Agents in Industry: The Best from the AAMAS 2005 Industry Track

Michal Pechoucek, Gerstner Laboratory, *Czech Technical University*
Simon G. Thompson, *British Telecom*

Agent technology provides industrial-applications developers with new abstractions for distributed-system development, new methodological tools, and a set of algorithms for creating autonomous, collaborative systems.

Over the past few years, a number of industrial applications have deployed agents. However, a substantial gap still exists between the cutting-edge research carried out mainly in university laboratories and research institutes and the domain-specific industrial applications that commercial organizations develop.

The articles in this department intend to give some indication of agent technology's readiness for commercial deployment, based primarily on the presentations and discussions at the inaugural Industry Track of AAMAS 2005—the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems.

Opportunities for agent deployment

Multiagent systems and autonomous-agent technologies provide a design-and-implementation paradigm for software solutions based on collective decision making in a community of autonomous, loosely coupled computational entities. Many agent development environments and agent integration platforms are available either commercially or in open-source format ready to deploy in commercial applications.

The agent research community has consolidated significantly in the last few years. In particular, the formalization of agent-based computing's foundations has positioned the domain in relation to adjacent fields of theoretical research such as formal logic, game theory, theorem proving and model checking, distributed and parallel computing, scalability, and complexity theory. The community is also involved in much research that's closer to potential business applications, such as the Semantic Web, open systems, and ubiquitous computing. The achievements in these fields form a solid foundation for technology transfer from university labs and research institutes to industrial applications.

The agent paradigm and the available agent techniques perform well in five types of domains. The first is competitive and noncooperative domains, where information-sharing restrictions prevent a centralized decision-making architecture—for example, e-commerce applications, supply-chain management, and e-business. In such domains, the agent paradigm is employed to design and describe mainly Web-based systems.

In the second type of domain, the data required for automated decision making aren't centrally available. The usual reasons for this are the geographical distribution of knowledge (for example, logistics, collaborative exploration, mobile and collective robotics, or pervasive systems) or environments where communication is partially or temporarily inaccessible. Other reasons include temporal distribution (for example, satellite networks where satellites have different views of the earth at different times of the day) and conceptual distribution (for example, in layered hierarchies, where entities at one layer might have no knowledge of events or processes at other layers, as in the Internet or supply chains).

The third type of domain requires survivable time-critical response and high robustness in distributed scenarios. Example domains include time-critical manufacturing or industrial-systems control that requires replanning or fast local reconfiguration to handle problems instantly.

The fourth type of domain involves simulation and modeling. Using agents for simulation has been common. Agents can be deployed either in simulations requiring easy migration to the real environment or where traditional simulation techniques are expensive.

The final type of domain involves open-systems engineering. Early agent deployment projects emphasized such domains, but the reality of the implementations delivered so far hasn't met expectations. Even though using ontologies and FIPA (Foundation for Intelligent Physical Agents) standards has addressed many syntax issues, semantic-integration issues remain problematic. Web services and Web technologies in general seem to have taken the lead in applications in this area.

In our experience, industrial organizations frequently request (and agent technology developers frequently provide) these functionalities:

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

- planning,
- scheduling,
- resource and strategic decision making,
- diagnostics,
- control and real-time replanning,
- software systems integration,
- interoperability,
- knowledge integration,
- ontologies, and
- simulation and modeling.

Despite some successful case studies in industry, agent technology has suffered from hype and a loss of momentum. This loss has seen many unique properties of the early tools and techniques subverted by changes in the commercial environment (for example, the Web's emergence as the key corporate application platform, the dot-com crash, and Microsoft's emergence as the dominant provider of desktop-personal-productivity software) or by the development of rival technologies such as Enterprise JavaBeans, overlay networks, and Web services. Recently, however, the momentum has reversed course as academic agent research programs have borne fruit and particularly as agents have incorporated advanced techniques from other AI areas. Key examples of such advances are the

- utilization of efficient, powerful planning algorithms,
- development of OWL,
- development of algorithms to reason about action in teams,
- development of efficient market-clearing mechanisms, and
- refinement of the general principles and architectures of agents—specifically, BDI (belief-desire-intention).

These techniques have enabled the implementation of applications having a clear advantage over traditional systems.

In addition, improved computer hardware and increased availability of open source components, especially for networking and software development, have made development of effective agent applications cheaper, faster, more reliable, and, above all, easier.

Another reason for agent technologies' increased momentum is that the human capital available to practitioners has developed rapidly as generations of students who have been exposed to the principles and possibilities of agents and AI have joined the workforce. Clearly, without the skills and vision to implement these techniques into practical solutions, progress is impossible.

The AAMAS Industry Track

Conferences organized by the agent research community frequently discuss "blue sky" research ideas (ideas that aim beyond immediate application), theoretical- and empirical-research results, and agent technology's potential and actual applicability. For example, AAMAS is an annual meeting of agent technology researchers and practitioners that has become the canonical forum for the presentation of new results in the field. The conference resulted from the merger of three successful conferences (the International Conference on Multiagent Systems; Agent Theories, Architectures, and Languages; and the International Conference on Autonomous Agents). In an encouraging development, for the first time, a special track at AAMAS 2005 covered research on the industrial application of agents.

The Industry Track featured reports on defense and exploration applications and reports from commercial business operations.[1]

In the aerospace applications session, NASA presented a monitoring agent for space shuttle launching criteria, and the Jet Propulsion Laboratory presented an autonomous science agent flying onboard the Earth Observing One spacecraft. Two presentations covered defense applications on autonomous control and teamwork of unmanned aerial vehicles, and one covered an agent-based simulation application for naval training.

In the logistics-and-transport session, the Catholic University of Lueven presented a decentralized approach to autonomous-guided-vehicle control for warehousing. Whitestein Technologies and Magenta Technology offered their solutions for transport optimization in industrial logistics. University Jaume I Castellón researchers described their concept of agent deployment for traffic management and control. The ECN (Energy Research Center of the Netherlands) reported on the successful application of agent technology in electricity infrastructure control.

