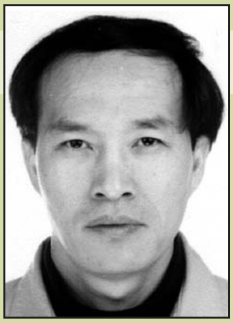

Editor: Fei-Yue Wang
University of Arizona and
Chinese Academy of Science
feiyue@sie.arizona.edu

Agent-Based Control for Networked Traffic Management Systems

Fei-Yue Wang, *University of Arizona and Chinese Academy of Sciences*

Agent or multiagent systems have evolved and diversified rapidly since their inception around the mid 1980s as the key concept and method in *distributed artificial intelligence*. They have become an established, promising research and application field drawing on and bringing together results and concepts from many disciplines, including AI, computer science, sociology, economics, organization and management science, and philosophy. Agent systems' success has lead to a modern definition of DAI: the study, construction, and application of multiagent systems—that is, systems in which interacting, intelligent agents pursue some set of goals or perform some set of tasks.[1]

However, multiagent systems have yet to achieve widespread use for controlling traffic management systems (see the sidebar for historical ties between agent research and traffic control). Since the introduction of modern control approaches, especially hierarchical control methods for urban traffic management problems, traditional control approaches based on functional decomposition have prevailed in both theoretical studies and practical applications.[2] Most research focuses on developing hierarchical structures, analytical modeling, and optimized algorithms that are effective for real-time traffic applications, as you can see from well-known traffic control systems such as Cronos, OPAC, Scoot, SCAT, Prodyn, and Rhodes.[3–5] Although those functional-decomposition-based systems are useful and successful for many traffic management problems, costs and difficulties associated with their development, operation, maintenance, expansion, and upgrading are often prohibitive and sometimes unnecessary, especially in the rapidly arriving age of connectivity. We need to rethink control systems and reinvestigate the use of simple task-oriented agents for traffic control and management of transportation systems.

Traffic and transportation management is well suited to an agent-based approach because of its geographically distributed nature and its alternating busy-idle operating characteristics. In future intelligent transportation systems, we should see autonomous intelligent agents traveling among traffic control centers, road intersections, highways, streets, vehicles, houses, offices, and malls, via the Internet or wireless and ad hoc networks. These agents will collect the right information at the right time and make smart decisions such that our transportation systems eventually will be "intelligent."

Recently, more studies have emerged on applying agent-based approaches to transportation. Examples include agents for implementing future carpooling, transportation scheduling, distributed control, and traffic simulation.[1,6] Although these problems are important, they don't yet systematically involve core ITS issues. Here I apply the concept of agent-based control for networked systems in control theory[7] to traffic and transportation management, calling for further research on and applications of an agent-based approach to ITS.

From control algorithms to control agents

In the past half century, control algorithms have been developed to run various industrial devices, including intersection traffic controllers. Normally, as the performance requirement increases, a control algorithm's complexity and cost increase too. For example, to make a refrigerator behave intelligently with enhanced functions requires a sophisticated control algorithm demanding more memory space and more processing power, and thus higher cost.

However, the age of connectivity will provide mobility that offers us the opportunity to control network-enabled devices with a new technology—agent-based control. This method decomposes a control algorithm into many simple task-oriented control agents distributed over a network, generally a wide-area network. Control agents that can be deployed and replaced over the network as operating conditions vary will run network-enabled devices. In this way, a network-enabled device can operate on a "control on demand" basis; that is, it will need to host only the operating agents, not all possible agents required for its operation. The idea is similar to that of "code on demand" in agent programming, which has been used successfully in many areas.

1541-1672/05/$20.00 © 2005 IEEE
Published by the IEEE Computer Society

So, agent-controlled network-enabled devices will require less memory and processing power than those operated by traditional control algorithms. This is significant because network-enabled devices are normally embedded systems for which memory space and processing power are key factors in determining their costs. Because manufacturers or service providers can develop and maintain control agents over the network efficiently and economically, agent-based control will lead to the construction of low-cost, yet reliable and intelligent, network-enabled devices for many applications.

