Auton Agent Multi-Agent Syst
DOI 10.1007/s10458-008-9050-0

# Industrial deployment of multi-agent technologies: review and selected case studies

**Michal Pěchouček · Vladimír Mařík**

**Abstract** This paper reports on industrial deployment of multi-agent systems and agent technology. It provides an overview of several application domains and an in-depth presentation of four specific case studies. The presented applications and deployment domains have been analyzed. The analysis indicates that despite strong industrial involvement in this field, the full potential of the agent technology has not been fully utilized yet and that not all of the developed agent concepts and agent techniques have been completely exploited in industrial practice. In the paper, the key obstacles for wider deployments are listed and potential future challenges are discussed.

**Keywords** Multi-agent systems · Agent technologies · Industrial applications · Control · Simulation · Planning

## 1 Introduction

The key motivation of this contribution is to provide the readers with compact information about how the various agent technologies and multi-agent systems paradigms have made it to the real industrial deployment. The addressed research problem is to clearly identify and assess the potential of practical use of agent technologies and report on successful industrial deployment. In order to provide high value to the readers, the analysis has been made in two steps: (i) four case studies, where distinct agent applications have been presented in greater level of detail and (ii) overview of 10 application areas and domains with information about and references to various projects and applications contributing to these areas. An attempt has been made to generalize the information and assess various application domains by the

---

M. Pěchouček (✉) · V. Mařík
Gerstner Laboratory, Agent Technology Group, Department of Cybernetics,
Czech Technical University, Prague, Czech Republic
e-mail: pechouc@labe.felk.cvut.cz

V. Mařík
Rockwell Automation Research Center, Prague, Czech Republic

use of agent concepts, required functionality, the level of application maturity, potential for reusability, hardware or software orientation of the application and requirements for legacy system integration. The research presented in this paper indicates that industry is involved in agent technology deployment and supports further interaction between academia and various application areas. There has been identified a high number of technology prototypes and agent system demonstrators. Despite of this fact, the developed agent concepts and agent techniques have not been fully utilized in industrial practice and only small selection of available technology has made it to the mature, on the market available applications.

It has been generally known that the agent research community provides powerful theories, algorithms and techniques that have got an immense potential for the deployment in various industrial applications. Sadly noted, the number of agent research centers and groups as much as the number of agent researchers by far outnumbers the commercial companies, individual CTOs, and entrepreneurs that have the courage to invest in adoption of these novel theoretical advancements in industrial problems. Despite our slight pessimism, there are several smaller start-up companies and a few big industrial organizations that integrate multi-agent systems concepts in their selected industrial operations.

While there is a reasonable amount of interaction between the research and industry, the main bottlenecks in fast and massive adoption of the agent-based solutions are (based on the common knowledge in the community and also by observation gathered through involvement in AgentLink III and own facilitation of the interaction between industry and agent research):

- limited awareness about the potentials of agent technology in industry—agents are used in few specialized disciplines, while they remain unused in the others where they fit,
- limited publicity of the successful industrial projects with agents,
- misunderstandings about the technology capabilities, over-expectations of the early industrial adopters and subsequent frustration (see Sect. 5),
- risk that comes with adoption of new technology that has not been proven in large scale industrial applications yet (i.e., “we would like to use it, but we do not want to be the first one”), and
- lack of design and development tools mature enough for industrial deployment.

The authors’ opinion is based on their long-term interaction with an international industry and agent researchers, participation in numerous industrial projects and organization of several different meetings encouraging the interaction between industry and agent researchers and commercial take-up of agent technologies (such as the series of HoloMAS conferences, AAMAS Industry Track, AgentLink Agent Technology Conferences, IFAC-INCOM Symposium, etc.).

The paper is structured into several parts. In Sect. 1.1 we list the abstract opportunities for agent technology deployment and in Sect. 1.2 the list of deployed agent concepts is presented. In Sect. 2 we introduce four case-study applications that shall illustrate four different use cases. In Sect. 3 we list typical application domains for agent deployment and provide references to finished and running projects, as well as to existing applications. In Sect. 4 we analyze appropriateness of the use of multi-agent technology and try to match the listed applications with the requirements provided in Sect. 1.1. The Sect. 5 provides our opinion on future trends and challenges for the industrial deployment of agent systems.

### 1.1 Opportunities for deployment of multi-agent systems

The very innovative and theory grounded computer science concept of autonomous agents and multi-agent systems (AAMAS) have different forms of practical applicability. In order to position the technical content of this paper with the current state of the art of multi-agent system deployment, let us distinguish among three principal directions.[1] AAMAS concepts support the applications in:

(1) *Agent Oriented Software Engineering*—provide designers and developers with a way of structuring an application around autonomous, communicative elements, and lead to the construction of software tools and infrastructures to support this metaphor.
(2) *Multi-Agent Techniques*—provide a selection of specific computational techniques and algorithms for dealing with collective of computational processes and complexity of interactions in dynamic and open environments.
(3) *Multi-Agent Simulation*—provide expressive models for representing complex and dynamic real-world environments, with the emphasis on capturing the interaction related properties of such systems.
(4) *Autonomy-Oriented Techniques*—provide set of artificial intelligence techniques supporting autonomous decision making of intelligent systems and methods of adjusting their decision making autonomy.

There is an active community around the concept of Agent Oriented Software Engineering (AOSE) providing various programming and software system development methodologies based on AAMAS concepts [34]. Among the best known AOSE methodologies, let us list GAIA [93], DESIRE [7], Tropos [24], ADEPT [35], SODA [54] or Prometheus [55,56]. The AOSE community also provided AUML, the Agent Unified Modeling Language [1], an extension of classical UML by creating additional elements to support the modeling of multi-agent systems. The AOSE methodologies contributed to a number of successful software deployment success stories. However, this paper is oriented towards direct application of multi-agent techniques, methods and algorithms in the industrial applications and thus it does not have the ambition to review the applications of the AOSE methods and paradigms. The reader is referred to the publications of highly successful workshop series of International Workshops on Agent Oriented Software Engineering (AOSE) and the International Journal of Agent-Oriented Software Engineering (IJAOSE). The summary of challenging research directions for the AOSE field can be found in [95].

The technical content of the paper will be more oriented towards industrial deployment of agent technologies and industrial applications of multi-agent simulation (points (2) and (3) from above) with the special focus on manufacturing, logistics and defense applications.

When analyzing the opportunities for agent technology deployment it is fair to distinguish between two slightly different sets of techniques: (i) techniques supporting interaction and collaboration of distributed multiple agents and (ii) techniques supporting agents' autonomy. Even though the combination of both aspects of agency is a desirable property of an agent-based system, in our experience (documented in this paper) industrial deployment emphasize either the distributed and collective aspects or the autonomy-oriented aspects of agency. Let us discuss the properties of the problems and the application requirements with respect to what the agent techniques can provide. We will list and label the properties for further referencing from Sect. 4.

*Distributed and collective* aspects of agency are considered to perform well in application domains with the following specific properties (*properties P1–P6*):

- *Decentralized scenarios*: Particularly suitable are the domains where the data and knowledge required for computation are not or cannot be available centrally or the process

---

[1] The authors do not have the ambition to present definitions of the four specific subfields of the AAMAS system. This breakdown list is used here for clearer understanding of the orientation of the technical content of the paper.

physical system control needs to be distributed. This can be the case in several situations (*properties P1—P3*):

  - *Geographical distribution* of knowledge and control (e.g., logistics, collaborative exploration, mobile and collective robotics, pervasive systems) or the environments with partial or temporary communication inaccessibility (where self-organization, local interaction and intelligent synchronization is needed in order to cope with communication inaccessibility)—*property P1.*
  - *Competitive domains*, with the restrictions on the information sharing (e.g., e-commerce applications, supply-chain management, and e-business)—*property P2.*
  - Domains with the requirements for *time–critical response* and *high robustness* in distributed environment (e.g., time–critical (soft- and/or hard-realtime) manufacturing or industrial systems control, with re-planning, or fast local reconfiguration)—*property P3.*
- *Simulation and modeling scenarios*: Using agents for simulation purposes has been very common, while the right justification was often missing. Agents shall be deployed in simulation exercises where we require, e.g., an easy migration from the simulation to deployment in real environment—*property P4.*
- *Open systems scenarios*: In scenarios requiring integration and interoperability among software systems that are not known a priori and whose source code may not be available—here the use of agent technologies, especially agent communication languages and interoperability standards is advisable—*property P5.*
- *Complex systems scenarios*: In scenarios requiring modeling, controlling or engineering of complex systems. Decomposition of the decision making into separate agents' reasoning and solving problems by means of negotiation represents a novel software development paradigm [25]. Complex system modeling is often understood as closely related to solving complex problems. Potentials for decreasing computational requirements for complex problem solving by means of paralleling the computational process within multiple agents is limited, but possible—*property P6.*

*Autonomy oriented* aspects of agency is appropriate in application domains with high requirements for systems with decision-making autonomy, when the user delegates the substantial amount of decision-making authority to the system and when the system is expected to cope independently with unexpected situations (also in the situation with long-term communication inaccessibility and interaction isolation of the autonomous entity)—*property P7.*

The properties $P1$ and $P7$ are somewhat linked. In the situations, where the communication infrastructure, a critical component required for collective decision making, is disrupted, some of the agents need to perform higher level of decision-making autonomy.

### 1.2 Deployed agent concepts

Prior to presenting and assessing the individual agent systems and applications domains, let us list the typical agent concepts used in agent technology deployments:

- *Coordination*—list of agent techniques (based mainly on dedicated coordination protocols and various collaboration enforcement mechanisms) that facilitates coordinated behavior between autonomous, while collaborative agents. Coordination usually supports conflict resolution and collision avoidance, resource sharing, plan merging, and various collective kinds of behavior.
- *Negotiation*—list of various negotiation and auctioning techniques that facilitate an agreement about a joint decision among several self-interested actors or agents. Here we

emphasize mainly negotiation protocols and mechanisms how individual actors shall act and what strategies shall they impose to optimize their individual utility.

- *Simulation*—techniques that allow inspection of collective behavior of the interactive actors, provided that the models of the individual agents are known. Here we count on the versatile simulation frameworks that allow long-run complex simulation and various "what-if" analyses of different problems. If distributed hardware system is modeled, agent-based simulation enables a close linkage between simulation and the real hardware machinery.
- *Interoperability*—set of techniques for achieving high level interoperability among software components developed by different designers, especially in the situation where the source code and complete models of behavior are not shared. Interoperability is studied on the level of physical connections via interaction protocols but also semantics of communication.
- *BDI Architecture*—well known agent reasoning architecture based on guiding the agents behavior on formally structured knowledge and mental states into (i) believes, (ii) desires and (iii) intentions. BDI architecture is a design concept, while it has been integrated into numerous agent programming languages.
- *Adjustable Autonomy and Policies*—set of techniques and methods for specifying and dynamic adjustment of decision making autonomy of the individual actors in a multi-agent system. Various formal frameworks for specifying policies have been proposed and numerous policy management systems have been designed by the agent community.
- *Organization*—techniques supporting agents in ability to organize autonomously in permanent or temporal interaction and collaboration structures (virtual organizations), assign roles, establish and follow norms, or comply with electronic institutions.
- *Meta-Reasoning and Distributed Learning*—in the multi-agent community there are various methods allowing an agent to form hypothesis about available agents. These methods work mainly with the logs of communication or past behavior of agents. Agent community also provides techniques for collaborative (distributed) learning, where agents may share learnt hypothesis or observed data. A typical application domain is distributed diagnostics.
- *Distributed Planning*—specific methods of collaboration and sharing information while planning operation among autonomous collaborating agents. Agent community provides methods for knowledge sharing, negotiation and collaboration during the five phases of distributed planning [20]: task decomposition, resource allocation, conflict resolution, individual planning, and plan merging. These methods are particularly suitable for the situations when the knowledge needed for planning is not available centrally.
- *Knowledge Sharing*—techniques assisting in sharing knowledge and understanding different types of knowledge among collaborative parties as well as methods allowing partial knowledge sharing in semi-trusted agent communities [59] (closely linked with distributed learning and distributed planning).
- *Trust and Reputation*—methods allowing each agent to build a trust model and share reputation information about agents. Trust and reputation is used in non-collaborative scenario where agents may perform non-trusted and deceptive behavior.

The concept of agents' physical mobility is not discussed in the paper.

### 1.3 Frequently requested functionalities

There is a wide range of functionalities often expected from agent systems deployment. The purely software agent based systems often perform planning, scheduling, simulation or other decision support like functionalities. Agent solutions that are closely linked with some kind of hardware often provide control, diagnostics or integration functionalities. In robotics application, it is often requested to provide collision avoidance and trajectory planning functionality. The ability of automatic reconfiguration of the plant equipment is highly desired in manufacturing tasks. Recently agent systems in the fields of (ad hoc) networking and information integration and retrieval are also expected to provide intrusion detection, prevention and response functionalities as well as various service facilitation and brokering services.

It is important not to mix the agent concept of distributed planning with the planning functionality of a possible multi-agent system (that can be provided by, e.g., multi-agent simulation coupled with coordination). Similarly, multi-agent simulation as an agent concept is equally different from the simulation functionality provided by an application.

The information provided in the Sects. 1.2 and 1.3 will be used for assessing the data collected in the text to follow. Assessment is provided in the form of a small table at the end of each technical paragraph and in Sect. 4.