In the manufacturing session, Rockwell Automation's presentation on agent-based industrial control represented a traditional manufacturing industry. Another presentation detailed a system that the DFKI (German Research Center for AI) developed for planning and monitoring steel production, and a Singapore Institute of Manufacturing Technology presentation dealt with semiconductor assembly. Two presentations in this session focused on general conclusions instead of specific applications. The Czech Technical University compared the success and potential of agent deployment in defense and manufacturing applications. Tom Wagner, Les Gasser, and Mike Luck gave a panel-like presentation on agent technology's potential impact.

The following short articles summarize what we consider to be the four best contributions to that track. Our goal is to convey the event's key contributions to a wider audience than just the attendees or those who have time to read the entire proceedings. Unfortunately, we couldn't include every significant presentation or mention every important discussion point, and many attendees might disagree with our perspective. However, we hope that this personal view serves as evidence of agent technology's real and considerable impact.

Reference

1. M. Pechoucek, D. Stainer, and S. Thompson, eds., *Proc. 4th Int'l Conf. Autonomous Agents and Multi-Agent Systems—AAMAS 2005 Industry Track*, ACM Press, 2005.

Variable-Autonomy Control of Teams of Uninhabited Air Vehicles

Jeremy W. Baxter and Graham S. Horn, *QinetiQ*

Uninhabited air vehicles are of particular interest to the defense sector because they could significantly reduce risk to aircrews. Current UAV systems typically require multiple operators to control a single platform. QinetiQ has been developing an approach that lets one operator control multiple platforms.

The basic concept is a decision-making partnership between a human operator and an intelligent uninhabited capability. (A capability is a collection of platforms, sensors, and weapons.) The human provides mission-level guidance to the pool of cooperating UAVs and takes on a largely supervisory role. The UAVs self-organize to achieve the goals the operator sets, such as to observe an area or to locate and destroy a high-value mobile target. Owing to regulatory or liability issues, a human must make

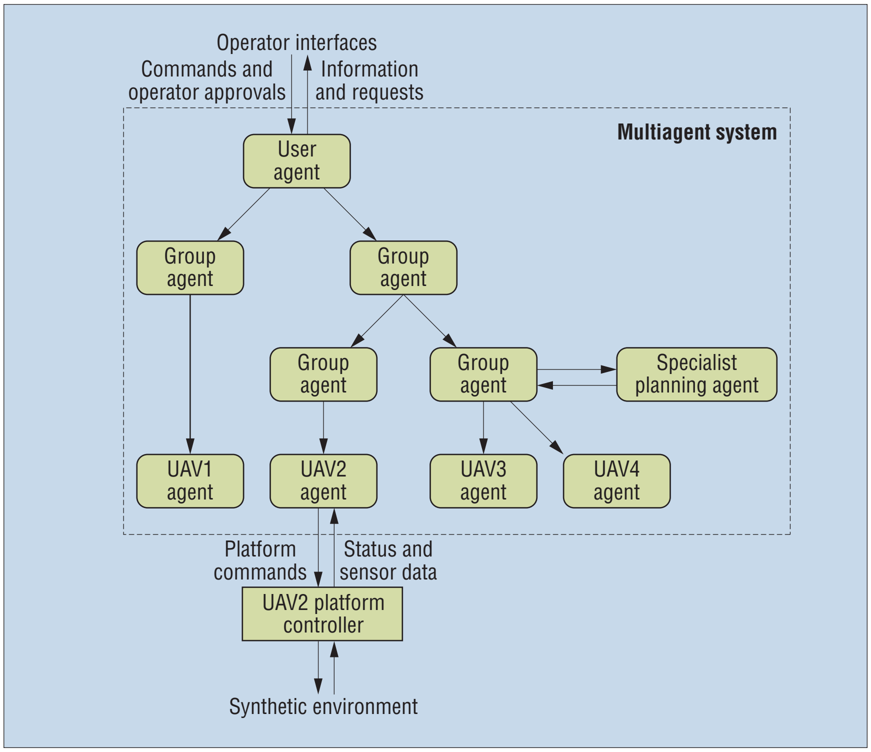


Figure 1. The main components of a trials system for controlling uninhabited air vehicles. Each UAV agent interacts with a platform controller that's connected to the synthetic environment.

some critical decisions, such as weapon release. So, the uninhabited capability must refer such decisions to the operator. We've implemented this concept using a variable-autonomy interface onto a multiagent system, as part of a larger trials system.

The trials system

We use the trials system (see figure 1) to evaluate potential concepts of use and technologies. It includes a synthetic-environment (SE) simulation that models real-world dynamic interactions. Human-in-the-loop trials let us capture the key requirements for the decision-making partnership. The system elements have evolved in response to feedback from trials (subjective comments and objective performance measures) and changes to the concepts of use.

A multiagent system provides a natural, powerful way to represent multiplatform tasks and sets of coordinated, cooperating agents. The trials system contains four types of agent (see figure 1). The *user agent* allocates individual UAVs to the tasks that the operator sets, and provides the operator with information. *Group agents* plan and coordinate a task's execution, sometimes calling on the capabilities of *specialist planning agents*. *UAV agents* interact directly with individual platforms, commanding the autopilot to undertake specific maneuvers and receiving status and sensor information. The user agent routes requests for critical decisions to the variable-autonomy interface. Depending on the autonomy level (set for each request type), the interface will either automatically grant permission to continue or defer to the operator.

Group agents embody the knowledge of how to plan and execute coordinated team tasks using a framework based on joint-intentions theory.[1] We originally designed the framework to enable robust execution of user orders by teams of entities in ground-based battlefield simulations.[2] It provides a solid grounding for the communication necessary to keep a team task coordinated. Originally, it contained only group and vehicle agents and didn't let an operator issue new tasks during execution (at start-up, it provided each team with a single order that could be decomposed into orders for subgroups). Adding the user agent allows for operator interaction and parallel tasking.