(You might ask, what will happen to a network-enabled device when its network is disconnected or not working? When electricity isn't available, you must use candles instead of lights—that is, go back to the pre-electrical age. Similarly, when connectivity isn't available, you'll have a network-enabled device operated by its current or default control agents with limited functionality and performance. That is, you'll go from the age of connectivity back to the conventional isolated devices of the electrical age.)

The step from control algorithms to control agents is a natural development of control engineering in the age of connectivity. Control will become an independent entity instead of an affiliated function in system design. The development of a theoretical framework for agent-based control systems will significantly advance knowledge of control engineering in this new age. It will also have a broad, significant impact on many real-world applications, especially for home automation, traffic control, and vehicle electronic systems or telematics. In such applications, real-time requirements aren't extremely high, systems often have a long "resting" or idle period, and network connectivity is available or emerging.

Historical Ties between Traffic Control and Agents

One of the first few applications of agent technology was for air traffic control and conflict management,[1] which is still an interesting topic in agent research, such as in the US Air Force Research Laboratory's Amber project (see www.mesa.afmc.af.mil/html/ambr.htm). Another early milestone in agent research and distributed-artificial-intelligence history was the Distributed Vehicle Monitoring Task. With the DVMT, Daniel Corkill and Victor Lesser introduced *blackboards* for communication of data and goals by geographically distributed agents, each capable of sensing some portion of an overall area to be monitored.[2,3]

References

1. S. Cammarata, D. McArthur, and R. Steeb, "Strategies of Cooperation in Distributed Problem Solving," *Proc. 8th Int'l Joint Conf. Artificial Intelligence* (IJCAI 83), 1983, pp. 767–770.
2. D.D. Corkill and V.R. Lesser, "The Use of Meta-level Control for Coordination in a Distributed Problem Solving Network," *Proc. 8th Int'l Joint Conf. Artificial Intelligence* (IJCAI 83), 1983, pp. 748–756.
3. E.H. Durfee, *Coordination of Distributed Problem Solvers*, Kluwer Academic, 1988.

