Multi-Agent Systems in the Industry Three Notable Cases in Italy

Federico Bergenti
Dipartimento di Matematica
Univesità degli Studi di Parma
Parma, Italy
Email: federico.bergenti@unipr.it

Eloisa Vargiu
Dipartimento di Ingegneria Elettrica ed Elettronica
Università degli Studi di Cagliari
Cagliari, Italy
Email: vargiu@diee.unica.it

***Abstract*—This paper reports on three notable examples of the use of multi-agent systems in the Italian Industry. First, we introduce the topic and we outline some examples of real-world agent-based software application. Then, we describe in details the use of multi-agent systems in three software packages for *(i)* personalized press reviews, *(ii)* monitoring boats in marine reserves, and *(iii)* advanced contact centers. Finally, we draw some conclusions and summarize the lesson learnt from described experiences. The discussion focuses on the benefits and problems that the choice of agent technology brought.**

I. Introduction

From the '80s the term "agent" has been adopted by a variety of sub-disciplines of artificial intelligence and computer science [1]. In particular, from the '90s we can talk of "Multi-Agent Systems" (MAS) in software engineering, data communications and concurrent systems research, as well as robotics, artificial intelligence and distributed artificial intelligence [2].

In 1996, FIPA[1] was established to produce software standards for heterogeneous, interacting agents and agent-based systems. Since its conception, FIPA has played a crucial role in the development of agents standards and has promoted a number of initiatives and events that contributed to the development and uptake of agent technology. According to FIPA reports many MAS and relative platforms have been developed; among others, let us recall here FIPA-OS [3], JACK [4], ZEUS [5] and JADE [6].

Nowadays, only few MAS solutions have been devised, deployed, and adopted in industrial applications [7]. As noted in [8], agent technologies have been concentrated in a small number of business sectors:

- *Simulation and training applications in defense domains*, e.g., the system developed by Agent Oriented Software to aid the Ministry of Defence in military training[2] and the NASA's OCA Mirroring System [9];
- *Network management*, e.g., IRIS, a tool for strategic security allocation in transport networks [10] and the system developed by Magenta to help a shipping company improve oil distribution shipping networks [11];
- *User interface and local interaction management in telecommunication networks*, e.g., the systems developed by Telecom Italia on WADE to implement a mediation layer between network elements and OSS and to provide step-by-step guidance to technicians performing maintenance operations in the fields [12];
- *Schedule planning and optimization in logistics and supply-chain management*, e.g., Living Systems Adaptive Transportation Networks, a system for logistics management developed by Whitestein Technologies [13] and the system for heat and sequence optimization in the supply chain of steel production proposed in [14];
- *Control system management in industrial plants*, e.g., the MAS solution applied in mass-production planning of car engines for Skoda Auto [15];
- *Simulation modeling to guide decision-makers in public policy domains*, e.g., the one developed by Eurobios to improve production schedules for a cardboard box manufacturer [16].

According to [17], the main bottlenecks that prevent a fast and massive adoption of agent-based solutions in real world applications are: *(i)* limited awareness about the potentials of agent technology; *(ii)* limited publicity of successful industrial projects carried out with the agent technology; *(iii)* misunderstandings about the effectiveness of agent-based solutions, characterized by over-expectations of the early industrial adopters and subsequent frustration; *(iv)* risks for adopting a technology that has not been already proven in large scale industrial applications; as well as *(v)* lack of mature enough design and development tools for industrial deployment.

More than ten years after the first FIPA specifications [18], we state that time is ripe to adopt MAS in the industry [19]. In fact, the MAS technology is already effective for deploying real applications from both a software engineering [20] and a technological perspective [21]. To this end, in this paper, we present and discuss our experience in using MAS for developing industrial applications. In particular, we describe in details the use of MAS in three software packages for *(i)* personalized press reviews, *(ii)* monitoring boats in marine reserves, and *(iii)* advanced contact centers.