## 2 Selected agent technology deployment case studies

In the following section of the article we are going to present several selected applications of multi-agent systems which have been developed by commercial organizations or by research institutions as a result of direct industrial requirements. In this section we have limited ourselves to the presentation of the selected application scenarios in which we were directly involved and that in our opinion represent a wide spectrum of agent deployment potentials.

In the rear of each subsection here we append a table generalizing the key agent concepts deployed, the main functionality of the application and specification of how mature the application is.

### 2.1 Agents in shipboard automation distributed control and diagnostics

Multi-agent technologies have been successfully deployed by Rockwell Automation, Inc. in flexible and distributed control of a ship equipment to enable reduced manning and reliable and survivable operations of a ship using the commercial-off-the-shelf products. The Shipboard Automation architecture is divided into three hierarchically organized levels [83]: The *Ship-level* is concerned with overall goals of the ship and communicates directly with the ship crew. The *Process-level* is aimed at optimizing the performance of the automation components and ensuring available services. The *Machine-level*, the lowest one in the hierarchy, is responsible for the real-time control, diagnostics, and system reconfiguration.

The first deployment of the shipboard automation architecture has been focused on the control of a chilled water system (CWS). To solve the technology migration problem, the firmware of the “classical” automation controllers (Rockwell Automation’s Logix family of controllers) has been extended to enable the execution of intelligent agents directly inside of the controller. The developed Autonomous Cooperative System (ACS) infrastructure enables to distribute agents (implemented in C++) over multiple controllers (one controller usually hosts 1-to-n agents) where they run in parallel and are able to interact with the low-level control tasks (written in relay ladder logic, IEC 61131-3).

There are more agents running on a single controller, and network of Logix controllers can host quite a large community of agents. The architecture of an agent consists of four main

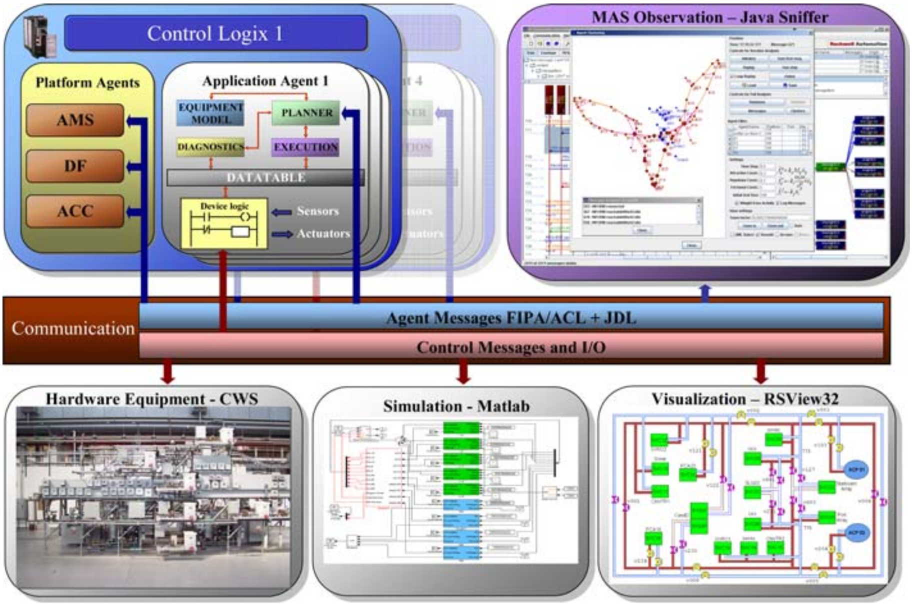


**Fig. 1** ACS runtime architecture

components: planner, device model, execution control, and diagnostic modules (see Fig. 1), where the device model and diagnostic modules are optional.

The planner reasons about the plans and events emerging in the physical domain and represents the core of an agent. It contains a set of pre-prepared "standard" plans (plan templates) which are modified by the actual data obtained from the controlled process, from an agent state, and from the other agents via message passing. Declarative style of agent behavior programming has been reevaluated and a procedural style of programming has been designed and used instead recently. This engine enables to use the full power of a programming language (Java or C++ in our case) together with a set of functions and attributes to interact with other agents, trigger planning processes, etc. which offers more flexibility and better performance than the declarative style. The device model provides the decision-making support by describing and evaluating the physical system configurations. It contains the information about the physical environment, its structure and capabilities. It is up-dated whenever certain changes in the agent's environment occur. The execution control module acts as a control proxy and translates committed plans into execution control actions carried out by the controller which hosts the given agent. The diagnostic part is responsible for detection of local events or disturbances in the physical systems and evaluates them according to the model. The diagnostic component of an agent may include a suite of data acquisition, signal processing, diagnostic, and prognostic algorithms. The diagnostic information gathered and evaluated locally can be accessed by any agent in the community. In such a way, the global diagnostics of the system based on distributed diagnostic approach can be achieved. The application agent structure in architecture presented in Fig. 1 is used to build agents on all the three levels mentioned above (ship-level, process-level, and machine-level). The high-level agent part is responsible for interactions among agents that represent sections of controlled system and low-level part is responsible for hard real-time control actions. The runtime architecture consists of (i) a collection of controllers (ControlLogix and FlexLogix) that host agents and control code, (ii)

MAS observation by Java Sniffer, (iii) hardware equipment that can be simulated (iv), and (v) visualization in RSView32 visualization tool.

Any of the agents can start to plan an action. This start is either triggered by the other (usually a higher-level) agent which orders to carry out a certain macro-action, or by the agent itself when conditions in the environment are evaluated—by confronting the data in both the device model and the diagnostic module with the planning knowledge contained in the planner. The planning process is carried out in three phases: Creation, Commitment, and Execution. During the Creation phase, an agent recognizes the current situation, creates a plan (selects a plan template and fills it with known facts), and possibly negotiates (using the Contract-Net-Protocol) with other agents to cover the whole plan. During the CNP, specialized agents are asked by the agent which started the planning process to offer the bids, the bids are evaluated by the agent-initiator and one or more of them is (are) selected to participate in the plan. The typical task to be negotiated is, e.g., cooling a device with a liquid using a network of pipelines and valves. Tree-like structure of agents potentially participating in the plan execution is being created during this phase. In the Commitment phase, the agents commit their resources to fulfill the task. In the last, Execution phase, the plan is executed by all participating agents.

The agents communicate in all the three phases of their activities using a Job Description Language (JDL) [83] as a content language for FIPA-ACL. Any agent can contact any other agent in the architecture using a JDL message. The FIPA-compliant part of the architecture explores the FIPA standard techniques to locate the agents. It includes a full implementation of Directory Facilitator (DF) that provides matchmaking and recruiting (to locate agents by capabilities) and the Agent Management Services (AMS) functionality (to locate agents by addresses). Moreover, the architecture supports multiple DF agents—this enabled to develop the Dynamic Hierarchical Teams architecture to increase the robustness of the system [82]. This architecture ensures availability of the matchmaking service despite the predefined number of failures since the structure has vertex and edge connectivity equal to the size of teams.

There has been implemented a special software tool, called the Agent Development Environment (ADE), helping the user to develop the distributed application. A very important part of this environment is the library of components called the Template Library (TL). The library is editable by the user, and each template of an agent can contain both the low-level control behavior (written in relay ladder logic, IEC 61131-3) and the higher-level intelligent behavior. Because of the "object-oriented" nature of the components in TL, the inheritance is strongly supported. To enable debugging of communication in the multi-agent system, a tool called Java Sniffer [84] was developed. The Java Sniffer receives messages from all the agents, reasons about this information and presents it from different points of view as supporting information for the designers and users. It is able to visualize messages as a low-level UML (Unified Modeling Language) sequential diagram and provides high-level view via dynamically created traceable workflow diagrams. Natural clusters of agents might be detected by dynamical clustering analysis and visualized through a special clustering/visualization interface to the designer [77]. The Rockwell's Java Sniffer became a freeware downloadable from the JADE's web site (http://jade.tilab.com) since its FIPA compliancy allows to interoperate with any JADE-based system.

The CWS application has been demonstrated on (i) the Reduced Scale Advanced Demonstrator (RSAD) model, which is a scaled-down version of a real U.S. Navy ship, (ii) table-top fluid system demonstrator and (iii) pure software simulator implemented in Matlab. The CWS application features two chilling plants with 16 chilling targets on board the ship (e.g., combat systems, communication systems, radar, and sonar equipment). Each water cooling

**Table 1** Shipboard automation — application overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, planning negotiation, distributed | Control, diagnostics, reconfiguration | Large-scale hardware demonstrator |

plant is represented by an agent, as well as each chilling target (heat source), each valve, and some parts of the piping system. The interoperability of the ACS agents has been verified by using the agents developed at the Johns Hopkins University on the Ship-level [44].

Many constraints, namely rather limited time for the decision making, constraints given by the properties of the physical equipment as well as limited number of acceptable equipment structures have brought strong requirements on agents' behavior. Only a few pre-prepared plan templates are stored in to the planners. The agents should generate new plans just by fast negotiations about the pre-prepared sub-plans. Usually just near-to-optimal solutions could be expected under these conditions. A specific validation approach (for avoiding unexpected emergent behavior during the technology real-life deployment) has been developed and applied [45]. The key element for implementation of this approach is a synchronizer for synchronizing data and time in (i) the simulated controlled process and (ii) the multi-agent control. The main advantages of the solution appreciated by the customers are the scalability of the solution, ability to reconfigure complex systems in fast way (within seconds) and the ability to continue in operation with partially damaged equipment. These properties are not achievable by using a centralized approach. The application requirements of the customer have been fully achieved.

The main capabilities of the agent-based solutions expected by the manufacturing industry are robustness of the highly distributed solutions, capability of re-planning and re-scheduling of operations on the fly, capability of the hard real-time reconfiguration of the manufacturing equipment (in the case of a local failure or a sudden change in the environment), simple way of extending both the hardware and software (plug-and-operate) when extending/modifying the manufacturing equipment with the goal to reduce the commissioning time, and software re-usability and a simple SW maintenance. These specific requirements can be hardly achieved by "classical" centralized approaches. See the Table 1 for the application overview.

## 2.2 Agent-based production planning of engine assembling

The concept of multi-agent systems has been successfully applied for planning manufacturing processes [60]. A European car company SkodaAuto, a member of the Volkswagen group, requested deployment of multi-agent planning technologies for planning their mass-production of car engines. Technology deployment has been coordinated by Gedas, s.r.o., (T-System software company) who implemented the final product, while the Gerstner Laboratory of the Czech Technical University in Prague (CTU), and CertiCon a.s. have contributed by the design, prototype development, and technical experience.

The factory layout consisted of three closely linked assembly lines (ZK line, RUMPF line, and ZP4 line), two operational storage units (Vehicle store and Conveyor store) and a final products storage capacity. The factory manufactures daily up to 2,000 pieces of engine heads for 2 and 4 cylinder 1,200 $\text{cm}^3$ engines, 2,000 pieces of 3 cylinder RUMPF engines, or 1,200 pieces of finished 3 and 4 cylinder 1,200 $\text{cm}^3$ engines (with 66% manufactured for sale). The engines can be assembled either from parts produced in the factory or from parts purchased externally.

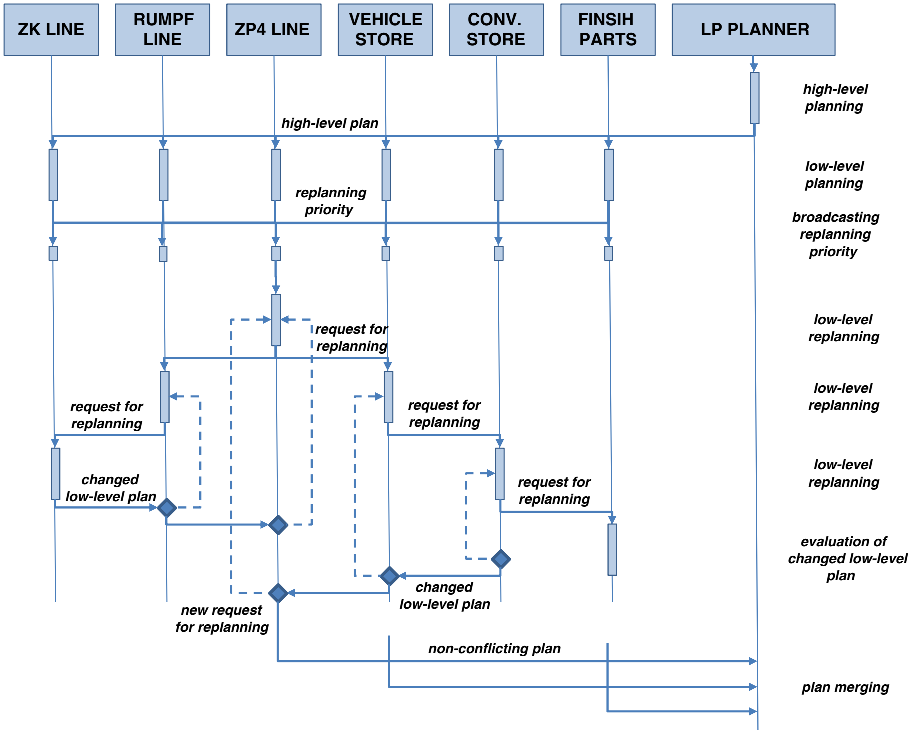


**Fig. 2** Integration of *high-level* and *low-level* planning and replanning by means of agent interaction

The functionality requirements for the final planning system were to provide a detailed production plans for a 6 weeks period, so that storage requirements and consequently also storage cost throughout the production chain were minimized, production type uniformity was maximized, tooling changes were minimized, and any unnecessary handling of products between successive steps of the production process was be minimized. The system shall be also open to integration with production monitoring and management tools and allow further system reconfiguration according to changes in the production processes themselves and support real-time re-planning in the case of demand changes or production anomalies.

