The Knowledge Engineering Review, Vol. 21:4, 345–392. © 2006, Cambridge University Press
doi:10.1017/S0269888906001020 Printed in the United Kingdom

Crossing the agent technology chasm: Lessons, experiences and challenges in commercial applications of agents

STEVE MUNROE[1], TIM MILLER[2], ROXANA A. BELECHEANU[3], MICHAL PĚCHOUČEK[4], PETER MCBURNEY[2] and MICHAEL LUCK[1]

[1]*School of Electronics and Computer Science, University of Southampton, UK;*
e-mail: rab2@ecs.soton.ac.uk, sjm@ecs.soton.ac.uk, mml@ecs.soton.ac.uk
[2]*Department of Computer Science, University of Liverpool, UK;*
e-mail: tim@csc.liv.ac.uk, p.j.mcburney@csc.liv.ac.uk
[3]*Knowledge Media Institute, The Open University, UK;*
e-mail: r.a.belecheanu@open.ac.uk
[4]*Department of Cybernetics, Czech Technical University in Prague, Czech Republic*
e-mail: pechouc@labe.felk.cvut.cz

Abstract

Agent software technologies are currently still in an early stage of market development, where, arguably, the majority of users adopting the technology are visionaries who have recognized the long-term potential of agent systems. Some current adopters also see short-term net commercial benefits from the technology, and more potential users will need to perceive such benefits if agent technologies are to become widely used. One way to assist potential adopters to assess the costs and benefits of agent technologies is through the sharing of actual deployment histories of these technologies. Working in collaboration with several companies and organizations in Europe and North America, we have studied deployed applications of agent technologies, and we present these case studies in detail in this paper. We also review the lessons learnt, and the key issues arising from the deployments, to guide decision-making in research, in development and in implementation of agent software technologies.

1 Introduction

Since Levitt (1965), marketing theorists and marketers have modelled the lifecycles of new products on the basis of the different types of customer they attract in different phases following launch. The adopters of new high-technology products in the early phases of product launch, for instance, typically show greater willingness to experiment (or, equivalently, greater proneness to risk) than do later customers of the same product. These differences in adopter characteristics were studied by Moore (1991) who argued that the differences between the very early adopters and the majority of adopters of new high-technology products are usually so great as to create a 'chasm' which technology providers need to bridge in their marketing strategies. To do this, providers of technologies need to reduce the costs and risks associated with adopting the technology; for example, by creating user-friendly interfaces, by developing tools and middleware to facilitate adoption, and so on.

Many observers see agent technologies as currently being in this chasm phase of development: adopted eagerly by enthusiasts of new technologies who share the vision, but not yet taken up by the majority of potential users who tend to be more pragmatic about new technologies (Luck *et al.*,

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

2005; Wagner *et al.*, 2005). The pragmatists wish to see the business costs and benefits associated with a new technology before they embrace it, and so a key task in 'crossing the chasm' is to provide an analysis of these likely costs and benefits. One way to provide this is through a catalogue of detailed case histories of agent technology deployment, i.e. detailed descriptions of typical applications including an assessment of the technological and business issues involved in the development and operation of agent technologies. In addition to providing guidance to potential adopters of agent technologies, these case studies may also be used to understand the concerns and constraints of adopters. Thus, such case studies may also guide commercial development efforts by technology providers, agent standards efforts by suppliers and users, and research efforts by agent computing researchers.

To date, agent technologies have been deployed in only a small number of industrial sectors, and only for particular, focused, applications. These applications have included: automated trading in online marketplaces, such as for financial products and commodities (Nicolaisen *et al.*, 2001); simulation and training applications in defence domains (Hill, 1997; Baxter & Hepplewhite, 1999); network management in utilities networks (Du *et al.*, 2003); user interface and local interaction management in telecommunication networks; schedule planning and optimization in logistics and supply-chain management (Dorer & Calisti, 2005; Himoff *et al.*, 2005); control system management in industrial plants, such as steel works (Jacobi *et al.*, 2005); and simulation modelling to guide decision-makers in public policy domains, such as transport and medicine (e.g. Bazzan, 2005).

While some of these applications may be implemented as closed systems inside a single company or organization (e.g., 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 organization. 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. These domains require agent solutions because of their ability to dynamically adapt to changes in the environment and thus to offer real-time optimization. For example, agent-based solutions have successfully replaced some conventional systems for network planning and transportation optimization, conventional systems which were limited in their ability to cope with the increasing complexity, and especially with the dynamics, of a markedly globalized transportation business (Dorer & Calisti, 2005).

As part of its industrial activity programme, AgentLink III[1] developed such a catalogue of case studies of agent applications, for the purpose of reducing the chasm that agent technology must cross. Working in collaboration with a range of companies to provide a selection of example applications, this paper draws out important issues arising from their development. In what follows, we present a summary of each of these case studies, and review the lessons learned, with the aim of identifying key problems and benefits. These summaries serve to illustrate the nature of the application and the range of sectors and solutions; the full set of case studies is available from the AgentLink III website.

The outline of this paper is as follows. Sections 2–10 present the details of the applications, discussing the companies that have helped in the development of the applications, the requirements of the applications and the use of agents in the application, as well as highlighting some of the challenges encountered and benefits provided by using agent technology in developing these applications. Specifically, these sections present and discuss the following.

- Section 2: Agent Oriented Software and their system used to aid the Ministry of Defence in military training.

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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

- Section 3: Eurobios and the simulation model they developed to improve production schedules for a cardboard box manufacturer.
- Section 4: Magenta and their approach to helping a shipping company improve their oil distribution shipping network.
- Section 5: Whitestein's system for logistics management.
- Section 6: Nutech and the solution they developed for an energy production and distribution company.
- Section 7: Acklin's approach to solving information exchange problems for insurance companies.
- Section 8: Rockwell Automation and their use of intelligent agents for chilled water systems for the US Navy.
- Section 9: The Combined Systems consortium and their application of agent technology to the field of crisis management.
- Section 10: Almende and their intelligent communications system, implemented using agent technology.

Section 11 presents a discussion of some of the lessons and experiences, drawn from these case studies, which were found to be of most importance and interest. Section 12 concludes the paper.

2 Agent Oriented Software and military training

This section describes the application of agent technology to modelling human variability in military environments. It outlines a system that was commissioned by the UK Ministry of Defence (MoD) and implemented by Agent Oriented Software (AOS) Ltd. The objective was to offer a simulation environment for studying how people alter their decision-making and interactions with other military personnel when influenced by various moderating factors, such as heat, tiredness, consumption of stimulants such as caffeine, as well as battlefield experience and cultural factors. Using agents to model human behavioural change in the military context was a novel approach, and the final system achieved a higher level of realism than previous synthetic environments in military training.

2.1 AOS

The AOS Group is a software company specializing in the development of intelligent systems, particularly for real-time distributed control. Founded in 1997 in Melbourne, Australia, AOS has successfully built up a multi-million dollar annual revenue stream through partnerships with a variety of defence-related organizations, using agents to simulate humans and human behaviour in a range of adverse conditions. Although its most successful applications have been in aerospace and defence, AOS has a broad portfolio of projects covering industries such as manufacturing, telecommunications, retailing, government, finance, application areas including distributed systems, scheduling, real-time systems (e.g. air traffic management and robotic manufacturing), call centre customer management, order management and provisioning, and business process management.

2.2 Human cognition modelling as application domain

Simulation and modelling are extensively used in a wide range of military applications, from the 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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

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.

2.3 Limitations of existing computer generated forces systems

In the last decade, significant advances have been made by the computer generated forces (CGF) and semi-automated forces (SAF) communities to make synthetic military environments more realistic. These have been mostly in terms of three-dimensional graphical interfaces for training, and high-fidelity sophisticated modelling of physical aspects of vehicle movement and damage caused by weaponry in different geographical environments.

However, human reaction, adaptability and decision-making in these environments are still little understood, and their modelling is still fairly simplistic. In particular, conventional software approaches make it difficult to model with sufficient fidelity the effect of moderating influences on human behaviour (e.g. external, environmental conditions such as heat, humidity or cold, or internal conditions such as caffeine intake, jet lag, etc.). Current CGF systems, for example Brawler (Shakir, 2002), CAEN (Rönnquist *et al*., 2000) and the SAF product family[2], do not adequately model such complex human behaviour because they derive knowledge from prescribed tactics, showing what people are meant to use in the field, and not what they actually use. These systems lack significant situation awareness capabilities, realistic sensitivity to moderating influences, or realistic mechanisms for learning from experience (adaptability). They use pre-scripted behaviours for controlling the entities within the simulation so they are designed to behave predictably, usually according to doctrine, which makes them unable to respond to unexpected events.

Current CGF systems also lack the ability to generate useful self-explanation (hence it is hard to trace system behaviour to see if it accurately illustrates human behaviour). More realistic information therefore needs to be incorporated into these systems to reflect more complex situations and human behaviour. Yet, because of this largely rule-based approach, these systems are difficult to change and integrate, as they require expert knowledge in a specialized area.

As a result, these limitations continue to restrict the wider use of CGF systems as a means of complementing conventional operational analysis and training techniques used by the armed forces of the UK, such as increasingly expensive military exercises.

2.4 The human variability in CGF project

A more suitable approach to modelling the cognition of military personnel is the agent metaphor. Agents are good for both interacting with humans and modelling human behaviour because of their rational and autonomous properties, as well as offering reliability in complex, distributed and real-time systems. While there has been some research, and in addition some implemented systems, that use agents to model military personnel performance in terms of situation awareness, choice of tactics, etc., little investigation has been undertaken to understand how cognitive processes change as a result of moderating influences. The human variability in CGF (HV-CGF) project focused precisely on this aspect, aiming to develop a framework for simulating behavioural changes of individuals and groups of British military personnel, when subjected to moderating influences.

2.4.1 JACK Intelligent Agents

Unlike other agent-based systems, the HV-CGF application was not built as a standalone system using a particular agent oriented methodology or framework, but rather as a set of components in order to facilitate integration with companies' legacy systems. The project built on the JACK Intelligent Agents™ toolkit developed by AOS, a commercial Java-based environment for developing and running multi-agent applications.

[2] OneSAF, see `http://www.onesaf.org/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

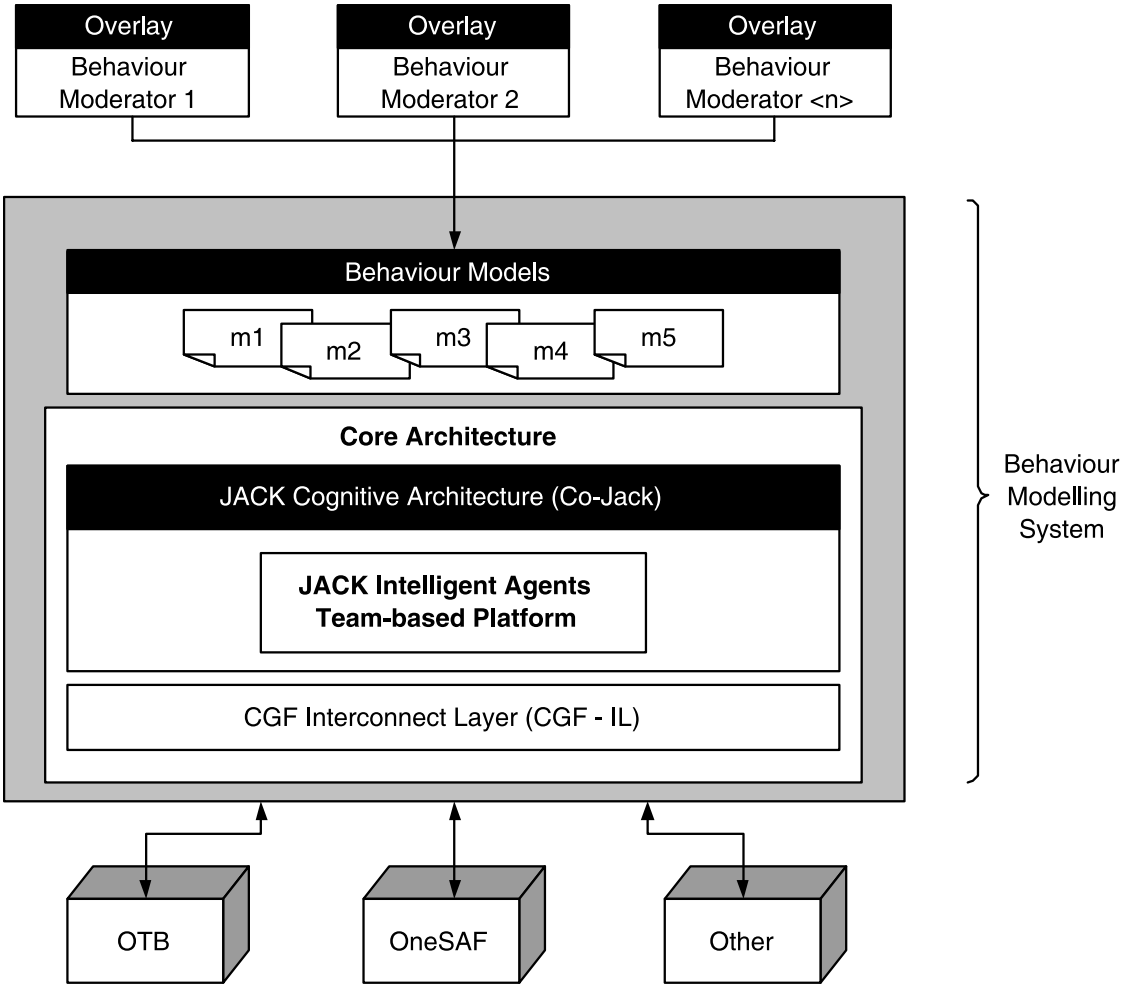


Figure 1 Layers of the HV-CGF system model (Source: AOS)

JACK was particularly appropriate for several reasons. It incorporates the belief–desire–intention (BDI) reasoning model, proven to effectively model certain types of behaviour, such as 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.

Despite these advantages, JACK and its applications had not previously addressed the problem of modelling moderating influences such as emotion and fatigue.

2.4.2 Solution framework

The main objectives in HV-CGF were first to extend the BDI paradigm by developing a cognitive architecture that includes psychologically based constraints and the influences of moderators, and second to develop a software system for validating this architecture and that could be integrated with CGF systems. To achieve these objectives, the HV-CGF team developed a multi-layer solution framework, the architecture of which is depicted in Figure 1.

At its core the framework has the JACK Cognitive Architecture (Co-JACK), a cognitive modelling layer on top of JACK, 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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

from different problem solving tactics. This layer can be used to develop various cognitive models.

The influence of internal and external moderators is then 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 implemented as behaviour moderator overlays that can be plugged into Co-JACK.

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 organized 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.

The integration of Co-JACK with different current CGF systems (e.g. OneSAF Test Bed (OTB)) is achieved via a CGF interconnection layer (CGF-IL), which is used to visualize project outputs using CGF entities (as each CGF entity is controlled by a JACK agent).

The multi-layer model has the advantage of allowing the independent development of each layer, with well-defined interfaces between layers, thus ensuring high cohesion and low coupling between the layer of behaviour moderators and the agents and CGF layers. In this model, each entity in a battle scenario (e.g. tank, soldier, tank commander, etc.) is a JACK agent, and agents are grouped in teams with associated team roles (e.g. commander, observer, etc.). Entities respond to events (e.g. attack the enemy) by selecting a particular tactic that depends on its role, on the state of its beliefs and on the rules of the mission. Modelling the moderating influences is done by changing the belief state. For example, an agent is ready to change its beliefs based on the tiredness of a soldier. The result of such a change is that the currently executing plan terminates and another plan's execution is started.

