Commercial Applications of Agents: Lessons, Experiences and Challenges

Roxana A. Belecheanu, Steve Munroe, Michael Luck, Terry Payne
School of Electronics and Computer Science
University of Southampton
Southampton, United Kingdom
{rab2, sjm, mml, trp}@ecs.soton.ac.uk

Tim Miller, Peter McBurney
Department of Computer Science
University of Liverpool
Liverpool, United Kingdom
{t.miller, p.j.mcburney}@csc.liv.ac.uk

Michal Pěchouček
Department of Cybernetics
Czech Technical University in Prague
166 27 Prague 6, Czech Republic
pechouc@labe.felk.cvut.cz

1. INTRODUCTION

As has been argued very eloquently and effectively, there is a *chasm* that needs to be crossed in the adoption of any new technology [8], and the marketing of such technologies must somehow try to bridge the gap that arises. Many currently see agent technologies in the middle of that chasm period of adoption, where addressing the visionaries as well as the pragmatists (or early adopters) who avoid risks but readily see the advantages of tested technologies [11], is necessary. As part of the needed marketing exercise, we believe that a catalogue of case studies of deployed applications with real quantified business benefit, can help convince those wavering pragmatists. Perhaps more importantly, the lessons that can be drawn from such case studies may also be used to guide the efforts of new commercial agent technology providers, both in developing the technology itself, and in understanding the concerns and constraints of early adopters.

As part of its industrial activity programme, AgentLink III[1] has been developing such a catalogue of case studies for exactly this purpose, working in collaboration with a range of companies to provide a selection of example applications, and draw out important issues arising from their development. In this paper, we summarise some of these case studies, and review the lessons learned, with the aim of identifying key problems. Due to space constraints, only very brief descriptions are provided in each case, but they serve to illustrate the nature of the application and the range of sectors and solutions; the full set of case studies is available from www.agentlink.org.

[1] http://www.agentlink.org

2. COMMERCIAL AGENT APPLICATIONS

To date, deployed applications of agent technologies have been concentrated in a small number of industrial sectors, and for particular, focused, applications. These have included: automated trading in online marketplaces, such as for financial products and commodities; simulation and training applications in defence domains; network management in utilities networks; user interface and local interaction management in telecommunication networks; schedule planning and optimisation in logistics and supply-chain management; control system management in industrial plants, such as steel works; and, simulation modelling to guide decision-makers in public policy domains, such as transport and medicine [5].

While some of these applications may be implemented as closed systems inside a single company or organisation (for example, agent-based simulation for delivery schedule decision-making), the domains to which agent technologies are most suited are those involving interaction between entities from more than one organisation. For example, automated purchase decisions along a supply-chain, require the participation of the companies active along that chain, so that implementing a successful agent-based application requires agreement and coordination from multiple companies. Moreover, to fully exploit the benefits of agent technology, other suitable domains are those involving a multitude of variables, with complex interdependencies, which change dynamically and thus cannot all be known at design time. This requires agent solutions because of their ability to dynamically adapt to changes in the environment and thus to offer real-time optimisation. For example, agent-based solutions successfully replaced conventional systems for network planning and transportation optimisation, because these were limited in their ability to cope with the increasing complexity, and especially with the dynamics, of a noticeably globalised transportation business [3].

In this section we describe a suite of agent applications provided by agent technology vendors for several commercial clients. In particular we describe the work of the following companies.

- Agent Oriented Software and their system used to aid the Ministry of Defence in military training.

- Eurobios and the simulation model they developed to improve production schedules for a cardboard box manufacturer.

- Magenta and their approach to helping a shipping company improve their oil distribution shipping network.

- Whitestein's system for logistics management.

- Nutech and the solution they developed for an energy production and distribution company.

- Acklin's approach to solving information exchange problems for insurance companies.

- Rockwell Automation and their use of intelligent agents for chilled water systems for the US Navy.

