An architecture for agent-based mobile Supply Chain Event Management

Anna Maria Jankowska*,
Karl Kurbel and Denny Schreber

Department of Business Informatics,
European University Viadrina,
D-15207 Frankfurt (Oder), Germany
E-mail: jankowska@uni-ffo.de
E-mail: kurbel.bi@uni-ffo.de
E-mail: schreber@uni-ffo.de
*Corresponding author

Abstract: Supply Chain Event Management (SCEM) is an approach aiming to fill the gap between Supply Chain Planning and Supply Chain Execution. The goal of SCEM applications is to monitor the states of supply chains by observing specific events and exceptions in real-time and alerting managers if problems occur. This paper presents an architecture for a mobile SCEM system based on software agents, Auto-ID technologies, and mobile computing. It introduces the main layers and components of this architecture. A special focus is placed on web services, mobile user interfaces, and integration of a multi-agent platform as innovative solutions to improve SCEM.

Keywords: mobile supply chain event management; SCEM; multi-agent systems; MASs; service-oriented architecture; SOA.

Reference to this paper should be made as follows: Jankowska, A.M., Kurbel, K. and Schreber, D. (2007) 'An architecture for agent-based mobile Supply Chain Event Management', *Int. J. Mobile Communications*, Vol. 5, No. 3, pp.243–258.

Biographical notes: Anna Maria Jankowska is a Research and Teaching Assistant at the Department of Business Informatics at the European University Viadrina in Frankfurt (Oder), Germany. Her research interests include mobile technologies and applications, software architectures, EAI and semantic web services.

Karl Kurbel is a Professor and Head of the Business Informatics Department at the European University Viadrina. He has authored several books and numerous papers on production planning and mobile business. His current research areas are mobile business, supply chain management, software agents, multimedia-based virtual education, and knowledge portals.

Denny Schreber is a Research Assistant at the Department of Business Informatics at the European University Viadrina in Frankfurt (Oder), Germany. His research interest is supply chain management, with emphasis on Supply Chain Event Management (SCEM) using agents and mobile technologies.

1 Introduction and problem description

The tendency to outsource non-core business processes, increasing complexity of supply chains, and the attendant lack of transparency led to the emergence of the Supply Chain Event Management (SCEM) concept. Conventional SCM systems are primarily focused on planning and are not flexible enough at the execution level to adequately respond to changes caused by supply chain disruptions. SCEM software is intended to track, monitor, and measure the condition of a supply chain by observing specific events and notifying process owners (Nissen, 2002).

In SCEM, two types of events can be distinguished: planned and unplanned events. Planned events (e.g., delivery of goods) are basic steps to measure SCM operations. Unplanned events (exceptions) disrupt established supply chain plans and their execution. They require real-time responses and appropriate adjustments of plans. Although exceptions account for only 10% of business operations, they require up to 90% of the workforce's time and can have serious economic implications (Bretzke, 2002). For example, in 2000 a fire damaged the plant of a supplier of key components for Nokia's and Ericsson's mobile phones (Lee, 2004). Nokia responded immediately with a change in the design of its phones that allowed for the use of chips from other suppliers, while Ericsson scaled back production. As a consequence Nokia gained a significant part of Ericsson's market share and Ericsson had to face a delay in the launch of a major new product.

The main purpose of our research was to contribute to the improvement of existing SCEM solutions by identifying the potential of emerging mobile computing technologies. In order to examine the benefits and drawbacks resulting from the application of those technologies to SCEM, we followed the 'construction and building' approach, as described by (Backlund, 2005). A prototypical SCEM system was developed to demonstrate the feasibility of our architectural approach. The aim of the new solution was to enable semi-automatic problem solving in SCEM, increase supply chain visibility, enable intra-organisational information exchange, and significantly decrease response times to supply chain disruptions. Intelligent agents and Multi-Agent Systems (MASs), Automatic Identification (Auto-ID) technologies, web services, and mobile computing were selected as technologies with the greatest potential for SCEM.

Intelligent agents can improve SCEM because they are able to coordinate the activities of loosely coupled, distributed entities (e.g., participants in a supply chain network). They can act independently and without complete a priori event knowledge (Davenport and Brooks, 2004). In addition, they are capable of interacting with the environment, adjusting the flow of goods, and modifying production schedules to increase an enterprise's agility.

Auto-ID technologies, particularly Radio Frequency Identification (RFID) tags, can significantly enhance supply chain visibility (Hanebeck and Tracey, 2003). Visibility in a supply chain means the availability of crucial information to all partners in the chain, for example, regarding stock levels, current orders, and quantities in production and transit. In SCEM such information is necessary to trigger events users should react to. The RFID tags deliver accurate, specific, and timely product information while simultaneously reducing the data collection workload. A variety of applications (e.g., tracking and tracing systems) can then use the collected data to track the objects from their origin to the destination and to elicit potential problems.

SCEM systems can be further enhanced by mobile computing and Service-Oriented Architectures (SOAs) (Sharma et al., 2003). Wireless devices enable access to critical business functionalities around the clock from any location, decreasing response time to disruptions and improving agility. A SOA provides the crucial functionalities as discoverable, network-addressable, platform-independent services. It guarantees better interaction possibilities among partners and enhances supply network visibility.

