Software Agents in International Traffic Insurance

Acklin and Interpolis
A Business Case Study

1. Introduction

This case study presents an agent-based application for the insurance industry. The process of handling insurance claims is heavily bureaucratic, with most of the information processing done by humans, resulting in long times for settlement. Many European insurance companies use heterogeneous information systems, with different means of storing and using data, hence data must be exchanged manually between companies when dealing with claims. Acklin B.V. 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 article gives an overview of the system, the development process, the challenges encountered and how they were overcome, and the main lessons learned.

2. Insurance Claim Handling for Green Card Traffic: Interpolis

Interpolis is a major Dutch insurance company, operating also in Ireland, Portugal and Luxembourg. The company has approximately 6,000 employees, more than 1 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 which covers the claim-

ant, 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 organisations. 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.

3. Acklin

Acklin BV, a Netherlands software company specialising in agent technology and its application to business process optimisation, identified this marked need in the insurance sector and attracted the interest of 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 optimisation solutions delivered for transportation and retail companies, travel agencies, local government, etc., Acklin convinced Interpolis that agent technology also had potential to solve the problems specific to this industry sector.

4. 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 re-start 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:

1. The first option was to build a central data store residing in Brussels, to which every insurance company can 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 synchronisation between different companies' back-office systems in order to pull data in a common format into the central data store. However, the main drawback was that companies may use the data store to access each others' information on different policies and then use the information to attract each other's 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 other's 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 the disadvantage that the final results of processing claims would still have to be transferred manually between back-offices. Additionally, 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 one chosen, was to develop a network of communicating agents to which the back-office of every insurance company can 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 email 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 email solution were that email was already in place, and no new usage and maintenance procedures needed to be introduced, except those related to the use of email by agents instead of people (for example, virtual email addresses had to be created by the email administrators). In addition, due to the large organisational 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 (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.

5. The KIR System

The KIR System was developed by Acklin for Interpolis and two of its collaborating companies: a Belgian (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:

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, and
3. an interface was designed to allow agents to have controlled access to the back-office.

5.1 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 specialised functions and knowledge.

- The *communication model* handles all the interactions between agents and other systems, with methods for message transportation, representation and interpretation.
- The *competence model* contains methods for the execution of the tasks for which the

agent is designed.

- The *self model* represents the knowledge that an agent has about its own capabilities (i.e. goals, tasks, jobs, states and competences).
- The *planner model* contains strategies that can be used to achieve an agent's goals.
- The *environment model* gives the agent a view on the world in which it operates (i.e. the other agents) and with which it interacts.

The implementation of each model depends on the type of agent; for example, for agent roles like payer and handler, the differences in capabilities are implemented only in the *competence* and *self* models.

The models allow the definition of a standardised interaction between agents via certain message exchange patterns. The *Incoming-Questions-Flow* illustrated in Figure 1 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 authorised to ask that question. If the sender is authorised, the self model checks if it recognises 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.

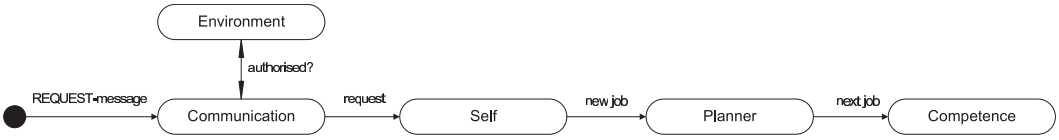

Figure 1: The *Incoming-Questions-Flow* describing the communication between the 5 capabilities model when receiving a REQUEST messages (Source: [1])

5.2 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 [2]). Figure 2 below 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 license plate, and
2. case identification, using policy number and local claim number.