2.1 Agent Oriented Software and Military Training

Simulation and modelling are extensively used in a wide range of military applications, from development, testing and acquisition of new systems and technologies to operation analysis and provision of training and mission rehearsal for combat situations. In particular, distributed simulation environments for use in training and analysis drive much of this work. For example, simulation exercises involve a few dozen real people with the remaining hundreds or even thousands of other battlefield entities being computer simulations. As the human element is a key success factor in combat situations, the value of computer models of combat are greatly affected by their ability to accurately represent the range and variability of expected human behaviour.

The Human Variability in Computer Generated Forces (HV-CGF) project, undertaken by Agent Oriented Software on behalf of the UK's Ministry of Defence (MoD), developed a framework for simulating behavioural changes of individuals and groups of military personnel when subjected to moderating influences such as caffeine and fatigue. The project built on the JACK Intelligent Agents toolkit, a commercial Java-based environment for developing and running multi-agent applications.

JACK was particularly appropriate for several reasons. It incorporates the Belief-Desire-Intention (BDI) reasoning model, proven to effectively model certain types of behaviour, like the application of standard operational procedures by trained staff. JACK also includes facilities for the creation of intelligent team behaviour through the advanced JACK Teams model, while the JACK Agent Language extends Java with constructs for agent characteristics, such as plans and events.

For the project, Agent Oriented Software built a layer on top of Jack, called the JACK Cognitive Architecture (CoJack), that provides cognitive modelling capabilities for enhancing agent functionality with psychological attributes. Using these attributes, cognitively-plausible agents can be developed to simulate human psychological parameters, such as the capacity of human memory to handle multiple concurrent tasks, the perception and speed of reaction to events from the battle-space, and the ability to choose from different problem-solving tactics. This layer can thus be used to develop various cognitive models.

The influence of internal and external moderators are modelled as an independent layer, through a set of *behaviour moderators*, one for each influencing factor that modifies the agent's psychological attributes. These were plugged directly into the JACK Cognitive Architecture.

Also part of the core architecture is the JACK Intelligent Agents Team-based Platform, which was implemented using JACK's team based agent capability, to model human variability at team level. It facilitates the design, configuration and execution of a collection of agents organised in military structures realistic to the UK army. Each team member is a rational agent able to execute actions such as doctrinal and non-doctrinal behaviour tactics, which are encoded as JACK agent graphical plans.

2.2 Eurobios and Manufacturing Management and Optimisation

SCA Packaging, a leading international manufacturer in the corrugated-box market, turned to Eurobios to provide an agent-based simulation model to explore different strategies for reducing stock levels without compromising delivery times, as well as evaluating the consequences of changes in its customer base. In order to operate successfully in a highly competitive market, the manufacturing company found itself under growing pressure to maintain a high quality of customer service and decrease delivery times, as well as increasing plant efficiency and reducing inventories of finished goods

Eurobios developed a solution for the management and optimisation of the company's manufacturing plant. In the plant, customer orders arrive simultaneously for a large variety of boxes, each with its own colour scheme and specific printing, and often to be delivered at very short notice. Plant managers have to deal with varying lead-times, varying seasonality and overlapping production pathways, while trying to avoid weekend shifts and overtime, keeping the warehouse stock levels down, and ensuring *on-time and in full delivery* [2]. In addition, the complexity of plant processes, combined with the difficulty of predicting customer behaviour and machine failure, makes managerial decisions difficult. Thus, the problems and the trade-offs arising in the management of the plant require a detailed understanding of the relationships between customer order patterns, factory capacity, machine speeds, order batching and warehouse size (to mention just a few of the relevant variables), which can only be achieved through the use of sophisticated simulation technology.

The simulation model developed by Eurobios combined agent technology with discrete event simulation [2]; agents represented interacting entities with behaviour rules and decision capabilities attached, while the simulation of the processes was achieved using events, occurring concurrently or consecutively.