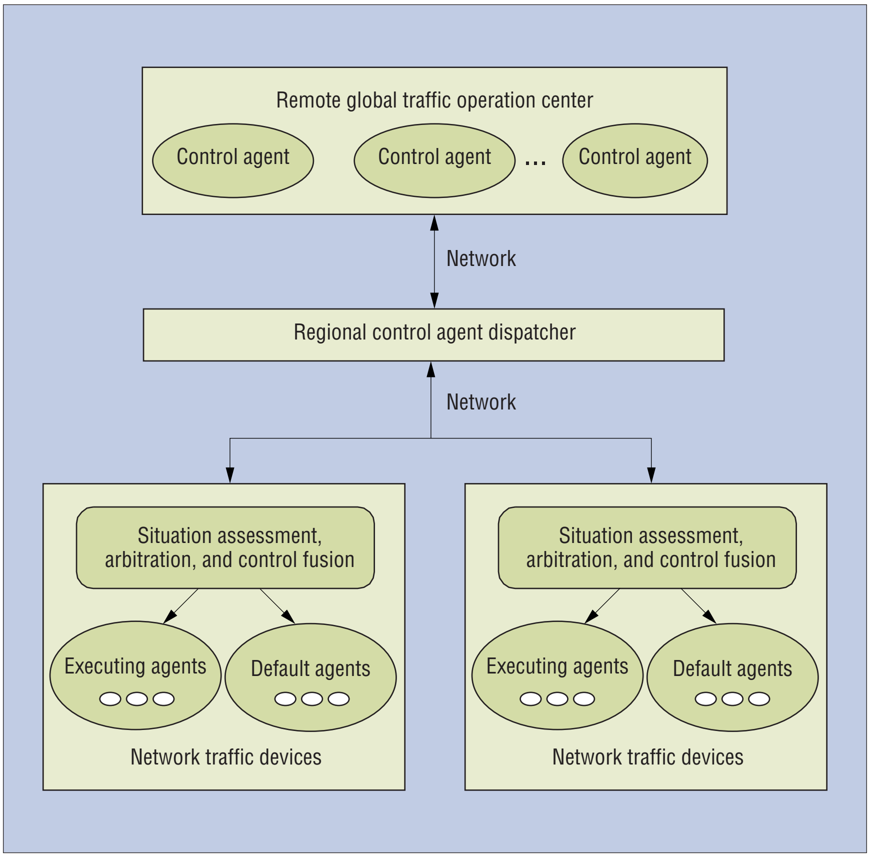

Figure 1. A hosting mechanism for traffic control agents.

A hosting mechanism for traffic control agents

In agent-based traffic control management, a traffic controller becomes a traffic agent host where different agents reside at different times in response to different traffic conditions. Figure 1 illustrates this hosting mechanism. Unlike in a traditional control system, where a control algorithm is an integral part of an isolated device and must be responsible for the entire operation, a control agent in an agent-based control system focuses on only a few specific operating conditions. Many simple task-specific control agents distributed over networks carry out the device's operation.

Default control agents reside in a traffic controller to ensure its basic operation and performance if connectivity isn't available. For example, we can use fixed-time control strategies to construct the default agents.