In addition to adding cognitive properties to agents, the JACK team infrastructure was also extended to allow redistribution of team roles at runtime. Other changes to the BDI model were also necessary because of human cognitive limitations in comparison with the execution performance of the agent system, in order to improve the realism of agent behaviour. For example, there are limitations to human cognition in terms of the number of options considered in decision-making; there are time delays and limits to the number of issues or beliefs that can be dealt with at the same time. By contrast, BDI agents can deal with a virtually unlimited number of plans concurrently, hold a virtually unlimited number of beliefs, etc. The model was not modified by limiting the number of plans or beliefs of the agent but by associating a larger time delay when deciding on the tactic used, when choosing between options, etc. It was validated through feedback from cognitive science and military experts and by running the model with data (e.g. plans) that had been used in experiments with other cognitive architectures (e.g. Soar) and then comparing the results.

2.5 Demonstration scenario

The system was demonstrated through a scenario chosen by the MoD Directorate of Analysis, Experimentation and Simulation (DAES) to have both high military relevance and operational feasibility. This scenario was a mission with six attack helicopters attacking a ground target. The objectives of the demonstration were to illustrate the concept of cognition and behaviour moderation in the Co-JACK environment both at individual and team levels. In particular, the demonstration showed how caffeine and sleep deprivation affect perception, situation awareness and decision-making, and how the selection of doctrinal and non-doctrinal behaviour is affected as a result. For example, when a soldier is tired, creating situation awareness may take longer than otherwise, and perception of the route for attack may be different. This was tested by showing how changes in psychological variables within Co-JACK determine changes in the behaviour of the

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

entities in the OneSAF system, thus demonstrating the integration of Co-JACK with the CGF system.

However, the demonstration also revealed two areas where the system required improvement, and which were therefore taken into a subsequent project phase for further development. The first was modelling the variability across agents of perception of the world as a result of moderation of behaviour. The second area concerned ease of use by non-experts, especially because demonstrating value to stakeholders was important. There was a fair amount of difficulty in using the system both in terms of writing accurate tactics for the model, and tracing and understanding overall system behaviour because of 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 can most usefully be presented for decision support or training were two of the objectives for further development.

2.6 Business model and project management framework

The project started in February 2003 as a partnership between AOS Limited (Cambridge, UK and Melbourne, Australia), QinetiQ (Farnborough and Malvern, UK), Pennsylvania State University (PA, USA), and the University of Melbourne (Australia). It was commissioned by the UK MoD under the sponsorship of the DAES.

2.6.1 Contract

The contract was won after a hotly contested invitation to tender, with a proposal that was supported both by a strong technical argument and a good business case. The technical competency of the proposal was evaluated by the Defence Science and Technology Organisation (DSTO), the commercial value was judged by a commercial arm of the MoD and final approval was given by the Director of the DAES.

On the technical side, AOS's expertise in agent technology and a good record of agent-based implementations for defence applications, combined with a strong relationship with the DSTO, were also positive factors in winning the contract. A strategic agreement established in 2001 between AOS and the DSTO, which covered the use by the DSTO of JACK Intelligent Agents™, thus bringing to AOS the benefit of collaborative evaluation and integration of intelligent agents into simulation systems used by DSTO, also contributed to this. However, the business value of the project was equally important, and the technical proposal had to be accompanied by a business case. This contained, for example, an outline of high-level objectives and the time plan of work breakdown for each of the two years, a list of necessary resources, (both in terms of staff and other financial resources such as the number of JACK licences necessary for contributing partners), as well as a statement of stakeholder relevance.

2.6.2 Team

The team included a Project Director, who also acted as the main customer liaison, a Project Leader responsible for the day-to-day management of the project, a Technical Architect, several Java developers and a Technical Author. Expert knowledge of human cognition and human behaviour modelling was provided by Professor Frank Ritter (Pennsylvania State University) and Emma Norling (Manchester Metropolitan University), while QinetiQ as subcontractor was responsible for integration with the CGF system, and for the acquisition, modelling and encoding of expert knowledge, and building of scenarios. As there had been few prior experiments and results on human behaviour in harsh environments (hot, dirty, wet), a team of subject matter experts was used to ensure realism. A Military and Technical Advisory Panel of experts from the Defence Science and Technology Laboratory (DSTL) and the UK Army was established, bringing both expertise on

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

the military domain (e.g. doctrine and tactics used in the army) and hands-on experience in combat situations.

2.6.3 Commercial model

The commercial model was largely determined by the framework of research, development and acquisition of the MoD. As part of a strategy of de-risking early technologies, the MoD has set up a framework of monitoring and assessment of technology, whereby leading-edge technology that is inherently risky is assessed against several levels of readiness. These levels cover phases such as concept formulation, proof-of-concept, validation in different environments, through to final deployment, usually spanning between eight and 12 years in total. The objective is to ensure that technology reaches the level of readiness where it can be made available to the MoD's equipment supplier base to incorporate it into defence products. According to this framework, the HV-CGF project was in early readiness stages (but at higher levels than previous AOS projects). Thus, the customer of the two-year project phase described in this case study was the DSTO, who monitored the project, ensured that technical requirements were met and monitored who bought the product. However, in order to increase the technology readiness level towards deployment in military applications, a second two-year phase was commissioned by the MoD and has recently begun. One of the aspects to be addressed when making the system more ready for acquisition is to improve its user-friendliness. Thus, one of the objectives of this phase is to develop a graphical interface to illustrate the system's utility for training and simulation.

2.7 Lessons and experiences

Mutual benefit was gained by the agent and defence communities, in that a new application area for agents was demonstrated, as well as showing agents to be an innovative but feasible solution to the problem of modelling moderated behaviour in distributed simulations. This achievement raised further interest in agent technology within the defence community, particularly in applying agents to other types of simulations to solve problems previously deemed difficult or unsolvable, thus pushing agent technology towards exploration and development of new areas.

The project raised several challenges, both technical and managerial. The technical challenges were mostly related to integration aspects, as follows.

1. The integration between the agent-based system and the CGF systems raised several problems. For example, there was a need to use different time models in the agent model and the CGF system in order to synchronize between the processes run by different entities. Other integration problems often resulted in having to reprogram parts of the software (e.g. in the CGF case).
2. Matching cognitive science concepts to computing concepts and then linking back and interpreting simulation results according to cognitive principles was also problematic, not least because using agents was a novel modelling approach for the project team. Validation was therefore an important step and was achieved by comparing the results of the agent-based simulation with those produced by other cognitive architectures that had previously been validated through experiments with real humans.
3. Because much of the moderation of behaviour is time-based (e.g. the influence of caffeine reduces over time), the integration of time-dependent variations in beliefs across all agents was difficult to model, because of the large number of variables involved and the associated statistical uncertainty.

In general, the integration across three different technologies (agent, cognition modelling and CGF) was one of the main risks of the project, and the use of a clear and regularly updated plan for mitigating the risks was important throughout the whole project lifespan.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

A series of non-technical challenges were also encountered, as follows.

1. The knowledge acquisition and engineering processes were difficult. Gathering information about the military environments in a form coherent with the underlying model, from military experts with little understanding of, or interest in, agent technology, and with limited time available for discussions, proved challenging. Subsequently, converting this acquired complex knowledge into rules added an extra difficulty.
2. Equally important was the managerial overhead in having to coordinate development and integration from several subcontractors at different sites.

3 Eurobios and manufacturing management and optimization

At the end of the 1990s, SCA Packaging, a leading international manufacturer in the corrugated-box market, found itself under growing pressure to maintain a high quality of customer service and delivery times, as well as plant efficiency and reduced inventories of finished goods. They turned to Eurobios to provide an agent-based solution to explore different strategies for reducing stock levels without compromising delivery times, as well as evaluating the consequences of changes in its customer base. An agent-based tool was developed by Eurobios to model and simulate plant operations at SCA Packaging. The tool was developed in 2001 and was used by SCA Packaging to make strategic decisions on the customer base and plant capacity. As a result, SCA Packaging experienced a reduction of warehouse levels by over 35% while maintaining delivery commitments. This section presents the application of agent-based modelling (ABM) to develop this tool.

3.1 SCA Packaging

SCA Packaging was established in 1929 and originally comprised ten forest industry companies producing sawn goods and paper pulp in northern Sweden. SCA has since grown and diversified into three main businesses, delivering hygiene products, forest products and packaging. In the packaging domain, SCA is Europe's leading provider of customer-specific packaging. It has operations in 30 countries and production at 320 plants worldwide (SCA, 2004), and its business covers the whole supply chain, from the production of raw material (containerboard), to the high-quality finished packaging product.

3.2 Eurobios

Eurobios is a company specializing in commercial applications of complexity science and complex adaptive systems for business optimization problems. Eurobios uses agent-based technology to build simulations and optimizations of complex business systems, with a strategic focus on optimizing the supply chain. The company builds on the expertise of an international team of 15–20 scientists/developers, experienced in all stages of software development, from algorithm design through to implementation. This expertise has been built up through an extensive portfolio of successful projects with clients such as Unilever, BP, Post Danmark, SCA Packaging, Kappa Packaging, Peugeot, Ferrovia, Deutsche Post World Net, as well as through subcontracting on large-scale projects for vendors such as CapGemini and BCG.

In the case of SCA Packaging, an extensive collaboration with Eurobios resulted in a suite of agent-based tools to simulate, analyse and optimize supply chain operations (e.g. transport optimization and production reallocation, replenishment policy tuning, etc.). The use of ABM was beneficial because agents have the potential to capture the complexity of real-world problems. With agents, complex operations and entities in the supply chain can be modelled in great detail, while the complexity of supply chain problems emerges bottom-up from the interactions between agents.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

3.3 Modelling and simulation of a corrugated-box plant at SCA Packaging UK

In the corrugated-box 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 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 (Darley & Sanders, 2005). 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 variables of plant management), which can only be achieved through the use of sophisticated simulation technology.

3.4 The agent-based model

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

3.4.1 The problem

The main objective of the model was to understand the impact of changes in the plant set-up, and changes in some of the main characteristics of the customer orders, on the running of the factory. Two key factors were used to analyse the impact on the plant. The first was On-Time–In-Full (OTIF), an important quality measure, representing the number of orders delivered to the customer on-time and in-full quantity. The second factor was the warehouse stock level, which is a combination of the finished goods stock (kept by agreement with some customers) and the goods deposited temporarily in the dispatch area before transport. However, the two measures (the OTIF and warehouse level) cannot both be optimized at the same time, in that it is difficult to reduce finished good stocks without compromising OTIF delivery. To maximize plant efficiency, a detailed scheduling of operations on machines is usually required, in order to ensure permanent high machine utilization. However, the more detailed the production plan, the more sensitive (and therefore the less robust) it is to disturbances such as machine failure, customer order irregularities (peaks and troughs) or unpredictable changes in orders, resulting in last-minute changes to the production plan. As a result, dispatches can be delayed or delivered in less quantity than required, thus negatively affecting the OTIF of the plant.

Delaying or missing dispatches can also have a major negative impact on customer satisfaction, as well as negative financial implications. While it is difficult to establish exactly the cost of missed dispatches, it is, instead, easier to compare it with the cost of inventory. In this case, being able to reduce inventory levels while keeping the OTIF (i.e. the inverse of the cost of missed dispatches) constant is one of the means of plant optimization. For this reason the trade-off between the OTIF and warehouse levels is a good operational measure of the efficiency of the plant and therefore for testing the agent-based model.

3.4.2 The model

In the Eurobios model, software agents represent different entities, ranging from functional and physical departments to plant machines and humans, while the physical flow of goods and customer orders are represented by information flow between the communicating agents. One of the key agent types in the model is the customer, which aims to model the handling of incoming orders, where an order is described by the product and quantity, delivery location and delivery data and

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

time. The products themselves have several different parameters (such as box size, unique print design and colour scheme, number of boxes per pallet shipped, agreed stock level for that product and guaranteed lead-time), and each customer has a number of such products for which it places orders of various amounts over the year.

Orders from the customer arrive at the plant through the sales agent (representing the salesperson), which deals with an incoming order by performing a predefined action associated with the type of that order. For example, if the order is a request for a certain quantity of a product from pre-agreed warehouse stock, then the stock level is checked, the requested quantity is shipped to the customer and a stock-replenishment order is triggered. The quantity that must be shipped is then moved to the dispatch area, which is also represented as an agent, and is the location where finished goods are temporarily deposited while waiting to be loaded and transported to the customer. However, if there is no stock, a production order is triggered instead. Production orders are then slotted into the current production plan according to different order handling policies and lead-time agreements.

The behaviours of the agents are constructed as parametrized 'if–then' rules encoding the behaviours observed in the factory. Thus, the agents representing the machines (e.g. corrugated board machines, converters, gluers, stitchers, palletizers) have deterministic behaviours, characterized by parameters such as speed and set-time, while the behaviours of the agents modelling the humans mimic the formalized or unformalized if–then rules that salespeople and factory managers use daily. Part of the simulation for the agents then consisted of modifying the if–then rules in relatively simple ways, such as changing their priorities (e.g. how willing an operations manager is to work a weekend shift), or changing the strategy according to which production orders are slotted into the production plan (e.g. simple planning strategies can be used, such as slotting the production order nearer the time at which the customer order arrives or nearer the due date of delivery, or more complex strategies can be used to make machine utilization more uniform or to optimize the transportation schedule).

The simple rule-based modelling approach that mimics human behaviour was preferred to sophisticated dynamic optimization techniques. The assumption was that the knowledge of people with 20 years of hands-on experience in the plant is invaluable when dealing with complex plant operations and unpredictable customer behaviour. Furthermore, rules of thumb are often the basis for decisions taken by plant operators, making statistical approaches irrelevant.

3.5 Demonstration scenario

As a strategy tool, the agent-based simulation was used to make decisions on the choice between two important customers. One of the customers (Customer A) was a high margin customer (i.e. high profit margin per box), with short lead times (shorter than the average plant lead time) and with high levels of stock. By contrast, the second customer (Customer B) was a low profit margin customer, but allowed for large contractual lead-times and with little stock of its products. A choice between these customers had to be made when Customer B decided to bring a new product to the UK market and required a large production volume (approximately 20% of the volume of the SCA plant). Making the right strategic decision in this difficult situation required accurate and fast analysis of the cost of serving these customers versus the amount of orders and therefore the revenue brought in by the customers.

Using the simulation results from the agent-based tool, a series of options were evaluated in order to deal with the cost and capacity problems. The options ranged from refusing business from a number of small existing customers to taking on a limited number of orders from Customer B, or completely rejecting all orders from Customer A. Despite expectations, the model offered a clear solution to the problem, which was to stop working for Customer A completely, and to accept all orders from Customer B, resulting in a drop of 40% in warehouse stock levels and an increase in OTIF and overall production efficiency.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

The reason for this choice was a subtle difference in order patterns between Customer A and Customer B. It was found that Customer A would often 'abuse' its already short lead-time by requesting even shorter lead-times on its orders, while Customer B would instead accept even longer lead-times than specified in the contract. In addition, Customer B would tend to place its orders on Mondays, Tuesdays or Wednesdays, while Customer A would do so only on Thursdays. This hidden order pattern often resulted in having to shift orders from Thursday to Friday or even to the following Monday, and thus having to allocate orders over the weekend in order to avoid changing other customer's delivery agreements, which in turn incurred extra labour costs.