2.3 Magenta and Vessel Scheduling

As in many other transportation domains, in the domain of shipping, a major problem is efficient cargo allocation and vessel scheduling. Commonly, such tasks are performed manually by human schedulers with no automated support. However, this causes two major problems, as follows.

1. The vast amount of information that must be considered to schedule large fleets of vessels in a dynamic

environment means that the complexity is high, potentially leading to costly mistakes. Reducing the frequency of such mistakes improves fleet utilisation, allowing increased profits while using the same resources.

2. There are many domain-specific constraints that impact on the scheduling process and, each time an employee involved in scheduling leaves the company, their expertise in relation to these domain-specific constraints leaves with them; reliance on human schedulers alone brings great dependencies causing the company to become inefficient, and requiring new employees to start building up expertise from scratch.

In the scheduling of ships, many factors must be taken into account. First, each vessel, cargo, and port has different constraints that may prevent certain schedules (for example, a vessel may be too big for a port, or a cargo too big for a vessel). Such constraints are *hard* in that they cannot be broken during the scheduling process. However, other *soft* constraints must also be considered. These are constraints that the scheduler can break only if there are no other reasonable solutions available such as, for example, reducing the amount of time that a vessel travels without cargo. In addition, the environment is dynamic, in that the cargo is changing constantly, and the vessels may fail, rendering them out of service for a period, and requiring a reschedule for their cargo over that time.

In addressing these concerns for a large shipping company, Magenta Technology developed the *Ocean i-Scheduler* vessel scheduling tool [4], which was built on top of Magenta's agent technology, using the multi-agent paradigm. The shipping company's original requirement for their application was simply an online system to allow direct booking of cargo to be transported. However, after discussions with Magenta and demonstrations of their agent technology, it was clear to the company that agents might enable the development of a much more useful system. A proof of concept application developed by Magenta convinced the customer to invest in the development of a full scale system. The customer was first impressed by the way that, using agent technology, the system's behaviour could be changed without having to shut it down or introduce new code. Second, they recognised how well the agent system itself could handle changes to cargo, vessels, and ports. Overall, the customer envisaged two benefits to implementing their system in this way.

1. The system would provide support to employees in the scheduling task, therefore reducing the possibility of costly mistakes.

2. The system would provide a formal model of the business knowledge needed to schedule shipping fleets, so that much less knowledge might be lost when scheduling employees leave the company.

In the development of the system, the client, together with Magenta Technology developers, used the agent paradigm also as a design tool. Each cargo, vessel, port, etc., was modelled as a real-world role. For example, the role of vessel was designed, taking into account the responsibilities and rights of such a vessel. Then, for each vessel in the client's fleet, a vessel agent was created. The developers claim that this led to a simplified design when compared to what might have been developed using traditional software engineering techniques.

2.4 Whitestein and Transportation Networks

In transportation, one of the keys factors affecting efficient operation is the optimisation of the use of available capacity. Strategic network planning and short-term route optimisation are used to allocate despatching plans but are normally performed manually, causing difficulties when plan deviations become necessary due to the occurrence of unexpected events.

The Living Systems Adaptive Transportation Networks (LS/ATN) application is a cost-based optimisation system for transport logistics [3]. Developed by Whitestein Technologies, LS/ATN is designed to provide automatic optimisation for large-scale transport companies, taking into account the many constraints on their vehicle fleets, cargo, and drivers. Although the agent solution accounts for only 20% of the entire system, agent technology plays a central role in the optimisation. Vehicle drivers send information specifying their location and proposed route, and the system determines if that vehicle can collect an additional load, or swap future loads with another vehicle in order to reduce cost. A negotiation is performed automatically by agents, with each agent representing one vehicle, using an auction-like protocol. The vehicle that can provide the cheapest delivery wins the auction, reducing the overall cost of cargo delivery and, in most cases, the combined distance travelled for all vehicles. The aim is to find a local optimum (that is, not European-wide), so that only vehicles travelling in close proximity to each other will be involved in negotiations.