The controller hosts a number of *executing*

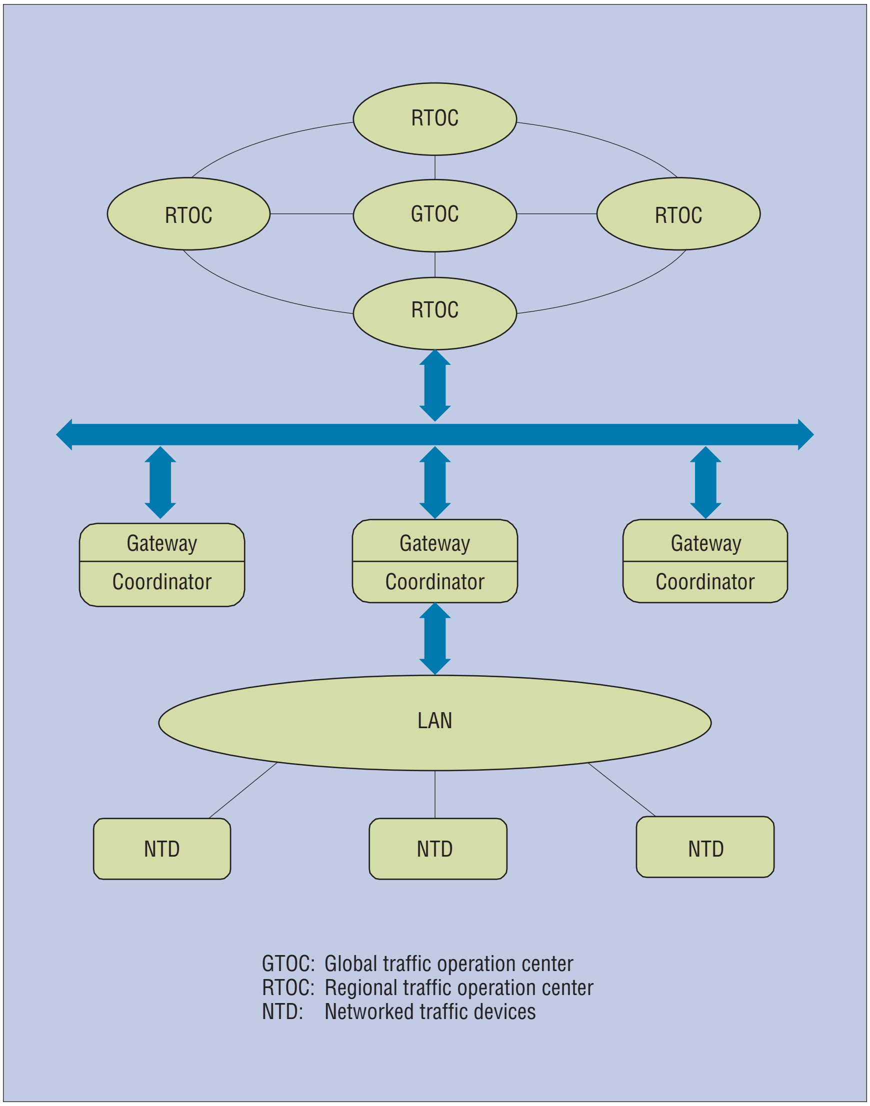

Figure 2. A system architecture for agent-based traffic management.

control agents that can deal with current traffic conditions with reasonable performance. The executing agents' decision-making process consists of *situation assessment*, *arbitration*, and *control fusion*. Situation assessment classifies the current external traffic and internal system states into predefined cases upon which control agents can make decisions regarding their actions. Executing agents will compete for the right to make the decision in a particular case. An arbitrator will determine which executing agent has the right to participate. If the arbitrator selects only one control agent, that agent will manage the traffic situation on behalf of all executing control agents. If the arbitrator selects multiple agents, they'll make their individual decisions, which a control fusion algorithm will combine into a single decision for the networked traffic management system.

The arbitrator also decides whether new control agents should replace some executing agents. New control agents reside outside the traffic controller at a remote *traffic operation center*; a *regional control agent dispatcher* (which I describe in more detail later) deploys them upon an arbitrator's request. A TOC can also recall default and executing control agents directly through the network.

Simple but effective algorithms for situation assessment, arbitration, and control fusion are key issues for the success of an agent-based control system for traffic management. To address these issues, we normally can use fuzzy logic or another rule-based approach.[8]

Owing to the local traffic devices' limited memory space and computing ability or to cost requirements, it's infeasible to make those devices powerful enough to host all possible control algorithms for high intelligence and performance. Agent techniques provide a novel way to address these problems, because an agent is a mobile object that can migrate from one network node to another one with its code, data, and thread. The proposed hosting mechanism enables the implementation and upgrading of new, better traffic management strategies without increasing or modifying the capacity of local hardware and software environments. This leads to low cost for high flexibility and intelligence for transportation system development, operation, maintenance, expansion, and upgrading in connected environments.

A system architecture for agent-based traffic management

To ensure a coherent control and communication mechanism among control agents, we must integrate and coordinate their activities. To this end, we can use the hierarchical architecture developed for intelligent control systems to divide an agent-based control system's structure into three levels: *organization*, *coordination*, and *execution*.[9] Figure 2 illustrates the architecture of agent-based control systems for operating and managing traffic and transportation systems in a distributed environment consisting of wide-area networks and local-area networks.

For example, consider an agent-based control system for networked urban intelligent traffic systems. A *global traffic operation center* develops and maintains various control agents for intersection traffic control, road incident detection, and other transportation activities. WANs connect the GTOC to several *regional traffic operation centers*. Each RTOC maintains a control agent repository and dispatches control agents to thousands of intersection controllers and traffic management devices at hundreds of locations in a region. At each location, a gateway downloads control agents from a WAN to traffic devices (controllers, sensors, and displayers) through a LAN and uploads information or requests from traffic devices to the RTOC. Each location also has a control agent coordinator for conducting cooperative control among traffic devices or even within a single device.

In this control system, the *agent organization level* mainly performs reasoning and planning for task sequences and organizes control agents to achieve specified transportation goals. It also develops and maintains control agents and provides protocols, algorithms, knowledge bases, and databases for agent communication, decision making, and learning. This level can incorporate methods from AI and knowledge systems.