Through 'what-if' simulation, the model made visible certain customer order patterns and the propagation of events across the plant as a result of the arrival of orders. It was found that the relationship between the days of the week when the orders of Customer A and Customer B were placed, the utilization of the machines during those days and the lead-time requested for those orders were crucial for the cost incurred for the two customers. The observed pattern was useful for efficiently managing other customers as well, and therefore strategically valuable for SCA.

3.6 Business case and project management framework

The project was commissioned by senior management at SCA Packaging, drawing on Eurobios' reputation of successful implementations of agent-based systems. In particular, the extensive experience of Eurobios with supply chain optimization using agents was valuable in convincing managers at SCA Packaging about the feasibility of using the technology. In the first instance, however, problems at SCA plants were largely ill-defined, and it was therefore necessary to establish exactly what needed to be addressed. While many solutions to improving plant operations could be implemented, finding the areas where the ABM approach was appropriate and where ABM tools could help was critical.

The project therefore started with a six-month exploratory phase, during which Eurobios and SCA Packaging collaborated in carrying out a feasibility assessment. This assessment involved choosing between two SCA plants to focus on, by analysing existing plant problems and ways of improving them. The first option was a plant located in the South-East of England, which was largely underperforming because of a series of known problems that could be solved through solutions simpler than ABM. The second option, which was chosen, was a plant located in Edinburgh, with higher performance than the previous plant, but where optimization was more challenging and thus had the potential to emphasize more clearly the benefits of using agent technology to SCA managers.

Development was then split into three phases in order to better monitor the progress and manage the risk, and each phase lasted for one month. During the first phase, a basic model of the plant was built and the model was run with real company data. The objective of this phase was to provide a preliminary demonstration of how the plant works, so the realism of the model and the level of detail of the data were less important. During the second phase the model was further elaborated to add more realism and more detailed data, while the objective of the third phase was to make the model as realistic as possible and to perform 'what-if' analyses based on different simulation scenarios. The model was thus developed iteratively, so that the level of complexity and detail were increased at each iteration. The final, complete model was then calibrated, by running it with real data (e.g. customer orders) and by comparing the results of the simulation with the real values of variables such as warehouse inventory levels and cost.

3.7 Lessons and experiences

3.7.1 Knowledge engineering

Knowledge engineering was necessary for data acquisition, model building (rule formalization) and model validation (validating the rules and correcting missing or wrong values in the data). In order

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

to produce changes at an operational level, such as changing how people make decisions in the day-to-day running of the plant, a high level of realism was necessary both for the model and the data. To ensure this realism, Eurobios collected more data and in more detail than strictly required by the model, reducing the chances of inconsistencies between the model and reality, but also adding to the knowledge acquisition and validation effort.

Implementations of systems for business performance optimization can sometimes require significant data acquisition effort, particularly in companies where data are spread between different enterprise applications. For example, with the advent of enterprise resource planning (ERP) systems, a higher level of data integration has been achieved across companies, and therefore, in companies with ERP systems in place, the knowledge acquisition effort is less significant. This acquisition effort is also sometimes due to the nature of the business applications generating the data. For example, some of these systems are mainly transactional and not designed for storing intermediate states of information between the transactions or the events that model the business processes, or for storing other types of information such as the time of generating the data, or 'what-if' analysis information. Furthermore, the effort of eliciting data of sufficient quality and detail can sometimes be underestimated because software engineers and business people have a different understanding of what it is that constitutes good data.

In the case of SCA, however, Eurobios found that the acquisition effort was slightly less significant than for other companies, and data were made available more easily to the developers, so that the model could be run with real data from the first stage of development.

The benefits of the knowledge engineering task went beyond the realm of software development. For SCA, an add-on benefit was the insight into different departments' work practices, which were sometimes unformalized or *ad hoc*, based on an everyday understanding of different events, such as customer order fluctuations. For example, an agreement between plant managers and one customer to change delivery dates during a busy period to avoid unwanted delays and customer dissatisfaction, but resulting in changes in stock levels, remained unknown to top management and was not included in the agent-based model, creating discrepancies between the simulation results and real warehouse stock levels. By encouraging different departments to talk to each other in the process of eliciting plant information, some of the communication barriers between functional departments and between plant management and top management were eliminated.

3.7.2 Validation

For a model of the complexity of the SCA plant, validation through extensive tests was mandatory. Diagnosing problems and tracking system behaviour was important not only for validating the model, but also to explain to SCA how the system works and to eliminate issues related to plant operations that were not properly understood. However, the task proved challenging for several reasons. First, agent-based systems may explore realms of behaviour outside people's expectations and often yield surprises (Buchanan, 2005). Second, there was a lack of control over the input data, combined with a lack of access to the company servers storing those data. Every optimization was done over data collected from company systems in real time through a multi-step, interactive process, (based around stock replenishment, optimizing stock policies for warehouses, etc.), and there was no simple mechanism for recording the input data for each run. Consequently, each simulation run was irreproducible, which made diagnosis and debugging more difficult.

3.7.3 The trade-off between design and implementation

Eurobios experienced a dilemma in determining the time and effort required for the design and that for the implementation of the agent-based tool. While spending considerable time in advance to ensure sufficient data accuracy and designing the application to ensure reusability, quality and robustness was important, it was 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. Because of the complexity of the model, there was a point after which spending time in design

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

brought no further value to the final system because there was a chance that what was being designed would later be invalidated.

4 Magenta and vessel scheduling

Tankers International[3] is one of the world's largest very large crude carrier (VLCC) oil tanker pools. Scheduling tankers manually is a difficult task, and was costing Tankers International time and money each year because of non-optimal solutions and because of employees leaving the company and taking their vast amounts of knowledge regarding scheduling with them. To improve their scheduling process, Tankers International teamed up with Magenta Technology[4] to create the Ocean i-Scheduler, an intelligent scheduler for cargo fleets. The Ocean i-Scheduler assists the human schedulers at Tankers International to plan and replan which cargoes are to be assigned to which vessels in their fleet. This section describes the application of agent technology to the development of this system.

4.1 Tankers International

Tankers International is one of the world's largest VLCC oil tanker pools. Based in London, Tankers International has a 46-strong VLCC fleet, which operates all over the world.

4.1.1 Problems facing Tankers International's schedulers

Vessel scheduling at Tankers International was being performed manually by human schedulers with no automated support. This was causing two major problems, as follows.

1. The vast amount of information that must be considered to schedule such a large fleet of vessels in a dynamic environment meant that the complexity was high, potentially leading to costly mistakes. Reducing the frequency of such mistakes improves fleet utilization, allowing the company to increase their profits while using the same resources.
2. Each time a scheduler employee leaves the company, all the expertise that they have learnt about scheduling leaves with them, and new employees must start from the beginning.

4.1.2 Why is the scheduling so complex?

At first glance, it may seem like a simple problem to schedule vessels to cargo. However, there are many considerations that need to be taken into account. Each vessel, cargo and port has its own constraints that may prevent certain schedules (e.g. a vessel may be too large for a port, or a cargo too large for a vessel). Such constraints are known as hard constraints. That is, they are the constraints that cannot be broken when scheduling.

However, there are many other contracts that need to be taken into consideration, which are known as soft constraints. These are constraints that the scheduler can break only if there are no other reasonable solutions available. Examples of soft constraints are such things as the following.

1. Reducing the amount of time that a vessel travels without cargo. After a delivery, a vessel must travel to its next load port. If a vessel can collect cargo on the way to this port, then it can recover more of its travel costs.
2. Particular stevedoring companies may have a preference not to load or unload cargo on weekends because they have to pay their workers higher wages. However, agreeing to a contract that will not profit the company but may increase their chances of future contracts for the vessel could be beneficial in the long term.

[3] See `http://www.tankersinternational.com/`.
[4] See `http://www.magenta-technology.com/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

Both hard and soft constraints can be either global, meaning that they apply to the entire system, or they may be specific to a particular vessel/cargo/port or type of vessel/cargo/port. These constraints may also change over time; for example, a port may upgrade their facilities. In addition to these constraints, 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. These complexities led Tankers International to search for a system that provides decision support for its scheduling employees.

4.2 Magenta Technology's agent technology

Magenta Technology are located in London, with a software development centre in Russia. In 2002 they launched the commercial aspects of their business with the aim of providing multi-agent technology to companies to run their applications in a commercial environment. Magenta Technology's agent technology is used to build and run the Ocean i-Scheduler presented in this paper. This agent technology has two main components, as follows:

1. Ontology management toolkit: This toolkit is used to create ontologies for a specific business domain.
2. Virtual marketplace engine: This is used to build, run and monitor agent systems that use the ontology defined in the ontology management toolkit.

Magenta Technology's agent technology is built on Sun's Java 2 Enterprise Edition (J2EE), thus providing an enterprise-ready platform on which the above two components can run. For more details regarding Magenta Technology's agent technology, see the Magenta Website (`http://www.magenta-technology.com/`) and Wooldridge (2004).

4.2.1 Ontology management toolkit

An ontology is a computer processable model of business concepts that also records the attributes of, instances of and relationships between these concepts, therefore providing a model of business knowledge. In addition, having an ontology provides a suitable set of terms for agent communication, ensuring that when one agent uses a particular term, the other agents in the system share the same meaning of that term.

Magenta Technology's ontology toolkit allows developers to define an ontology using a graphical user interface. The ontology is modelled as a graph, with the nodes in the graph representing a concept or instance, and the edges in the graph representing the relationship between two concepts/instances, as shown below. The toolkit also provides automatic translation of ontologies into other formats, such as J2EE with method stubs, and XML. This translation provides an excellent means to take an ontology as a static document, and put it to use in a running system.

4.2.2 Virtual marketplace engine

Magenta Technology's virtual marketplace engine is a component for building, running and monitoring multi-agent systems. Once an ontology has been agreed upon and developed using the ontology management toolkit, the developers define the types of roles that will be in the system and how an agent fulfilling each role will make decisions using the knowledge in the ontology. The virtual marketplace engine provides support for principles and components that are common in multi-agent systems. For example, it provides a convenient way for sending messages between agents to save the developers from implementing such functionality. In addition, the platform provides utilities to monitor and debug agent applications at runtime.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

4.3 Agent technology in the Ocean i-Scheduler

The Tankers International application was built using Magenta Technology's agent technology using the multi-agent paradigm. Tankers International's original idea for an application was an online system that allows people to book their cargo to be transported. However, after being introduced to Magenta Technology and their agent technology, Tankers International decided that this technology may allow the development of a much more useful system. A proof of concept application developed by Magenta Technology convinced Tankers International to invest in such a system.

Tankers International were first impressed by the way that, using agent technology, applications could change behaviour without having to shut down the system or introduce new code, by editing the ontology at runtime. They were also impressed by how well the agent system itself could handle changes to cargo, vessels and ports. Tankers International envisaged two benefits to implementing their system using Magenta Technology's agent technology.

1. The system would provide support for scheduling to its employees, therefore reducing the probability of costly mistakes.
2. The ontology would provide a formal model of the business knowledge needed to schedule fleets; therefore when scheduling employees leave, the knowledge lost is greatly reduced.

In the development of the system, Tankers International together with Magenta Technology developers used the agent paradigm 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 that a vessel has. Then, for each vessel in the Tankers International fleet, a vessel agent was created. The developers claim that this led to a simplified design when compared to one they would have developed using other software engineering techniques.

Estimating the cost of developing the application is difficult. First, Magenta Technology's original agent platform required the use of Delphi instead of J2EE. The first version of the Tankers International application was developed using this Delphi version, and after Magenta Technology's change to J2EE, Tankers International re-implemented the system using the new version. Second, the development was split over the two companies.

Magenta Technology estimate that the second version of the application used approximately 10,000 person hours at Magenta Technology alone. Tankers International estimate that this version used an additional 4000 person hours of their time. However, we must take into account that a Delphi version had already been implemented. This gave developers from both companies experience not only in building multi-agent systems, but also meant that they had already solved many application-specific problems prior to developing the J2EE version, therefore reducing development time.

4.4 Lessons and experiences

From meetings with the developers regarding the design and implementation of the system, it seems as though agent technology has provided them with a means of developing flexible systems that can sufficiently handle changes in the environment. Tankers International are clearly impressed with both the agent paradigm and Magenta Technology's agent technology. The Ocean i-Scheduler is currently in the beta stage of testing, so Tankers International could not provide us with any quantitative data showing an improvement in the scheduling of their vessels, although they are convinced that their application will benefit their business in the long term.

In any busy industrial enterprise, there are always difficulties in integrating new working practices, and training and familiarization must play a vital role in the adoption of any new technology; Tankers International are facing this problem with the Ocean i-Scheduler. Even though the system has impressed Tankers International management, they admit that there is still an effort

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

to be made in training and familiarizing the scheduling staff to use and trust the output of the system. The human schedulers working at Tankers International have a great deal of experience in their area, and for them to accept and trust the recommendations produced by the scheduling software, a period of familiarization and day-to-day interaction with the system will be necessary.

From a development perspective, we have noted the following:

1. The developers did not use current agent standards for such things as agent communication, and have developed their own languages and protocols.
2. Agent technology not developed by Magenta Technology did not appear to play any role in the development.

These design and development choices reflect the closed nature of the agent system. In other words, standardized methodologies and communications languages were not important as the system does not inter-operate with other agents systems.

4.5 Further information

As part of Magenta Technology's commitment to further developing the Ocean i-Scheduler to improve the features and functionality, Magenta Technology is releasing Ocean i-Scheduler v2.1 in Spring 2005. This will also benefit from new platform software to increase agent interactions and performance. Features of the new release will include real-time scheduling, Web-based multi-user environment with powerful visualizations such as interactive maps, Gantt-charts, user-specific dashboards, KPI charts, additional reports and a simple manual rework facility to allow the user to take full control. Magenta Technology are also building related i-Scheduler software for truck logistics and project management. Many others are possible beyond these as well.

5 Whitestein and transportation networks

In transportation networks, unnecessary costs are incurred when coordination between dispatchers fails, and when vehicles are not utilized to their capacity as often as possible. To improve this, transport plans and schedules must be easily adaptable, even in response to unforeseen events. This section presents a case study of an agent-based system for dynamic transport optimization. Living Systems–Adaptive Transportation Networks (LS/ATN) was developed by Whitestein Technologies Inc. for a European logistics company, to help reduce the cost of transportation.

5.1 Whitestein Technologies

Whitestein Technologies is a Switzerland-based software company, founded in 1999, which specializes in agent-based solutions for the production and logistics domains. Today, Whitestein has a team of more than 80 people at four European locations. The core logistics product offered by Whitestein is LS/ATN. It is based on the Living Agents Runtime System (LARS), a development toolkit for agent-based optimization. LARS was initially developed by Living Systems GmbH, Germany, a pioneer of industry-grade software agent technologies, which was acquired by Whitestein Technologies in 2003. Since then, the core functionality of LARS has been extended and enhanced through other in-house development at Whitestein and is now offered as a new agent development platform, the Living Systems Technology Suite (LS/TS). A set of advanced solutions for complex telecommunications problems, as well as professional services are also part of the company portfolio.

5.2 The transportation problem

Optimal utilization of capacity is one of the most important factors in the transportation business. To this end, logistics companies implement computer-based tools for planning of the transportation network both at a strategic level and for short-term route optimization. Traditional off-the-shelf

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

software solutions are able to automatically create dispatching plans, but cannot adequately handle unexpected events and produce the necessary plan deviations in real time. In cases of last minute changes of orders or unexpected unavailability of trucks because of traffic jams, breakdowns or accidents, static planning systems cannot be used, and human effort is needed to adapt the dispatch plans and control their execution. This is because these planning systems are designed for relatively stable and not overly complex transportation networks, where transportation processes are mostly repetitive and therefore analytical optimization methods are applicable (Dorer & Calisti, 2005).