2.5 NuTech Solutions and Plant Optimisation

Recent years have seen substantial change in the energy transformation market caused by deregulation, fluctuating energy prices, changes in customer demand and changes in production and delivery costs. In this context, a large energy supply company sought out NuTech Solutions to develop a system that would enable them to address a number of key concerns. Essentially, the company needed to optimise its operations so that it could schedule the production and distribution of liquid oxygen and liquid nitrogen from more than 40 plants to 10,000 customer sites.

The Supply Chain Production Optimiser system developed by NuTech combined the power of ant system optimisation and genetic algorithm search techniques, and is one of the largest and most successful applications of these techniques used in industry to date. The system takes into account the production schedules of all of the company's oil production plants and adapts them to projected energy prices, weather changes, client demand and desired inventory levels. Utilising data representing seven day power prices, customer demand projections, daily power costs and efficiency measures from every plant in the network, the system evaluates production costs based on forecast demand to determine which plants should produce which products. In addition, the system can model and take into consideration potential plant production capability, maintenance needs and power issues that may impact upon the generated schedules. All of this information can then be fed into the energy company's *operations control centre* to enable adjustments to be made to the operation of the plants to meet production requirements across the whole network.

2.6 Acklin and Information Management for Insurance

The process of handling transnational insurance claims is heavily bureaucratic, with most of the information processing done by humans, resulting in long times for settlement. European insurance companies use heterogeneous information systems, with different means of storing and using data, so that the data must be exchanged manually between companies when dealing with claims, which involve motor vehicle accidents in one country by drivers from another. Acklin, B.V. developed an agent-based system to automate the exchange of insurance information between human operators in the claim handling process [10] concerning trans-national vehicle accident claims. The main customer was a major Dutch insurance company, but other insurance companies involved in the claim handling process with this customer needed also to implement this system for interoperability reasons.

The KIR system developed by Acklin consists of a network of communicating agents to which the back-office of every insurance company can connect in order to exchange information. The communication between agents is done by email, as a means of message transport, because it was the only technology that was supported by all the insurance companies implementing this automation solution.

The agents were designed and implemented using the Five Capabilities model, also known as conceptual separation of concerns, which is composed of the following models:

The communication model handles all the interactions between agents and other systems, and it has methods for message transportation, representation and interpretation.

The competence model contains methods for the execution of the tasks for which the agent is designed.

The self model represents the knowledge that an agent has about its own capabilities (i.e. goals, tasks, jobs, states and competencies).

The planner model contains planning strategies that can be used to achieve the goals.

The environment model gives the agent a view on the world in which it operates (i.e. the other agents) and with which it interacts.

The advantage of this approach was that each model could be implemented independently, to reflect the type of agent; for example, for agent roles like payer and handler, the differences in capabilities were implemented only in the competence and self models.

The introduction of the KIR system resulted in a work pressure release of three people at the customer site and reduced the process of identification of client and claim information from 6 months to 2 minutes. The success of this system led the customer to request its implementation at European level.

2.7 Rockwell Automation and Chilled Water System

For many years now, Rockwell Automation has been investigating the use of multi-agent systems in manufacturing and various control and diagnostics applications [7]. Based on the Autonomous Cooperative System (ACS) proprietary agent platform [6], Rockwell Automation developed an intelligent application for distributed control of a shipboard Chilled Water System (CWS) for vessels of the US Navy [9].

In the CWS application, intelligent agents are deployed within commercial off-the-shelf (COTS) Programmable Logical Controllers (PLCs). Specifically, the set of functionalities of the standard Rockwell Automation ControlLogix and FlexLogix controllers was extended so that they can host a collection of various intelligent agents. The deployed agents carry out negotiations and collective decision-making aimed at dynamic reconfiguration of the Chilled Water System and fluid rerouting as a result of possible failures in the system.