The trials scenario

The scenario for the trials is a time-critical targeting mission against a high-value mobile target. The system deploys a package of four UAVs, containing a variety of sensors and weapons, to locate and destroy the target. The operator is the pilot of a single-seat fighter. The mission consists of two main phases: search and attack.

One specialist planning agent produces plans for the search phase. It expands a set of possible target positions into regions that a moving target could reach in the next few minutes. The agent plans routes that let the UAVs efficiently search these regions with short-range sensors and take images of potential targets that the operator will classify.

The attack phase can begin when the operator has classified a ground entity as the high-value target. Another specialist planning agent provides access to a dynamic scheduler[3] that allocates UAVs to the tasks they must execute during the attack phase: release the weapon and gather images of the target to see if the weapon has destroyed it.

We've used the multiagent system in three trials. A single pilot was able to successfully control the UAV team to complete the missions. Including the agents in the trials system has allowed the quicker completion of more complicated missions, with reduced operator workload.

Acknowledgments

This research was part of the UK Ministry of Defence Output 3 research program on behalf of the Director Equipment Capability—Deep Target Attack. We gratefully acknowledge their support. We're part of a QinetiQ team that's developing and implementing the decision-making partnership concept; we focus on the multiagent-system element.

References

1. H. Levesque, P. Cohen, and J. Nunes, "On Acting Together," *Proc. 8th Nat'l Conf. Artificial Intelligence* (AAAI 90), AAAI Press, 1990, pp. 94–99.
2. J.W. Baxter and G.S. Horn, "Executing Group Tasks despite Losses and Failures," *Proc. 10th Conf. Computer Generated Forces and Behavioral Representation*, 2001, pp. 205–214.
3. M.J.A. Strens and N. Windelinckx, "Combining Planning with Reinforcement Learning for Multi-Robot Task Allocation," *Adaptive Agents and MAS II*, D. Kudenko et al., eds., LNAI 3394, Springer, 2005, pp. 260–274.

The PowerMatcher: Multiagent Control of Electricity Demand and Supply

Koen Kok, Cor Warmer, and René Kamphuis, *Energy Research Center of the Netherlands*

Distributed generation of electricity is providing an increasing part of the worldwide energy supply. DG consists of different sources of electric power connected to the distribution network or to a customer site. This approach is distinct from the traditional central-plant model for electricity generation and delivery. Examples of DG are photovoltaic solar systems, small and medium-scale wind turbine farms, and the combined generation of heat and power (CHP).

When the share of DG increases in a geographical area, clustered control of DG by common ICT (information and communication technology) systems can add value. As a result, distribution networks are expected to evolve from a hierarchically controlled structure into a network of networks, in which a vast number of system parts communicate with and influence each other. The number of components actively involved in coordination will be huge. Centralized control of such a complex system will reach the limits of scalability and communication overhead.

A key technology for solving this problem is market-based control. In market-based control, many control agents competitively negotiate and trade on an electronic market to optimally achieve their local control action goals. Use of market-based control in the electricity infrastructure opens the possibility for distributed coordination in addition to the existing central coordination.

The PowerMatcher

The PowerMatcher method provides market-based control for *supply-and-demand matching* (SDM) in electricity networks with a high share of DG. It's based partly on earlier research by Fredrik Ygge and Hans Akkermans;[1] Hans Akkermans, Jos Schreinamakers, and Koen Kok;[2] and Per Carlsson.[3] In this method, a control agent represents each device. The agent tries to operate the device process in an economically optimal way, within the process's constraints. The agents negotiate their electricity consumption or production on an electronic exchange market. The resulting market price determines the power volume allocated to each device.

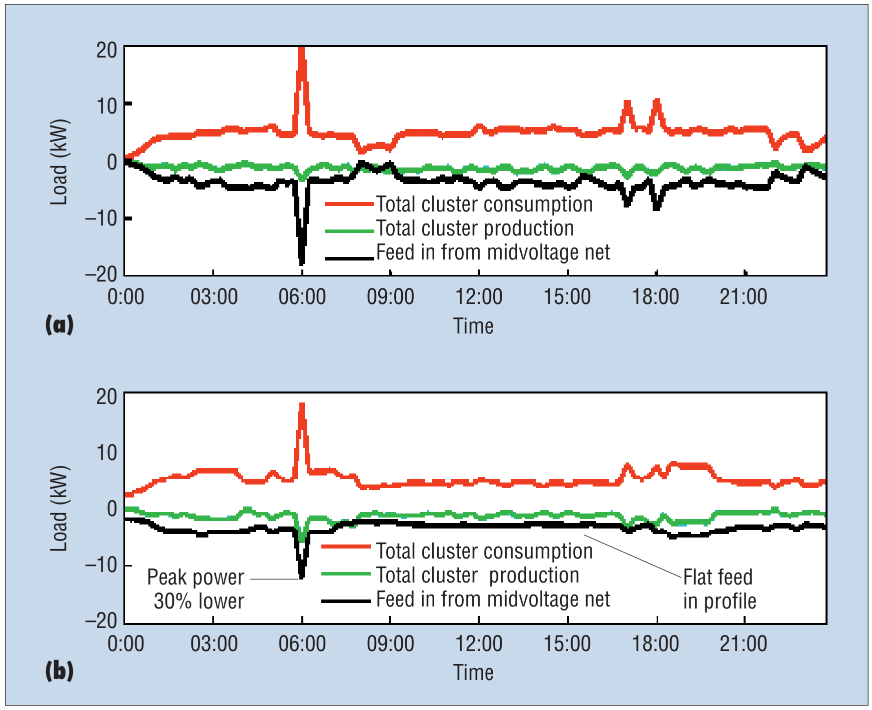


Figure 2. Results of a simulation of residential electricity distribution: (a) free-running devices; (b) market-based control agents match supply and demand. Multiagent control leads to peak load reduction and power profile smoothing.