Planning systems need to find optimal routes in response to transportation requests arriving simultaneously from many customers. Transportation orders contain information about the location where the product must be delivered, the product quantity, and the time window allowed for pickup and delivery. The challenge lies in allocating a limited number of trucks of varying capacity and available at different locations, so that transportation time and costs are minimal, while the numbers of on-time pickups and deliveries, and therefore customer satisfaction, are maximized.

5.3 The LS/ATN system

LS/ATN was initially developed for a German subsidiary of a major European logistics company, and later contracted and customized for another smaller logistics customer. LS/ATN supports, optimizes and, to a large extent, automates the dispatching process of logistics companies.

The complexity of the truck allocation problem is a result of the dynamic nature of transportation:

1. transportation requests are not all known in advance, in that new orders can arrive at any time during the execution of optimization;
2. old orders can change during optimization;
3. the size of the order may be correctly known only when the driver arrives at the pickup location.

The optimization problem consists in finding a delivery plan, which is a set of routes and the associated transportation schedule for each route. A schedule must specify the locations and the times at which trucks must arrive and depart from each of these locations. Other input parameters that characterize the transportation problem are related to the product to be transported (e.g. order type, volume, weight, loading time, etc.) and the trucks available for transport (e.g. truck type, volume, loading and unloading equipment available, and staff and tariff).

5.3.1 Agent-based optimization

The design of the agent-based system takes into account the geographically dispersed nature of transportation and the way that European logistics companies approach the transportation problem. The agents therefore represent geographical regions, while cargo load is modelled as information (objects) flowing between agents. Thus, transportation operations are allocated to different dispatching regions, with each region controlled by an AgentRegionManager, while a broker agent (the AgentDistributor) deals with incoming transportation requests.

Optimization is implemented as a two-step process and aims to achieve the lowest transportation cost.

1. In the first step, a local optimization is performed. Every incoming transportation request is received by the AgentDistributor and allocated to the AgentRegionManager of the region containing the pickup location. The AgentRegionManager then searches for an optimal solution within that region, using a hill-climbing algorithm.
2. In the second step, global optimization is performed. The solution found at step (1) is changed by bilateral negotiation between the AgentRegionManagers, in order to find improvements to the current solution. In this negotiation, agents ask each other questions in order to determine

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

the availability of trucks for a certain load, or the availability of load for a certain number of trucks. Communication is restricted to the agents of the starting region and the end region, and is designed for the exchange of a minimum amount of information, in order to keep the overhead time and memory at minimum. For example, for a route which begins in Germany and ends in France, the negotiation and information exchange is only between the Germany agent and the France agent, and not with any other agent. Other negotiation solutions were also implemented and tested (e.g. between three and four agents), but they were found to significantly increase the solution space and therefore the search time, while the quality of the final solution was less than 0.1% higher than the bilateral negotiation.

While the optimization function is mainly based on cost, other objectives include reducing the number of trucks by utilizing them more efficiently, and reducing the length of the routes. In addition, the optimized solution must satisfy a set of constraints, which the algorithm calculating the routes must consider. Some of these constraints are compulsory (hard constraints), such as capacity and weight limitations of the truck, opening hours of customers, that pickup date is before delivery date, that pickup and delivery are performed by the same truck, etc. Other constraints (soft constraints) can be violated against a cost penalty, such as finding the earliest possible pickup time or the earliest delivery time.

Optimization is dynamic, being performed every time a new transportation request is entered in the system. This allows the system to dynamically adapt transport plans and schedules to possible deviations and unforeseen events. Distributing the optimization task among the agents ensures scalability with increases in the number of requests, and improves robustness by avoiding a single point of failure. The negotiation and coordination between agents have the potential to reduce overall transport costs and to improve resource consumption.

5.3.2 System integration

The system can be used both standalone for simulations purposes, supporting planning decisions, and also to automate the optimization task, when running as an integrated tool with telematics systems and transportation management systems (TMSs). As an operational production environment, the agent-based system must be linked to several back-end systems, as follows:

1. the back end is integrated with an order acquisition system;
2. the final results are automatically sent to the specific billing and reporting TMS subsystems;
3. geographical data necessary to calculate distances and time between locations are gathered and displayed in real time from geographic systems.

In addition, the dispatcher also has the ability to trace the execution of plans and routes of trucks, so that a decision can be taken whenever a delay occurs. Information on times at which a pickup or delivery occurred is usually fed back automatically to LS/ATN via a telematics system. The link with the telematics system is implemented through an agent that proactively informs the dispatcher whether loading or unloading were according to plan, but information about plan changes can also be entered manually by the dispatcher when it is obtained, for example through a phone call from the truck driver. Real-time tracking of pickup and delivery times is used by LS/ATN to react to plan changes and update current plans. For example, when an order is modified or is cancelled, if the corresponding route is still being planned, the agent re-optimizes the plan and informs the dispatcher. If the route is being executed, the agent only suggests a change to the transport plan but leaves the final decision to the dispatcher.

5.4 Business case and project management framework

The development of LS/ATN was a major project, involving more than 20 people over a period of three years. The project started with a one-year investigation phase, aimed at analysing the business

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

and technical requirements, to evaluate different solutions and to establish the value of agent technology in comparison with existing TMS. In this stage, it was also important to the customer to ensure that Whitestein could deliver both standard large information technology (IT) projects, successfully addressing complexity and scalability requirements, as well as providing dynamic optimization in real time.

From a business perspective, Whitestein proposed a detailed business case containing projections of cost and return on investment, for a subset of the business operations of the customer, over a period of 10 to 12 months. In technical terms, the LS/ATN system was compared with traditional TMS technology. In such traditional TMS, order information is collected, transportation routes are determined by an automatic management module, route information is then manually dispatched to the truck drivers and finally payment is calculated and truck companies are invoiced. The advantage of LS/ATN is that it provides automatic and optimized dispatching support and an automatic link to truck's telematics systems. While some TMS also address the optimization problem, they use either linear programming or standard industry software packages for logistics (e.g. iLog). Moreover, traditional TMS are designed for medium-size logistics companies and therefore optimization addresses only the regional and not the European level. In addition, some of them are only single-user applications.

A six-month period followed in which the system specification was developed, through collaboration between Whitestein and the customer, involving frequent face-to-face meetings (up to two or three times weekly) with local business and technical representatives. During the specification process, one key aspect addressed was system usability. To ensure easy adoption and operationalization of the final system, LS/ATN user interfaces had to share functionality and design features with the user interfaces of traditional TMSs, both in terms of static features and features that allow the user to design and change the interface elements (e.g. windows, columns, etc.). In addition, the system needed to have only an assisting role, with the dispatcher retaining full control over the dispatch information, given the experimental nature of the project. The dispatcher thus needed to be able to override, at any time, system decisions and have the visual and functional facility to move all the trucks into manual dispatcher mode, thus falling back on the electronic switchboard.

5.5 Lessons and experiences

From Whitestein's experience, the successful adoption of agent technology depends on several factors.

First, it was important to help people understand the business value of an agent-based system, to explain to them how agent technology works and how the optimization problem is approached. However, it was also essential that optimization was not overemphasized in relation to other technical features necessary for the system to compete on the TMS market. It was thus necessary to ensure that innovation was introduced only where needed and was balanced by standard features and functionality provided by TMS tools. In the process of selling agent technology, the agent metaphor was useful in attracting customer interest, as it was found to be an intuitive way of modelling the business domain, through the one-to-one mapping between agent roles and business roles (e.g. the dispatcher).

Building a good customer relationship was crucial and required significant effort and cooperation from both sides. For example, more than three years were needed for the developers to completely dig into the domain and understand its special challenges in detail.

The challenge throughout the project was to deliver software of industry-standard quality, and generally to compete with vendors of existing TMS tools. It was thus important that development and deployment were approached with traditional software engineering methods and standards, and traditional project and risk management techniques used for IT implementations.

A major challenge in operationalization was getting users to accept and trust the results of the optimization, and thus to start using the system on a daily basis.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

6 NuTech Solutions and plant optimization

Air Liquide America is a world leader in the highly competitive energy transformation market. To improve their business, Air Liquide America recognized the need for production schedules that are responsive to a large number of ever-changing constraints. To help with this scheduling, they approach NuTech Solutions, a software development company that focuses on solving difficult logistic business problems. Together they developed an optimization and production scheduler that uses a hybrid of agent-based technology and evolutionary computing techniques. This system helped Air Liquide America maintain their position as the market leader in the highly competitive energy transformation market. This section presents the work of NuTech and Air Liquide America on this system.

6.1 Air Liquide America

Founded in 1902, the Air Liquide Group[5] is the world leader in the production, distribution and transformation of industrial and medical gases. Their global presence (130 subsidiaries in more than 65 countries) allows the company to combine the resources and expertise of a global enterprise with a powerful local presence based on independent customer-focused teams. Sales in 2003 totalled €8394 million euros, with gases and services accounting for €7388 million. The group is listed on the Paris Stock Exchange, is a member of Eurostoxx 50 and has 31,900 employees worldwide as of December 2003. Air Liquide America[6], the US division of the Air Liquide Group, currently serves more than 8000 customers nationwide from more than 75 sources, and with more than 400 transportation units.

6.1.1 The challenge facing Air Liquide America

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. With a commitment to maintain their position as market leader, Air Liquide America sought to develop a system that would enable them to meet the growing challenges facing the industry.

6.1.2 The core problem

At the heart of the challenges facing Air Liquide America was a need for real-time evaluation of their supply chain assets, involving the development of methods to responsively construct schedules for the production of products at the many different plants owned by the company. Such schedules must be responsive to a large number of dynamic constraints, and thus have to be modifiable on the fly. Linked to this is the problem of finding optimal product routing policies from Air Liquide America's plants to its customer sites. Again, these policies had to be responsive to variations in customer demand and the changing constraints placed on the plants producing the products. These two core challenges formed the driving force behind the development of a system that would allow them to efficiently schedule the production and distribution of the products from more than 40 plants to 10,000 customer sites. In summary, the proposed system had to enable Air Liquide America to:

1. value and optimize their assets and supply chain;
2. reduce production and distribution costs;
3. deploy systems to capture lows in energy pricing;
4. deliver products cost effectively, where and when demanded by customers;

[5] See `http://www.airliquide.com/`.
[6] See `http://www.us.airliquide.com/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

5. project energy prices, customer demand and inventory levels;
6. make optimal risk decisions that have significant impact on profits.

As a consequence of meeting the above objectives, Air Liquide America expected to make annual gains of $20 million, resulting from new market opportunities and operational savings. It was within this context that Air Liquide America sought out NuTech Solutions and, with their collaboration, Air Liquide America hoped to develop a system that would meet the upcoming challenges facing them.

6.2 NuTech Solutions

Founded in 1999 upon the cutting-edge work of several world-renowned scientists, NuTech Solutions[7] has a mission to develop dynamic technology solutions able to intelligently learn and adapt. The company has six offices in three countries, and has helped over 50 corporate and government clients solve a range of problems. NuTech Solutions' activity spans a wide range of sectors including the automotive industry, consumer products, financial services, communications, manufacturing and energy, and has provided advanced predictive analysis and profit optimization software products for many Global 1000 companies.

6.2.1 NuTech Solutions and BiosGroup

In 2003 NuTech Solutions acquired BiosGroup, a world leader in the science of complexity and complex adaptive systems. BiosGroup has raised over $20 million in venture capital financing by providing simulation and modelling tools for government and industry bodies to solve complex and difficult problems. The acquisition of BiosGroup provided NuTech Solutions with cutting edge expertise in the science of complexity that is helping NuTech Solutions to accelerate the development of next generation solutions.

6.2.2 NuTech Solutions' Intelligent Business Engines™

At the core of NuTech Solutions' approach is a suite of Intelligent Business Engines™. The engines analyse, predict, optimize and adapt to solve challenging business problems while creating tangible business benefits. Intelligent Business Engines™ have been successfully applied to many areas, including data preparation and mining, modelling, simulation and forecasting, optimization, and prediction and forecasting.

6.3 NuTech Solutions and Air Liquide America

Having identified the challenges they were facing, Air Liquide America approached NuTech Solutions with a business case describing the core problem and their solution requirements. After a period of validating the business case, employees from both Air Liquide America and NuTech spent 15 months working on the initial development phase of the project. This phase required a large amount of interaction between the two companies (at times requiring virtual meetings with up to eight participants). Such intense interaction was necessary to facilitate the transfer of vital domain knowledge from Air Liquide America to NuTech Solutions. In these early stages of the collaboration, between three and four members of staff from NuTech Solutions were required, each of whom brought important skills to the project, including expertise in analysing optimization problems, expertise in database management and integration, and expertise in GUI design to provide the front-end systems for Air Liquide America's staff.

[7] See `http://www.nutechsolutions.com/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

6.3.1 The solution technologies

After considering a number of traditional operations research and mathematical programming solutions, NuTech Solutions determined that the nature of the problem required an innovative approach. The highly nonlinear aspects of the problem meant that it was not amenable to traditional analytical modelling techniques and, thus, demanded the use of non-traditional methods. This led the NuTech Solutions team to consider adopting an approach based around simulation. Furthermore, the complexity of the problem meant that it was unlikely that any one solution approach would be likely to succeed alone.

Based on these considerations, NuTech Solutions decided that the optimal approach was to construct a hybrid system utilizing agent-based and evolutionary computation approaches. At the core of the system is an ant system optimization approach that has the task of discovering efficient distribution routes for Air Liquide America's products from plant to customer. Coupled to this is a genetic algorithm, which searches for highly optimal production level schedules for the individual plants. Underpinning these two techniques are a host of expert heuristics derived and modelled on the knowledge and expertise developed by Air Liquide America. Combining all these techniques, NuTech Solutions developed a unique product called the supply chain production optimizer that is able to respond and adapt to all the interfaces across Air Liquide America's energy transformation business.

6.3.2 Ant system optimization

The agent-based modelling technique used by NuTech for Air Liquide is an ant system optimization technique. Ant system optimization exploits discoveries found in the study of real ant colonies and their self-organizing capabilities. When foraging for food, ant colonies quickly converge upon the shortest possible paths to food. This is achieved by each individual ant laying down pheromones that other ants follow. As more ants travel down the same path the pheromone trace becomes stronger and thus attracts more ants, until only the shortest and most used paths remain. NuTech models relevant components of Air Liquid's gas plants as ants, for example pipes and transport vehicles, which then enables the plant to be modelled as an ant system searching for optimal routing policies.

6.3.3 Genetic algorithms

Coupled to the ant system optimizer is a genetic algorithm (GA). GAs are computational models of evolutionary processes that can perform a highly parallel search across large, multi-dimensional search spaces. Specifically, GAs mimic the evolutionary processes of mating and mutation across a population of individuals, called a generation. By modelling aspects of the routing solutions given by the ant system optimizer as individuals of a population of solutions, the GA can then combine different solutions through simulated mating, in which aspects of two solutions are combined, and perform small mutations on each to generate new solutions which form the next generation. Within each generation, there can be many hundreds of solutions, each of which is then tested for fitness given an evaluation criterion, and the best of these can then be mated to produce the next generation of solutions. This process continues over many generations until the most optimal routing policies are found.