The rest of the paper is organized as follows: in Section II, Section III, and Section IV we describe and discuss the system

[1]http://www.fipa.org
[2]http://ats.business.gov.au/Company/CompanyProfile.aspx?ID=22

for creating personalized press reviews, for monitoring boats in marine reserves, and for managing advanced contact centers, respectively. In particular, for each system we first illustrate the corresponding scenario, then, we present the proposed solution, and, finally, we discuss the underlying motivations in adopting a MAS-based approach. Section V ends the paper with some conclusions and a summary of the lesson learnt from described experiences.

II. Personalized Press Reviews

In this section, we present a MAS explicitly devoted to generate press reviews by *(i)* extracting articles from Italian online newspapers, *(ii)* classifying them using text categorization according to user's preferences, and *(iii)* providing suitable feedback mechanisms [22]. The system has been developed and deployed together with Arcadia Design[3] under the project DMC (Digital Media Center) ordered by Cosmic Blue Team[4].

A. The Scenario

The World Wide Web offers a growing amount of information and data coming from different and heterogeneous sources. As a consequence, it becomes more and more difficult for Web users to select contents according to their interests, especially if contents are frequently updated (e.g., news, newspaper articles, reuters, RSS feeds, and blogs). Supporting users in handling the enormous and widespread amount of Web information is becoming a primary issue. To this end, several online services have been proposed (e.g., Google News[5] and PRESSToday[6]). Unfortunately, they allow users to choose their interests among macro-areas (e.g. *economics*, *politics*, and *sport*), which is often inadequate to express what the user is really interested in. Moreover, existing systems typically do not provide a feedback mechanism able to allow the user to specify non-relevant items—with the goal of progressively adapting the system to her/his actual interests.

Fig. 1. An example of results provided by the personalized press review system.

[3]http://www.arcadiadesign.it
[4]http://www.cbt.it
[5]http://news.google.com
[6]http://www.presstoday.com

B. The Implemented MAS

To generate press reviews, the system is organized in three layers, each aimed at performing a specific information-retrieval step:

- *Information Extraction*. To perform information extraction, we use several wrapper agents, each associated with a specific information source: the Reuters portal[7], The Times[8], The New York Times[9], the Reuters document collection, and the taxonomy adopted during the classification phase. Once extracted, all the information is suitably encoded to facilitate the text categorization task. To this end, all non-informative words, e.g., prepositions, conjunctions, pronouns and very common verbs are removed using a stop-word list. After that, a standard stemming algorithm removes the most common morphological and inflectional suffixes. Then, for each category of the taxonomy, feature selection, based on the information-gain heuristics, has been adopted to reduce the dimensionality of the feature space.
- *Hierarchical Text Categorization*. To perform hierarchical text categorization, we adopt the Progressive Filtering approach proposed in [23]. Each node of a given taxonomy is a classifier entrusted with recognizing all corresponding relevant inputs. Any given input traverses the taxonomy as a *token*, starting from the root. If the current classifier recognizes the token as relevant, it passes it on to all its children (if any). The typical result consists of activating one or more pipelines of classifiers within the taxonomy.
- *User's Feedback*. When an irrelevant article is evidenced by the user, it is immediately embedded in the training set of a *k*-NN classifier that implements the user feedback. A suitable check performed on this training set after inserting the negative example allows to trigger a procedure entrusted with keeping the number of negative and positive examples balanced. In particular, when the ratio between negative and positive examples exceeds a given threshold (by default set to 1.1), some examples are randomly extracted from the set of truely positive examples and embedded in the above training set.

The prototype of the system has been devised through X.MAS [24]—a generic multi-agent architecture, built upon JADE [6], devised to make it easier the implementation of information retrieval and information filtering applications. Through the user interface, the user can set *(i)* the source from which news will be extracted, and *(ii)* the topics s/he is interested in. As for the newspaper headlines, the user can choose among the Reuters portal, The Times, and The New York Times. As for the topics of interest, the user can select one or more categories in accordance with the given RCV1 taxonomy. First, information agents able to handle the selected newspaper headlines extract the news. Then, all agents that embody a classifiers trained on the selected topics are involved

[7]http://www.reuters.com
[8]http://www.the-times.co.uk/
[9]http://www.nytimes.com/

to perform text categorization. Finally, the system supplies the user with the selected news through suitable interface agents (see Figure 1). The user can provide a feedback to the system by selecting all non-relevant news (i.e., false positives). This feedback is important to let the system adapting to the actual interests of the corresponding user.