The CWS application has been demonstrated on the Reduced Scale Advanced Demonstrator (RSAD) model, a Rockwell Automation developed reconfigurable fluid system test platform, which is a scaled-down version of a real ship. It features two chilling plants with 16 chilling targets on board the ship (for example, combat systems, communication systems, and radar and sonar equipment). Each water cooling plant is represented by an agent, as well as each chilling target(heat source), each valve, and some parts of the piping system.

3. LESSONS LEARNED

Having presented an overview of these agent-based applications, we now turn to discuss the main lessons learned from their development and deployment, and from the successful collaborations between the software companies and their industry clients. In this section, we present some of the challenges encountered and the ways in which they were overcome, in terms of selling the technology to the customer, knowledge acquisition and knowledge engineering, integration with legacy systems, validation, usability and operationalisation, and standards.

3.1 Selling Agent Technology

One of the important factors in securing a contract for an agent-based application was selling the agent technology to the right customer in the right industry sector. To achieve this, it was generally necessary to focus on industries who were forced to transform under competitive pressures and on companies who were looking to streamline some of their processes in order to reduce costs and become more efficient. Not only did agent technology need to address these demands, but it had also to deliver visible business or commercial value to the customer and sometimes even to their cooperating partners. In particular, success in securing such contracts often required helping companies to achieve a more profound transformation of how they operate and not just an incremental improvement.

Because of the disruptive nature of agent technology, a good relationship with customers was also important, as was having a history of successful implementations of agent-based systems, and experience of modelling of the particular problem domain. For example, the extensive experience of Eurobios with supply chain optimisation using agents was valuable in convincing managers at SCA Packaging of the feasibility of an agent-based solution. The alternative model was to find *pioneering* customers, or *early adopters* who understood the potential business value of the agent-based application and were willing to invest in it (such as the insurance company in the Acklin case study and the MoD), even if they did not have competence in the specifics of the technology itself.

The agent metaphor itself was also useful in attracting customer interest, as it was found to be an intuitive way of modelling some business domains (for example, in the insurance case and the packaging plant) and easier to understand than analytical or mathematical models typically used for optimisation. With concepts such as roles and responsibilities, agent-oriented approaches to problem and system description are much closer to the ways in which managers conceive of business domains than are traditional software engineering descriptions. By mapping organisational functions onto agents using roles and capabilities, and by mapping business logic and process flows onto message exchange patterns between agents, it was easy to reason about and better understand company processes, especially when visualisation was used in early model-building stages. This aspect particularly appealed to senior management and process engineers, and made it easier to convince them to invest in the agent-based solution, even when the customer had no prior exposure to agents. The same cannot be said about using agents for cognitive modelling for the first time in a defence application, where building cognitive science concepts on top of agent concepts, and then linking back and interpreting simulation results according to cognitive principles was problematic. However, by successfully completing the project and demonstrating the feasibility of agent technology in modelling moderated behaviour in distributed simulations, further interest was raised within the defence community in applying agents to other types of simulations, thus pushing agent technology towards exploitation and development of new areas.

3.2 Knowledge Acquisition and Knowledge Engineering

In modelling and simulation applications for planning, such as the corrugated-box factory modelling of Eurobios, the cargo fleet scheduling of Magenta, or the production planning and routing optimisation of NuTech, knowledge acquisition and model building are often performed simultaneously and iteratively. To build a model that matches reality as accurately as possible requires gathering information about different operations and processes from documents and company databases, as well as eliciting and formalising knowledge from domain experts. For large scale and complex problems, the model is built incrementally, in several steps, from a simple basic model through to a detailed and complex one that approaches reality. The model and underlying knowledge are then validated by experts, after which more knowledge acquisition is undertaken to refine or correct the rules where necessary.