6.3.4 The supply chain production optimizer

Exploiting the combined power of ant system optimization and GA search, the supply chain production optimizer 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 Air Liquide America's plants and adapts them to projected energy prices, weather changes, client demand and desired inventory levels. Utilizing data representing seven-day power prices, customer demand projections, daily power costs and efficiency measures from every plant in the Air Liquide America

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

network, the system evaluates production costs based on forecasted demand to determine which plants should produce which products. On top of this, 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 to the Air Liquide America's Operations Control Center to enable adjustments to be made to the operation of the plants to meet production requirements across the whole network.

6.4 Lessons and experiences

6.4.1 Communication

In order for the collaboration between NuTech Solutions and Air Liquide America to be effective, much learning on both sides had to take place. One of the difficulties facing the project was eliciting information from Air Liquide America about the domain in a form that was computationally representable. Much of the initial developmental phase of the project was spent establishing, and making explicit, the wealth of background knowledge and assumptions that Air Liquide America's staff brought to their day-to-day problems and challenges. Making sure that the NuTech engineers understood the language and terminology used by Air Liquide America and ensuring that errors in understanding were caught and corrected became a key focus in the early stages of the project.

6.4.2 A Role for Agent Technology

Many of NuTech Solutions' successes with agent-based approaches arise in domains that require planning under uncertainty. In such domains, traditional planning approaches can result in plans that are brittle and prone to failure. Uncertain environments require robust plans that, while not always provably optimal, assure that the system will not fail in the face of environmental change. One way NuTech Solutions discovers these plans is through agent-based simulation, in which parts of the system are represented by agents. Once a simulation is constructed, candidate plans can be tested across a wide variety of simulated events, enabling the discovery of robust plans that guarantee a high degree of performance over varying circumstances. Using the multi-agent-based ant system optimization approach allowed NuTech Solutions to develop flexible routing strategies that allowed products to be transported efficiently through the network even when unexpected events occurred.

7 Acklin and information management for insurance

The process of handling insurance claims is heavily bureaucratic, with most of the information processing carried out by humans, resulting in long times for settlement. Many European insurance companies use heterogeneous information systems, with different means of storing and using data, and hence data must be exchanged manually between companies when dealing with claims. Acklin BV developed an agent-based system to automate the process of information exchange between a Dutch, a German and a Belgian insurance company when handling cross-European motor vehicle accident insurance claims. This section discusses Acklin's use of agent technology in this system.

7.1 Acklin

Acklin BV, a Netherlands software company specializing in agent technology and its application to business process optimization, identified this marked need in the insurance sector and attracted the interest of Dutch insurance company Interpolis in investing in an agent-based solution to reduce processing time and costs. With a strong skill set in software engineering, research and business consulting, and a strong record of successful optimization solutions delivered for transportation and retail companies, travel agencies, local government, etc., Acklin convinced Interpolis that agent technology also had the potential to solve the problems specific to this industry sector.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

7.2 Insurance claim handling for green card traffic: Interpolis

Interpolis is a major Dutch insurance company, also operating in Ireland, Portugal and Luxembourg. The company has approximately 6000 employees, more than one million private customers and 100,000 companies in its portfolio, and offers a wide range of life and non-life insurance products. Car accident insurance is one of the services offered and, within this sector, Interpolis operates as part of a network of 17 commercial insurance companies from different European countries. The process for handling transnational vehicle accident claims in Europe is called green card traffic. This process is initiated by the green card bureau in the country where the accident occurred, and then delegated to the necessary national insurance companies in the network. It then requires an exchange of information between the insurance company in the country where the claim originates and the insurance company that covers the claimant, as well as with car repair companies and medical experts. This information includes details about the car accident, car registration and driver details, insurance policy, medical information, etc., and is typically exchanged between claim handlers by telephone, fax or email. The reason is that legacy systems and back-office data at insurance companies are heterogeneous in the way they store and process data, because language translations are often required, and because of access limitations and confidentiality concerns on data held by each company.

Being a bureaucratic process that relies heavily on human effort for identifying and processing information, international claim handling is therefore time-consuming and costly. A typical claim settlement at Interpolis, for example, can take up to six months and can involve four to six organizations. However, a new European Commission directive introduced in 2003 required insurance companies to settle claims submissions within three months of the date of an accident or be heavily fined. In this context, Interpolis decided to investigate ways to automate and streamline the handling of information in the claims settlement process.

7.3 Requirements analysis and solution selection

The project started with an in-depth analysis of the business and technical requirements, carried out through close collaboration between Acklin and a range of functional experts at Interpolis (including network and database administrators, process engineers and insurance experts). The objective was to gain an understanding of the technological infrastructure (e.g. communication protocols, databases, etc.) and of the maintenance procedures (e.g. login and closing down of applications) in insurance companies, and to determine the most appropriate architecture and implementation tools for this context.

Specifically, the three main requirements were data privacy, system robustness and compliance with industry procedures. To maintain data privacy between insurance companies, the specific requirements were that agents should have no direct access to the back-office of any insurance company and should not be able to query databases of other insurance companies without a claim case. System robustness was equally important and demanded that agents should be able to recover from failure and be able to restart execution from the point of failure. To comply with existing IT procedures, agents would have to operate within time-windows and have shut-down and start-up procedures, synchronously with the time-windows in which each back-office operates; this is because back-office systems are typically operational only 18 or so hours per day, to allow for software maintenance and backups.

Given these requirements, several technical solutions were investigated, as follows:

1. The first option was to build a central data store residing in Brussels, to which every insurance company could connect in order to upload and retrieve data. Not only was this solution hugely expensive (having to connect 28 insurance companies), but it also had disadvantages from a technical point of view. One of these was the need for data mapping and synchronization between different companies' back-office systems in order to pull data into a common format into the central data store. However, the main drawback was that companies may use the data

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

store to access each others' information on different policies and then use the information to attract each others' customers, for example by offering cheaper insurance packages. This aspect made explicit the requirement that companies should retain control over what information to share, so that companies would have to make specific and limited requests for information from one another, rather than automatically or directly accessing each others' databases.
2. The second option was to give Web-based access to each individual back-office, but developing a unique Web interface for 28 insurance companies would have raised problems in terms of choice of language, functionality and incompatibility of procedures (e.g. login procedures); it would also have had the disadvantage that the final results of processing claims would still have to be transferred manually between back-offices. In addition, the solution was not even applicable to some companies, as they were not technically ready to connect to the Internet.
3. The third option, which was the option chosen, was to develop a network of communicating agents to which the back-office of every insurance company could connect in order to exchange information. In order to be robust, the agents needed to provide a fairly simple, unsophisticated application, with a minimum level of intelligence, without developing autonomous or negotiation capabilities.

Regarding communication between agents, the insurance companies chose e-mail as a means of message transport, because it was the only technology that all companies supported. Using CORBA as a middleware solution for message exchange was also an option, but the diverse states and levels of technology at user companies prevented it. Similarly, a design solution based on Web services was also rejected on grounds of speed, effort of installation of new infrastructure and procedures and, not least, because of a lack of trust in Internet connectivity from a data security point of view. In contrast, the benefits of an e-mail solution were that e-mail was already in place, and no new usage and maintenance procedures needed to be introduced, except those related to the use of e-mail by agents instead of people (e.g. virtual e-mail addresses had to be created by the e-mail administrators). In addition, because of the large organizational hierarchy of Interpolis, with several levels of business units, each having its own IT system, introducing a new communication medium would require integration with many IT systems and approval from the corresponding IT sub-departments, and would therefore be a cumbersome, costly and consequently risky option.

The use of existing agent frameworks and toolkits was also investigated. One option was to use JADE (`http://jade.tilab.com`), but many of its features were not needed, and the functionality of the KIR system was too specific to require such a general framework. The fact that JADE was open-source, with some components still in the process of being validated, also raised concerns about its robustness, while the use of RMI as an invocation method in JADE could not be supported by the infrastructure of the insurance companies. Another option was the agent toolkit from Tryllian (`http://www.tryllian.com`), but this would have required the installation of additional servers, which was less appealing than developing the agent system from the beginning as a thin client.

7.4 The KIR System

The KIR System was developed by Acklin for Interpolis and two of its collaborating companies: a Belgian insurance company (KBC) and a German insurance company (R+V). The system consists of a network of communicating agents which exchange insurance information on behalf of insurance companies, and which are able to connect to the back-office of every insurance company. A multi-stage software engineering approach was applied, as follows:

1. the process of collaboration between insurance companies was analysed;
2. the activities to be performed in the course of this process were mapped onto agent behaviour and agent communicative acts;
3. an interface was designed to allow agents to have controlled access to the back-office.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

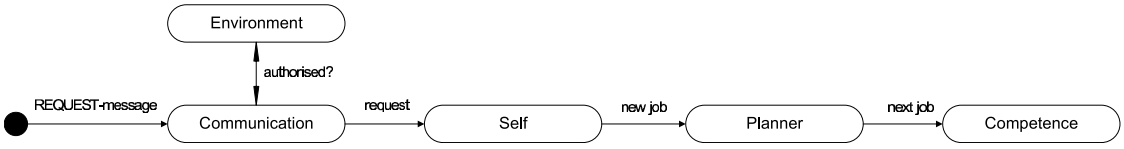


Figure 2 The Incoming-Questions-Flow describing the communication between the five capabilities model when receiving a REQUEST messages (Source: van Aart *et al.*, 2002)

7.4.1 The five capabilities model

Every agent was designed and implemented using the five capabilities model, based on a conceptual separation of concerns, because it has the potential to better explain to managers and software engineers at Interpolis the value of agent technology. The five capabilities of communication, competence, self, planner and environment are implemented as separate models within each agent, in the form of specialized functions and knowledge.

1. The communication model handles all the interactions between agents and other systems, with methods for message transportation, representation and interpretation.
2. The competence model contains methods for the execution of the tasks for which the agent is designed.
3. The self model represents the knowledge that an agent has about its own capabilities (i.e. goals, tasks, jobs, states and competences).
4. The planner model contains strategies that can be used to achieve an agent's goals.
5. 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 implementation of each model depends on the type of agent; for example, for agent roles such as payer and handler, the differences in capabilities are implemented only in the competence and self models.

The models allow the definition of a standardized interaction between agents via certain message exchange patterns. The Incoming-Questions-Flow illustrated in Figure 2 is one such pattern, in which message exchange is initiated by an agent requesting another agent to provide certain information, such as looking up a person's file. When the request message is sent, the communication model of the receiving agent checks the validity of the message and whether it is sent to the right agent, and it requests the environment model to verify whether the sender is authorized to ask that question. If the sender is authorized, the self model checks if it recognizes the content of the message, otherwise it sends a failure message to the sender through the communication model. If the message is in response to an existing conversation, the self model defines a new job and sends it to the planner; if it is a new request, the planner creates a new job and forwards it to the competence model. In turn, the competence model then selects the appropriate execution method for that job and performs it, after which an Outgoing-Answers-Flow is initiated by the communication model in order to send the results of the execution back to the original requesting agent.

7.4.2 Inter-agent communication

The communication between agents was designed to reflect the green card traffic process of information exchange. The agent interaction protocols used were based on the FIPA REQUEST protocol (see FIPA, 2002). Figure 3 shows the Agent-UML representation of the information exchange process between a handling and paying bureau, where the interaction between humans by way of speech and handwriting is replaced by communicative acts sent between agents. The process consists of two identification tasks:

1. client identification, using green card number and licence plate;
2. case identification, using policy number and local claim number.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

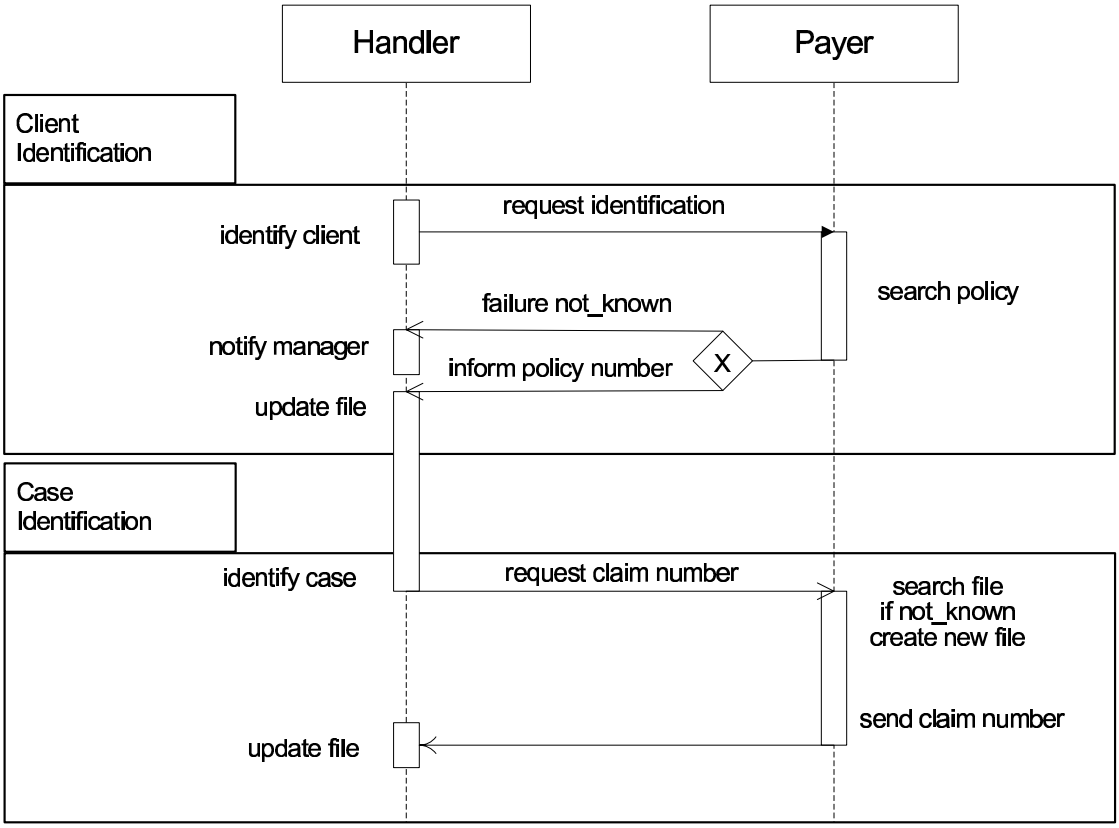


Figure 3 The green card traffic illustrated in a UML activity diagram (Source: van Aart *et al.*, 2002)

The interaction is initiated by a request for client information from one insurance company to another. The company making this request (here called the handler) is the company to which the claim is first sent by the person involved in an accident, and the office which makes this request is the handling bureau. The company receiving this request (here called the payer) deals with it in its paying bureau. In the KIR system, the manager of the claim handling department delegates the two identification tasks to an agent rather than of to a human claim handler. The handler agent then begins the process with a REQUEST message to the payer agent for identification of the client, based on the licence plate and green card number. The payer agent validates the identification and can respond with (1) a FAILURE message containing not known, which means that the client is not known or (2) an INFORM message containing a policy number, meaning that the payer has identified the local insurance taker. A REQUEST message for claim identification is then sent by the handler agent, asking for all known data from the file of the payer. The payer agent can respond with an INFORM message containing a policy number, which means that the payer agent has either created a new file with locally known data, or a file already existed for the case. The latter can occur when the insured party had previously registered this accident. In both cases, the payer agent sends a claim number, which is the key to the file of the accident.

This communication infrastructure between agents was implemented using e-mail as means of message transport, for reasons of inter-company interoperability and because it would facilitate the transition of KIR to other insurance companies, in the future, if necessary.