C. The Role of Agents

The motivation for adopting a MAS lies in the fact that a centralized classification system might be quickly overwhelmed by a large and dynamic document stream, such as daily-updated online news [25]. Furthermore, the Web is intrinsically a distributed system and offers the opportunity to take advantage of distributed computing paradigms and distributed knowledge resources.

Let us also note that an information retrieval system must take into account several issues, such as: *(i)* how to deal with different information sources and to integrate new information sources without re-writing significant parts of it, *(ii)* how to suitably encode data in order to put into evidence the informative content useful to discriminate among categories, *(iii)* how to control the imbalance between relevant and irrelevant articles, *(iv)* how to allow the user to specify her/his preferences, and *(v)* how to exploit the user's feedback to improve the overall performance of the system. The above problems are typically strongly interdependent in state-of-the-art systems. To better concentrate on these aspects separately, we adopted a layered multiagent architecture, able to promote the decoupling among all aspects deemed relevant.

III. Monitoring Boats in Marine Reserves

Under the project "A Multiagent System for Monitoring Intrusions in Marine Reserves" (POR Sardegna 2000/2006, Asse 3 - Misura 3.13) supported by Regione Autonoma della Sardegna, we experimented a MAS-based solution with the goal of monitoring and signaling intrusion in marine reserves. In the corresponding system, developed and deployed together with the companies SETI S.N.C.[10] and ICHNOWARE S.A.S., authorized boats are equipped with suitable devices able to transmit (through GSM technology) their position (through GPS technology). In this way, the corresponding scenario encompasses two kinds of boats: authorized, recognizable by the GPS+GSM devices, and unauthorized. Both kinds of boats are expected to be identified by a digital radar able to detect their position in the protected area. Comparing the positions sent by boats with those detected by the radar allows to identify unauthorized boats.

A. The Scenario

In the summertime, in Sardinia and in its small archipelago, tourists sometimes sail in protected or forbidden areas close to the coast. Monitoring such areas with the goal of discriminating between authorized and unauthorized boats is quite complicated. In fact, along Sardinian coasts, there are two-hundred tourist harbors with about thirteen thousand places available for boats and several services for boat owners. Monitoring large areas without suitable resources (such as radars) can be highly uneconomic, since staff operators would be (and typically are) compelled to directly patrol them over time. A typical solution consists of using a radar system controlled by a central unit located ashore in a strategical position. Radar signals allow to detect the positions of the boats that sail in the controlled area.

[10]http://www.setiweb.it

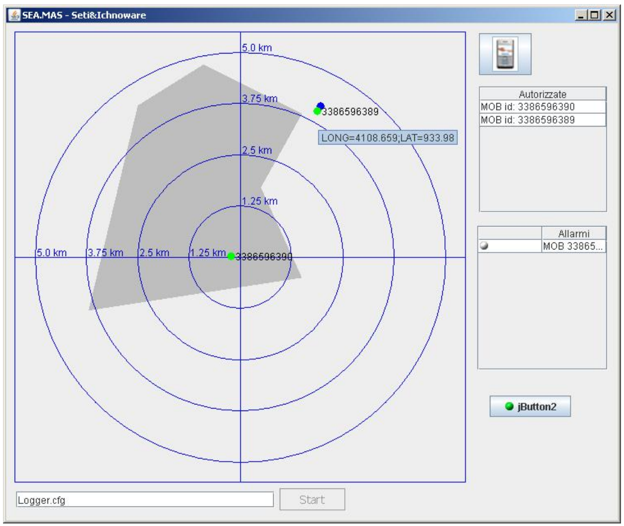

Fig. 2. A snapshot of SEA.MAS user interface

B. The Implemented MAS

The MAS aimed at monitoring boats in marine reserves has been called SEA.MAS [26] to highlight the fact that it stems from X.MAS. The adoption of X.MAS comes from the fact that the problem of monitoring and signaling intrusions in marine reserves can be seen as a particular information retrieval task: radar and GPS+GSM devices are information sources, while authorized and unauthorized boats are categories to be discriminated.