These tasks pose particular challenges but also offer substantial benefits both to software developers and customers. For business performance optimisation systems, for example, data acquisition often requires significant effort in companies where data is spread between different enterprise applications. This effort is even higher when there are no existing integration applications (such as enterprise resource planning, or ERP) in a company, and where the enterprise applications do not automatically generate the information necessary for optimisation. Such is the case of transaction-based applications, (as experienced by Eurobios at SCA Packaging, for example), which are not designed for storing intermediate states of information between transactions or for storing the events that model the business processes; neither are they able to store other types of information like the time of generating the data, or what-if analysis information, typically used in the modelling and optimisation task.

In terms of eliciting domain knowledge, specific challenges were encountered.

- First, the process of making explicit that knowledge which is complex, unformalised and developed through years of experience by people working in the field was one of the most challenging and time-consuming aspects of the acquisition task in some of the cases we considered.
- Second, clarifying domain terminology to the software developers was often necessary and was clearly very important, especially in the early stages.
- Third, the effort to obtain data of sufficient quality and detail was sometimes underestimated, and several iterations were necessary because the software engineers and business managers often had a different appreciation of what it is that constitutes good data.
- Additional problems came from the fact that domain experts often had little understanding of, or interest in, agent technology, and had limited time available for discussion (as in the case of the defence application).

Sometimes minor alterations to the documented processes were introduced on an *ad hoc* basis at an operational level without informing management. For example, changes to delivery dates for a particular customer were introduced by plant operators at SCA Packaging, based on their everyday understanding of customer order fluctuations. Consequently, there were often discrepancies between the results of simulations and real data from the domain (such as the warehouse stock levels in the case of the cardboard-box factory), which had to be corrected.

However, discovering such discrepancies was one of the benefits of the knowledge acquisition and engineering activities. Indeed, the impact was broader than just the software model, and reached beyond the realm of software development. Not only did different departments gain an insight into each other's work practices, but by encouraging them to talk to each other in the process of eliciting knowledge, some of the communication barriers between functional departments and between different management roles in the company were eliminated.

3.3 Integration with Legacy Systems

A key concern of many companies when deploying and integrating new IT systems is data security and the stability of the systems already in place (or legacy systems). For these reasons and because of the current early stage of agent technology adoption, agent-based applications for simulation and optimisation were used as tools to assist human decision-making and not operationally to control or automate processes. The only integration with legacy systems was therefore at a database level and not with any planning or scheduling applications in use at client companies. The only example of process automation with agents was the exchange of information in insurance claim handling, but no actual integration with insurance systems was present here either. A contrasting case is the Co-JACK system developed by AOS, where the agent-based system was integrated with other simulation environments in order to take advantage of well developed user interfaces and high-fidelity graphical modelling of various geographical environments. Two main integration approaches were identified, as follows.

- The vertical approach, used by AOS, by which the Co-JACK agent system was built using a layered architecture, and the bottom layer of the architecture (the CGF Interconnection Layer) was used for integration with existing defence CGF systems (such as the OneSAF Test Bed).
- The horizontal integration approach, by which every agent interfaces with company databases, was used by Acklin. In order to minimise the interference between agents and the back-office of the insurance company, Acklin developed an interfacing component that maps execution commands from an agent to the back-office, and returned results from the back-office to the agent. Transferring data between agents and the interfacing component was done through a buffer where instructions and reports are read and written by agents. This solution enabled agents to have controlled, but not direct, access to the required functionality and data from the legacy system. The interfacing component was developed by the IT department of the insurance company, thus sharing the risk with the client and allowing the client to retain control over company systems and data.

3.4 Validation

Validation is crucial to ensure the accuracy, correctness and realism of any model and of its outputs. Due to the complexity of the processes being modelled, extensive tests are usually necessary for diagnosing problems and tracking system behaviour, not only for validation purposes but also to help the customer understand how the system works. However, the task is challenging, not least because agent-based systems may explore realms of behaviour which go beyond people's intuition and expectations and therefore results often seem surprising [1]. Testing and validation are even more problematic when the system must access company databases for data input in real-time and when this interaction occurs over multiple steps (for example, in the Eurobios case), because simulations become irreproducible, making diagnosis and debugging more difficult.