From the viewpoint of controllability, devices that produce or consume electricity fall into six classes, each having a specific agent strategy. We look at three in this article. The first class consists of *stochastic-operation devices*, such as solar and wind energy systems, where the power exchanged with the grid behaves stochastically. The second class is *shiftable-operation devices*, which must run for a certain amount of time regardless of the exact moment and thus are shiftable in time. An example of such a device is a ventilation system in a utility building that needs to run for 20 minutes each hour. The third class comprises *user action devices*, whose operations result from a user's direct action. Examples include audio and video devices, lighting, and computers.

Local agents' self-interested behavior causes electricity consumption to shift toward moments of low electricity prices and causes production to shift toward moments of high prices. So, SDM emerges on the global-system level.

A simulation

To investigate distributed SDM's impact for a residential area, we simulated a cluster of 40 houses, all connected to the same segment of a low-voltage distribution network. Heat pumps (electricity consumers) heated 20 of the dwellings; micro-CHP units heated the other 20. The simulation treated washing machines as shiftable-operation devices with a predefined operational time window, photovoltaic solar cells as stochastic-operation devices, and lighting as user action devices.

Figure 2 shows the result of a typical simulation run. In both plots, a single plotline indicates the total consumption and production, and we treat production as negative consumption. In figure 2a, all devices are free running; in figure 2b, the market-based control agents match supply and demand.

This simulation shows that our method can exploit flexibility in device operation through agent bids in an electronic power market. The peak in electricity demand is substantially lower in the controlled case. From the viewpoint of network operations, this result is important, because the highest expected peak demand determines the needed network capacity (transformers and cables). Reducing this peak reduces network investments. Furthermore, introducing SDM results in a flatter, smoother

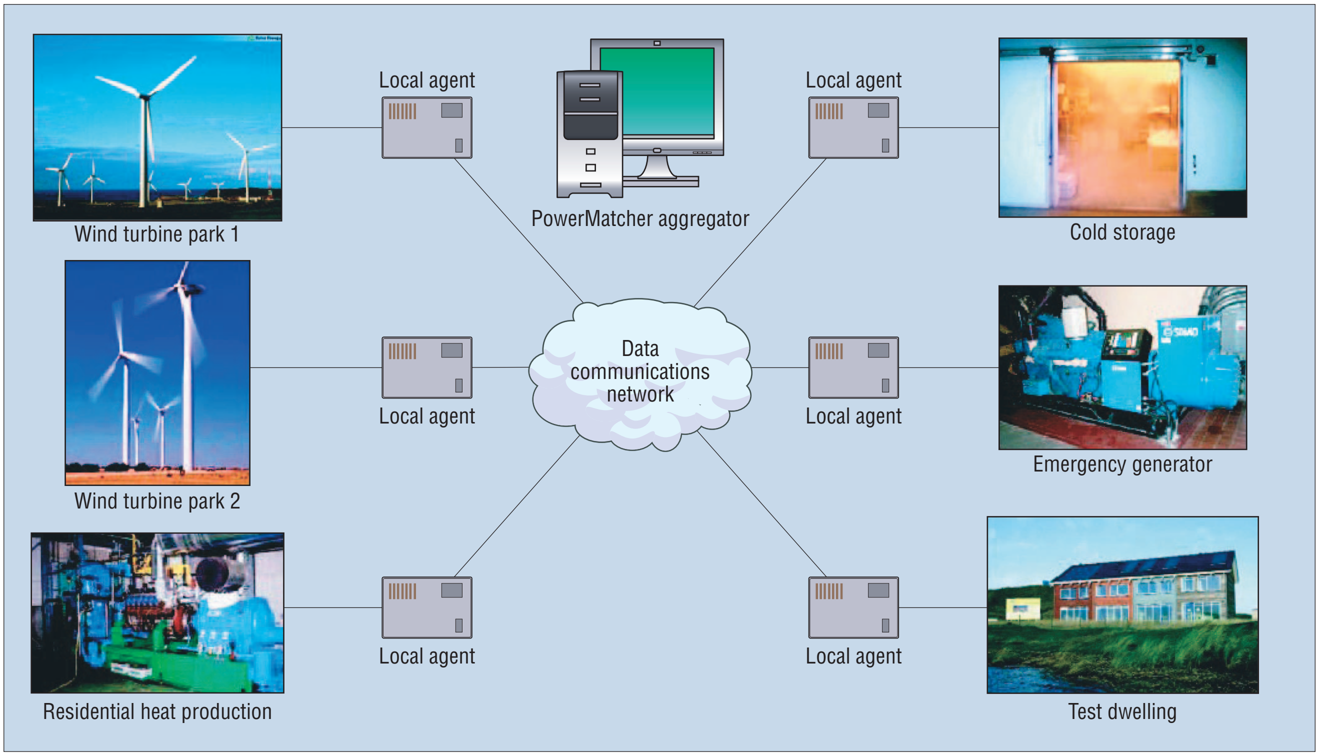


Figure 3. Control agents balancing a real-world commercial-trade portfolio.

profile of the electricity fed in from the midvoltage network. This result is interesting from the viewpoint of electricity trading, where increased predictability of both production and consumption adds value.

Field testing

We're investigating the PowerMatcher in real-life environments for two different business cases. One aims to automatically reduce the imbalance in a commercial trader's real-world portfolio by aggregating medium-sized industrial electricity producing and consuming installations (see figure 3). In this experiment, overproduction and underproduction of wind parks induce price changes on the cluster's electronic market. The other devices' control agents react to this with counteractions, which restore the cluster's energy balance. The first test results indicate a decrease of the total power imbalance by approximately 25 percent. Reduction of unpredictability in the trade portfolio reduces imbalance costs charged to the trader by the independent network operator.

The other field test, on a cluster of micro-CHP units operating as a virtual power plant, demonstrates their ability to contribute to a common control goal. This experiment uses 15 domestic heating systems at consumer premises. The virtual power plant can provide value though electricity trading or local-grid-operation support.

