristic Classification". Artificial Intelligence, 59, 191–6 (1993)
5. Cybenko, G.: Agent-Based Systems Engineering. TASK Program Technical Proposal (2000)
6. Forgy, C., L.: Rete: a fast Algorithm for the many pattern/many object pattern match problem. Artificial Intelligence, 19, 17–37 (1982)
7. Frazier, T., Newman, A.: Intelligent Roadway Vehicle System. REF for the DARPA ISO TASK Program (2000)
8. Frazier, T., Newman, A., Roszman, L.: IVRS REF Scenarios. DARPA Progress Report, Dec (2000)
9. Godbole, D., Lygeros, J., Sastry, S.: Hierarchical Hybrid Control: a Case Study. Proceedings of the CDC (1994)
10. Godbole, D., Lygeros, J., Sastry, S.: Verified Hybrid Controllers for Automated Vehicles. IEEE Transactions on Automatic Control (1998)
11. Kotz, D., Jiang, G., Gray, R., Cybenko, G., Peterson, R.: Performance Analysis of Mobile Agents for Filtering Data Streams on Wireless Networks. Dartmouth College Tech Rep.
12. Moizumi, K., Cybenko, G.: The Traveling Agent Problem. To appear in Mathematics of Control, Signals and Systems (2000)
13. Varaiya, P.: Smart Cars on Smart Roads: Problems of Control. IEEE Transactions on Automatic Control, 38 (2) (1993) 195–207
14. Varaiya, P. and Horowitz, R.: Control Design of an Automated Highway System. In Proceedings of the IEEE (2000)

15. Wooldridge, M.: Intelligent Agents. In Gerhard Weiss, ed., Multiagent Systems, Weiss, MIT Press (2000) 27–77