Due to very high complexity of the planning problem and several nonlinear constrains a classical operational research methods cannot be used. Instead, the planning problem has been decomposed into *high-level planning* and *low-level planning* processes. The former represents the solution to a substantially relaxed planning problem based linear programming [60]. A coarse, 6-weeks semi-optimal production plan is provided as a result of the high-level planning. This plan however does not comply with the various non-linear local production constrains such as knowledge of late arrival of material, rump-up and completion phase of a daily production (respecting the different numbers of shifts), change of tooling, and many others.

During the low-level planning the production process is simulated by a multi-agent system to detect conflicts and inconsistencies in the high-level plan (see the diagram in Fig. 2). The planning agent sends the high-level production plans to the agents representing the physical entities on the shop floor. The agents use their local resource allocation mechanisms to assign appropriate processes and continually consult dependencies among them. Detected inconsis-

**Table 2** Production planning of engine assembling—application overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Negotiation, distributed planning, (simulation) | Planning and scheduling | In real operation |

tencies trigger local re-planning algorithms. All agents receive their goals prescribed by the high-level plan. The difference between the necessary and the nominal load defines *replanning priority*. The agent with the highest priority performs replanning (by means of classical search methods) so that all constrains are satisfied, and requests appropriate changes from the agents whose plans are linked by resources. These agents perform replanning in the same manner. The revised plans are sent back to the requestor. This process can be iterated until the changes in the plans of the individual agents comply with all the predefined non-linear constrains. Even though this negotiation process has not been theoretically proved for cycles' avoidance, practical experiments have validated its operation.

The described production planning system in SkodaAuto is an important part of a modular Manufacturing Execution System (MES), which is designed to cover in successive steps all of the 11 areas of functional model of MES. Besides necessary interfaces between the company ERP systems, the developed MES system contains modules supporting quality management, production surveillance, production scheduling, and long-term planning. All these components have been fully tested and are now being introduced to real manufacturing process (Autumn 2006). For the implementation of our current long-term planner, a free third party linear programming based solver (LP PLANNER) was used, together with the communication and data transformation wrapper. The whole scheduling takes less than 1 s on a standard PC (with 28 days, 50 products, and 3 machines considered). This completely satisfies the performance requirements. The short-term scheduler has been fully developed at Gedas, s.r.o.

One of the most difficult tasks was to make the decision concerning the granularity of the agents in the design phase. Only a small group of "heavy-duty" planning agents each capable of carrying out complex planning tasks has been designed. The key deployed agent concept in this case study is distributed planning (mainly on the different levels of planning granularity and different level of compliance to the imposed constrains) and negotiation among the agents when resolving the conflicts and mutual replanning interdependencies. See the Table 2 for the application overview.

## 2.3 Agents in air traffic control

With the ever rising use of UAVs in nowadays military and civilian operations, there is an increasing demand for intelligent and unmanned deconfliction mechanisms that control parts of the air traffic in decentralized and autonomous manners. The Gerstner Laboratory, CTU has been working with the Air Force Research Laboratory, NY on technology deployment exercise that is supposed to deliver an agent-based demonstrator (here referred to as AGENTFLY) providing an experimental testbed for various UAV collision avoidance strategies.

AGENTFLY is a software prototype of multi-agent planning system supporting the free-flight based collision avoidance among multiple aerial vehicles. All aerial assets in AGENTFLY are modeled as asset containers hosting multiple intelligent software agents. Each container is responsible for its own flight operation. The operation of each vehicle is specified by an unlimited number of time-specific, geographical waypoints. The operation is tentatively planned before take-off without consideration of possible collisions with other flying objects. During the flight performance, the software agents hosted by the asset containers detect possible collisions and engage in peer-to-peer negotiation aimed at sophisticated replanning in order to avoid the collisions. The aim of this agent deployment is to demonstrate readiness of multi-agent technology for distributed, flexible, and collision-free coordination among heterogeneous, autonomous aerial assets (manned as well as unmanned) with a potential to (i) fly a higher number of aircrafts, (ii) decrease requirements for off-board control operators and (iii) allow a flexible combination of cooperative and non-cooperative collision avoidance.

AGENTFLY is build on top of the AGLOBE (http://agents.felk.cvut.cz/aglobe/) multi-agent platform [76] developed at the Gerstner Laboratory. AGLOBE provides flexible middleware supporting seamless interaction among heterogeneous software, hardware and human actors. AGLOBE outperforms available multi-agent integration systems (e.g., JADE) by its ability to model and integrate rich physical environments in which agents interact, by its support of full code migration, by its model of communication inaccessibility and by its support for scalable experiments.

AGENTFLY provides a wide range of integrated functionalities. It provides a distributed model of flight simulation and control and a time-constrained way-point flight planning algorithm avoiding specified (static and dynamic) no-flight zones. AGENTFLY also provides a flexible, multi-layer collision avoidance architecture that dynamically adjusts the flight plans to changes in the flight environment. Besides collision avoidance algorithms it also provides set of algorithms implementing collective flights in formations. The collective flight and collision avoidance functionalities are based on deployed multi-agent negotiation protocols with known theoretical properties. AGENTFLY also provides connectors to external data (Landsat images, airports monitors, no-flight zones, cities), 2D/3D visualization including a web-client access component, and a multi-user operator that facilitates real-time control of selected assets.

AGENTFLY provides four distinct collision avoidance (CA) algorithms linked by a flexible mechanism managing the autonomy of individual assets and selecting the best collision avoidance strategy in real time [61]:

- *Rule-based CA algorithm* (RBCA) is a domain dependent algorithm based on the Visual Flight Rules defined by FAA. Upon the collision threat detection, the collision type is determined on the basis of the angle between the direction vectors of the concerned aircrafts. Each collision type has a predefined fixed maneuver which is then applied in the replanning process. Visual flight rule-based changes to flight plans are done by both assets independently because the second asset detects the possible collision with the first asset from its point of view.
- *Iterative Peer-to-peer Collision Avoidance* (IPPCA) deploys multi-agent negotiation theories (namely Monotonic Concession Protocol with the Zeuthen Strategy) aimed at finding the optimal CA maneuver [92,96]. The software agents on each asset generate a set of viable CA maneuvers and compute costs associated with each maneuver (based on e.g., the total length of the flight plan, time deviations for mission way-points, altitude changes, curvature, flight priority, fuel status, possible damage or type of load). The agents negotiate such a combination of maneuvers that minimizes their joint cost associated with avoiding the collision.
- *Multi-party CA algorithm* (MPCA) extends the above presented CA algorithm by allowing several assets to negotiate about collective CA avoidance maneuver. This algorithm

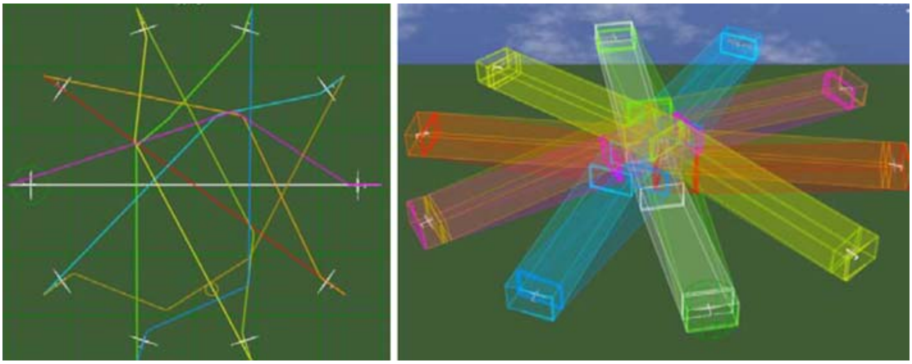


**Fig. 3** Collisions avoidance among 10 aircrafts implemented by RBCA (left) and IPPCA (right)

minimizes the effects of CA maneuvers causing conflicts in future trajectories with other flying assets. While requiring more computational resources, this strategy has shown to provide more efficient free-flight collision free trajectories.

- *Non-cooperative CA Algorithm* (NCCA) supports collision avoidance in the case when communication between aircrafts is not possible. Such a situation can arise e.g., when on-board communication devices are temporarily unavailable or when an asset avoids a hostile flying object. This algorithm is based on modeling/predicting the future airspace occupancy of the non-cooperative object and representing it in terms of dynamic no-flight zones. Based on this information, the algorithm performs continuous re-planning.

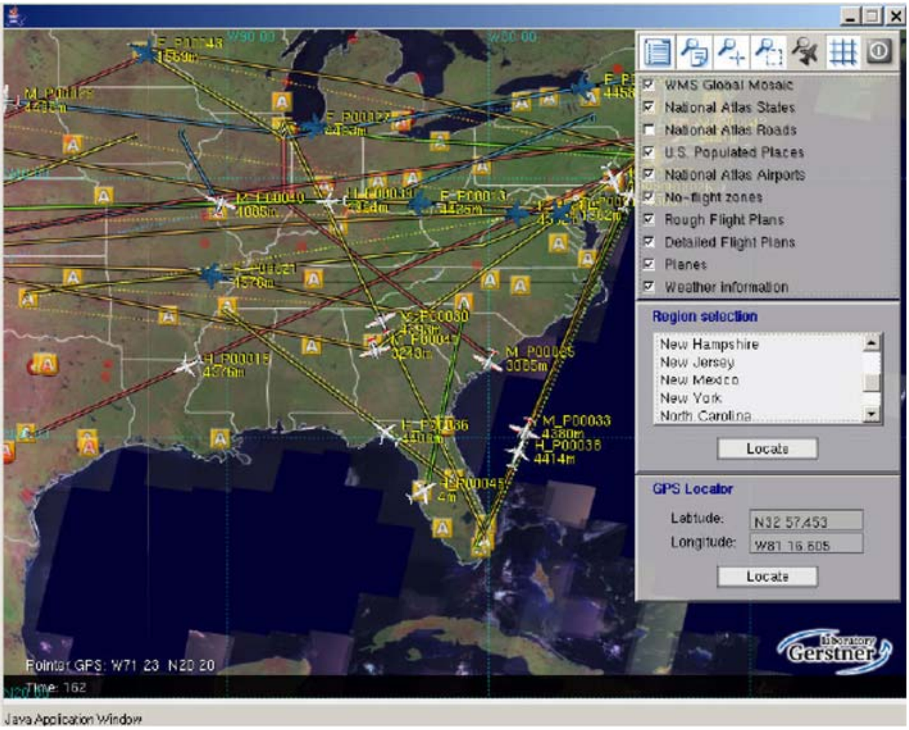
AGENTFLY has been tested on a number of highly complex scenarios and scalability situations [74], such as 10 aircraft with their planned trajectories to meet all in the middle (see Fig. 3), 200 randomly generated assets in a restricted flight area, one asset flying against a formation of 13 assets, landing scenarios or flying through tunnels, or various combat scenarios based on collective flight performance.

AGENTFLY is integrated with publicly accessible real-time data such as mosaic of Landsat7 satellite images obtained from NASA website, data from U.S. Geological Survey (USGS) including detailed vector shapes of U.S. state boundaries, U.S. airports, GPS coordinates with the corresponding average numbers of landings per year and a set of more than 26 thousand major U.S. highway segments or Geographic Names Information System (GNIS) database as well as a set of more than 24 thousand U.S. populated places (see Fig. 4). The publicly available, 10 min delayed information about air traffic in the Los Angeles International Airport was used for modeling non-cooperative flying objects.

This industrial deployment exercise clearly illustrates that the different methods of coordination, multi-agent negotiation and multi-agent simulation have strong application potential in the collective robotics domain and in the domain of air traffic control especially with the autonomous unmanned assets. See the Table 3 for the application overview.

### 2.4 Agent deployment in RFID enabled material handling control

MAST is an advanced agent-based simulation tool developed by Rockwell Automation originally for the purpose of a simulation of material handling in flexible manufacturing. MAST (see Fig. 5) Integrates agents modeling basic components of the material handling systems like conveyor belts, diverters, manufacturing cells, and AGVs. The goal of each transportation process is to find an optimal transportation route within the transportation system. The tool



☑ WMS Global Mosaic
☑ National Atlas States
☐ National Atlas Roads
☑ U.S. Populated Places
☑ National Atlas Airports
☑ No-flight zones
☑ Rough Flight Plans
☑ Detailed Flight Plans
☑ Planes
☑ Weather information

New Hampshire
New Jersey
New Mexico
New York
North Carolina

Latitude: N32 57.453
Longitude: W81 16.605

**Fig. 4** AGENTFLY integration with publicly available geographical data

**Table 3** UAV deconfliction task—application overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, negotiation, autonomy, distributed planning, simulation | Trajectory planning, collision avoidance | Agent-based software prototype |

models very high structural flexibility in the routing as well as a strong fault tolerance. The MAST system enables to emulate any component failure resulting in the relevant CNP-like negotiation processes. An alternative routing is found as a result of these processes. New components can be added, removed, damaged or repaired during the run-time. There is a specific technology that enables switching the simulation to real-time control carried out in standard industrial PLCs. This technology enables a smooth step-by-step shift from simulation in MAST to real-time control carried out by Rockwell Automation's ControlLogix controllers. This technology uses identical communication standards and data structures for knowledge representation in both the PC-hosted simulation and the PLC-based implementation.