Acknowledgments

The European Commission partially supported this research in the context of the EUSUSTDEV Project NNE5-2001-00906, called CRISP (Distributed Intelligence in *Cr*itical *I*nfrastructures for *S*ustainable *P*ower).

References

1. F. Ygge and J.M. Akkermans, "Resource-Oriented Multi-Commodity Market Algorithms," *Autonomous Agents and Multiagent Systems*, vol. 3, no. 1, 2000, pp. 53–71.

2. J.M. Akkermans, J.F. Schreinemakers, and J.K. Kok, "Emergence of Control in a Large-Scale Society of Economic Physical Agents," *Proc. 3rd Int'l Joint Conf. Autonomous Agents and Multiagent Systems* (AAMAS 04), IEEE CS Press, 2004, pp. 1230–1231.

3. P. Carlsson, "Algorithms for Electronic Power Markets," PhD thesis, Dept. of Information Technology, Uppsala Univ., 2004.

Manufacturing Agents at Rockwell Automation

Vladimír Marík and Pavel Vrba, *Rockwell Automation Research Center, Prague*
Kenwood H. Hall and Francisco P. Maturana, *Rockwell Automation Advanced Technology Laboratory*

As the complexity of manufacturing business environments grows, multiagent-systems (MAS) technology is becoming increasingly important for development of highly distributed, robust, and flexible industrial-control architectures. From an MAS viewpoint, the manufacturing system is a community of highly distributed, autonomous, efficiently cooperating, and asynchronously communicating units—agents—integrated by the plug-and-operate approach.

Rockwell Automation Inc. manufactures industrial-automation technology, leading the US market in discrete automation and control products. In 1995, its first industrial-agent project optimized machine load balancing and increased reliability of a steel rod bar mill. It is currently applying MAS technology to its flagship product, ControlLogix programmable logic controllers (PLCs), and developing the MAST

(Manufacturing Agent Simulation Tool) agent simulation infrastructure to support design and validation.

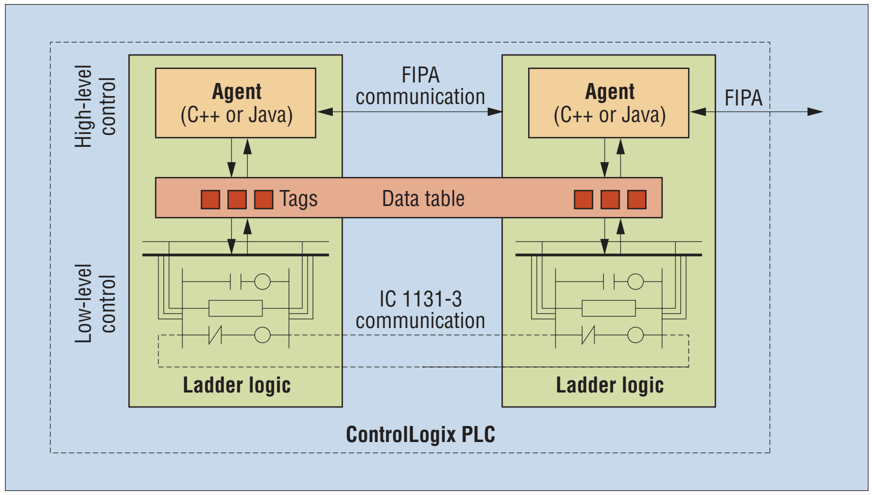


Figure 4. A real-time control agent architecture for the ControlLogix programmable logic controller. (FIPA stands for the Foundation for Intelligent Physical Agents; IEC stands for the International Electrotechnical Commission.)

Real-time control agents and simulation

Rockwell Automation's control agent architecture usually implements each agent as a module that encapsulates both the real-time control subsystem and the software agent (see figure 4). The RT control subsystem directly handles the information from physical sensors and actuators in real time and is programmed in a low-level language (usually the ladder logic programming language). The software agent is implemented in a higher-level programming language (usually C++ or Java) and handles decision making and negotiation.

The important part of this solution is an efficient runtime interface allowing both information transfer from the RT control subsystem (I/O and other control or diagnostics data) to the software agents and propagation of the agents' control actions to the RT control subsystem. To simplify this system's integration with existing industrial-automation-control architectures based on PLCs, we gave the agents direct access to the PLC's data memory so that they can observe and influence the RT control subsystem directly.

When thinking of a real industrial deployment of agents that requires high reliability and strict adherence to real-time constraints, you must abandon the idea of hosting the software agents on a PC and interfacing them to PLCs. Therefore, we have modified the ControlLogix PLC's firmware so that the C++ agents can run directly inside the PLC in parallel with the ladder logic code. Rockwell has developed the Autonomous Cooperative System as a C++-based agent platform dedicated to ControlLogix PLCs. The ACS lets us distribute the agents across several PLCs (one PLC usually hosts several agents). It also supports agent management services (registration, deregistration, services lookup, and so on) and ensures the transport of messages conforming to FIPA (Foundation for Intelligent Physical Agents) standards among the agents.

The ACS's first application was the development of a reconfigurable control system for a US Navy ship's *chilled-water system*[1] that increased the system's survivability. An individual agent controls each element of the physical CWS equipment (valve, cooling unit, piping section, and so on). When an agent's built-in diagnostic module detects a failure, the agent initiates negotiation with other agents to reconfigure the CWS—for example, finding an alternative path for water in the piping section to avoid a broken part.

To test and validate the agent-based control system before deploying it in a manufacturing environment, simulation is indispensable. The simulation must emulate the manufacturing equipment or processes; for this, we strongly prefer commercially available simulators such as Matlab or Arena. Once the simulation proves that the agent-based control system is mature enough to deploy, we replace the simulation with the real physical system. This shift must be as smooth as possible, preferably without any modifications to the agent code developed for the simulation. Because the agents will interact with the physical system by sharing the control data in the PLC memory, we use this mechanism also to share data with the simulation.