The first step for customizing X.MAS to a specific application consists of extending each abstract class with the goal of providing the required features and capabilities:

- *Information level.* Information sources are the digital radar and GPS+GSM devices. For each information source, a suitable information agent has been devised to embody the information provided therein. GPS- and radar-signals are retrieved by suitable information agents, devised to extract the actual position according to GPS and NMEA standards.
- *Filter level.* Filter agents are aimed at encoding the information extracted by the information agents. The encoding activity consists of creating events containing the position of the detected boats and their identification code, when available. Moreover, filter agents are devoted to avoid two kinds of redundancy: information detected more than once from the same device (caching) or throughout different devices (information overloading).

- *Task level*. A task agent is created for each boat, the underlying motivation being the need to centralize the knowledge regarding the position of a boat and its state. As for the position, events are classified as belonging either to anonymous sources or to known sources. For known sources the state reports their identification code and—when available—further information, i.e., a description of the boat and/or owner's data. The main tasks of the agents belonging to this level are: *(i)* to follow a boat position during its navigation, also dealing with any temporary lack of signal; *(ii)* to promptly alerting the interface agents in the event that unauthorized boats are identified; and *(iii)* to handle messages coming from the interface level, e.g. false alarm notification.
- *Interface level*. Suitable interface agents allow the system administrator and staff operators to interact with the system. In both cases, the corresponding interface agent is aimed at getting a feedback from the user, for instance to inform relevant agents about changes occurred in the environment or about faults that might occur in devices located on the authorized boats. User feedback can also be used to improve the overall ability of discriminating among authorized and unauthorized boats. This kind of user feedback is performed through a simple solution based on the k-NN technology. When either a false positive or a false negative is evidenced by the user, it is immediately embedded in the training set of the k-NN classifier that implements the feedback.

SEA.MAS has been experimented in a marine reserve located in the North of Sardinia. The interface agent (see Figure 2) represents in different colors different states: authorized, unauthorized, not-detected, under verification. In case of intrusion, a sound is generated together with the position of unauthorized boats; such a signal can be forwarded to the security patrol, whose primary goal is to catch intruders. SEA.MAS involves a number of agents that in practice is proportional to the number of boats being monitored. In fact, whereas the number of middle-, information-, filter-, and interface-level agents is fixed (i.e., one agent for each middle-span level, two agents at the information level, one agent at the filter level, and typically one agent at the interface level), a task agent is instantiated for each boat. This fact does not generate any scalability problem for two main reasons: the number of boats sealing in marine reserves is typically less than a hundred at a time and, if needed, agents could be distributed on several nodes. In practice, the maximum number of boats in the selected marine reserve was 20.

C. The Role of Agents

Since monitoring boats requires to involve entities able to cooperate each other, move in the environment, and adapt to changes that may occur, an agent solution could help in the development of such a system.

As for cooperation, SEA.MAS agents can horizontally and vertically cooperate. The former kind of cooperation occurs among agents belonging to a specific level in accordance with the following schemes: pipeline, centralized composition, and distributed composition. The latter is performed—across levels—throughout middle agents, which support communication among requesters and providers belonging to adjacent levels of the architecture.

As for mobility, all involved agents can be mobile, if needed. In fact, in case of a large number of agents (i.e., boats) this requirement becomes mandatory in order to handle the computational complexity. Thus, mobility permits the run-time deployment of agents in a distributed architecture.

As for adaptivity, task agents are able to adapt their behavior in order to avoid losing boats in case of signal absence (i.e., areas devoid of GSM signal).

IV. Advanced Contact Centers

Back in 2002 FRAMeTech S.R.L.[11] developed the Mercury Contact Center Suite (Mercury for short) to provide its customers with a full-featured, cost-effective solution for inbound/outbound contact centers and automatic telephony services. The product has been successfully adopted in the last 8 years from a large number of customers ranging from SMEs to Large Enterprises to implement automatic and semi-automatic services up to 60 concurrent users and 1,000 calls/hour. We can say now that Mercury is a mature product with a solid architecture and no foreseen limitations in the scalability of functionality and performances.

From the functional point of view, Mercury is a point of convergence for common communication media: telephone (analog, digital and IP-based), faxes and e-mails. From a technological point of view, Mercury is completely Web-based and implemented taking advantage of innovative, open-source technologies like Java.