In addition, questions of how much time and effort to spend in system design and model building, as opposed to implementation and validation are common difficulties, and were experienced by Eurobios. On the one hand, spending considerable time in advance to ensure sufficient data accuracy, and designing the application to ensure reusability, quality and robustness was important. However, it was also found that only through simulations during the model calibration and validation stages did certain issues surface that would not have been discovered at design time. For reasons of complexity, there was a point after which spending more time in design brought no further value to the final system because there was a chance that what was being designed would later be invalidated.

In contrast, in the case of the Co-JACK system, experiences with model validation and overall system testing were different and more challenging, firstly because they had to address the problem of integration with the CGF system, and secondly because the realism of human behaviour modelling had to be proven without being able to test the system in an actual combat situation. Testing was therefore achieved by showing how changes in psychological variables within Co-JACK determine changes in the behaviour of the entities in the OneSAF system, thus testing the link with the CGF system. The agent-based model was validated by comparing the results of the simulation with those produced by other cognitive architectures that had previously been validated through experiments with humans.

3.5 Usability and operationalisation

Training and familiarisation play a vital role in the adoption of any new technology; this is true for agent technology, too. When an agent-based system is used to assist decision-makers (as with human schedulers in the Whitestein and Magenta applications), trusting the results of an optimisation or a simulation is essential, especially since decision-makers have a great deal of experience in their area. A period of familiarisation and day-to-day interaction with the system is therefore necessary for users to accept and have confidence in the recommendations of the software, and to rely on it when making critical decisions.

At this user-oriented level, usability was also an important aspect of operationalisation. To encourage day-to-day use in a seamless manner, customers may require that user interfaces are designed in a similar way to, or share some features with, standard software packages used in that particular industry. For example, the experience of Whitestein was that transportation management systems have specific interface designs and functionality, which the agent-based system for transport route optimisation had to provide.

In the Co-JACK application, demonstrations with non-expert users initially showed a fair amount of difficulty in the use of the system both in terms of writing accurate tactics for the model and tracing and understanding overall system behaviour. This was due to the complexity of inferences at the agent level, and of the interactions between agents. Thus, dealing with the combinatorial explosion of different outcomes in the battle-space, and how this explosion could most usefully be presented for decision support or training, were two of the objectives identified for further development during a subsequent project phase.

3.6 Agent Standards

While agent standards, such as FIPA[2], are developing and becoming more mature, it is interesting to note that in many of the projects outlined in this article, the developers did not use current agent standards for such elements as agent communication, and have developed their own languages and protocols. These design decisions reflect the closed nature of many of the systems. Where systems are open, in that they involve different organisations, they are not open in the sense that agents developed outside of these organisations can participate in the system. Even so, with these organisations developing their own interaction languages and protocols, this is a sign that the state-of-the-art in this area does not appeal to industry partners as much as many would hope, and that perhaps work is required in this field to make it more relevant and applicable to industry.

[2]http://www.fipa.org

4. CONCLUSIONS

This paper has reviewed and discussed some of the most important lessons and experiences of developing agent-based applications for niche markets, in the logistics, production, defence and insurance sectors. The experiences and challenges faced are largely determined by the nature of the business operations in the industry domain, the characteristics of the problem to be solved and the issues related to the organisational and technical infrastructure of the client company. Nevertheless, commonalities were observed across these applications, which led us to believe that sharing these lessons may also be useful to other companies considering agent-based implementations in the future. The common aspects can be summarised as follows.

- From a business perspective, technology adoption decisions should be strategic, based on long term benefits and supported by a clear business evaluation. A software developer wishing to sell agent technology to a new company must put forward a strong business case, identifying the commercial and technical value of an agent solution for that company, and the advantages of agent technology over other technologies already in use in the relevant industry, but without underestimating them.