MAST has been deployed in two material handling testbeds provided by the Center for Distributed Automatic Control (CDAC), the University of Cambridge, and the Automation and Control Institute (ACIN), Vienna University of Technology. The Cambridge testbed required modeling components like Fanuc M6i robots, gates in the Montech system, rack storage, and RFID (Radio Frequency Identification) readers. There have also been intro-

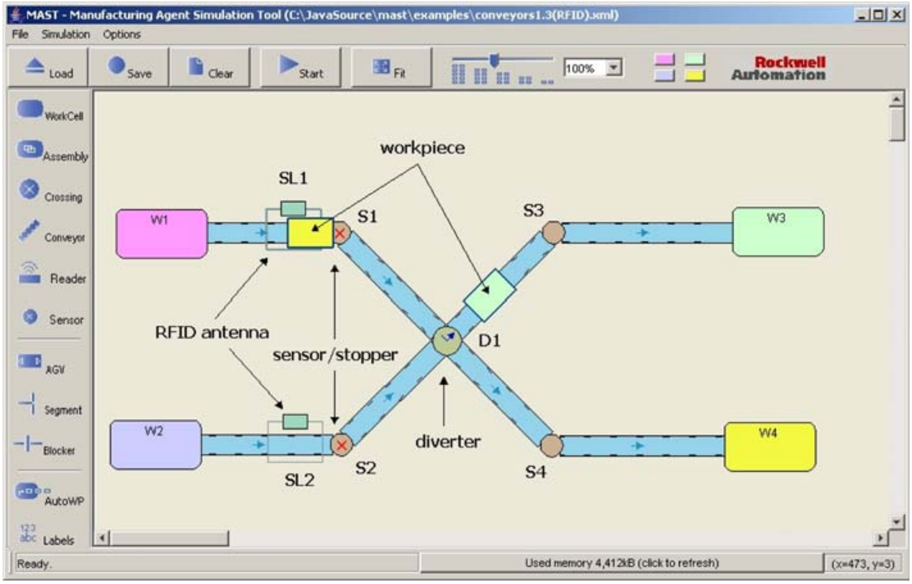


**Fig. 5** Layout of components in the RFID-based navigation in a conveyor transportation system

duced agents representing individual manufactured products that participate proactively in the negotiations to efficiently control the corresponding manufacturing processes. One of the first use-cases of deployed MAST was assembling customized Gillette gift boxes. The agents representing products (boxes) communicate and negotiate proactively with the other agents engaged in the manufacturing process representing available resources. In this case, the manufacturing process involves packing a box with different grooming items such as razors, deodorants, and gels. The agents negotiate by means of CNP over such issues as finding out which storage location can provide the needed items and which robot is able to pack them.

The *RFID agents* directly collect data from RFID readers, filter the data and store them in their internal real-time storage [88]. The other agents can then subscribe for being informed about particular RFID data events. It was suggested that the functionality of the RFID agent aimed at collecting and filtering data from RFID readers is internally ensured by a special module—named *RFID Manager*—plugged into the agent. The RFID Manager is designed as a standalone Java class (with a well-defined interface) to be used as a module to other Java applications. The exchange of information among various RFID agents enables to filter the RFID information (double reading, reading a tag by two different readers at the same time) and to fully understand the content of the distributed RFID reading processes. The position of a (semi)product can be understood from readings accomplished through distributed antennas connected with different physical readers. Linking the agent solutions with RFID technology is expected to lead to a real breakthrough in the material handling and manufacturing systems as this might lead to precise localization of the (semi)products as well as a direct involvement of the (semi-)product agents into the negotiation processes.

The pioneering testbeds have documented the viability and efficiency of multi-agent approach [22,46]. It is documented that the transportation paths (conveyors, pipelines, AGVs, etc.) and their switching and sensing elements (diverters, valves, crossings, storages, tag

**Table 4** Material handling task—application overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Simulation, coordination, negotiation, (interoperability) | Simulation, control | Agent-based software prototype, demonstrator |

readers, vibro-sensors, pressure sensors, etc.) can be easily represented by agents and their interaction can be organized in the way of the agent communication. It has been also illustrated that simulation plays an important role in the development of such systems and especially in taking the technology from the lab to the real factory [89]. The direct re-usability of the simulation code for control/diagnostic purposes belongs to the key factors reducing the commissioning time and expenses. The re-usability is enabled only in the case, the physical equipment is also simulated (at least at the very beginning of the simulation process) to validate the control/diagnostics. Validation systems including the synchronization subsystems (to synchronize the synchronous by its nature equipment with asynchronous activities of agents), input/output configuration and appropriate visualization systems are under development.

The introduction of the RFID technology complements the capabilities of the agent-based systems by capability to provide information about the actual position of the items which are the subject of manufacturing or transportation process. Besides active RFID tagging would directly involve these items into the communication and negotiation processes. See the Table 4 for the application overview.

## 3 List of reported agent technology industry deployments

In this section of the paper we plan to provide the readers with a more comprehensive list of agent technology industry deployments that have been presented recently within the multi-agent community. Again, in the rear of each subsection here we append a table generalizing the key agent concepts deployed, the main functionality of the discussed application domain together with specification of how mature the discussed applications are.

### 3.1 Manufacturing control

For automotive and similar types of industries (aimed at mass-production of individually customized products), highly variable customization requirements, changes in technology as well as equipment failures resulting in needs to change plans and schedules frequently, seem to be quite obvious characteristics of every-day operations. All these requirements and emergency situations can be conveniently and flexibly handled by the agent technology.

A specific class of agent systems, the holonic systems, are using the concept of agents' reactivity and are able to perform system's reconfiguration, in order to achieve pre-programmed situations [42]. The holon is understood as a specific type of a reactive agents which with pre-scripted behavior. Holons could be jointly linked to enhance their decision making power (principle of holarchy). Unlike agents the holons are directly linked with a part of the physical world (valve, pipeline, motor etc.) [18]. The negotiation scenarios, if any, are very modest and are based on simple message exchange. Because of the real-time nature of the tasks, simple, but robust holonic systems are applied with advantage in discrete manufacturing [41]. Majority of these systems explore the IEC-61499 standard [17,40]. This standard known as function blocks has been developed by the Holonic Manufacturing Systems (HMS) consortium within the Intelligent Manufacturing Systems (IMS) initiative for the holonic-based real-time control [26]. The standard is based on the function block part of the well-known IEC-1131-3 standard for languages in programmable logic controllers (PLCs). The major advantage is the separation between the data flow and the event flow amongst various function blocks. Multiple function blocks would be logically grouped together, across multiple devices into an application, to perform the desired process control [10].

One of the most active company in the field of holonic real-time control is currently ProFactor which is investigating (together with the ACIN, University of Vienna) various engineering methods for reconfiguration of real/time control systems [78,97]. The recently established IMS Community of Common Interest — OOONEIDA — focuses namely at creation of the technological object-oriented infrastructure and builds a repository of software components to integrate control knowledge. Its unification and standardization efforts represent in certain sense continuation in the HMS activities [8]. The main problem in the area of holonic real-time systems is still the lack of unified engineering methodology [9,31]. The contributions in this direction are coordinated by the 4DIAC initiative.

Commercial products based on the IEC-61499 are provided by the Canadian ISaGRAPH company, acquired by Rockwell Automation recently. Several interesting applications of the holonic systems based on IEC-61499 have been reported recently, e.g., in the area of baggage handling [5] or in the downtimeless control of energy distribution systems [28]. All the current applications document the efficiency of the holonic approach especially in the real-time reconfiguration tasks.

Semi-real operation of a small agent-based production line at Daimler Chrysler, Stuttgart, Germany, demonstrated very high flexibility, increased throughput, robustness, and reliability of the agent-based manufacturing facility [12]. Significantly increased investment costs were declared as well.

The agent-based solution was developed for bar steel milling process at BHP Billiton, Melbourne by Rockwell Automation in 1990’s. Because of the safety concerns and possible damage to equipment the risk was too high to enable direct control by the agent technology. The agent-based system did not control the bar mill but instead recommended a configuration to the operator. Although the agent system performed very well in all the tests, to release the system for production would require testing all the steel recipes with all possible configurations of cooling boxes [27].

There are running research efforts investigating the use of stigmergy-based approaches in manufacturing coordination and control [37], while no industrial deployment has been reported yet.

Interoperability, on both the networking and semantic levels, is one of the key issues for the manufacturing companies. Besides promotion of well-defined communication standards, the research in ontology/semantic interoperability principles for manufacturing practice has also been started. Rockwell Automation is investigating methods for semi-automatic ontology translation and integration. In [53] the integrating ontologies for material handling tasks utilizing the OWL language exemplify the potential of exploring explicit semantics for working with heterogeneous ontology structures in industrial environment. See the Table 5 for the application overview.

## 3.2 Logistics

Logistics is a typical example of a domain, where the information and data required for efficient planning are not available centrally. In the case of logistics this fact is caused by

**Table 5** Manufacturing control—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, negotiation, distributed planning, simulation, interoperability | Control, simulation, diagnostics | Agent-based software prototypes, hardware demonstrators, semi-real plant deployments |

frequent changes of the respective information and its geographical availability. In the case of the supply chain the needed information cannot be centralized due to requirements for planning knowledge privacy and limited information disclosure. Currently, the resource availability information in logistics scenarios is concentrated in a single point where planning and re-planning is implemented. However, often, especially in the scenarios with partial communication inaccessibility [75] this approach is far from an optimal one. Similarly, in dynamic, multi-tier supply-chain environment, the business partners may not be willing to provide a complete set of information to a centralized planning point. It needs to be noted that there are industrial applications in this domain that use the concept of multi-agent planning as an alternative to the traditional AI planning and scheduling techniques.

Besides numerous research projects in the field of agent-based logistics, there is also a substantial commercial success reported in the multi-agent research community. The typical example of an early adopter of the multi-agent technology in these domains reported Whitestein Technologies, who provided Living Systems/Adaptive Transport Network (LS/ATN) technology to ABX, European logistics company. They have decided for the use of agent-based solution here in order to (i) achieve performance scalability, (ii) reflect the geographical distribution of the nodes, (iii) provide local re-planning without the need to rebuild the whole plan and (iv) increase robustness so that a single point of failure would be avoided. At Whitestein they carried out several performance tests to determine the overall cost saving potentials of the LS/ATN system. Based on the analyzed 3500 transportation requests, 11.7% cost saving was achieved (for more details see [19]).

Successful commercial deployment has been reported by Magenta who managed to build an agent-based system i-scheduler. I-scheduler is a logistics scheduling system developed for Tankers International providing a decision making support to 46 Very Large Crude Carriers (VLCC) [30]. The i-schedulers use the concept of virtual marketplace, where during a typical execution of the system, there are about 1,000 agents running. Besides the scheduling of ship operations, i-schedulers were also successfully employed in a road transportation application to be used by several UK road logistics operators. The application was tested on two sets of client data with 50 and 200 trucks [29].

DARPA has supported via the UltraLog programme the development of a Cougaar multi-agent integration platform. Cougaar has been designed to support planning and replanning of large-scale massive defense logistics operations. The key emphasis in designing of Cougaar was put on the survivability of distributed agent-based systems operating in extremely chaotic environments. The tangible output of the UltraLog pogramme is an open source multi-agent platform and development framework implemented as eclipse plug-in (see http://www.cougaar.org/).

A very specific application domain of the agent-based algorithms is logistics planning of the rescue operations and humanitarian relief missions [4,52,59,81]. Here the agent concepts meet the concept of robotics, thus some efforts need to be spent on hardware integration. Also the important challenge is to work here with rather incomplete information and partially shared knowledge of the participating actors. Recently, the U.S. Army, CERDEC have supported the research/demonstration effort in integrating the state-of-the-art planning tech-

**Table 6** Logistics—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, negotiation, distributed planning, simulation | Planning and scheduling | Software products, deployed systems |

niques with the advanced multi-agent technology so that planning and coordination in very dynamic environments would be facilitated. Such environments are typically decentralized, with partial knowledge sharing, with varying interaction availability, opportunistic and again very dynamic. See the Table 6 for the application overview.

### 3.3 Production planning

An application domain similar to logistics and supply chains is production planning. Here, the advancements of the reported multi-agent technology deployment are also noteworthy. Besides mass-oriented production (see Sect. 2.2) the agent technologies were successfully applied in the domain of project-oriented production, where the objects of the manufacturing process tend to be unique and the planning process regards mainly a single individual project at a time [57]. ExPlanTech is one of the successfully deployed production planning multi-agent systems that was developed in the Gerstner Laboratory, CTU. ExPlanTech performs production processes monitoring and data collection, models individual production units in the factory (such as design departments, workshops and machines) and carries out negotiation-based resource allocation for projects with different due dates and priorities. Similarly to the above listed logistics-oriented multi-agent planning system, the main motivation of industrial deployment of ExPlanTech was to (i) provide effective production plans for particular projects, (ii) integrate production data distributed over the shop-floor and (iii) allow local replanning and reconfiguration of allocated resources. The important use case of the ExPlanTech system was a "*what-if*" kind of analysis that provides the user with decision making support modeling how changes in resource availability (e.g., hiring new people), changes in individual projects' due dates and projects' priorities affects the global operation of the factory. In cooperation with the CERTICON software company (CZ), the Gerstner Laboratory made use of ExPlanTech in Modelarna Liaz, a pattern shop that manufactures casts, forms, and moulds for the leading European car makers. ExPlanTech was integrated with their local ERP system and is currently in a daily use [62]. Later, the use of ExPlanTech has been extended to support also company extra-enterprise operation for efficient planning of their supply chain relations.