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 re-

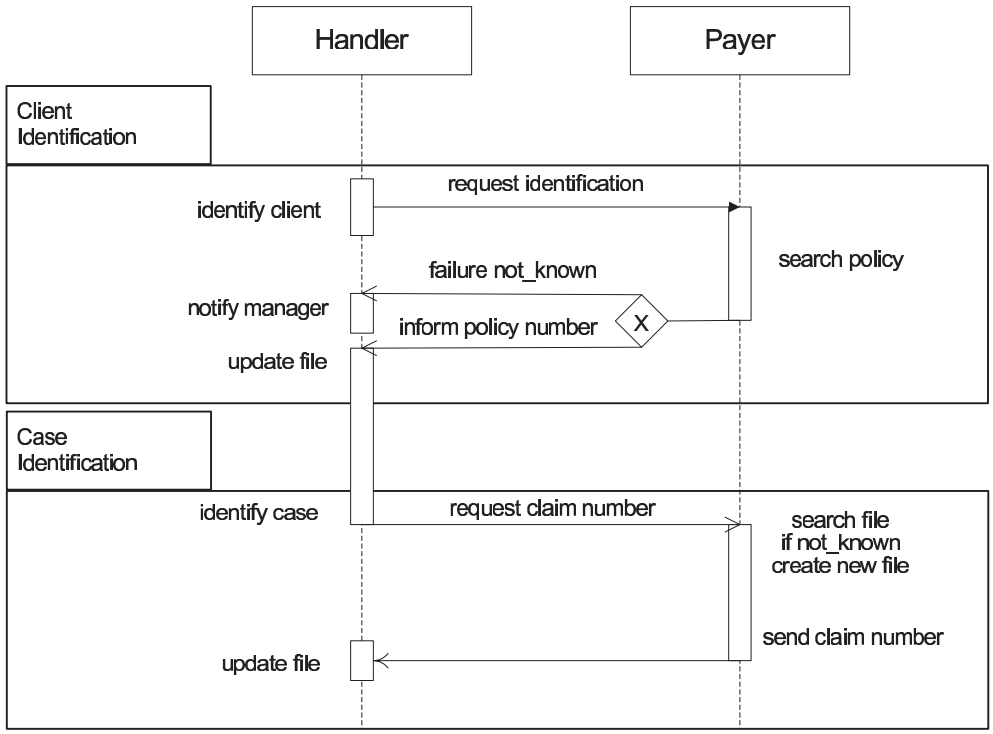

Figure 2: The *Green card traffic* illustrated in an UML activity diagram (Source: [1])

quest 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 instead 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 license 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 happen 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 email 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.

5.3 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 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 license 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.

6. 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 centralised 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 organisational functions, and approval had to be obtained from process managers and from the IT department of Interpolis. The overall decision process required ap-

proximately 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 specialising 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 due to 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 customised to the specific technical requirements of the contracting insurance company, and coupled via a specific interface to the back-office of that company. This model allows Acklin to sell customisation and maintenance as part of a consultancy services package after selling the core system. Customisation 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 messages 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. Lessons and Experiences

1. The agent metaphor was found to be a useful and natural means to model the insurance business. By mapping organisational functions onto agents using roles and capabilities, and by mapping business logic and process flows onto message exchange patterns between agents, it was easy to reason about and better understand company processes. 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.

2. The success of the system was even more important given the traditional difficulties to overcome when introducing new technology simultaneously in several large organisations running legacy IT infrastructures. The difficulties were not only related to technology incompatibility, but also to the organisational model of the insurance companies: since 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 (for example, increased maintenance work for database administrators).

3. 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. Summary

We described an agent-based application to automate the exchange of information between insurance claim handlers. The system was developed by Acklin B.V. for Interpolis, a Dutch insurance company, and was designed using the 5-Capabilities agent model and FIPA inter-agent communication protocols. The introduction of the system at Interpolis and two other insurance companies reduced the process of identification of client and claim from 6 months to less then 2 minutes. This success, combined with the strength of the agent paradigm and the simplicity of the application design, led Interpolis to commission a European-wide implementation of the system.

References

[1] van Aart, C. J., Van Marcke, K., Pels, R. F. and Smulders, J. L. F. C, International insurance traffic with software agents. In F. van Harmelen (Editor), Proceedings of the 15th European Conference on Artificial Intelligence, IOS Press, Amsterdam, 2002.

[2] FIPA, Communicative Act Library Specification. Standard SC00037J. Foundation for Intelligent Physical Agents (www.fipa.org), 2002. www.fipa.org

Roxana Belecheanu and Steve Munroe
School of Electronics and Computer Science
University of Southampton
Southampton SO17 1BJ
United Kingdom
{rab2, sjm}@ecs.soton.ac.uk

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

Chris van Aart
Y'All BV
Grotestraat 182
5141 HD Waalwijk
The Netherlands
chris@yall.nl

www.acklin.nl
www.interpolis.com

www.y-all.nl