- From a technical perspective, while the specific agent solutions considered here address specific performance requirements like dynamic adaptivity and real-time response, they also had to satisfy requirements common to any software implementation for industrial applications, such as accuracy of results, interoperability, robustness and user-friendliness. It is thus important that development and deployment is approached with traditional software engineering methods and standards, and traditional project and risk management techniques used for IT implementations.

More generally, it can be concluded that successful adoption of agent technology depends on identifying the specific key conceptual and technical aspects related to agents that can address real world problems, even if these problems require unsophisticated solutions, and introducing innovation only where necessary.

Acknowledgements

This paper is a personal view based on interviews conducted with relevant experts from each of the companies mentioned: Tony Delaney and Martyn Fletcher from Agent Oriented Software; Vince Darley from Eurobios; Nadezhda Yakounina of Magenta Technology; Christian Danneger and Klaus Dorer from Whitestein Technologies; Bryan Woody and David Davis from NuTech; Chris van Aart from Acklin and Y'All. The interviews were undertaken as part of the case study effort of AgentLink III, the European Commission's Coordination Action on Agent-Based Computing. We are grateful to all those who contributed. The full set of case studies is available from www.agentlink.org where more complete presentations are also given.

5. REFERENCES

[1] M. Buchanan. Special report: Supermodels to the rescue. *strategy+business*, Spring, 2005.

[2] V. Darley and D. Sanders. An agent-based model of a corrugated-box factory: the trade-off between finished goods stock and on-time-in-full delivery. In H. Coelho and B. Espinasse, editors, *Proceedings of the Fifth Workshop on Agent-Based Simulation*, 2004.

[3] K. Dorer and M. Calisti. An adaptive solution to dynamic transport optimiation. In M. Pechoucek, D. Steiner, and S. Thompson, editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track*, pages 45–51. ACM Press, 2005.

[4] J. Himoff, P. Skobelev, and M. Wooldridge. MAGENTA technology: Multi-agent systems for industrial logistics. In M. Pechoucek, D. Steiner, and S. Thompson, editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track*, pages 60–66. ACM Press, 2005.

[5] M. Luck, P. McBurney, O. Shehory, and S. Willmott, editors. *Agent Technology: Computing as Interaction. A Roadmap for Agent Based Computing*. University of Southampton on behalf of AgentLink III, 2005.

[6] F. Maturana, P. Šlechta, P. Vrba, P. Tichý, R. Staron, D. Carnahan, F. Discenzo, V. Mařk, and K. Hall. A specification to build distributed reconfigurable manufacturing systems using intelligent agents. In *The 3rd International Conference on Reconfigurable Manufacturing (CIRP 2005)*, Ann Arbor, MI, 2005.

[7] F. P. Maturana, R. J. Staron, and K. H. Hall. Methodologies and tools for intelligent agents in distributed control. *IEEE Intelligent Systems*, 20(1):42–49, 2005.

[8] G. Moore. *Crossing the Chasm*. HarperCollins, 1991.

[9] P. Tichý, P. Šlechta, F. P. Maturana, R. J. Staron, K. Hall, V. Mařk, and F. M. Discenzo. Multi-agent technology for robust control of shipboard chilled water system. In *Proceedings of International Federation of Automatic Control (IFAC) Conference on Control Applications in Marine Systems (CAMS 2004)*, Ancona, Italy, 2004.

[10] C. J. van Aart, K. van Marcke, R. F. Pels, and J. L. F. C. Smulders. International insurance traffic with software agents. In F. van Harmelen, editor, *Proceedings of the Fifteenth European Conference on Artificial Intelligence*. IOS Press, 2002.

[11] T. Wagner, L. Gasser, and M. Luck. Impact for agents. In M. Pechoucek, D. Steiner, and S. Thompson, editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track*, pages 93–99. ACM Press, 2005.