For example, for agent-based control of the CWS, we implemented the simulation in Matlab and Simulink. After verification and testing, we successfully deployed the unchanged agent-based control system to control the valves, cooling units, and other actual equipment of a scaled-down physical model of the ship.

MAST

This simulation environment, which Rockwell Automation developed and implemented in Java, serves mainly as an agent-based demo implementation for material handling in flexible manufacturing. The developed agent library represents material-handling systems' basic components such as work cells, conveyor belts, and switches (diverters). Agent cooperation focuses on finding the optimal transportation routes in the system. The proposed solutions provide fault tolerance and structural flexibility. You can emulate any component failure (for example, any conveyor belt failure), which causes the agents to negotiate an alternative route to avoid the broken component. You can add new components (representing new transportation capabilities) to the system or remove existing ones on the fly.

Recently, Rockwell Automation has extended the MAST environment to simulate the holonic-packing-cell testbed at the University of Cambridge's Centre for Distributed Automation and Control (see figure 5). They have extended MAST's agent library with a set of agents to represent and control particular components of the lab's equipment such as a Fanuc M6i robot, a storage unit, a gate in a Montech conveyor system, a gantry robot, rack storage, and RFID (radio frequency identification) readers. More important, an agent represents each manufactured product—in this case, a customized Gillette gift box. This product agent autonomously and proactively controls its own production process by negotiating with the other agents. In this case, the process involves packing a box with differ-

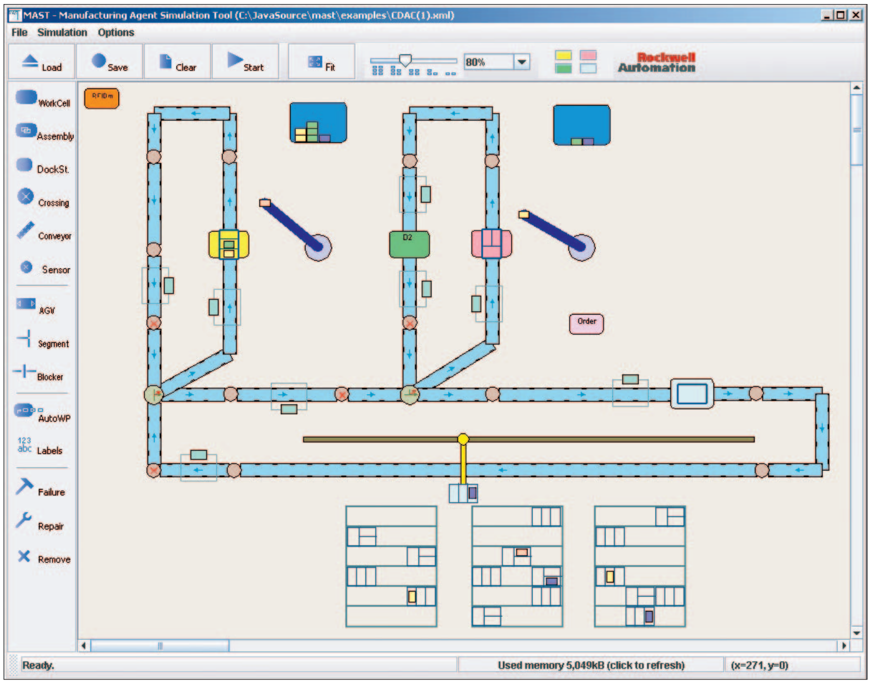


Figure 5. A simulation of the Cambridge packing cell in the MAST simulation environment.

ent grooming items such as gels, deodorants, and razors. The agents negotiate over such issues as which storage location can provide the requested items and which robot will pack them.

The industrial case studies in this article illustrate that you can effectively employ MAS technology to design the next generation of large-scale, robust, and flexible manufacturing control systems. Features such as fully decentralized decision making, dynamic lookup for suitable service providers, or embedded support for simulations go far beyond the capabilities of classic centralized and hierarchical industrial-control systems.

Reference

1. F.P. Maturana, R.J. Staron, and K.H. Hall, "Methodologies and Tools for Intelligent Agents in Distributed Control," *IEEE Intelligent Systems*, vol. 20, no. 1, 2005, pp. 42–49.

Adaptive, Dynamic Transport Optimization

Klaus Dorer and Monique Calisti,
Whitestein Technologies

Logistic networks' increasing complexity and dynamic nature motivates a cost-sensitive rethinking of process and optimization strategies.[1] This goal requires not only efficient processes but also IT solutions that can deliver the required flexibility and dynamically respond to change and customization.

Living Systems Adaptive Transportation Networks is a comprehensive agent-based solution for optimization and dispatching of full and part truck loads, including tracking and real-time event handling. LS/ATN includes

- a real-time route optimizer,
- an event management system that informs dispatchers about a wide range of events as they occur (or, proactively, if expected events don't occur),
- a tracking facility that provides accurate data about the progress of orders, and
- a simulation mode that assists in tactical and strategic decision making.

Agent-based optimization

Finding optimal routes for serving transportation requests from a (usually large) set of customers is a complex problem. A limited number of available trucks must pick up and deliver transportation orders at specific customer locations. The trucks can be of different types and capacities and are usually available at different locations. Truck drivers must observe drive time restrictions. Pickup and delivery must occur within specific time windows, even though time constraints can potentially be violated within some tolerated degree (soft constraints). The problem is highly dynamic, not only because transportation requests aren't all known in advance but also because various unpredictable events can affect previously defined plans. Trucks might be delayed owing to traffic jams or other unforeseen problems or even become temporarily unavailable.[2]

You can distribute the solving of this transport optimization problem among multiple interacting software agents to

- achieve scalability with growing sizes of problem instances,
- directly reflect the distributed nature of transportation organizations and decision-making centers,
- facilitate the handling of local deviations without having to propagate local changes and recompute the whole solution, and
- increase robustness (avoiding a single point of failure).