7.4.3 Interface to the back-office

In order to ensure data security and the stability of the back-office, the interaction between the agents and the back-office had to be minimal. The chosen approach was to develop a transducer as an interface between the agent and the back-office, designed to map execution commands from the agent to the back-office and to map results from the back-office to the agent. This solution enabled

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

agents to have controlled, but not direct, access to the required functionality and data from the legacy systems. The transfer of data between the agents and the transducer was done through a blackboard, which is a buffer where instructions and reports are read and written by agents. For example, some of the actions that an agent can execute using the transducer are to search a file by policy number, to create a file using policy and licence plate number or to retrieve data from a file using the policy number. A blackboard structure enables agents to communicate without being present at the same time and place.

The transducer was developed by the IT department of Interpolis, based on a specification of the functionality of the interface given by Acklin. While many approaches could have been used to ensure controlled access to the legacy system, the advantage of this approach is that, in order to install an agent at an insurance company, only the transducer must be configured.

7.5 Business case and project management framework

A business case was prepared in support of the investment decision. Being one of the first agent-based systems in the insurance sector, and because Interpolis had no prior experience of agent technology, the investment was viewed as an experiment, and the decision to invest was made purely on financial grounds rather than the technical benefits brought by advanced technology. The factors taken into account were the cost of acquiring, deploying and maintaining the agent-based system and the savings expected to arise from using the system (determined from the revenue gained per year from using the system with a certain number of customers) and how much of this revenue might be used to recover the investment and over what period of time. The business case also included an analysis of potential commercial risks arising from developing a new system with totally new technology, and from outsourcing the software development to a small and new IT company such as Acklin.

The project fee was determined by Acklin based on a fixed-price strategy, aiming to cover the estimated cost of development in terms of equipment and human resources, while also taking a small financial risk. Although the system was cheaper and easier to build than a centralized data store, in hindsight, the opinion was that the system was priced lower than would have been acceptable in the insurance sector, given the usually high costs of acquiring traditional IT insurance systems. However, the low project fee encouraged the decision of Interpolis to invest in the system, and counterbalanced the risk taken by using a novel technical approach.

While the investment decision was made by top management at Interpolis, the decision process also involved other organizational functions, and approval had to be obtained from process managers and from the IT department of Interpolis. The overall decision process required approximately one year, with final approval being given by the database administrator. During the entire project period, Interpolis engaged approximately 40 people of which 10 participated extensively. On the Acklin side, a commercial expert worked with managers at Interpolis, a software engineer specializing in agent technology was responsible for communication with process engineers at Interpolis and for translating insurance process models onto processes at agent and inter-agent levels, and a software programmer was responsible for the implementation of the system.

Interpolis was largely the initiator of the automation effort, but the German and Belgian companies were easily convinced to participate, not only because of the small implementation cost, but also driven by the need to be interoperable and the fact that the new system establishes a whole new communication infrastructure that all companies must share in order to collaborate. Interpolis also attempted to bring other smaller collaborating insurance companies to make a similar investment, but these rejected the option because of an insufficient number of claim cases to generate the necessary cost savings.

The product strategy adopted by Acklin for the insurance market was to build an agent-based system for automating cross-company business transactions as a set of components which could afterwards be customized to the specific technical requirements of the contracting insurance

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

company, and coupled via a specific interface to the back-office of that company. This model allows Acklin to sell customization and maintenance as part of a consultancy services package after selling the core system. Customization typically consists of language translations, configuring the set and the format of the attributes exchanged between insurance companies, or providing the functionality that allows the customer themselves to change the set of attributes (e.g. via a configuration file). Other functionality added would sometimes be the control and maintenance capabilities that allow tracing the number and types of message exchanged between companies, or implementing policies to limit the frequency of messages that a company can send in order to request information from another company.

7.6 Lessons and experiences

The agent metaphor was found to be a useful and natural means to model the insurance business. By mapping organizational 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. This aspect appealed particularly to top managers and process engineers at Interpolis, and made it easier to convince them to invest in the agent-based solution.

The success of the system was even more important given the traditional difficulties to overcome when introducing new technology simultaneously in several large organizations running legacy IT infrastructures. The difficulties were not only related to technology incompatibility, but also to the organizational model of the insurance companies. Because the IT departments of all five insurance companies in the Netherlands were merged to create a separate IT company, any new system implementation is first outsourced to this IT company and only if the results are unsatisfactory is a third-party IT developer brought in. In these circumstances, more intensive negotiation between Acklin and the IT experts at Interpolis was needed to overcome a slight cultural rigidity. Thus, implementing a new IT solution through a third-party software house may have made in-house IT experts at the client company sometimes feel that the quality of their work was less appreciated; this was perhaps increased because of uncertainty about the comparative advantages of an agent-based system compared to the systems developed and maintained by the IT department. In addition, there was a perceived inconvenience of having to integrate an external application with company databases because of the extra effort required (e.g. increased maintenance work for database administrators).

The agent metaphor was used not only as a design approach but also as a way to manage the project, by dividing development into stages according to the implementation of each agent. This approach made it easier to plan the implementation of each agent individually rather than the implementation of the overall system layer by layer.

8 Rockwell Automation and chilled water system

This section presents a case study based on the development of a control application using agent technology developed in Rockwell Automation, the autonomous cooperative system (ACS) agent platform (Maturana *et al.*, 2005). The application presented involves intelligent distributed control of a shipboard chilled water system (CWS) for vessels of the US-Navy (Tichý *et al.*, 2004).

8.1 Rockwell Automation

Rockwell Automation, Inc.[8] is one of the world's leading companies in the field of industrial automation and control. Its Prague-based subsidiary, Rockwell Automation, s.r.o, Prague, has 50

[8] http://www.rockwellautomation.com/.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

employees, mainly researchers and developers. The Research Centre Prague, as a division of the Rockwell Automation s.r.o., is a part of the Advanced Technology organization of Rockwell Automation. Research activities focus on intelligent control, distributed discrete manufacturing solutions, system diagnostics and software architectures for real-time control. Specifically, for more than a decade, research has concentrated on designing highly distributed, dynamically reconfigurable and intelligent manufacturing control infrastructures drawing on multi-agent systems and holonic systems paradigms. The main focus has been aimed at agent-based control at the shop-floor level where the agents encapsulate both the low-level control part responsible for real-time control of the physical manufacturing equipment and the software agent part in charge of the high-level decision-making and negotiation. Particular attention has been paid both to the design of the general simulation framework enabling testing of the agent control system behaviour in an emulated manufacturing environment and also to the smooth transfer to actual deployment. Significant cooperation in this area of research has been established between Rockwell Automation and the Centre for Distributed Automatic Control at Cambridge University, UK, and with the ACIN Institute at Vienna University of Technology, Austria.

8.2 Application description

The CWS pilot system is based on the reduced scale advanced demonstrator (RSAD) model, a reconfigurable fluid system test platform. The RSAD has an integrated control architecture that includes Rockwell Automation technology for control and visualization. The RSAD model is currently configured as a CWS.

The physical layout of the RSAD CWS is a scaled-down version from a real ship. There is one chiller per zone, i.e. currently two plants. There are 16 heat sources on board the ship (e.g. combat systems, communication systems, and radar and sonar equipment) that must be kept cool enough, i.e. below the equipment's shutdown temperature, in order to operate. Each water cooling plant is represented by an agent, as well as each heat source, each valve and some parts of the piping system. Within the RSAD, immersion heaters provide the energy to increase the temperature of the heat sources. A temperature sensor at each source provides temperature input to the control system. The main circulation piping is looped to provide alternative paths from any chilled water plant to any heat source.

8.2.1 System requirements

To design a highly distributed shipboard automation system, the following four fundamental requirements are considered.

1. Reduced manning. This is intended to create a system with less human intervention and more intelligent components capable of making decisions on behalf of the equipment.
2. Flexible distributed control. This is understood as the capability of the system to adapt its components' operations to respond to dynamically changing conditions without using predefined recipes. To achieve flexible distributed control, extensions to the control and network software are required to enable creation of component-level intelligence that increases robustness of the automation system.
3. Commercial-off-the-shelf (COTS). This aspect addresses the cost reduction and system life cycle requirements of new shipboard systems. Under the COTS scope, a ship can be maintained at any friendly location in the world.
4. Reliable and survivable operation. As the system becomes more autonomous and self-determined, it is required to augment the level of diagnosability of the components in a distributed manner. The system must be decentralized to avoid a single point of failure. The decision-making process should be placed as close as possible to the component level so that a disconnected system should be able to provide local decisions in the case of failure.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

8.2.2 Agent technology deployment

For this type of application, classical or distributed control methodology is applied using programmable logical controllers (PLCs). Nevertheless, in this case there were requirements for survivability and flexibility and these are more or less features of multi-agent systems. Although it is possible to use backup of PLCs, the control application has to be designed so that all possible combination of failures that might occur are avoided. The control application can be physically distributed over several controllers, but the presence of any centralized point in this structure becomes a single point of failure that has to be avoided.

Agents are distributed according to the physical location of the hardware equipment. The hardware partitioning follows the wiring configuration of the input/output (I/O) signals in such a way that the components are independent of one another. Single-point-of-failure nodes, from the hardware distribution perspective, are avoided. Other rules need to be satisfied to be free from single-point-of-failure nodes (e.g. use no remote I/O, deploy the smallest possible number of agents per component and controller, and use no mapped inter-controller data). With these rules in mind, the 68 agents for this application are deployed within 23 industrial controllers.

8.3 Autonomous cooperative system description

There was a requirement to use COTS PLCs. Because none of the existing multi-agent system platforms was applicable for PLCs at that time, design and implementation of a custom multi-agent system platform was needed; this became the autonomous cooperative system (ACS) (Maturana *et al.*, 2005). Standard Rockwell Automation ControlLogix and Flexlogix controllers were augmented with multi-agent system capabilities. Each controller is able to host a collection of various agents, where agents of the same type are downloaded only once and instances differ only by their name and configuration.

8.3.1 Main features of agent platform

There are two implementations of ACS available: C++ and Java. The first is able to run in programmable logical controllers and also in their software emulations running on PCs. The Java version is able to run in any Java SE 1.5.

A structure of middle-agents (DF and AMS agents in this case) called dynamic hierarchical teams (DHTs) were designed and implemented (Tichý, 2003). This structure has a user-defined level of fault tolerance and is moreover fixed scalable, i.e. the structure can be extended by fixed known cost. This is an important aspect for the fault tolerance of the whole system as these middle-agents are used to find suitable cooperation partners among agents.

There are two levels of agent programming and execution within one agent, as follows:

1. High-level, written in a high-level programming language (such as C++ or Java). This is the core of the agent, the part that is able to communicate with other agents, form plans, undertake reasoning, etc.
2. Low-level, written in relay ladder logic, a standard control language. This part is responsible for real-time control to guarantee response times and safety of the control system. This part is also responsible for generating events to the high-level part of the agent as notifications of some state or condition requiring the agent's attention.

The presence of these two levels is essential for fault tolerance. The low-level ensures safe functionality of a control system even when the high-level part is not working properly or when there is a need for fast reaction to some critical situation, perhaps before the high-level part is able to make a more accurate decision.

So far, a declarative style of programming agent behaviour has been used and successful implementation of several agent-based control systems undertaken. Nevertheless, this approach has

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

been reevaluated and it became clear that the declarative style has low flexibility as any additional feature has to be included into all parts of the system. Thus, a procedural style of programming has been designed and a procedural engine has been created. This engine enables use of the full power of a programming language (Java or C++ in this case) together with a set of functions and attributes to interact with other agents, trigger planning processes, and so on.

8.3.2 Agent development environment

The development environment (DE) (Staron *et al.*, 2004) is a software tool that assists the user in programming and deployment of the distributed application, for example creation and deployment of agents, I/O connections, and automatic code generation. The DE introduces the following dimensions into the development phase.

1. It allows the user to specify the physical and behavioural aspects of the application in a manner completely independent of the control system.
2. It enables the user to specify a multi-processor control system in a manner completely independent of the application that is to run on it.
3. It assists the user in combining an application with a control system.
4. It generates the control code and behaviour descriptions for each agent in the system.
5. It combines the code for all agents assigned to each processor in the system.
6. It augments each controller automatically to handle the communications to other controllers as a result of the program distribution.
7. It communicates with all the controllers involved in an application, for their programming and configuration, and for subsequent monitoring and editing.

The DE is based on a library of components called the template library (TL). The library is editable by the user, and each template can contain both low-level control behaviour (written in ladder diagram) and higher-level intelligent behaviour. The model for the control behaviour supports an 'object' view of the components in that, for example, inheritance is supported. The TL author can express, for example, that 'a Radar is a type of CombatSystem'. Each instance of Radar inherits all ladder data definitions and the logic from the CombatSystem template. Each library is targeted to a specific application domain, for example material handling systems or shipboard CWSs, and so can be used to create control systems for multiple similar facilities (F) in the same domain. In this way, the effort to build the library is amortized over all the facilities built with the library, and each facility reaps the benefits of having control system software that is structured, well-defined, predictable and tested.

The user creates a facility from components of the TL and customizes their parameters. Next, the user establishes a control system (CS) that describes all the controllers, I/O cards and networks, plus their interconnections. After the TL, F and CS parts are completed, the user generates and compiles the code. We currently support relay ladder logic (IEC 1131–3) for low-level agent part and C++ code for high-level agent part, plus we are prepared to support also Java code. After the agent assignment, the user downloads the software into the controllers.

Because TL, F and CS editors are independent, it is possible to change only the necessary parts when the system needs to incorporate changes. For example, a new controller can be added by the CS editor and subsequently have some agents assigned to it. The system is regenerated in a manner consistent with all modifications.

8.4 Lessons and experiences

There were several key lessons from this project. The first is that for actual deployment of such an agent system it is important to simulate its behaviour prior to running it in a real environment. Both a software simulation environment (using Matlab) and a small-scale hardware simulation

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

environment were created for testing purposes. Simulation is especially important for a multi-agent system as an emergent behaviour may appear in this case and its identification and study without simulation can be hard.

The second lesson concerns the issue of standards. During the development of the multi-agent system platform close attention was paid the FIPA agent system standards. The FIPA standards were given low priority in the beginning as they bring with them a huge overhead of resources. Subsequently, as development proceeded, the system was made increasingly FIPA-compliant. Standardization is important when two or more systems (multi-agent system platforms in this case) need to be interconnected, and this was the case for other applications.

Nevertheless, this standards-compliance effort is not for free. The first issue is the overhead that it brings. For example, the FIPA agent communication language (ACL) has to be adopted on top of messages. These are then embedded as a content of this ACL message structure. Agents have to be able to use the FIPA ACL semantic language (SL) for mutual understanding of communication, which represents a further overhead. The overhead is not only in computation time and memory consumption, but also in the effort of the people developing the system. Moreover, SL is impractical as its notation is not suitable for extensive parsing and working with the message in a form of the collections of objects. In summary, all these factors have to be carefully evaluated before adopting the FIPA ACL standards. One of the possible compromises is to use standards for external communication only and avoid them for internal purposes.

A third lesson was that, as the system becomes more complex, there is a need for a visualization and debugging tool to observe the internal communication among the agents and behaviour of the system and to discover potential problems. This need led to the development of a tool that receives messages from all agents in the system, reasons about the information, and presents it from different points of view and from different levels of perspective. All these parts of the visualization screen are interconnected. For example, it is possible to identify some problem in the workflow window, select the appropriate part of the conversation among the agents and receive the list of messages involved in the conversation. Using the tool, the user can also configure running agents by sending service messages to them.