A. The Scenario

Mercury is an open platform that is verticalized to meet the needs of single customers. Just to give an example of the service scenarios that Mercury addresses, we can mention the common case of inbound contact centers. The basic functionality of an inbound contact center are: *(i)* to answer telephone calls from multiple lines, *(ii)* to provide the caller with an interactive menu, *(iii)* to hold on music the call if no user is available, *(iv)* to dispatch queued calls to the best user via, e.g., skill based routing, and, *(v)* to provide the chosen user with all available details on the caller in order to meet her/his requests.

Mercury advocates a similar approach also for outbound contact centers, which are services that provide, at least, the functionality *(i)* to allow interfacing a (possibly external) database to collect a list of names and numbers to call, *(ii)* to place calls to the selected numbers ensuring that, at answer-time, a good user with the right skills and capabilities would be eventually available, *(iii)* to provide the chosen user with all available details on the called party in order to value the call.

[11] http://www.frametech.it

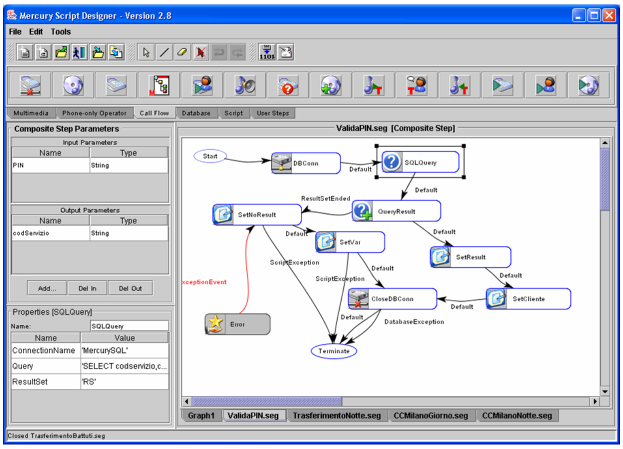

Fig. 3. Mercury Contact Center Suite workflow editor

Mentioned examples are very simple cases of semi-automatic scenarios, because a user is always involved in the so-called *workflow* of the service. It is worth mentioning that Mercury supports also fully automatic services, e.g., for automatic calls to personnel on duty in quality processes, and for self-serve banking services.

B. The Implemented MAS

Even if the analysis and specification of requirements of Mercury were done using a use-case driven approach, the design of the whole system was centered around agent-oriented concepts. Unfortunately, the concrete implementation of the product could not exploit the available agent-oriented technologies, e.g., JADE [6], because of the initial tight non-functional requirements. Actually, the product still ships embedded onboard of a low-end PBX (Private Branch Exchange).

In order to discuss the role of agents in the design of Mercury, we first need to outline the architecture of the product. The main modules of the architecture are:

1) *Contact Manager* module, in charge of providing a converged interface for sending commands to and processing events from available communication channels[12].
2) *User Manager* module, that manages the lifecycle of users in the system and that provides access to user profiles.
3) *ACD (Automatic Contact Dispatcher)* module, that manages the allocation and dynamic rerouting of contacts to available users via customizable skill-based policies.
4) *Workflow Manager* module, in charge of coordinating the execution of scripts that control the behavior of services.

Agents are used to implement all such modules, as follows.

The Contact Manager module is made of a group of agents, one for each inbound/outbound communication channel, that are in charge of the supervision of the physical state of channels and of their relative expected behaviors. Such agents realize *(i)* functionality that are common to all channels, e.g., activation/deactivation and notification of state change, and *(ii)* channel-specific functionality, e.g., agents in charge of managing a telephone line can send commands to dial a number or to answer a call, and they can also inform interested parties of changes in the state of the line, e.g., when the line is ringing or when line is put on hold.

The User Manager module associates an agent to each and every active user in the system and such agents are in charge of *(i)* managing negotiations on behalf of users, and *(ii)* maintaining users' skill and capability profiles. Moreover, User Manager agents are also meant to implement groups of users with similar skills and capabilities in order to provide a means for dynamic allocation of inbound/outbound contacts.

