# Use Case 1 in Service Industry – Business Cases
# Agent-based System for an Insurance Company

**KW Partner:** UPM

# 1 Overview

**Challenge**
*The traditional way of handling insurance claims involves different experts in different departments using different approaches, making the process very costly.*
**Solution**
*Chain integration for insurance claim handling can make the entire process more economical*
**Why a Semantic solution**
*Ontologies are required to formalize the vocabularies of companies from different domains such as insurance and car repair.*
**Key Business Benefits**
*Reduce errors in the interactions between companies due to inconsistency in their databases. Ease the addition of more participating companies in the interaction chain.*
**Business Partners**
*Insurance agencies*
*Damage repair companies*

**Keys components**

Existing Software
*Local databases*
*Local data input, processing and distribution*
Research and Development
*Agent-based systems*
*Ontology development*
*Ontology mapping*
Technology locks
*Gap between agent technology and ontology technology*
*Semantic interoperability issues including ontological translations*
*Working with (for industry fairly new and therefore not industry-strength supported) languages such as RDF and OWL*

For one of the largest insurance companies in the Netherlands, Acklin BV has built an agent-based system to handle all communication related to car damages with a damage repair company.

Insurance claims handling involves a costly process where each aspect of a claim is examined by different experts in different departments using different approaches. One of the processes within the claim handling process is repairing car damages.

The agent-based system requires an extension with ontology technology as a means to optimize interactions between two companies that have different processes and employ technologies of various degrees of sophistication. In designing and implementing the system, some requirements could be identified that raise the need for such a semantic solution.

# 2 Current Practices and Technologies

## *2.1 Typical business practices*

The insurance market hugely relies on a traditional way of claims handling. Every aspect of a claim will often be dealt with by a different specialized person working in a different department of the company. The input, processing and distribution of data are treated by each part of the organization in their own traditional way, making the process being very costly. Nowadays, the insurance market is looking more and more for ways to economize the process of claims handling. Because the process of claims handling involves many different parties such as the victim(s), witnesses, surveyors, lawyers, insurance companies, middlemen and doctors there is a growing need for chain integration.

Acklin solved the business case using agent technology. The system is composed of two agents. One at the insurance company (the insurance-agent) and one at the damage repair company (the repair-agent). The insurance-agent sends data about insurance policies and car information to the repair-agent. The repair-agent on its return sends information related to performed jobs and invoices to the insurance-agent. When a car needs to be repaired, the repair-agent has to ask for permission at the insurance-agent. The agents communicate with each other using a peer-to-peer technique.

## *2.2 System requirements Analysis*

When designing and implementing the system several (unexpected) issues delayed the progress of the project. The most interesting issue was the use of different vocabularies by the two companies, especially the use of the word “cause”. The repair company used the word “cause” to indicate the reason of car damage e.g. (freely translated) “car has been hit by another car”, “car ran into an obstacle” or “car broke down”. The insurance company used the word “cause” to indicate why they had to pay for the repair of a damage, e.g. (freely translated) “repair a part of the car” or “replace a part of the car”. The approach taken to make a translation between the different vocabularies was to write down three tables: one with the vocabulary of the insurance company, one with the vocabulary of the damage repair company and one translation table. The insurance company complained about faults in the table of the repair company and vice versa. After several meetings, the companies started to understand that their view on the world was not unique. From there, they also learned the concept of ontology and the technique of ontology translation.

However, there seems to be a gap between agent technology and ontology technology. Engineering intelligent agents typically involves dealing with distributed environments, including complex message passing in interactions and autonomy. Although, some pieces of ontology technology enable simple client-server architecture, most ontology technology seems to be engineered around a centralistic concept having central databases and one major ontology server. In this case, new pieces of technology have to be built in order to have agents working with ontologies, distributed databases and concept translations.

Connecting two or more systems to each other will always raise interoperability issues. Although agreeing on using the same transport protocol (e.g. FTP, HTTP) and content language (e.g. fixed width, XML) is hard, a lot of companies are not aware of semantic interoperability issues. Furthermore, companies are not (yet) aware of techniques and methods available for handing semantic interoperability issues, such as translations. Companies do understand (sometimes after a while) that they use different vocabularies in their processes and databases. The technical languages used in these companies are database schemas in SQL and XML. Therefore, most ontology techniques and methods are unavailable for these companies, because ontology research tends to focus on working with (for industry fairly new and therefore not industry-strength supported) languages such as RDF and OWL. Of course these languages are needed, however the methods to reason with knowledge stored in these language should also be needed in the less expressive languages, such as SQL and proprietary XML, because today these are the languages most used.

## *2.3 Review of the current systems*

AcklinQ is a commercial product aimed at supporting agent-based cross-organization, technology and process information logistics.
For more information see http://www.acklin.nl/products/acklinq/acklinq.pdf