The cooperation between the Saarstahl AG and DFKI GmbH led to the development of the AgentSteel System for on-line planning of the steel production [32]. This system explores the InteRRaP multi-layered generic agent architecture [48] that is capable of integrating reactive and deliberative behavior. The Three-tier architecture is used as an internal system implementation vehicle. The system is fully integrated with the IT-environment of Saarstahl AG—the selected Web Services are explored to build the "external" service-oriented architecture for flexible system integration. See the Table 7 for the application overview.

### 3.4 Simulation

Due to agents' natural modeling capability, simulation is one of the favorite deployment scenarios of multi-agent technologies. Engineers tend to use agents as an alternative to classical simulation and modeling techniques due to its enormous expressivity and run-time recon-

**Table 7** Production planning—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, distributed planning, simulation, interoperability | Planning, scheduling | Agent-based software prototypes, deployed systems |

figuration possibilities. Besides this, agents are used for modeling and simulation due to an easy technology migration from simulation to real-life control and decision making. Identical (or very similar) software environment can be used for simulating but also for the real installation. Another key reason for strong calls for industrial use of agent-based simulations is that the real-life deployment of massive industrial multi-agent system is not a trivial task and it requires an elaborate rump-up phase. Such a rump-up phase calls for complex testing, debugging, and simulating software tools. And yet, agent-based systems can be simulated only by agent-based simulation that allows emulation of the system behavior, studying its long-term stability and testing alternative solutions in a very safe way. That is why potentials of agent-based simulation research are closely watched by industry [42].

Rockwell Automation's MAST system (introduced in Sect. 2.4) that has been recently extended to access the control data held in an automation controller is one of the manufacturing-oriented multi-agent simulation systems. Other notable systems are for example the ABAS System, developed jointly by the Tampere University of Technology and Schneider Electric as the first agent-based simulation tool aimed at visualization and simulation of the operation of robot in the 3D manufacturing space [39]. The elementary building units of the system are called actors and they represent particular assembly operations. Actors create clusters representing the given assembly operation. The clusters are built under the support of the *recruiter* agents.

LostWax software company achieved commercial success with its Aerogility product which has been presented during the Agent Technology Conference 2005 in Stockholm [91]. Aerogility is a very complex multi-agent system that models long-term operations of engines and provides aerospace companies with the decision support they need to manage the complex balance of aftermarket resources. Aerogility has been produced for RollsRoyce. Agent-based simulation was used for simulation of the 716 GWh per year district heating system in Gavle region, Sweden. The simulation multi-agent system DHEMOS developed at Blekinge Institute of Technology has been used as a model for testing operation of the ABSINTH—an agent-based system for monitoring and control of district heating systems [90].

SCA Packaging, a leading international manufacturer of the corrugated-boxes, deployed in cooperation with Eurobios an agent based simulation of its manufacturing processes that was designed to assist the company management in finding alternative strategies for reducing stock levels without compromising delivery time [3].

CADENCE Design Systems GmBH in collaboration with CERTICON software company are currently deploying A-globe multi-agent system developed by Gerstner Laboratory, CTU in a project aimed at simulation of the chip design process. The final simulation system is planned to facilitate evaluation and efficient planning of the chip design process and to optimally explore expensive manpower resources [21].

For the same reasons as listed above, agent-based simulation was used also in the domain of collective robotics. Institute for Human and Machine Cognition (IHMC) in collaboration with the Gerstner Laboratory, CTU developed an agent-based simulation of robotics mine-

**Table 8** Simulation—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, negotiation, simulation, adjustable autonomy | Simulation, control, planning | Agent-based software prototypes, deployed products |

sweeping exercise of collective underwater robots. The multi-agent coordination technology has been successfully migrated from simulation to the robotic environment [67].

Rockwell Automation also used the presented technology for simulating drinking water and waste water distribution in a municipal distribution network. The expected energy savings in the extent of 23–69% in comparison to the current "classical" centralized solution have been reported [43]. See the Table 8 for the application overview.

### 3.5 Agent based UAV control

The reasons for increased popularity of agent-based approach in Unmanned Aerial Vehicle control were listed in Sect. 2.3. Apart from AFRL, QuinetiQ also reported advancements in automated Uninhabited Aerial Vehicle control [2]. The main motivation of their project is to demonstrate that the UAV fleet can be controlled by a limited number of operators (ideally one). The human operator is expected to run a collection of UAVs on the mission level, while the UAVs self-organize in order to achieve the mission goals (such as observe an area, locate or observe a target). At QuinetiQ they have developed a multi-agent system that integrates methods of deliberative and reactive planning and that has been tested in three successful human-in-the-loop testing trials. A single pilot was able to successfully control four UAVs with numerous sensors and several weapons and complete the mission consisting of search and attack operations.

At Australian Defense Science and Technology Organization (DSTO) in collaboration with the University of Melbourne, they have reported on successful design and development of agent-based, autonomous UAV mission control system. The approach taken here integrates classical agent-based programming of real-time controllers that extended functionality of standard autopilot Flight Control System (FCS) with intelligent decision-making capabilities. The system has been developed in JACK programming environment and makes a good use of classical BDI modeling structures. The system implementation matches the OODA loop model of military decision making [36] that is widely known by military pilots. Unlike in the QuinetiQ case, DSTO focused on a single UAV control and successfully conducted a flight test of the Codarra Avatar UAV in 2004. SRI International, funded by the U.S. Army is currently investigating potentials of multi-agent deployment in UAV Airspace Management by comparing functionality of (i) fully centralized air-traffic control, (ii) centralized planning but distributed deconfliction, (iii) distributed deconfliction with centrally provided data, and (iv) fully distributed air-traffic control and deconfliction with the non-cooperative air vehicles [65]. Rockwell Scientific (overtaken by Teledyne Scientific recently) is investigating the concept of market-based collaborations for UAV operations within an effort funded by U.S. Army/AATD. The developed Autonomous Collaborative Mission System (ACMS) makes a use of two stage task allocation protocol that explores classical first-price-one-round auctions [15].

There is a lot of research supporting the agent based UAV control projects from the (i) agent integration infrastructures (e.g., [76]), (ii) autonomous interactions and collective decision making (e.g., [80], (iii) from the point of multi-agent robotics [65], and finally (iv) from the field of swarm intelligence [58]. Synergic contribution in all these distinct, while still

**Table 9** UAV control—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Coordination, negotiation, simulation, adjustable autonomy, BDI architecture | Planning simulation, control, deconfliction | Agent-based software prototypes, deployed systems |

interrelated disciplines, may provide a breakthrough in the collective UAV control industry. See the Table 9 for the application overview.

### 3.6 Space exploration applications

In various different space exploration applications the concept of intelligent agents and multi-agent system has been applied. We are not aware of any reported application that would implement collective behavior of several different space missions. Instead of requirements for collective/distributed decision making functionality, the space exploration applications share very high requirements for intelligent system autonomy and ability to operate with only partial, higher level instructions also provided in non-timely fashion. The reasoning systems are expected to follow their mission objectives (regularly updated) and are able to update and revise their operation according to the unexpected situations without consulting the ground stations. Classical agent architectures such as BDI architectures [64] or reactive and deliberative planning thus have a high application potential in this kind of domains.

At the Spaceport Processing System Branch at NASA Kennedy Space Center they have developed an intelligent agent application that processes a ground processing telemetry stream in order to increase situation awareness for the space shuttle count-down experts [70]. The system provides a set of automatic alerts and identifies violation of the launch commit criteria and assist in troubleshooting of possible problems. The system is based on classical rule-based programming such as JESS and thus the reuse of the research concepts provided by the international agent community is limited.

Yet higher level of deployment of classical multi-agent technologies was reported by Jet Propulsion Laboratory (JPL), California Institute of Technology. An Autonomous Science Agent deployed onboard the Earth Observing One Spacecraft [16] was developed. This agent has been designed to detect and monitor scientifically interesting events on the Earth (such as volcanoes, floods, snow melts, etc.). This agent application deploys various AI techniques for data analysis and image processing and also mechanisms for reactive and deliberative planning and robust execution. The agent is in operation since 2003. This application is a standalone piece of software without any sophisticated negotiation mechanisms and social knowledge maintenance mechanisms. Unlike at NASA Kennedy Space Center, at JPL their application based on various AI mechanisms for autonomous and deliberative reasoning have been integrated.

Besides these two agent applications, various different systems have been tested in space such as Remote Agent Experiment (RAX) onboard NASA Deep Space One mission [50], PROBA—European Space Agency (ESA) demonstrating onboard autonomy, or IDEA—recent NASA Ames Research Center autonomous reactive planning and executing architecture [49].

Activities of humans and machines in organizations were modeled at NASA Ames Research Center with the aim to understand work processes and workflows. A multi-agent modeling and simulation environment Brahms [72] has been successfully used in the work

**Table 10** Space exploration—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| BDI, adjustable autonomy | Control, planning, simulation | Prototypes, deployed systems |

**Table 11** Training—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Simulation, adjustable autonomy, BDI | Simulation, training | Agent-based software prototypes |

practice of the Apollo 12 astronauts in the deployment of the Apollo Lunar Surface Experiments Package (ALSEP) on the Moon [73]. See the Table 10 for the application overview.

### 3.7 Training

Researchers are deploying the agent technology rightfully in, e.g., the domain of training. Development of such applications is often supported by international defense industry and makes an important use of the mental state modeling approaches developed by the multi-agent community (such as BDI architectures). A cognitive agent for a naval training simulation environment has been developed by TNO Defense, Security and Safety and Vrije Universiteit Amsterdam [87]. A BDI empowered multi-agent environment JACK has been used for development of an agent-based system simulating behavior of soldiers on an individual and team levels. This application has been integrated with Computer Generation Forces environment used by UK Ministry of Defense [3]. The multi-agent system DEFACTO [69] has been used for development of a multi-agent based tool for training incident commanders for large scale disasters in collaboration of University of Southern California and Los Angeles Fire Department (LAFD). See the Table 11 for the application overview.

### 3.8 Distributed diagnostics

Multi-agent systems contribute to the field of distributed diagnostics and partial hypothesis fusion that is in central interest of various industrial companies, e.g., in the case of complex on-board car diagnostics (current research efforts by, e.g., Denso) also on the level of understanding an aircraft, ship or manufacturing workshop failures from the global perspective. The direct physical linkage of the control and diagnostic hardware and software is a natural opportunity for the deployment of agent-based systems. Real-time agent-based diagnostic solution was demonstrated by Rockwell Automation within the frame of the Shipboard Automation project. The proposed agent architecture has been verified at the chilled water system of a ship [45]. The developed diagnostic algorithms used in a distributed way by individual agents documented that there is possible not only to detect certain type of failure (water leakage/blockage, valve failure, etc.), but also to dynamically localize it in the system even with a very small set of sensors.

The Gerstner laboratory, CTU developed in cooperation with Institute for Human and Machine Cognition (IHMC) within the framework of NASA funded research project a root cause detection solution that provides an agent-based model of NASA hydrogen manufacturing facility. This model is used for diagnostic of the manufacturing alarm situations [11,68].

**Table 12** Distributed diagnostics—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Distributed learning, meta-reasoning, knowledge sharing, interoperability | Diagnostics, simulation, data collection | Agent-based software prototype, hardware demonstrators |

**Table 13** Networking—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Distributed learning, meta-reasoning, knowledge sharing, trust, negotiation, coordination | Security, intrusion detection service provisioning, data streaming, ad-hoc networks | Research concepts, agent-based software prototype |

The Office for Naval Research supported a research in distributed learning in the civilian naval domain, where different learning, semi-collaborative agents monitor different parts of Mediterranean Sea and sharing selected observations and selected hypotheses [86].

Loosely related is a use of agent technologies for aircraft maintenance. At CMU, a prototype agent infrastructure integrating various components of wearable computing and timely distribution and collection of geographically distributed data [71] was developed. The system is based on the multi-agent platform RETSINA [79]. See the Table 12 for the application overview.

## 3.9 Networking

Industrial community has ineligible expectations from deployment of agents in the field of networking. We can see two different problems in this area, namely (i) service provisioning and (ii) network security. The first problem relates closely the mobile ad-hoc networking (e.g., MANET [47]), bandwidth sharing and data streaming applications in commercial (e.g., telecommunication) and various military applications. Even though we are not aware of any such agent application available on the market, there are several applied-research and technology-transfer projects supported that provide preliminary results in this area, such as [13,63,94]. U.S. Army has been recently supporting the application oriented, demonstration research investigating the use of trust modeling and multi-agent reflection in the area of network security [66].

Recently, there has been a notable uptake of allowing ad-hoc car-to-car communication (C2CC) supported by major automotive companies such as DaimlerChrysler or Volkswagen [51]. It is generally believed that the various multi-agent techniques oriented towards interoperability, knowledge sharing, but also agent techniques supporting session management, trust and handling communication inaccessibility can play a vital role in this class of applications. In despite of numerous research projects in the field of multi-agent network security and intrusion detection area (using various meta-reasoning and data mining techniques) we are not aware of any truly multi-agent solution to be any near to industrial prototype or a demonstrator. See the Table 13 for the application overview.