In particular, the LS/ATN architecture reflects how logistics companies manage this domain's increasing complexity. A transportation business is usually divided into dispatching regions. Transportation requests arriving at a region are first tentatively allocated and possibly optimized in that region. If orders' pickups or deliveries occur in different regions, these other regions are also informed and asked to handle the request in case they can provide a cheaper solution to transport the order.

In LS/ATN, distinct software agents represent different regions. A local *AgentRegionManager* manages trucks starting in its region. A centralized *AgentDistributor* distributes incoming transport requests according to their pickup location. When receiving a new order, an AgentRegionManager generates a valid solution (that is, a transportation plan specifying which orders to combine into which routes, and which trucks will handle those routes). To do this, LS/ATN uses a contract net protocol to sequentially insert transportation requests.[3] The system checks all available trucks in that region both to verify their capability to transport the order and to determine the cost. However, this approach could produce suboptimal plans—for example, because the "best fitting" truck is already full. So, to improve the solution, the system schedules cyclic transfers between trucks.[4,5] A cyclic transfer is an exchange of orders between routes—transfer requests among regions are triggered when trucks have routes spanning different regions. A simple strategy to select an order transfer is a hill-climbing approach that selects the most cost-saving transfers from a neighborhood of possible transfers. This hill-climbing process continues with all changed routes until the system can't perform any more cost-saving exchanges.

The LS/ATN design's main advantage stems from its direct mapping to today's transport business organizations and its good scalability. Moreover, its computational overhead is also lower than a fully distributed solution using one agent per truck.[6] Its main drawback is degradation of the solution, compared to a fully centralized approach (from a global-optimum perspective). However, a fully centralized solution often wouldn't be feasible in real-world scenarios.

Figure 6 illustrates details of a route that LS/ATN generated.

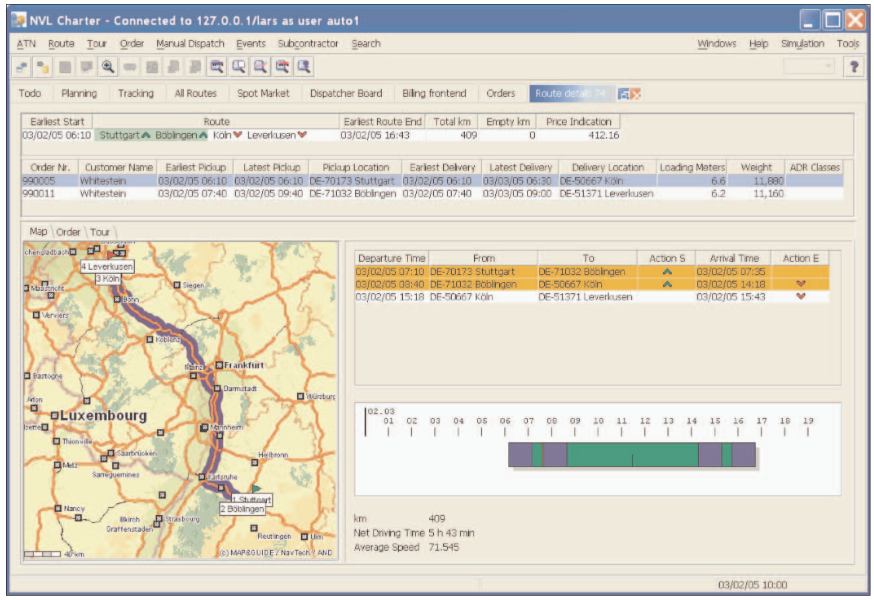


Figure 6. Details of a route that LS/ATN (Living Systems Adaptive Transportation Networks) computed, including route and order information, a map of the route, and schedule information in tabular and graphical form.

Results

We ran extensive empirical tests for ABX, a European logistics company, to determine what cost savings LS/ATN could provide. The cost model and constraint checker took into account real-world costs and constraints, thereby enabling comparison of the agent-based solution's optimization results with real transport plans that professional dispatchers created manually. The analyzed data set contained roughly 3,500 real business transportation requests (orders).

LS/ATN decreased costs 11.7 percent; 4.2 percent of this stemmed from fewer driven kilometers. Another 2.2 percent of the savings came from significantly increasing the number of consecutive routes that are cheaper to sell on the spot market. The rest of the savings stemmed from the LS/ATN solution preferring routes starting in regions where trucks are cheaper to buy. An additional important achievement is that the LS/ATN solution used 25.5 percent fewer trucks than the manual solution. This is due to higher utilization of the trucks and longer utilization, on average, of each truck. Today, ABX uses LS/ATN in its day-to-day operations.

LS/ATN draws its strength from a multiagent system core. Built on a bottom-up optimization philosophy, goal-directed agents interact to solve subproblems that, when consolidated, result in a solution to the overall problem. Similar to human decision making, solutions to problems arise from the interaction of individual decision makers (represented by software agents), each with its own local knowledge. Traditional IT systems' centralized, rule-based nature imposes intrinsic limits on dealing successfully with unpredictability. Multiagent systems don't have this limitation because collaborating agents quickly adapt to changing circumstances and operational constraints.

References

1. "UK Consumer Products Industry Cites Cost Reduction as Its Biggest Logistics Challenge," *Exel News*, 2002, www.exel.com/exel/home/media/-news/newsreleases/pressreleasecostreduction.htm.

2. M.W.P. Savelsbergh and M. Sol, "The General Pickup and Delivery Problem," *Transportation Science*, vol. 29, no. 1, 1995, pp. 17–29.

3. J.-J. Jaw et al., "A Heuristic Algorithm for the Multi-Vehicle Advance Request Dial-a-Ride Problem with Time Windows," *Transportation Research*, vol. 20 B, no. 3, 1986, pp. 243–257.

4. P.M. Thompson and H.N. Psaraftis, "Cyclic Transfer Algorithm for Multivehicle Routing and Scheduling Problems," *Operations Research*, vol. 41, no. 5, 1993, pp. 935–946.