On the issue of simulation, the manufacturing agent simulation tool (MAST; Vrba & Mařík, 2005) was developed separately from the ACS system. MAST represents a new generation of simulation systems with embedded multi-agent systems principles aimed at the manufacturing domain. It runs on top of the JADE system and it has been intended as an agent-based demonstration application that would illustrate, for some typical manufacturing task, the major benefits of the deployment of agent technology. To show the robustness and flexibility of the agent solution, attention was paid to failure detection and recovery. A failure of any component can be emulated (e.g. a failure of the conveyor belt) causing the agents to start negotiations on alternative transportation paths while avoiding the broken component. An easy-to-understand visualization helps the user to observe the overall behaviour of the system during the simulation.

Since the project began, an enormous amount of time has been spent on the design of the generic planning engine. This planning engine was originally based on a declarative programming language, which imposed many limitations. Any addition to this language resulted in changes required for all related tools. Thus, a lesson learnt was that moving to a procedural language gave the benefit of overcoming these limitations and stabilizing development of the related tools. During the deployment of the multi-agent system, a key lesson was that a fault-tolerant multi-agent system must satisfy the following characteristics:

1. Reliable communication. The system has to guarantee that no messages are lost or duplicated. Every message must be delivered, or else the sender must be notified if a message cannot be delivered.
2. Fault-tolerant agent platform. As the agent platform is an environment where agents live, the possible failure of some agent should affect neither this environment nor other agents in this system.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

3. Fault-tolerant knowledge. Knowledge about other agents is a fundamental part of a multi-agent system. Thus, this knowledge has to be present at configuration time or dynamically obtained in a fault-tolerant manner. For example, having only one directory facilitator that manages this knowledge creates a single point of failure in the system.
4. Physical distribution. It should be possible to physically distribute the agents of a multi-agent system and allow communication among them. This distribution increases fault tolerance in the case of a hardware failures, power failures, etc.

Because agent technology is more powerful than classical control, it is important not to limit its power by blindly fulfilling the initial requirements of customers, especially procedures that they are using with classical control. For example, there are standard procedures for performing specific tasks that a person has to follow, usually in a form of recipe. Nevertheless, it could be possible to use more advanced techniques in a multi-agent system that guarantee the same result with increased performance.

What would be the main guidance while considering implementation of an agent system? In the beginning, ensure that agent technology provides some advantage over classical control in considered implementation. The benefit can be not only from improved functionality, fault tolerance and scalability. The benefit can also be the ability to reuse an already developed agent solution to solve a similar problem. The first implementation is likely to be harder than before, but subsequent implementations may just be a matter of different parameter configuration.

9 The combined system and multi-agent systems in crisis management

The Combined Systems (Chaotic Open world Multi-agent Based Intelligently NEtworkedDecision support Systems) project[9] is a research project consisting of several multidisciplinary partners whose aim is to explore and design engineering guidelines for large-scale, open systems in a constantly changing environment. The validation scenario of the project is taken from the field of crisis management, and outlines a use case in which poisonous material has been accidentally released into a city. Early observation of the problems at hand and communication between crisis management experts and people 'on the ground' are both crucial, as is alerting the public of the danger and providing emergency services such as medical support to people affected. The Combined Systems project has provided an intelligent decision support system, using multi-agent technology, that can collaboratively save lives, stabilize the cause of the incident and conserve the surrounding infrastructure. The decision support system is the focus of this section.

The purpose of this paper is to give examples of deployed, industry-based applications using agents. While the Combined System is targeted at industry, funding is mainly government provided, and the system needs are not directly industry driven. However, this case study is included in the series because it contributes to the theme of industry projects. The size and complexity of the validation scenarios in the Combined Systems project, combined with the many 'off-the-shelf' solutions that are being used in the project, make it a suitable candidate.

9.1 Project information

The Combined Systems project is funded by the Dutch government. At a total of 40 person-years (10 person-years each year for four years), the Combined Systems project is quite large by research project standards.

Expertise within the project varies, with nine different partners from various backgrounds. Those areas that are strongly represented in the project include experts from communication, infrastructure, autonomous systems, human factors and multi-agent systems, with crisis management experts regularly consulted.

[9] See `http://combined.decis.nl/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

The project is hosted by the DECIS laboratory[10] in the Netherlands. The DECIS laboratory consists of four major partners: Delft University of Technology[11], Thales Research and Technology Netherlands[12], University of Amsterdam[13] and the Netherlands Organization for Applied Scientific Research[14]. These partners have further invited several other commercial companies to assist on the project: Y'All,[15] Inology[16] and Lithp Systems[17].

9.2 Validation scenario

A large-scale validation scenario has been developed that describes a real-world problem, in which many of the aspects of a multi-agent system are useful. The scenario is based on real-world reports from the crisis management domain. It begins with two ships colliding in Rotterdam Harbour, one of which is transporting poisonous material. This material is leaked from the ship, and the wind starts to spread the material over the city of Rotterdam.

Evacuation of the effected areas is imperative, and early observation of what is happening in the harbour and in the city is crucial to crisis management experts. Getting medical aid to the affected individuals is also of utmost importance.

The Combined Systems project has developed this case study because it provides a sufficiently non-trivial use case to research the types of method and support needed for large-scale, dynamic systems, is feasible to implement the Combined Demonstration System, and is a good case for demonstration.

9.2.1 Communication

The Combined Demonstration System uses mobile devices in crisis management tasks. In the 21st century, communication using mobile devices is useful in such cases, because such devices are ubiquitous, and transferring data to and from them is fast and often more reliable than many other means of communication.

For the purpose of observation, mobile devices provide distributed perceptions of the crisis at hand. That is, individuals can perceive their local environment, but the problem would often be more widespread, so different individuals in different areas surrounding the harbour can give crisis management experts a better idea of the overall problem[18]. Individuals in different areas can be located by the mobile networks, and can be asked to provide information as to what they see, and what the possible causes of any problems may be. Critical thinking (Schraagen *et al.*, 2005) is used to help extract information from the individuals on the ground.

To alert the public, mobile phones can be used to broadcast messages informing the public of the situation, and giving advice and instructions as to what action they should take. In addition, mobile phones can be used by individuals to register information, such as their medical abilities, or people in need of medical assistance. This will help with getting assistance to individuals in need at a faster rate. Of course, problems arise when using mobile devices for communication. Most notably, connections between mobile devices and the rest of the network can be lost intermittently, which means the data that the people on the ground are trying to send to the crisis management experts can be lost if the system is not designed robustly enough. To ensure robust communication between agents, distributed coding techniques are applied (Miletić & DeWilde, 2004).

[10] See `http://www.decis.nl/`.
[11] See `http://www.tudelft.nl/`.
[12] See `http://www.thalesgroup.com/ga/research_and_technology/policy.htm`.
[13] See `http://www.uva.nl/`.
[14] See `http://www.tno.nl/`.
[15] See `http://www.yall.nl/`.
[16] See `http://www.inology.nl/`.
[17] See `http://www.lithp.nl/`.
[18] Similar effects were seen in the July 2005 attacks on the London underground, in which individuals were able to use their mobile phones to send images and text to police and the media.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

The icon language (Tatomir, 2003) is used to communicate in a language-neutral way. Instead of using text, icons are presented, and the meaning of each icon can be inferred from the icon itself. In general, icons bear a resemblance to the concept that they are representing, such as an icon of a car. Icons are used both to present information to people on the ground, as well as for them to construct sentences to be relayed back to the crisis management experts.

Traffic is rerouted away from the crisis scene without causing traffic jams. This is realized by means of dynamic vehicle routing based on indirect communication between vehicles and their environment (Tatomir *et al*., 2005).

9.2.2 Ad hoc organizations and dynamic planning

Information coming from individuals on the ground can be used in many ways. Clearly, crisis management experts can meet at a central meeting point, and use the distributed perceptions to obtain a clearer idea of what is happening, and what actions need to be taken. However, this information can also be used for other things.

For example, after individuals have been registered, the system looks for emergency professionals with certain abilities, such as medical skills, who are located in an area near to people that require treatment that they can provide. Emergency services can be directed to specific areas based on the information that has been received, and information can be used for collaboration between emergency services.

The formation of organizations to respond to information and needs is managed by the system itself. Such self-managed aspects of the Combined Demonstration System are far more flexible than their static counterparts, as the services provided by the system can change to suit the context of particular problems. Planning and coordinating medical logistics dynamically greatly improves the efficiency of getting services to areas, which is crucial in such crises.

9.3 Agents in the Combined Systems project

While the Combined Systems project is not aimed solely at autonomous agents and multi-agent systems, agents play some of the most central roles in the system, particularly for information logistics and coordination. Traditional software development strategies would not be effective in handling the vast amounts of dynamic data and decision-making in the system. This is the primary reason for choosing the validation scenario discussed in Section 9.2.

9.3.1 Agents in communication

Agents provide three important capabilities in the Combined System communication. First, information agents help to organize data as it is sent in from the people on the ground. Second, network agents help to alleviate some of the problems with the volatile network connection encountered with mobile devices. Third, coordination agents are responsible for coordinating the overall operation.

The agents are responsible for performing initial analysis on the data that is received from individuals on the ground. Upon receiving new information, specialized agents assess this information and aggregate it in order to give crisis management experts a better view of the overall situation. For example, the data can be organized based on the time and location of the individual that sent the data, so the managers can judge if and how fast the gas is spreading. Semantic engines are used to group messages into categories based on their content, as well as to filter out messages with duplicate meanings.

To help alleviate some of the network communication problems, agents monitor network connections and send information at appropriate times. For example, if an individual on the ground enters information into their mobile device, and then attempts to send the information, rather than report to the individual that the information cannot be sent because a connection to the recipient

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

is not available, the agent with the responsibility of sending the information will keep it until a connection becomes available.

Flexibility is important in the system, because in a time of crisis, it is likely that networks will be overloaded, and that the individuals on the ground be distracted by other events and activities, and may not be able to continue checking to see if their device has connectivity.

9.3.2 Agents in organization and planning

Dynamic organization and planning is a major contribution of the Combined Demonstration System. Agent-based mobile devices are applied to automatically construct *ad hoc* wireless networks. Communication between the mobile devices and centralized servers may not be possible at certain times, but local communication may still be a possibility. This will give the advantage that individuals on the ground can browse lists of people that are in their area, and receive information about the type of emergency or specialist service they can provide or receive from each other. Matching individuals with medical and specialist capabilities with individuals requiring medical attention can significantly decrease the time taken to provide first aid, which increases the likelihood of patient survival.

Such organizational awareness is a major step up from coordinating these services centrally, either automatically or manually (Oomes, 2004). Global connectivity is not required, and information can be gathered, collated and sent to individuals at a much faster rate. The inclusion of last-minute information in such planning also allows for more services to deployed in a more efficient manner. Again, semantic engines are used to filter out messages containing the same content.

Performing tasks such as matching medical professionals on the ground with individuals in need is a straightforward task once the necessary information is available. The concept of agents and roles provides an excellent metaphor for specifying, designing and implementing this behaviour, as the system structure can be mapped to the real-world, human-based structure that would otherwise be used to fulfil this behaviour.

9.3.3 Agent and other technologies

Another positive aspect of the implementation of the demonstration system is the use of agent technology within the system. The implementation-level technology that is used is quite standard and largely based on open-source software. While the agent system is highly collaborative and handles dynamic data well, the agents themselves are not 'super-intelligent'. The open-source Cougaar Agent Architecture[19] is being used to develop agents. The five capabilities model, discussed in Section 7.4.1, is used for analysing the capabilities of the agents in the system. In the Combined Demonstration System, important entities in the real world are represented by an agent; for example, a patient in need of medical assistance, or an ambulance travelling to the effected area. Some agents work as a sensors, whether it be via hardware or via human input. Additional agents are also engaged in running the system, such as agents to manage deployment of emergency services.

OpenMap[20] is used for handling geo-spatial information. OpenMap is an open-source toolkit for building applications that require the mapping and handling of geographical information. Combining the agent metaphor with OpenMap allows the system to 'see through the eyes of the victim', thus giving the system a way to view data in a distributed and localized way, and to help management agents understand the local perspective of individuals in the system.

[19] See `http://www.cougaar.org/`.
[20] See `http://openmap.bbn.com/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

9.4 Lessons and experiences

The dynamic nature of the system requires autonomous and flexible entities that can be modelled using the agent metaphor. As with many other large-scale systems, agents make up only a small percentage of the overall system, yet they provide some of the most important parts of the system. In particular, they provide a suitable abstraction for the system's functions and dynamics.

The current state-of-the-art of open-source support for agent-based systems, such as the JADE and Cougaar development platforms, are mature enough for industrial agent development. Although taking time to understand the agent frameworks was valuable, it was costly in terms of time and funds.

It is also clear that the use of current agent standards are somewhat limited in the project. The choice not to use many standards reflects the closed nature of the system. That is, the agents in the system are designed and developed within the project only, and the system does not inter-operate with other agents from other crisis management systems.

9.5 Further information

The Combined Demonstration System has already been demonstrated at several roadshows, and the feedback from the invited guests was positive. At each iteration, the integration of new aspects of the system is working well, a common theme found in the development of agent-based systems.

Further information on the Combined Systems project can be found at the Website (`http://combined.decis.nl/`). In addition to more information on the topics discussed in this section, the page contains information about the other research topics in the project, information about the validation scenarios used in the project, the partners participating in the project and publications available for the public to download.

10 Almende and ASK

This section examines an intelligent communications system (ASK), developed by Almende[21] and marketed by ASK Community Systems[22] (ASK CS), which provides several mechanisms for matching users requiring information or services with potential suppliers. The system offers intelligent and self-learning solutions using dynamic classification, scheduling and feedback mechanisms, and is targeted at intelligent routing, dynamic resource planning and distributed knowledge management applications. Almende and ASK CS work in a synergistic collaboration, with Almende providing the continuing development of the ASK system using feedback given by ASK CS gained through its experience with deploying the system with clients.

10.1 Almende

Almende was founded in 2000 in Rotterdam. Its name translates into English as commons, which is some resource, usually understood to be a piece of land, over which other people—often neighbouring landowners—could exercise one of a number of traditional rights, such as allowing their cattle to graze upon it. Almende uses this metaphor to describe the types of area they tackle, developing tools and products that can help companies and businesses to manage the common resources at their disposal. Their approach focuses on self-organization techniques, adaptive company management and innovation, and their main technology areas include systems for human–agent networks, GAs and distributed systems in general. Almende also collaborates with a wide range of knowledge institutes and universities around the world, and develops products for

[21] See `http://www.almende.com/`.
[22] See `http://www.ask-cs.nl/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

businesses or organizations where communication with customers, suppliers and clients plays a key role.

10.1.1 The application area

Effective communication is vital if businesses and organizations are to operate efficiently and competitively. Getting hold of the right person with the right information at the right time is clearly important if business goals are to be met and problems are to be solved. However, obtaining such effective communication is difficult when people are on the move, are not free to talk or do not have the necessary information to hand. With the increased penetration of mobile communications devices, however, it is possible to streamline the communication process and make it much more efficient than traditional methods in which people simply attempt to call directly and hope that the person they are contacting is available and in a position to provide the necessary information or assistance. By exploiting the ubiquity and power of mobile devices, Almende offers communication solutions that can greatly increase the effectiveness of companies to provide accurate information between its employees and customers.