**Table 14** Supply chain integration and virtual enterprises—application domain overview

| Agent concepts | Functionality | Application maturity |
|---|---|---|
| Knowledge sharing, negotiation, coordination, interoperability, trust | Planning, integration | Research concepts, agent-based software prototype, |

### 3.10 Supply chain integration and virtual enterprises

We see a long-term potential in the use of multi-agent technologies in supply chain integration and virtual enterprises lifecycle support. In this domain, the supplier and the customer are mutually dependent on the shared data and knowledge. The quality and the volume of mutually transparent data contributed to cost efficiency of their collaboration. This fact suggests the centralization of coordination processes as the best strategy. However, at the same time the trading partners may be reluctant to share some company sensitive information, since its disclosure may cause their competitive disadvantage. In such a semi-trusted environment the various auctioning and contracting techniques may be applicable. Similarly there is a notable potential of applying results of ontology-related research and interoperability initiatives. Apart from extension of ExPlanTech (see Sect. 3.3) to extra-enterprise production planning and involving suppliers of Modelarna Liaz into their planning process, we are not aware of any commercially successful application in this area. It needs to be noted that there are several international research projects investigating deployment of agent technology in virtual organizations (VO), supply chain management and inter-enterprise interoperability provisioning.

Conoise project seeks to support robust and resilient virtual organization formation and operation. It aims to provide mechanisms to assure effective operation of VOs in the face of disruptive and potentially malicious entities in dynamic, open and competitive environments. British Telecom is involved in this project. Ecolead integrated project investigates and supports deployment of technologies for integration of collaborative networks of organizations. In Ecolead, big number of industrial partners are involved, e.g., Virtuelle Fabrik, Gruppo Formula, LogicaCMG Nederland, France Telecom or Siemens Austria. Athena is another major EU integrated project, contributing to interoperability provisioning of the networked organizations and virtual enterprises. Athena integrates high number of industrial partners such as SAP, IBM or Siemens (for full list refer to the project website).

The vision of virtual enterprises as a goal-oriented coalition of cooperating manufacturing and other bodies which explores the MAS ideas has been very popular (especially in Europe), but neither platform nor realistic reportable test case are available yet. See the Table 14 for the application overview.

## 4 Appropriateness of agent technology for industry

This section provides a unified view on appropriateness and level of deployment of agent technology in industrial practice based on the analysis of the above discussed applications. Here we also discuss the different aspects of application functionality often requested from the agent applications as well as what are the frequently deployed agent concepts in those applications. We analyze the listed applications alongside the following criteria (See Table 15):

**Table 15** Generalized analysis of agent deployment domains

| Domain | Agent concepts | Integration | Application maturity | Hardware integration | Functionality | Agent platform | Involved company |
|---|---|---|---|---|---|---|---|
| Manufacturing control | Coordination, negotiation, distributed planning, simulation, interoperability | Yes | Demonstrator | Critical | Control, simulation, diagnostics | Proprietary, JADE | Rockwell Automation, DaimlerChrysler, BHP Billiton |
| Logistics | Coordination, negotiation, distributed planning, simulation | No | Operational system | Yes | Planning and scheduling | Proprietary, MAGENTA Cougaar, (RETSINA) | Magenta, Whitestein, cougaarsoftware |
| Production planning | Coordination, distributed planning, simulation, interoperability | No | Operational system | Important | Planning, scheduling | Proprietary, JADE | Volkswagen, Liaz, SkodaAuto, CERTICON |
| Simulation | Coordination, negotiation, simulation, adjustable autonomy | No | Demonstrator | Yes | Simulation, control, planning | Proprietary, JADE, A-GLOBE | Rockwell Automation, CADENCE, CERTICON, SCA Packaging |
| UAV control | Coordination, negotiation, simulation, adjustable autonomy, BDI architecture | No | Prototype | Yes | Planning simulation, control, deconfliction | Proprietary, A-GLOBE, JACK | AFRL, DSTO, SRI, U.S. ARMY, Quinetiq |
| Space exploration | BDI architecture, adjustable autonomy | No | Operational system | Important | Control, planning, simulation | Proprietary, | NASA, JPL |
| Training | Simulation, adjustable autonomy, BDI | No | Demonstrator | No | Simulation, training | Proprietary, JACK, DEFACTO | AOS, TNO, LFAD, UKMOD |
| Automotive | Distributed learning, meta-reasoning, knowledge sharing | No | Prototype | Important | Diagnostics, simulation, ad-hoc networking | N/A (RETSINA, A-GLOBE) | Denso, DaimlerChrysler, Volkswagen, BMW |
| Supply-chains | Knowledge sharing, auctioning, trust, interoperability | Important | Prototype | No | Integration, planning, coordination | JADE, web services | Virtuell Fabrik, Siemens, LogicaCMG, Liaz, Global Infotek, SAP, IBM, CERTICON |

- *types of agent concepts deployment:* which existing research concepts provided by multi-agent community have been deployed—see Sect. 1.2,
- *legacy system integration:* denotes whether the legacy system integration is one of the key requirements for the area at hand or not,
- *level of application/product maturity:* classification according to whether the application is a prototype (+), demonstrator, or fully operational system, based on the data provided by the authors,
- *hardware integration:* criterion specifying whether the agent technology is solely on the software level or whether hardware components have to be also integrated,
- *types of agent functionality:* specific application functionality—see Sect. 1.3 and
- *agent platform:* specification whether the application is build on top of an existing, reusable multi-agent platform or whether it is build on top of proprietary agent infrastructure.

The mapping from the application domains to requested functionalities and to the deployed agent concepts have been widely discussed in the main body of this article. An interesting observation: In all domains, we have encountered that either the multi-agent oriented concepts or the autonomy oriented aspects were requested. The number of applications with both concepts exploited is very limited. The predominant functionality requested by industry is planning and simulation. These are the functionalities requested in a short-term perspective especially by companies which are motivated to save costs by streamlining their business process. In manufacturing and automotive industry they also request control and diagnostics functionality to be provided. In supply chains and logistics there is a broad potential for integration functionality to be deployed.

The requirements for standardized legacy system integration are obvious only in the supply chain and virtual enterprise applications and in parts in the applications for manufacturing. While the most of the applications are software-based, the manufacturing and networking applications require serious hardware migration (from classical to agent-oriented) and integration processes.

The highest level of maturity and readiness to enter the market has been achieved by logistics, production planning, and space exploration applications only. The level of maturity has been identified as particularly poor in automotive industry, UAV control and supply chain domains. An interesting observation is that the level of application maturity goes against the level of agent concept deployment. In our judgment the use of the multi-agent concept as a design metaphor for massive centralized problem solving systems has not been yet appropriately justified. We are not aware (with the exceptions of LS/ATN testing performed by Whitestein Technologies) of many measurement and testing that would advocate that performance of such systems is better in comparison to already existing methods. This is particularly true about the logistics and several production planning and simulation systems. Paradoxically these are closest to the market and some of them are in the form of applications running at clients' sites. On the other hand the community is busy developing demonstrators and prototypes that exploit better the potentials of multi-agent technology in truly distributed environments—such as integration of supply chains, air-traffic control or car-to-car communication. Unfortunately, we are not aware of any such mature applications to be reported until now.

We have identified some potential for reusability of the developed technology mainly on the level of the agent development and platform integration, especially in the domain of manufacturing, air traffic control, and supply chains. However, it has been identified in [3] that the agent technology has higher chances to enter the market in the form of an application rather than a platform or a development environment. This hypothesis is also supported by the fact that very little of the numerous agent integration platforms, listed on the AgentLink website, were used in the discussed systems or applications. Several agent platforms, developed in the past by major industrial companies, are no longer supported and only few academic agent platforms were used as a basis for software prototypes, demonstrators and commercial systems. The most of the discussed applications are based on proprietary integration platforms or do not use any reusable agent middleware at all. Some of the discusses applications and prototype systems deploy JADE, developed by Telecom Italia labs, a leading open-source agent development environment on the market, JACK provided by Agent Oriented Software, A-globe, a CPL agent development framework supporting environment simulation, interaction inaccessibility and full agent migration, developed at The Gerstner Laboratory, Czech Technical University, RETSINA, a Carnegie-Mellon University flagship multi-agent system or DEFACTO, a system provided by the University of Southern California.

The applications listed in this study mostly support one of the agent deployment justifications from Sect. 1.1. A large number of applications complying with the property $P4$ were oriented towards *simulation* functionality with (demonstrated) further extension towards running physical hardware—especially in manufacturing (Rockwell Automation, DaimlerChrysler) and collaborative robotics domains (e.g., UAV traffic control or underwater vehicles coordination). When deployed on physical hardware the use of agent technology was justified also by the fact that the computation supporting the required decision making has been *decentralized* and *geographically distributed* (property $P1$). In the Shipboard Automation project, the agent technology also proved to increase robustness of system performance in time-critical situations and limited the single point of failure danger. This corresponds to the property $P3$. Agent technology was used here very appropriately. Then there was an important collection of applications that used the multi-agent simulation as a heuristic or an alternative technology for *complex problem solving*—property $P6$. These applications (mainly in production planning and logistics) were deployed in industry and are used in routine operation and clearly made commercial success (SkodaAUTO/Volkswagen, Magenta, Whitestein Technology). From the research point of view it needs to be noted that these applications do not report clear numerical analysis and comparison with existing classical planning methods. Yet another important class of application made a use of the concept of *autonomy*, especially BDI architecture—they complied with the property $P7$. These applications were used mainly for the space exploration problems and also for training purposes. In our judgments, the deployed techniques were applied here correctly. Sadly, in our study we have not reported any application from *competitive domains* or that would hugely require *interoperability*—properties $P2$ and $P5$, with an exception of several application-oriented research activities and demonstration projects from the field of supply chain management, manufacturing and virtual organizations.

The agent technology in the investigated industrial domains provided various different functionality as opposed to the traditional software solutions. In manufacturing control and production planning, the agent technology provided solution based on decentralized approach to control. This is particularly suitable in the real-time domains, where processing large volumes of operational data is not made possible and some decisions (mainly related to replanning and reconfiguration) need to be made autonomously. Even though that distribution of real-time data is viewed as the key advantage for logistics domains, this has not been the case yet. Instead, in logistics the agents are used as an alternative computational paradigm for resource allocation problems. In the field of simulation, the multi-agent concept outperforms classical simulation technology by its support of migration to real-life control and decision making tasks, provided that these tasks are performed in a decentralized fashion (such as UAV control). In the space exploration as much as in training the agent technology provides the concept of autonomy, often based on various models of computational psychology (e.g., BDI).

## 5 Future potentials and challenges

Despite numerous agent applications listed in this paper, we see a substantial potential for agent technology deployment in several specific domains. With a progress of the agent technology transfer and the applied research in the multi-agent system area we expect an increasing demand for agent applications in the domains that fit better the specific capabilities the agents can provide. At the same time we suppose that multi-agent technologies will be less used in the areas where we fail to soundly justify a specific added value provided by the agents. We expect the following future trends in multi-agent system deployment:

*Simulation*: Based on the properties listed in the Sect. 3.4, it is generally assumed that there is a potential of wider deployment of multi-agent systems in the field of simulation. While agent based simulation cannot compete easily with classical simulation tools used for, e.g., simulating economical behavior, agent-based simulation technology shall be used primarily in the applications that require smooth transformation from the agent-based simulation to the agent-based control. There have been applications where the identical set of software agents have been used for simulation purposes and subsequently for running real hardware (UAV vehicles, conveyor belts in a factory, etc.).

*Hardware*: This is yet another reason why we expect that the agent technology will play an important role in the applications that are closely linked to hardware devices. This is true especially in manufacturing, collaborative robotics, and networking. Here the arguments for deployment of decentralized computation solution are stronger than in isolated software applications. In particular, we expect a stronger breakthrough with the agent applications linked to the utilization of RFID technology.

*Software technology*: The progress in software technology, software architectures and tools will also significantly influence the future trends in development and deployment of agent-oriented solutions. Industry technology roadmaps declare future integration of the agent-technology with the service-oriented architectures (SOA) and semantic web technologies, covering such aspects like security, search, messaging, service orchestration, choreography and others [33].

*Autonomy*: An important domain for future deployment of multi-agent systems will likely be connected closely with the requirements for higher autonomy. In particular, we foresee more massive integration of the viable concepts of adjustable autonomy, policies [6], agent-human interactions, and mixed initiative interaction [14] in multi-agent systems. The autonomy related concepts are expected to be applicable in domains such as UAV traffic management and free-flight implementation [85], automotive industry, various defense applications, grid-computing, and resource sharing of computationally demanding processes. Concepts closely related to autonomy are mainly computational reflection [23], self reconfiguration, and self-healing of various computational processes. Closer interaction with autonomic computing community is highly recommended.

*Verification and Testing*: For more successful breakthrough of multi-agent systems in industrial practice more elaborate methods of verification and testing of multi-agent operations would be required. Here, the connection with more fundamental agent research is inevitable. Similarly, we expect calls for more general approaches to multi-agent debugging as a support to the agent systems developers. Debugging support of many multi-agent development environments is currently in their nappies and is based on the concept of the communication sniffer. It is expected that more complex tools for investigating run-time properties, relations and roles of multi-agent interaction are likely to emerge in the agent development community.

*Visualization*: An important field that may enjoy the increased popularity is visualization of multi-agent operations [77]. Even thought the users prefer the agent concepts to be hidden from their graphical user interface, often a visualization of the multi-agent community operation, especially in connection with visualization of the geographical environment is appreciated by the customers. A general, reusable set of tools may make the technology more accessible to industrial users.