The *agent coordination level* is the interface between the organization and execution levels. Generally, it consists of a dispatcher and coordinators. The dispatcher receives control agents from the organization level and deploys control agents to the appropriate coordinator through WANs on a control-on-demand basis. A LAN connects a coordinator to several networked traffic devices. The coordinator downloads control agents to devices and initiates collaboration between those agents. Normally, the dispatcher and coordinators are at different geographical places, the dispatcher is close to the GTOC, and the coordinators are close to networked traffic devices.

The *agent execution level* consists of embedded hardware and software units for deploying, replacing, hosting, and running control agents. Generally, this level consists of many *field-programmable-and-configurable devices* and is distributed among LAN-linked local systems connected by WANs. *Application-specific operating systems* (ASOSs) will be the key to effective, reliable real-time traffic operations at this level.

The proposed hierarchical, distributed architecture will enable the implementation of “local simple, remote complex” design of networked systems.[7] That is, we use simple strategies for local implementation while keeping complex facilities for remote enhancement such as refining, learning, optimization, and updating.

Software platforms and operation protocols

In the proposed three-level system architecture, the GTOC coordinates all RTOCs to achieve global traffic management objectives and optimize the entire system’s performance. To address the control agent management tasks, a GTOC software platform should provide these functions:

- A *control agent naming service* keeps track of control agent location to facilitate interagent communication.
- A *control agent factory* designs and creates control agents to implement intelligence for dynamic traffic control.
- A *control agent repository* stores and maintains control agents.
- *Performance management* monitors control agents and system runtime situations.
- *Task dependency* analyzes and decomposes the dependency relationships among various control and service tasks and provides a basis for failure recovery.
- *Resource allocation* manages system resource reservation and allocation mechanisms and implementation.
- An *interoperability and legacy management facility* integrates existing out-of-date or heterogeneous traffic systems.

Clearly, an effective global traffic operation center software platform must provide a coherent mechanism for the operations of agent-based distributed control systems.

- *Quality of services* builds negotiation mechanisms to guarantee hard to soft real-time requirements for traffic operations.
- *Failure recovery management* looks up failure causes and corresponding recovery schemes on the basis of service dependency analysis.
- A *security service* provides mutual-trust mechanisms for agents.
- *System management* provides the general system management services.

A regional control agent dispatcher coordinates regional network communications, implements control agent caching, and provides regional coordination of control agents. The local *gateway and coordinator* provides a single integration point for a range of agent-based services in local network environments. It works as a network communication gateway and coordinates local control agents.

Clearly, an effective GTOC software platform must provide a coherent mechanism for the operations of agent-based distributed control systems. We can combine mobile-agent programming and CORBA (common object request broker architecture) to create such a flexible software platform and achieve control on demand. That is, the control functionality can be deployed to the controller according to the demands of various control scenarios. In this software platform, we could encapsulate the required control algorithms or control parameters into mobile agents that could migrate from remote traffic control centers to field traffic devices or from one field device to another.

This software platform should benefit from the complementary properties of mobile agents and the CORBA communication infrastructure. On the one hand, the mobile agent expands CORBA’s static-object concept. That is, an object can move from one node to another until it completes its task. Also, mobile agents’ “dynamical adaptability” lets them sense their execution environment and react autonomously to changes. In addition, the asynchronous, autonomous execution of mobile agents makes them suitable for fragile network environments.

On the other hand, CORBA provides mobile agents necessary services such as these:

- The *naming service* keeps track of agent locations.
- The *lifecycle service* defines services and conventions for creating, deleting, copying, and moving CORBA objects.
- The *externalization service* provides a standardized mechanism for mobile-agent migration during which the state of mobile agents could be recorded into, or reconstructed from, a network data stream.
- The *security service* guarantees mobile agents’ execution.

Furthermore, the adoption of CORBA as the underlining communication infrastructure facilitates the integration of heterogeneous control systems and legacy control systems that could be non-mobile-agent-based or even non-object-oriented.