5. S. Mitrovic-Minic, *Pickup and Delivery Problem with Time Windows: A Survey*, tech. report TR 1998-12, School of Computing Science, Simon Fraser Univ., 1998.

6. K. Fischer, "Cooperative Transportation Scheduling: An Application Domain for DAI," *Applied Artificial Intelligence*, vol. 10, 1996, pp. 1–34.

Conclusions and Lessons Learned

Michal Pechoucek and Simon G. Thompson

The AAMAS 2005 Industry Track included several discussions in various formats, ranging from formal debates to philosophical discussions in the small hours of the morning. These discussions explored how the agent community could improve its relevance and impact to build on successes so far. While a reasonable amount of interaction occurs between agent researchers and industry, industrial adoption of agent-based solutions faces these main bottlenecks:

- Limited awareness about the potential of agent technology in industry. Agents are used in a few specialized disciplines and remain unused in others where they might be appropriate.

Michal Pechoucek is the principal investigator and head of the Gerstner Laboratory Agent Technology Group, an associate professor in artificial intelligence at the Czech Technical University in Prague, and a part-time senior consultant for CertiCon. Contact him at pechouc@labe.felk.cvut.cz.

Simon G. Thompson is a principal research scientist and research group leader at the Intelligent Research Center in British Telecom's research department. He's also a visiting research fellow at the University of Southampton. Contact him at simon.2.thompson@bt.com.

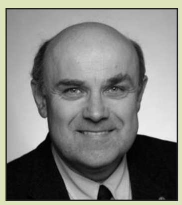
Jeremy W. Baxter is a lead researcher at QinetiQ. Contact him at jwbaxter@qinetiq.com.

Graham S. Horn is a researcher at QinetiQ. Contact him at ghorn@qinetiq.com.

Koen Kok is a scientific researcher in intelligent energy management at the Energy Research Center of the Netherlands. Contact him at j.kok@ecn.nl.

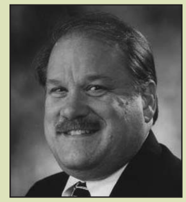
Co Warmer is a scientific researcher in intelligent energy management at the Energy Research Center of the Netherlands. Contact him at warmer@ecn.nl.

René Kamphuis is a scientific researcher in intelligent energy management at the Energy Research Center of the Netherlands. Contact him at kamphuis@ecn.nl.



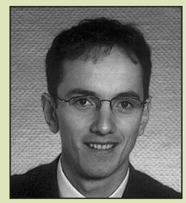
Vladimír Marík is the managing director of the Rockwell Automation Research Center, Prague, Czech Republic. Contact him at vmarik@ra.rockwell.com.

Pavel Vrba is the lead of the Agent Technology Group at the Rockwell Automation Research Center, Prague, Czech Republic. Contact him at pvrba@ra.rockwell.com.



Kenwood H. Hall is the vice president for Architectures and Systems Technology at Rockwell Automation. Contact him at khhall@ra.rockwell.com.

Francisco P. Maturana is the Agent Infrastructure Lead at the Rockwell Automation Advanced Technology Laboratory, Cleveland. Contact him at fpmaturana@ra.rockwell.com.



Klaus Dorer is a senior researcher at Whitestein Technologies. Contact him at kdo@whitestein.com.

Monique Calisti is the vice president of R&D at Whitestein Technologies. Contact her at mca@whitestein.com.

- Limited publicity of successful industrial projects with agents.
- Misunderstandings about agent technology's capabilities, which led to early industrial adopters' unrealistic expectations and subsequent frustration.

Some common unrealistic expectations and inappropriate uses of agent technology fall into seven main categories:[1]

- *Complexity*. People often expect that agent technology can help solve very complex (perhaps NP-hard) problems. In our experience, this is obviously incorrect, although partitioning problems using the agent abstraction can often lead to approximate solutions with lower computational demands.
- *Black box*. People often view agent technology as a black-box technology (like neural networks or genetic algorithms) that you can insert to solve a particular complex problem. However, agent technology provides primarily system concepts and design paradigms that are useful in well-defined classes of problems.
- *Intelligence*. People sometimes think that agents can directly deal with problem solving and domain-specific intelligence. However, agent researchers' prime concern is the agents' collective behavior and decision making, and agent research often overlooks the technology's application to real-life problems.
- *Agentification*. People think that you can fully automate agent integration and legacy system encapsulation. However, no sophisticated mechanism exists that can encapsulate any legacy system fully automatically. Common current solutions involve alternative technologies (for example, Web services).
- *Learning*. People frequently overestimate multiagent systems' potential for learning. They often think that an agent should be superadaptable and able to accommodate to any requested behavior (this expectation is closely connected to those of intelligence and agentification).
- *Interoperability*. Standards and interoperability are computationally expensive. It isn't wise to use full FIPA (Foundation

for Intelligent Physical Agents) compliance in systems where full openness isn't necessary (for example, in simulation and modeling).
- *Mobility*. People often claim that agent mobility is inevitable and more essential than is actually the case. Often, migration of data or simple communication is sufficient, rather than migration of an agent's code and state.

While the Industry Track attendees appreciated the presentations' high technical quality, they frequently expressed one specific concern. The presented applications demonstrated only a technical, revenue, or efficiency advantage. These application's developers can rightly claim that their solutions are superior to those that were the previous state of the art. However, in business, this isn't the fundamental test of value. Businesses assess an advantage by its return on investment, but the Industry Track presented no evidence of this. This point (in a slightly different form) also applies to fields such as defense and medicine, where a precise commercial quantification of agent technology isn't possible. These fields have well-known benchmarks and metrics for evaluating system performance, but agent-based systems rarely, if ever, prove superior according to these criteria.

Reference

1. M. Pechoucek, M. Rehak, and V. Marik, "Expectations and Deployment of Agent Technology in Manufacturing and Defence: Case Studies," *Proc. 4th Int'l Conf. Autonomous Agents and Multi-Agent Systems—AAMAS 2005 Industry Track*, ACM Press, 2005.