10.2 The ASK system

The ASK system allows people to contact each other via phone, e-mail or other means based on information stored in a database. The database maintains information about each person in the system and provides contextual knowledge that enables the ASK system to provide the right type of contact between individuals depending on their current availability. For example, if a person is out of the office, ASK might route the call through to a mobile device. Similarly, if several possible people can address the needs of the person attempting to find help, then ASK can select between the candidates based on their current availability. ASK allows many different types of query to be made that can then be used to organize a particular service for a user. For example, calender information can be imported to allow the routing of queries in a way that is sensitive to peoples' activities. ASK also allows users to provide feedback on the service provided so that subsequent service provision can be improved. Furthermore, the system is fully distributable and each component can be duplicated to provide increased redundancy, thus minimizing the impact of failure in any one component.

Currently, the ASK system has been deployed in several companies, such as Pfizer, RaboBank, Vestia and Start, to improve their communications. From September 2005, a new company, ASK Communications Systems, was founded to handle the commercial side of marketing the ASK system.

10.2.1 The main three ASK components

At the highest level, the ASK system consists of three components: a Web interface, a database and the ASK engine. The Web interface provides the front end through which the system can be configured, and online access provided. The database stores the information for each person in the system that is used to determine the best way for contact to be established. The final component, the ASK engine, consists of a set of software agents, and is where the core functionality of the ASK system resides.

10.2.2 The ASK engine

Within the ASK engine are several agents, each of which provides a distinct service for the system. If we consider the system as a pipeline through which requests for contacts pass, then the following agents each carry out their role in the following order.

First, incoming requests are processed and classified by the Reception agent. Once a request is classified it is passed by the Reception to the Matcher agent, which matches the caller with a person

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

that can provide assistance (requester and responder in the terminology of ASK). Matching can be problematic as the preferences of the requester and candidate responders must be taken into consideration. The Matcher tries to find several potential responders and selects between them using one of four possible methods, as follows:

1. Round robin: the Matcher randomly selects a responder from the set of candidates available.
2. Last spoken: the Matcher selects the responder that was selected previously.
3. Rating: the Matcher uses feedback provided by the requester about potential responders and selects the one with the highest rating.
4. Friendly rating: the Matcher again selects based on the received ratings, but occasionally randomly selects a different responder in order to provide them with the opportunity to improve their rating.

Because application domains differ in their requirements, the Matcher can be configured to make selections in different ways. Once a match has been found, the Executor agent is called; this has the role of determining how best the requester and responder can be connected, and involves the use of information stored about the responder, such as location, the skills and information available to the responder, availability for different levels of emergency, and so on. Once the best method of connection is determined, the Resource Manager agent makes the connection between the requester and responder in the manner identified by the Executor. For example, if the communication medium identified by the Executor is, say, for example, then the Resource Manager sends a text message to the responder indicating the nature of the request. Possible communications methods that ASK supports are analogue and ISDN telephony, GSM, VOIP, SMS and e-mail.

The final main component within the ASK engine is the Scheduler agent, which is responsible for the learning capabilities of ASK. Once a connection with a requester and a responder is finished, the Scheduler can contact both to obtain feedback about the interaction. In addition, the Scheduler is responsible for ensuring that there are enough potential responders available. It does this by contacting potential responders when their available numbers fall below a given threshold, in which case a request can be made for them to become available.

10.3 The development of ASK

The development of ASK took place over two years. Initially, ASK was a project sale, in that it began as a pilot project for a telecommunications company to investigate how computational systems could be used to reduce the cognitive load placed on support staff, and to improve the efficiency of Communications between departments. However, subsequent development enabled the commercialization of ASK as a full product. Around six people were involved in its development: one architect, three to four coders and one customer relations manager. A large amount of time was spent developing the sales pitch for ASK, as it was not immediately apparent to customers how the system could add value. Adopting ASK involves a complete change to the way a company works, in terms of how it organizes its internal and customer communications. For this reason, a clear sales strategy had to be developed to enable the benefits of the system to be demonstrated to potential clients.

10.3.1 Agents in ASK

The use of agent technology in ASK helps to provide the necessary autonomy for the various different components of the system. For example, the Matcher can autonomously decide which responder is most suitable for a given requester based on the requirements of the requester and the stored information about the responders. The agents in the system are lightweight and relatively simple, and this is entirely appropriate.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

10.4 Lessons and experiences

Although Almende finds that using the agent metaphor helps to facilitate a client's understanding of what ASK does, pushing the agent aspect too far can make the client wary of the system. This happens because the perception becomes one of an autonomous system that may begin to act in unpredictable ways that the client cannot control. The simple nature of the agents does not hinder the goals of the system designers, and allays the fears of the client that the system may begin to act in ways that are unexpected.

11 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 operationalization, and standards.

11.1 Selling agent technology

One of the important factors in securing a contract for an agent-based application is 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 optimization 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 optimization. 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 organizational 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 visualization 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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

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.

There is also a problem with overemphasizing agent technology and its role in development, and a suitable balance must be found. For example, Whitestein noted that pushing agent technology too much in relation to other approaches led clients to infer that much of the system was experimental, which could lead to people becoming wary. Almende noted that emphasizing the automony of agent systems led its client to infer that the system would behave in unpredictable ways that the client could not control.

11.2 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 optimization 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 formalizing 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 model 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 to both software developers and customers. For business performance optimization systems, for example, data acquisition often requires significant effort in companies where data are 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 optimization. Such is the case of transaction-based applications (e.g. as experienced by Eurobios at SCA Packaging), 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 such as the time of generating the data, nor what-if analysis information, typically used in the modelling and optimization task.

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

1. The process of making explicit that knowledge which is complex, unformalized 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.
2. Clarifying domain terminology to the software developers was often necessary and was clearly important, especially in the early stages.
3. 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.
4. 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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

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.

11.3 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 optimization 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:

1. 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 OTB).
2. The horizontal integration approach, by which every agent interfaces with company databases, was used by Acklin. In order to minimize 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.

11.4 Validation

Validation is crucial to ensure the accuracy, correctness and realism of any model and of its outputs. Because of 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 (Buchanan, 2005). 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 (e.g. 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

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

experienced by Eurobios. On the one hand, spending considerable time in advance to ensure sufficient data accuracy, and designing the application to ensure re-usability, quality and robustness was important. On the other hand, 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, first because they had to address the problem of integration with the CGF system, and second 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.

11.5 Usability and operationalization

Training and familiarization 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 optimization or a simulation is essential, especially as decision-makers have a great deal of experience in their area. A period of familiarization 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 operationalization. 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 optimization 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 in terms of both writing accurate tactics for the model and tracing and understanding overall system behaviour. This was because of 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.

11.6 Agent standards

While agent standards, such as IEEE FIPA[23], are developing and becoming more mature, it is interesting to note that in many of the projects outlined in this paper, 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 organizations, they are not open in the sense that agents developed outside of these organizations can participate in the system; even more so, with these organizations developing their own interaction languages and protocols.

[23] See `http://www.fipa.org/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

Even in the development of the CWS, in which Rockwell used FIPA standards for inter-agent communication, during the early stages of development, the standards were not used sparingly due to the high overhead associated with them.

These points signal a situation in which 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.

12 Conclusion

This paper has presented and discussed a series of case studies aimed at helping agent software technologies cross the chasm phase of new technology adoption, where many people currently see the technology being. In addition to reviewing and discussing relevant aspects of each of these case studies, we have explored the key lessons and experiences of developing agent-based applications for a variety of domains, in the logistics, production, defence and insurance industries. The experiences and challenges faced are largely determined by the nature of the business operations in the industry domain concerned, the characteristics of the problem to be solved, and issues related to the organizational and technical infrastructure of the adopting 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 summarized as follows:

1. From a business perspective, technology adoption decisions should be strategic, based on perceived long-term benefits and supported by a clear business evaluation. A software developer wishing to sell agent technology to a new company must propose a strong business case, identifying the commercial and technical value of an agent-based solution for that company, and the relative advantages of agent technology over other technologies already in use in the relevant industry. This must be done without underestimating the likely costs—financial, technical and organizational—of an agent-based solution.
2. From a technical perspective, while the specific agent solutions considered here address specific performance requirements, such as dynamic adaptivity and real-time response, these solutions also had to satisfy requirements common to any software implementation for industrial applications, such as accuracy of results, inter-operability, robustness and user-friendliness. It is therefore 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, we believe from undertaking these case studies that the successful adoption of agent software technologies depends crucially on first identifying the specific conceptual and technical aspects of the application domain to which software agents and multi-agent systems best lend themselves. It may be the case that the solutions required are relatively unsophisticated, technically, or that unsophisticated alternatives to agent technologies could be deployed in these domains. The agent paradigm is of most value in such cases, in our view, when the specific features of the domain or of the application support an agent-based approach. These tend to be domains where there are multiple stakeholders or decision-makers, or where there are potentially conflicting information, interests or goals. It is in such domains where the encapsulation of execution control provided by agent technologies will be of greatest value to a software development team.

Acknowledgements

The views expressed in this paper are those of the authors, and are based on interviews conducted with relevant experts from each of the companies mentioned: Tony Delaney and Martyn Fletcher from AOS; Vince Darley from Eurobios; Nadezhda Yakounina from Magenta Technology;

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

Christian Danneger and Klaus Dorer from Whitestein Technologies; Bryan Woody and David Davis from NuTech; Chris van Aart from Acklin and Y'All; Paul Burghardt and Hakan Elgin from Thales Research & Technology; and Pavel Tichý, Vladimír Mařík and Pavel Vrba from Rockwell Automation Research Centre. 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 most grateful to all those who contributed. The full set of case studies is available from the AgentLink III Website (`http://www.agentlink.org/`).

This paper is a revised and extended version of a paper originally presented at the Industrial Track of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006), held in Hakodate, Japan, in May 2006 (Belecheanu *et al*., 2006). We are grateful for financial support received from the European Commission under the Information Society Technologies programme, through projects AgentLink III (IST-FP6-002006 CA) and PIPS (IST-FP6-507019).

References

Baxter, J and Hepplewhite, R, 1999, Agents in tank battle simulations. *Communications of the ACM* **42**(3), 74–75.

Bazzan, ALC, 2005, A distributed approach for coordination of traffic signal agents. *Journal of Autonomous Agents and Multiagent Systems* **10**(2), 131–164.

Belecheanu, R, Munroe, S, Luck, M, Payne, T, Miller, T, Pechoucek, M and McBurney, P, 2006, Commercial applications of agents: lessons, experiences and challenges. In *Industrial Track, Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006), Hakodare, Japan, May 2006*. New York: ACM Press, pp. 1549–1555.

Buchanan, M, Spring, 2005, Special Report: supermodels to the rescue. *Strategy+Business*.

Darley, V and Sanders, D, 2004, An agent-based model of a corrugated-box factory: The trade-off between finished goods stock and on-time-in-full delivery. In Coelho, H and Espinasse, B (eds.), *Proceedings of the 5th Workshop on Agent-Based Simulation, Lisboa, Portugal, 3–5 May 2004*.

Dorer, K and Calisti, M, 2005, An adaptive solution to dynamic transport optimization. In Pechoucek, M, Steiner, D and Thompson, S (eds.), *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track, Utrecht, the Netherlands, July 2005*. New York: ACM Press, pp. 45–51.

Du, T, Li, E and Chang, A-P, 2003, Mobile agents in distributed network management. *Communications of the ACM* **46**(7), 127–132.

FIPA. 3 December 2002 Communicative Act Library Specification. Standard SC00037J, Foundation for Intelligent Physical Agents.

Hill, R, Chen, J, Gratch, J, Rosenbloom, P and Tambe, M, 1997, Intelligent agents for the synthetic battlefield. In *Joint Proceedings of the 14th National Conference on Artificial Intelligence and the 9th Conference on Innovative Applications of Artificial Intelligence, Providence, RI*, pp. 1006–1012.

Himoff, J, Skobelev, P and Wooldridge, M, 2005, MAGENTA technology: Multi-agent systems for industrial logistics. In Pechoucek, M, Steiner, D and Thompson, S (eds.), *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track, Utrecht, the Netherlands, July 2005*. New York: ACM Press, pp. 60–66.

Jacobi, S, Madrigal-Mora, C, Leon-Soto, E and Fischer, K, 2005, Agentsteel: an agent-based online system for the planning and observation of steel production. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems, Utrecht, the Netherlands, July 2005*. New York: ACM Press, pp. 114–119.

Levitt, T, 1965, Exploit the product life cycle. *Harvard Business Review* **43**(6), 81–94.

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

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

Miletić, F and DeWilde, P, 2004, Distributed coding in multi-agent systems. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Den Haag, the Netherlands, October 2004*, vol. 6. pp. 5929–5934.

Moore, G, 1991, *Crossing the Chasm*. New York: HarperCollins.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press

Nicolaisen, J, Petrov, V and Tesfatsion, L, 2001, Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation* **5**(5), 504–523.

Oomes, A, 2004, Organisation awareness in crisis management. In *Proceedings of the International Workshop on Information Systems for Crisis Response and Management (ISCRAM), Brussels, Belgium, April 2004*, pp. 63–68.

Rönnquist, R, Lucas, A and Howden, N, 2000, The Simulation Agent Infrastructure (SAI)—incorporating intelligent agents into the CAEN Close Action Simulator. In *Proceedings of the SimTecT Conference, Sydney, Australia, 28 February–2 March 2000*.

SCA. 2004 SCA environmental and social report. `http://www.sca.com/Pdf/env-report04gb.pdf`.

Schraagen, J, Eikelboom, A, van Dongen, K and te Brake, G, 2005, Experimental evaluation of a critical thinking tool to support decision making in crisis situations. In *Proceedings of the 2nd International Conference on Information Systems for Crisis Response and Management, Brussels, Belgium, 18–20 April 2005*.

Shakir, A, 2002, Assessment of models of human decision-making for air combat analysis. Technical Report DSTL/PUB03152/1.0, Defence Science and Technology Laboratory, Farnborough, UK.

Staron, RJ, Maturana, FP, Tichý, P and Šlechta, P, 2004, Use of an agent type library for the design and implementation of highly flexible control systems. In *Proceedings of the 8th World Multiconference on Systemics, Cybernetics and Informatics*, pp. 18–21.

Tatomir, B, Fitrianie, S, Paltanea, M and Rothkrantz, L, 2005, Dynamic routing in traffic networks and MANETs using ant based algorithms. In *Proceedings of the 7th International Conference on Artificial Evolution, Lille, France.*

Tatomir, I, December 2003 Iconic communication. BSc thesis, Delft University of Technology, Delft, The Netherlands.

Tichý, P, 2003, Social knowledge in multi-agent systems. PhD thesis.

Tichý, P, Šlechta, P, Maturana, FP, Staron, RJ, Hall, K, Mařík, V and Discenzo, FM, 2004, 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*.

van Aart, CJ, van Marcke, K, Pels, RF and Smulders, JLFC, 2002, International insurance traffic with software agents. In van Harmelen, F (ed.), *Proceedings of the 15th European Conference on Artificial Intelligence*. Amsterdam: IOS Press, pp. 623–627.

Vrba, P and Mařík, V, 2005, Simulation in agent-based manufacturing control systems. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1718–1723.

Wagner, T, Gasser, L and Luck, M, 2005, Impact for agents. In Pechoucek, M, Steiner, D and Thompson, S (eds.), *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems: Industry Track*. New York: ACM Press, pp. 93–99.

Wooldridge, M, 2004, Responding to real-world complexity: Introduction to multi-agent technology. Magenta white paper. Available from `http://www.magenta-technology.com/resources/downloads/`.

https://doi.org/10.1017/S0269888906001020 Published online by Cambridge University Press