*Interoperability*: Interoperability in an open environment has been emphasized in the late 20th century. Our opinion is that its importance was by large overestimated, which has led to less active operation of various standard bodies, such as FIPA recently. The situation is that there are simply many less applications requiring openness and full interoperability. Nevertheless, some industrial clients require, e.g., FIPA compliancy as an evidence of quality and reliability. The truth is so that higher level of interoperability in closed systems (which applies to most of the applications listed in this contribution) is expensive. It requires both—higher development costs and higher computational resources when running the applications. Thus we advise to carry out sensible analysis of benefits that the application may enjoy from interoperability versus the required costs. On the other hand, we expect a further rise in investments towards semantic interoperability and knowledge sharing in various multi-agent applications. Here, closer collaboration with the semantic web community and exploration of its results is assumed [38].

*Security & Safety*: Both the manufacturing and defense industries require communication among the agents being secure and safe. These aspects are in the focus of attention and currently do represent serious obstacles for broader deployment of agent technology in practice. The infrastructure overheads required still represent a significant burden on the agents' community. The future solutions seem to follow two tracks: distributed semantics/ontologies make the message traffic simpler, more secure and safer; meta-level units equipped by reasoning capabilities might contribute to both the security (intrusion detection) and the safety (completeness checking).

## 6 Conclusions

As already emphasized, the field of agent research has advanced substantially during the last couple of years and has established as a rigorous and respected branch of artificial intelligence and computer science. The basic research community is well structured and provides fair measures of scientific quality (e.g., AAMAS conference, JAAMAS journal, and others). Unfortunately there is a gap between fundamental researchers and industrial users of agent technology. There were attempts to bridge this gap by organizing of the various events such as AAMAS industry track, HoloMAS, ATC, and others. The early adopters need to report about their case studies, success stories and failures back to the research community and share their experience across the industrial community.

What we found interesting is that specific sectors of industry, especially those that are motivated to streamline their business process and cut the costs, require solutions to be delivered in the short period of time [3]. This may lead to misuse (and experiences show that it does) followed by dissatisfaction at the use multi-agent technology. We thus suggest extending the research-to-development lifecycle by allowing implementation of the prototype and demonstrators, verifying their properties prior to undertake massive investment in the development of a product. Our experience is that the defense industries are much more generous in this respect and thus less sensitive to the wrong use of this technology.

There are many obstacles to widespread adoption of agent-based technology in industry—compare to [27,41,42]:

- *Thinking:* There is a lack of skill in "distributed thinking" as the current educational systems trains the future engineers to consider mainly (or exclusively) the centralized approaches and strictly centralized systems. There are very few courses in distributed systems.
- *Risks:* The industry is "afraid" of emergent behavior of multi-agent systems without any central unit. In industry, there are no formal algorithms or procedures guaranteeing that the distributed systems would behave as desired. The only way how to verify this is by simulation, but it is impossible to simulate all the system modes for all the configurations.
- *Costs:* Immediate costs of adoption of the agent-based technology are also higher than in the "classical" centralized systems. Much more flexible and intelligent systems need to be developed, and higher investments in both the hardware and software are needed. This was clearly documented by the first manufacturing units brought into the real or semi-real operation (e.g., the steel bar mill).
- *Vendor centric view:* All the reported solutions have been developed by the developers/suppliers who are also expected to maintain the systems. Until the end-users are able to develop and maintain intelligent agents by themselves in a straightforward way, these solutions will remain to be accepted with serious difficulties. It is necessary to develop such frameworks and tools which would enable the end-users to provide in a smooth and simple way just the problem specific knowledge needed for the system operation.

The other related suggestion is not to oversell the technology and deploy it only in the problems where its use would be appropriately justified. In order to sell the technology better and enjoy a substantial market success, we suggest providing the potential customers with grounded numerical analysis of the benefits from agent technology deployment. It is mistaken to assume that mere comparison of the situation before and after agents' deployment provides responsible justification. Functionality of the agent application needs to be compared to alterative planning, simulation or diagnostic technologies available on the market [27]. We advise not to argue only by claiming that 'agents are novel and groundbreaking technology ready to solve all your problems.'

Research and industry shall work together on providing a set of viable test-cases that the researchers can use for validation and verification of the agent concepts. The popular test-beds and competitions such as RoboCup Soccer or Trading Agent Competition shall be taken as good examples. Similarly, we understand that there are strong calls for verification methodologies that would validate whether the deployed agent-based system correspond to the original requirements for the application.

**Acknowledgements** The authors wish to acknowledge the support of the members AgentLink III Management Committee for their strong involvement in the AgentLink Industry Action work package (namely Peter McBurney, Michael Luck, Steve Willmott, Monique Calisti, Terry Payne, Onn Shehory, Simon Thompson, Andrea Omicini and Wiebe v/d Hoek) and also the agent technology companies for their continual support of AgentLink. The authors thank to the Dušan Pavlíček, Pavel Tichý and Pavel Vrba for proof reading of the manuscript. The authors also like to thank to Kenwood H. Hall, VP for Advanced Technology, Rockwell Automation for many hours of stimulating discussions. Special thanks go to Paul Losiewicz (U.S. Air Force) and Ray McGowan (U.S. Army) for evangelizing the agent technology in the U.S. defense industry (supporting the applications in 2.3 and 3.9, respectively).

## References

1. Bauer, B., Muller, J. P., & Odell, J. (2004). Agent UML: A formalism for specifying multiagent interaction. In *Agent-oriented software engineering* (pp. 91–103). Berlin: Springer.
2. Baxter, J., & Horn, G. (2005). Controlling teams of uninhabited air vehicles. In *Proceedings of AAMAS-05 – Industry Track* (pp. 27–33). ACM Press.
3. Belecheanu, R. A., Munroe, S., Luck, M., Miller, T., McBurney, P., & Pěchouček, M. (2006). Commercial applications of agents: Lessons, experiences and challenges. In *Proceedings of AAMAS-06 – Industry Track* (pp. 1549–1555). ACM Press.
4. Berry, D., Usmani, A., Torero, J., Tate, A., McLaughlin, S., Potter, S., et al. (2005). FireGrid: Integrated emergency response and fire safety engineering for the future built environment. In *Workshop on Ubiquitous Computing and e-Research*. Edinburgh, UK: National eScience Centre.
5. Black, G., & Vyatkin, V. (2007). On practical implementation of holonic control principles in baggage handling systems using IEC 61499. In *Proceedings of HoloMAS'07 Conference: Holonic and Multi-Agent Systems for Manufacturing. LNAI No. 4659* (pp. 314–325). Heidelberg: Springer Verlag.
6. Bradshaw, J. M., Feltovich, P., Jung, H., Kulkarni, S., Taysom, W., & Uszok, A. (2004). Dimensions of adjustable autonomy and mixed-initiative interaction. In *LNCS No. 2969* (pp. 17–39). Heidelberg: Springer Verlag.
7. Brazier, F. M. T., Dunin Keplicz, B., Jennings, N., & Treur. J. (1997). DESIRE: Modelling multi-agent systems in a compositional formal framework. *International Journal of Cooperative Information Systems, 6*, 67–94.
8. Brennan, R. W. (2005). An initial automation object repository for OOONEIDA. In *LNAI No. 3593* (pp. 154–164). Heidelberg: Springer Verlag.
9. Brennan, R. W. (2007). Toward real-time distributed intelligent control: A survey of research themes and applications. *IEEE Transactions on System Man and Cybernetics, Part C, 37*(5), 744–765.
10. Brennan, R. W., & Norrie, D. H. (2001). Agents, holons and function blocks: Distributed intelligent control in manufacturing. *Journal of Applied Systems Studies, 2*(1), 1–19.
11. Bunch, L., Breedy, M., Bradshaw, J., Carvalho, M., & Suri, N. (2005). KARMEN: Multi-agent monitoring and notification for complex processes. In *LNAI No. 3593* (pp. 197–206). Heidelberg: Springer Verlag
12. Bussmann, S., & Sieverding, J. (2001). Holonic control of an engine assembly plant – An industrial evaluation. In *Proceedings of IEEE Conference on SMC*, Tucson, AZ.
13. Carvalho, M., Pěchouček, M., & Suri, N. (2006). A mobile agent-based middleware for opportunistic resource allocation and communications. In *LNCS No. 3890*. Heidleberg: Springer Verlag.
14. Chambers, N., Allen, J., & Galescu, L. (2005). A dialogue-based approach to multi-robot tam control. In *Proceedings of Third International Naval Research Labs Multi-Robot Systems Workshop*, Washington, DC.
15. Chen, Y., Gregory, B., Easley, M., Peot, M., Lee, J., & Altshuler, T. (2006). Market-based collaborations for unmanned air vehicles operations. In *Proceedings of IEEE Workshop on Distributed Intelligent Systems* (pp. 273–278). Prague.
16. Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castaño, R., et al. (2005). Lessons learned from the autonomous sciencecraft experiment. In *Proceedings of AAMAS-05 – Industry Track* (pp. 11–18). ACM Press.
17. Christensen, J. H. (1994). Holonic manufacturing systems: Initial architecture and standards directions. In *Proceedings of 1st European Conference on Holonic Manufacturing Systems*, Hannover, pp. 1–20.
18. Deen, M. S. (Ed.) (2003). *Agent based manufacturing: Advances in the holonic approach*. Berlin: Springer Verlag.
19. Dorer, K., & Calisti, M. (2005). An adaptive solution to dynamic transport optimization. In *Proceedings of AAMAS-05 – Industry Track*. ACM Press.
20. Durfee, E. H. (1999). Distributed problem solving and planning. In *A Modern Approach to Distributed Artificial Intelligence* (Chapter 3). The MIT Press
21. Ermolayev, V., & Matzke W.-E.(2007). Performance in industrial holonic systems. In *Proceedings of HoloMAS'07 Conference: Holonic and Multi-Agent Systems for Manufacturing. LNAI No. 4659* (pp. 383–386). Heidelberg: Springer-Verlag.
22. Fletcher, M., McFarlane, D., Thorne, A., Jarvis, D., & Lucas, A. (2003). Evaluating a holonic packing cell. In *LNAI No. 2744* (pp. 246–257). Heidelberg: Springer Verlag.
23. Foltýn, L., Tožička, J., Rollo, M., Pěchouček, M., & Jisl, P. (2006). Reflective-cognitive architecture: From abstract concept to self-adapting agent. In *Proceedings of IEEE Workshop on Distributed Intelligent Systems* (pp. 326–334). Prague.

24. Giorgini, P., Kolp, M., Mylopoulos, J., & Pistore, M. (2004). The tropos methodology: An overview. In F. Bergenti, M.-P. Gleizes, & F. Zambonelli (Eds.), *Methodologies and Software Engineering For Agent Systems*. Kluwer Academic Publishing.
25. Giorgini, P., Müller, P., & Odell, J., (2003). Agent-oriented software engineering. In *LNCS No. 2935*. Heidelberg: Springer.
26. Gruver, W. A., Kotak, D. B., van Leeuwen, E. H., & Norrie, D. H. (2003). Holonic manufacturing systems: Phase II. In *LNAI No. 2744* (pp. 1–14). Heidelberg: Springer-Verlag
27. Hall, K. H., Staron, J. R., & Vrba, P. (2005). Experience with holonic and agent-based control systems and their adoption by industry. In *LNAI No. 3593* (pp. 1–10). Heidelberg: Springer-Verlag.
28. Hegny, I., Holzer, R., Grabmair, G., Zoitl, A., Auinger, F., & Wahlmuller, E. (2007). A distributed energy management approach for autonomous power supply systems. In *Proceedings of INDIN'07: 5th International Conference on Industrial Informatics* (pp. 1111–1116). IEEE.
29. Himoff, J., Rzevski, G., & Skobelev, P. (2006). Multi-agent logistics i-Scheduler for road transportation. In *Proceedings of AAMAS-06 – Industry Track* (pp. 1514–1521). ACM Press.
30. Himoff, J., Skobelev, P., & Wooldridge, M. (2005). MAGENTA technology: Multi-agent systems for industrial logistics. In *Proceedings of AAMAS-05 – Industry Track* (pp. 60–66). ACM Press.
31. Hummer, O., Sunder, Ch., Strasser, T., Rooker, M. N., & Kerbleder, G. (2007). Downtimeless system evolution: Current state and future trends. In *Proceedings of INDIN'07: 5th International Conference on Industrial Informatics* (pp. 1123–1128). IEEE.
32. Jacobi, S., Madrigal, C., Leon-Soto, E., & Fischer, K. (2005). AgentSteel: An agent-based online system for the planning and observation of steel production. In *Proceedings of AAMAS-05 – Industry Track* (pp. 114–119). ACM Press.
33. James, F., & Smit, H. (2005). Service-oriented paradigms in industrial automation. *IEEE Transactions on Industrial Informatics, 1*(1), 62–70.
34. Jennings, N. (1999). Agent-oriented software engineering. In *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99), LNCS No. 1647* (pp. 1–7). Heidelberg: Springer-Verlag.
35. Jennings, N. R., Norman, T. J., Faratin, P., O'Brien P., & Odgers, B. (2000). Implementing a business process management system using ADEPT: A real-world case study. *Applied Artificial Intelligence Journal, 14*(5), 421–490.
36. Karim, S., & Heinze. C. (2005). Experiences with the design and implementation of an agent-based autonomous UAV controller. In *Proceedings of AAMAS-05 – Industry Track* (pp. 19–26). ACM Press.
37. Karuna, H., Valckenaers, P., Verstraete1, P., Zamfirescu C., & Van Brussel, H. (2005). Emergent forecasting using a stigmergy approach in manufacturing coordination and control. In S. A. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, & R. Nagpal (Eds.), *Engineering Self-Organising Systems: Methodologies and Applications* (pp. 210–222). Heidelberg: Springer Verlag.
38. Lastra, J. L. M., & Delamer, I. M. (2006). Semantic web services in factory automation: Fundamental insights and research roadmap. *IEEE Trans. Industrial Informatics, 2*(1), 1–12.
39. Lastra, J. L. M., Torres, E. L., & Colombo A. W. (2005). A 3D visualization and simulation framework for intelligent physical agents. In *LNAI No. 3593* (pp. 11–22). Heidelberg: Springer Verlag.
40. Lewis, R. W. (2001). *Modeling Control Systems Using IEC 61499*. IEE Publishing.
41. Mařík, V., Fletcher, M., & Pěchouček, M. (2002). Holons & agents: Recent developments and mutual impacts. In *LNAI No. 2322* (pp. 233–267). Heidelberg: Springer Verlag.
42. Mařík, V., & McFarlane, D. (2005). Industrial adoption of agent-based technologies. *IEEE Intelligent Systems, 20*(1), pp. 27–35.
43. Maturana, F. P., Kotina, R., Staron, R., Tichý, P., & Vrba, P. (2006). Agent-based water/waste water control system architecture. In: *Proceedings IADIS International Conference on Applied Computing*, San Sebastian.
44. Maturana, F. P., Staron, R. J., Discenzo, F. M., Hall, K., Tichý, P., Šlechta, P., et al. (2004). Integrating multi-agent systems: A case study. In L. M. Camarinha-Matos (Ed.), *Emerging Solutions for Future Manufacturing Systems* (pp. 99–108). New York: Springer Science+Business Media.
45. Maturana, F. P., Staron, R. J., Tichý, P., Šlechta, P., & Vrba, P. (2005). A strategy to implement and validate industrial applications of holonic systems. In *LNAI No. 3593* (pp. 111–120). Heidelberg: Springer Verlag.
46. Maturana, F., Tichý, P., Šlechta, P., Staron, R. J., Discenzo, F. M., Hall, K. H., et al. (2003). Cost-based dynamic reconfiguration system for evolving hierarchies. In: *LNAI No. 2744* (pp. 310–320). Heidelberg: Springer Verlag.
47. McDonald, N., & Znati, T. (1999). A mobility-based framework for adaptive clustering in wireless ad-hoc networks. *IEEE Journal on Selected Areas in Communication, 17*(8), 1466–1487.
48. Muller, J. P. (1996). The design of intelligent agents: A layered approach. In *LNAI No. 1177*. Heidelberg: Springer Verlag.