The ACD module is a standard component of contact center systems that manages a set of queues and that is in charge of the dynamic allocation of contacts to users. Mercury uses such a standard approach and it provides an agent for each ACD queue. Dynamic dispatching of, e.g., incoming calls, is therefore a matter of negotiation between the relative Contact Manager line agent and available ACD agent in order to maximize the overall quality of the service. Mercury has sophisticated dynamic policies meant to maximize the Quality of Service (QoS) in terms of, e.g., the amount of time spent in queue, the length of the queue, the workload of personnel and the skills and capabilities of personnel.

The Workflow Manager module provides an agent for each active service to ensure that service activities are properly orchestrated. Mercury provides a graphical editor (see Figure 3) in the style of the popular WADE editor [12], to model the workflow of a service in terms of a *flowchart with asynchronous exceptions*. Once a service gets activated, e.g., because of an incoming call on a tall free number, a specific agent for the service is created to ensure that the call would follow the nominal steps as sketched in the relative flowchart. Unfortunately, asynchronous events may break the nominal flow of event, e.g., the caller drops before choosing an entry in the interactive menu, and the service agent is in charge of managing the event and performing needed actions to ensure a proper termination of the contact.

C. The Role of Agents

Common agent-oriented abstractions, e.g., dynamic capability-based coordination, dynamic contracting and goal-directed workflows, are ubiquitous in the design of Mercury, as described previously in this section.

Basically, we see agents used in the design of Mercury with two peculiar roles. The first is to provide a set of uniform abstractions that can effectively model all aspects of the product with no loss of specific details. This is the case, e.g., of the unifying view that agents provide to different types of communication channels in the scope of the Contact Manager: all media are modeled in terms of agents and the diverse causes of faults and unexpected behavior, that are very specific of single types of channels, are all uniformly modeled.

[12]Mercury adheres to the common nomenclature that names "contact" the actual event of communication, e.g., a telephone call, and *not* the person involved in the communication.

Then, agents are also used in the design of Mercury to embed value-added functionality into the coherent framework of agents abstracting away from low-level details, that are left unspecified in order to support future enhancements. This is the case, e.g., of the specific contracting mechanism that ACD agents and Contact Manager agents use: the high-level design of the product leaves it unspecified in order to ensure a proper place for future enhancement, e.g., to support predictive dialing of outbound calls using Markov Decision Processes.

V. Conclusion

In this paper we reported on three Italian industrial use cases developed by resorting to MAS technology.

The lesson learnt in adopting MAS technology in the industry is that agent technology is applicable to a wide range of industry problems. Moreover, an important added value in using MAS-based solutions is that they highly reduce complexity. Let us also note that, in principle, designing with agents is easier for business analysts than using any other mainstream technologies and that information-technology practitioners can easily make the transition from, e.g., everyday object-orientation to agent-orientation [27].

Summarizing, from our experience we can state that MAS technology is definitely effective in the design and concrete realization of industrial-strength software applications.

Acknowledgment

The authors would like to thank all researchers and practitioners that were involved in realization of the three case studies for their precious contributions.

References

[1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 2003.

[2] M. J. Wooldridge, "The logical modelling of computational multi-agent systems," Ph.D. dissertation, 1992.

[3] P. Buckle, T. Moore, S. Robertshaw, A. Treadway, S. Tarkoma, and S. Poslad, "Scalability in multi-agent systems: The fipa-os perspective," in *Selected papers from the UKMAS Workshop on Foundations and Applications of Multi-Agent Systems*. London, UK: Springer-Verlag, 2002, pp. 110–130.

[4] P. Busetta, R. Ronnquist, A. Hodgson, and A. Lucas, "Jack intelligent agents - components for intelligent agents in java," *AgentLink News*, vol. 2, 1999.

[5] H. S. Nwana, D. T. Ndumu, L. C. Lee, and J. C. Collis, "Zeus: A toolkit for building distributed multiagent systems," *Applied Artificial Intelligence: An International Journal*, vol. 13, pp. 129–185, 1999.

[6] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "JADE - a java agent development framework," in *Multi-Agent Programming*, 2005, pp. 125–147.