We need an operation protocol for agent-based control for traffic management so that third parties can develop and provide various traffic agents for services. The *Open Services Gateway Initiative* (www.osgi.org) is an open specification for the

Fei-Yue Wang is a professor at the University of Arizona's Systems & Industrial Engineering Department and the director of the university's Program in Advanced Research of Complex Systems. He's also the director of the Chinese Academy of Sciences' Key Laboratory of Complex Systems and Intelligence Science. Contact him at the Systems & Industrial Eng. Dept., Univ. of Arizona, Tucson, AZ 85721; feiyue@sie. arizona. edu.

delivery of multiple services over WANs to local networks and devices. The OSGi model's architecture proposes *service gateways*, which connect smart devices and enable them to seamlessly communicate with each other or to receive a variety of services from providers. A service gateway is a bridge between an external network, typically the Internet, and a local network consisting of, for example, Ethernet or a field bus. It provides flexible interfaces and an environment for service providers to deploy and maintain services to users at the end of local networks. So, the scenario considered by OSGi is almost identical to that of agent-based control systems. Consequently, OSGi architectures and modules have great potential for direct applications in agent-based control for networked traffic and transportation management.

Agent-based control methods provide a cheap, reliable, and flexible approach for intelligent, effective control and management of traffic and transportation systems in connected environments. We should develop a comprehensive operational framework that will connect networked traffic devices, enabling them to seamlessly communicate and cost-effectively receive a variety of traffic-related services from providers via the Internet. To this end, we must work on such goals as

- a theoretical framework for agent-based control systems for traffic management,
- a software platform for agent-based distributed traffic control systems, and
- a hardware platform for remote programmable and configurable traffic devices.

To reach these goals, we can use mobile-agent theory, behavior programming, intelligent control, an OSGi-based infrastructure, and ASOS-based embedded devices. Based on those ideas, a prototype system called aDAPTS (*A*gent-Based *D*istributed and *A*daptive *P*latforms for *T*ransportation *S*ystems) has been developed at the Chinese Academy of Sciences. ■

Acknowledgments

This work is supported partly by grants 60334020 and 60125310 from China's National Natural Science Foundation and grant 2002CB 312200 from China's Ministry of Science and Technology.

References

1. G. Weiss, ed., *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 2000.
2. G.N. Saridis and C.S.G. Lee, "On Hierarchically Intelligent Control and Management of Traffic Systems," *Proc. Eng. Foundation Conf. Research Directions in Computer Control of Urban Traffic Systems*, 1979, pp. 209–218.
3. F.-Y. Wang et al., "Toward Intelligent Transportation Systems for the 2008 Olympics," *IEEE Intelligent Systems*, vol. 18, no. 6, 2003, pp. 8–11.
4. M. Papageorgiou et al., "Review of Road Traffic Control Strategies," *Proc. IEEE*, vol. 92, no. 12, 2003, pp. 2043–2067.
5. P.M. Mirchandani and F.-Y. Wang, "RHODES to Intelligent Transportation Systems," *IEEE Intelligent Systems*, vol. 20, no. 1, 2005, pp. 10–15.
6. N.R. Jennings, K. Sycara, and M. Woodbridge, "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, 1998, pp. 7–38.
7. F.-Y. Wang and C.-H. Wang, "Agent-Based Control Systems for Operation and Management of Intelligent Network-Enabled Devices," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, 2003, pp. 5028–5033.
8. F.-Y. Wang and H.-M. Kim, "Implementing Adaptive Fuzzy Logic Controllers with Neural Networks: A Design Paradigm," *J. Intelligent and Fuzzy Systems*, vol. 3, no. 2, 1995, pp. 165–180.
9. F.-Y. Wang and G.N. Saridis, "A Coordination Theory for Intelligent Machines," *Automatica*, vol. 26., no. 5, 1990, pp. 833–844.