49. Muscettola, N., Dorais, G., Fry, C., Levinson, R., & Plaunt, C. (2002). IDEA: Planning at the core of autonomous reactive agents. In *Proceedings of the Workshops at the AIPS-2002 Conference*, Tolouse, France.
50. Muscettola, N., Nayak, P., Pell, B., & Williams, B. (1998). Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence, 103*(1–2), 5–48.
51. Nolte, T., Hansson, H., & Bello, L. (2005). Automotive communications past, current and future. In *Proceedings of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'05)* (pp. 985–992). IEEE Industrial Electronics Society.
52. Nourbakhsh, I., Lewis, M., Sycara, K., Koes, M., Yong, M., & Burion, S. (2005). Human-robot teaming for search and rescue. *IEEE Pervasive Computing, 4*(1), 72–78.
53. Obitko, M., & Mařík, V. (2005). Integrating transportation ontologies using semantic web languages. In *LNAI No. 3593* (pp. 99–110). Heidelberg: Springer Verlag.
54. Omicini, A. (2001). SODA: Societies and infrastructures in the analysis and design of agent-based systems. In *Agent-Oriented Software Engineering, Proceedings First International Workshop AOSE 2000, LNCS No. 1957* (pp. 185–193). Heidelberg: Springer Verlag.
55. Padgham, L., & Winikoff, M.(2002). Prometheus: A methodology for developing intelligent agents. In *Proceedings of the First Imitational Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*. ACM Press.
56. Padgham, L., & Winikoff, M. (2004). *Developing intelligent agent systems: A practical guide*. Wiley.
57. Paolucci, M., & Sacile, R. (2005). *Agent-based manufacturing and control systems*. Boca Raton: CRC Press.
58. Parunak, H. V. D., Brueckner, S., & Odell, J. (2003). Swarming coordination of multiple UAV's for collaborative sensing. In *Proceedings of Second AIAA Unmanned Unlimited Systems Technologies and Operations Aerospace Land and Sea Conference and Workshop & Exhibit*, San Diego.
59. Pěchouček, M., Mařík, V., & Bárta, J. (2002). A knowledge-based approach to coalition formation. *IEEE Intelligent Systems, 17*, 17–25.
60. Pěchouček, M., Rehák, M., Charvát, P., Vlček, T., & Kolář, M. (2007). Multi-agent planning in mass-oriented production. *IEEE Transactions System, Man and Cybernetics, Part C, 37*(3), 386–395.
61. Pěchouček, M., Šišlák, D., Pavlíček, D., & Uller, M. (2006). Autonomous agents for air-traffic deconfliction. In *Proceedings AAMAS-06 – Industry Track* (pp. 1498–1505). ACM Press.
62. Pěchouček, M., Vokřínek, J., & Bečvář, P. (2005). ExPlanTech: Multiagent support for manufacturing decision making. In *IEEE Intelligent Systems, 20*(1), 67–74.
63. Peysakhov, M., Dugan, C., Modi, J., & Regli, W. (2006). Quorum sensing on mobile ad-hoc networks. In *Proceedings of AAMAS-06* (pp. 1104–1106). ACM Press.
64. Rao, A., & Georgeff, M. (1995). BDI Agents: From theory to practice. In *Proceedings the First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco.
65. Regis, V., Osher, Y., & Andrew, A. (2006). UAV airspace management system UAMS. In *Proceedings of AAMAS-06 – Demo Track* (pp. 1449–1450). ACM Press.
66. Rehak, M., Pechoucek, M., Bartos, K., Grill, M., Celeda, P., & Krmicek, V. (2008). CAMNEP: An intrusion detection system for high-speed networks. In *Progress in informatics*, pp. 65–74.
67. Rollo, M., Novák, P., & Jisl, P. (2005). Simulation of underwater surveillance by a team of autonomous robots. In *LNAI No. 3593* (pp. 207–220). Heidelberg: Springer Verlag.
68. Rollo, M., Novák, P., Kubalík, J. & Pěchouček, M. (2004). Alarm root cause detection system. In *Emerging Solutions for Future Manufacturing Systems* (pp. 109–116). New York: Springer.
69. Schurr, N., Marecki, J., Tambe, M., & Scerri, P. (2005). Demonstration of DEFACTO: Training tool for incident commanders. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (The Netherlands, July 25–29, 2005), AAMAS'05 (pp. 147–148). New York: ACM.
70. Semmel, G., Davis, S., Leucht, K., Rowe, D., Kelly, A., & Bölöni, L. (2005). Launch commit criteria monitoring agent. In *Proceedings of AAMAS-05 – Industry Track* (pp. 3–10). ACM Press.
71. Shehory, O., Sycara, K., Sukthankar, G., & Mukherjee, V. (1999). Agent aided aircraft maintenance. In *Proceedings Third International Conference on Autonomous Agents*, Seattle, Washington, USA.
72. Sierhuis, M., Clancey, W. J., & Sims, H. M. (2002). Multiagent modeling and simulation in human-robot mission pperations work system design. In *Proceedings of the Hawai'i International Conference on System Sciences*. IEEE.
73. Sierhuis, M., Clancey, W. J., van Hoof, R., & de Hoog, R. (2000). Modeling and simulating human activity. In *Autonomous Agents 2000 Workshop on Intelligent Agents for Computer Supported Co-operative Work: Technology and Risks*, Barcelona. Spain.

74. Šišlák, D., Pěchouček, M., Volf, P., Pavlíček, D., Samek, J., Mařík, V., & Losiewicz, P. (2008). AGENTFLY: Towards multi-agent technology in free flight air traffic control. In *Defence Industry Application of Autonomous Agents and Multi-Agent Systems*, Birkhäuser Verlag Publishing (accepted for publication).
75. Šišlák, D., Rehák, M., Pěchouček, M., & Benda, P. (2005). Optimizing agents operation in partially inaccessible and disruptive environment. In *Proceedings Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference*.
76. Šišlák, D., Rehák, M., Pěchouček, M., Rollo, M., & Pavlíček, D. (2005). A-globe: Agent development platform with inaccessibility and mobility support. In R. Unland, M. Klusch, & M. Calisti (Eds.), *Software Agent-Based Applications, Platforms and Development Kits* (pp. 21–46). Berlin: Birkhauser Verlag.
77. Staron, R. J., Tichý, P., Sindelář, R., & Maturana, F. P. (2007). Methods to observe the clustering of agents within a multi-agent system. In *Proceedings of HoloMAS'07 Conference: Holonic and Multi-Agent Systems for Manufacturin, LNAI No. 4659* (pp. 127–136). Heidelberg: Springer Verlag.
78. Strasser, T., Zoitl, A., Auinger, F., & Sunder Ch. (2005). Towards engineering methods for reconfiguration of distributed real-time control systems based on the reference model of IEC 61499. In *LNAI No. 3593* (pp. 165–175). Heidelberg: Springer Verlag.
79. Sycara, K., Giampapa, J. A., Langley, B. K., & Paolucci, M. (2003). The RETSINA MAS, a case study. In *LNCS No. 2603* (pp. 232–250). Heidelberg: Springer Verlag.
80. Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research, 7*: 83–124.
81. Tate, A. (2006). The helpful environment: Geographically dispersed intelligent agents that collaborate. *IEEE Intelligent Systems, 21*(3), 57–61.
82. Tichý, P. (2004). Fault tolerant and fixed scalable structure of middle agents. Computational logic in multi-agent systems. In *4th International Workshop, CLIMA IV, LNCS No. 3259* (pp. 53–62). Heidelberg: Springer-Verlag.
83. Tichý, P., Šlechta, P., Maturana, F., & Balasubramanian, S. (2002). Industrial MAS for planning and control. In *LNAI No. 2322* (pp. 280–295). Heidelberg: Springer Verlag.
84. Tichý, P., Šlechta, P., Staron, R. J., Maturana, F. P., & Hall, K. H. (2006). Multiagent technology for fault tolerance and flexible control. *IEEE Transactions on System, Man and Cybernetics, Part C, 36*(5), 700–705.
85. Tomlin, C., Pappas, G., & Sastry, S. (1997). Conflict resolution for air traffic management: A case study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*.
86. Tožička, J., Jakob, M., & Pěchouček, M. (2006). Market-inspired approach to collaborative learning. In *LNAI No. 4149* (pp. 213–227). Heidelberg:Springer Verlag.
87. van Doesburg, W., Heuvelink, A., & van den Broek, E. (2005). TACOP: A cognitive agent for a naval training simulation environment. In *Proceedings of AAMAS-05 – Industry Track* (pp. 34–41). ACM Press.
88. Vrba, P., & Macůrek, F., & Mařík, V. (2006). Using radio frequency identification in agent-based control systems for industrial applications. In *Proceedings of 12th IFAC Symp INCOM*, pp. 459–464.
89. Vrba, P., & Mařík, V. (2005). From holonic control to virtual enterprises: The multi-agent approach. In R. Zurawski (Ed.), *The Industrial Information Technology – Handbook* (pp. 107–109). Boca Raton: CRC Press.
90. Wernstedt, F., & Davidsson, P. (2002). An agent-based approach to monitoring and control of district heating systems. In *LNAI No. 2358* (pp. 801–812). Heidelberg: Springer Verlag.
91. Woghiren, K. (2005). Modeling the aerospace aftermarket with multi-agents systems. Invited talk at *Agent Technology Conference*, available from http://www.agentlink.org/agents-stockholm/presentations/Lost%20Wax.pdf, Stockholm.
92. Wollkind, S., Valasek, J., & Ioerger, T. R. (2004). Automated conflict resolution for air traffic management using cooperative multi-agent negotiation. In *Proceedings of the American Institute of Aeronautics and Astronautics Conference on Guidance, Navigation, and Control*, Providence, RI.
93. Wooldridge, M., Jennings, N. J., & Kinny, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems, 3*(3), 285–312.
94. Yu, B., Scerri, P., & Sycara, K. (2006). Scalable and reliable data delivery in mobile ad hoc sensor networks. In *Proceedings of AAMAS-06* (pp. 1071–1078). ACM Press.
95. Zambonelli, F., & Omicini, A. (2004). Challenges and research directions in agent-oriented software engineering. *Journal of Autonomous Agents and Multi-Agent Systems, 9*(3), 253–287.
96. Zlotkin, G., & Rosenschein, J. S. (1998). Negotiation and task sharing among autonomous agents in cooperative domains. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 912–917). San Mateo, CA: Morgan Kaufmann
97. Zoitl, A., Strasser, T., Hall, K. H., Staron, R. J., Sunder, Ch., & Favre-Bulle, B. (2007). The past, present, and future of IEC 61499. In *Proceedings of HoloMAS'07 Conference: Holonic and Multi-Agent Systems for Manufacturing. LNAI No. 4659* (pp. 1–14). Heidelberg: Springer Verlag.