[7] J. McKean, H. Shorter, M. Luck, P. McBurney, and S. Willmott, "Technology diffusion: analysing the diffusion of agent technologies," *Journal Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, pp. 372–396, 2008.

[8] R. A. Belecheanu, S. Munroe, M. Luck, T. Payne, T. Miller, P. McBurney, and M. Pěchouček, "Commercial applications of agents: lessons, experiences and challenges," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2006, pp. 1549–1555.

[9] M. Sierhuis and W. J. Clancey, "Nasas oca mirroring system an application of multiagent systems in mission control," in *AAMAS '09: Proceedings of the 8th international joint conference on Autonomous agents and multiagent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 85–92.

[10] J. Tsai, S. Rathi, C. Kiekintveld, O. Ordez, and M. Tambe, "Iris-atoolforstrategic security allocation in transportation networks," in *AAMAS '09: Proceedings of the 8th international joint conference on Autonomous agents and multiagent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 85–92.

[11] J. Himoff, P. Skobelev, and M. Wooldridge, "Magenta technology: multi-agent systems for industrial logistics," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 60–66.

[12] G. Caire, D. Gotta, and M. Banzi, "Wade: a software platform to develop mission critical applications exploiting agents and workflows," in *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 29–36.

[13] K. Dorer and M. Calisti, "An adaptive solution to dynamic transport optimization," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 45–51.

[14] S. Jacobi, D. Raber, and K. Fischer, "Masdispoxt: heat and sequence optimisation based on simulated trading inside the supply chain of steel production," in *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 23–26.

[15] M. Pěchouček, M. Rehak, P. Charvat, T. Vlcek, and M. Kolar, "Multi-agent planning in mass-oriented production," *IEEE Transactions System, Man and Cybernetics*, vol. Part C, 37, no. 3, pp. 386–395, 2007.

[16] V. Darley, P. V. Tessin, and D. Sanders, "An agent-based model of a corrugated box factory: The tradeoff between finished-goods- stock and on-time-in-full delivery," in *Proceedings of the Fifth Workshop on Agent-Based Simulation*. H. Coelho and B. Espinasse, 2004.

[17] M. Pěchouček and V. Marik, "Industrial deployment of multi-agent technologies: review and selected case studies," *Journal Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 3, pp. 397–431, 2008.

[18] "Foundation for intelligent physical agents. specifications," 1997.

[19] D. Weyns, A. Helleboogh, and T. Holvoet, "How to get multi-agent systems accepted in industry?" *Journal Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 4, pp. 383–390, 2009.

[20] F. Zambonelli and A. Omicini, "Challenges and research directions in agent-oriented software engineering," *Journal of Autonomous Agents and Multiagent Systems*, vol. 9, pp. 253–283, 2004.

[21] F. Bergenti, G. Rimassa, A. Poggi, and P. Turci, "Middleware and programming support for agent systems," in *Proceedings of the 2nd International Symposium from Agent Theory to Agent Implementation*, 2002, pp. 617–622.

[22] A. Addis, G. Cherhi, A. Manconi, and E. Vargiu, "A multiagent system for personalized press reviews," in *Distributed Agent-Based Retrieval Tools*, A. Soro, G. Armano, and G. Paddeu, Eds. Polimetrica, 2006, pp. 67–86.

[23] A. Addis, G. Armano, and E. Vargiu, "Using progressive filtering to deal with information overload," in *7th International Workshop on Text-based Information Retrieval*, 2010.

[24] A. Addis, G. Armano, and E. Vargiu, "From a generic multiagent architecture to multiagent information retrieval systems," in *AT2AI-6, Sixth International Workshop, From Agent Theory to Agent Implementation*, 2008, pp. 3–9.

[25] Y. Fu, W. Ke, and J. Mostafa, "Automated text classification using a multi-agent framework," in *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*. New York, NY, USA: ACM Press, 2005, pp. 157–158.

[26] G. Armano and E. Vargiu, *Post-proceedings of PROMAS 2009*. Springer Verlag, 2010, ch. A MultiAgent System for Monitoring Boats in Marine Reserves, p. in press.

[27] S. S. Benfield, J. Hendrickson, and D. Galanti, "Making a strong business case for multiagent technology," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2006, pp. 10–15.