The focus of this paper is an architectural approach for agent-based mobile SCEM. The paper is organised as follows: In the next section, the state-of-the-art of architectures for mobile SCEM is presented. Section 3 outlines the concepts of software architecture and technologies that can be used in agent-based SCEM systems with mobile access. In Section 4, a multi-layer architecture for agent-based mobile SCEM is introduced. Crucial components of this architecture such as web services, the Agent Platform (AP), Auto-ID technologies, and mobile user interfaces are discussed. In the final section, some conclusions are drawn and issues for further research and development are outlined.

2 State-of-the-art in mobile SCEM

Several approaches deal with the management of disruptions in supply networks. In the software-agent domain, research is focused mainly on tracking and tracing of goods and resources in supply networks. Existing approaches are, for example, ECTL-Monitor (Hofmann et al., 1999), PAMAS (Bodendorf and Zimmermann, 2005), and Dialog (Kärkkäinen et al., 2003). Visibility can be improved through permanent resource tracking in the whole network with the help of Auto-ID and mobile technologies. This is the case, for example, in the Dialog system. However, these systems do not support automatic or even semi-automatic problem solving with software agents.

Agent-based approaches for automatic problem solving are PROVE (Szirbik et al., 2002), CoagenS (Dangelmeier et al., 2004) and Agent.Enterprise (Frey et al., 2003). In the latter approach, several MAS interact in production planning, supply chain planning, supply chain controlling, and tracking and tracing. Another interesting approach for supply chain controlling with problem-solving capabilities using web services to interoperate with various supply network members is presented by Speyerer and Zeller (2004).

In practice, current approaches like Oracle’s Business Activity Monitoring (Oracle, 2005) support mobile monitoring of resources, but often only on the intra-enterprise level. The problem of integrating all partners in a supply network, i.e., inter-enterprise monitoring, is still to be realised. Open standards have to be developed so that partners can collaborate and exchange monitoring data. One way to approach SCEM is to combine existing components. In the SAP world, for example, the SAP NetWeaver Platform, the SAP Event Manager, the SAP Auto-ID Infrastructure, and the SAP Mobile Infrastructure are available. Using these components, organisations can monitor resources in their supply networks and integrate mobile technologies as well (Blanchard, 2004).

While current approaches in both research and practice support tracking, tracing, and notification, more research effort is necessary to automate the investigation and solution of these problems – particularly through the application of software agents in today’s complex and distributed supply networks. Furthermore, there is a need for solutions that would facilitate information exchange between enterprises and improve the integration of supply network partners.

3 Concepts and technologies for a mobile SCEM architecture based on MAS

3.1 Information systems architectures

Although there is no common agreement on the definition of the term, most authors describe software architecture as a kind of blueprint. In addition to specifying the organisation of software elements and their interrelationships, this blueprint documents the principles governing the architecture's design and evolution (Dickmann, 1995; Risi and Rossi, 2004).

Architectures are strictly associated with the notion of layers. Current application systems are often composed of three major layers: the presentation layer, the application logic layer, and the data (services) layer (Britton, 2000). The presentation layer provides the user interface and is responsible for the interaction between the user and the front-end device. The application or business logic layer contains the business rules that drive the given enterprise. The services layer offers general services needed by the other layers such as database services, file services, or communication services. The functionalities of these three layers can be assigned to logical entities called tiers. Most current systems are designed with three-tier or multi-tier architectures.

In the last few years, component orientation has become an increasingly popular paradigm. In component-based architectures, software systems are assembled from building blocks called software components. Due to the decreasing cost of reliable, high-bandwidth internet connections, and the shortcomings of technologies such as DCOM and CORBA, component-based architectures have evolved towards SOAs, implemented via web services.

A SOA focuses on configuring entities (services, registries, contracts, and proxies) in a way that maximises the loose coupling and reuse of software components (McGovern et al., 2003). In a SOA, software functionality is represented by discoverable services on a network. A service is defined as behaviour that is provided by a component according to an interface that governs how other components can interact with it.

An important requirement for SCEM is visibility regarding the operations of the partners and the movement of goods between the partners. Dissimilar applications running on heterogeneous platforms, legacy systems lacking common interfaces, security issues, and unwillingness to disclose underlying systems functionalities slow down the process of improving supply chain visibility. The migration to web services and SOAs helps enterprises to implement real-time collaborative SCM and to improve their productivity, particularly in the area of Enterprise Resource Planning (ERP) and supply chain execution.

Web services are considered as a technology promoting interoperability and ad-hoc integration of applications, while software agents are seen as the key technology to discover and execute web services and to combine them into processes. ERP systems integrated with SCM systems can deliver static data required by the agents (e.g., product details, business partners, etc.) and provide transactional data to monitor and control the

SCM processes. Agents can find solutions to exceptional situations in supply chains if they have access to the core ERP functionalities.

Integrating an AP and an ERP system is difficult because ERP system Application Programming Interfaces (APIs) are usually proprietary if available at all. A web services Façade offering platform-independent access to the ERP system's functions can facilitate the integration. If agents are able to access different ERP systems, they can search for solutions to supply chain problems. For example, if a supplier fails to provide certain goods then the manufacturer's agents can search for alternative partners that expose their systems' functionalities as web services. If an agent finds a potential partner, it may initiate a negotiation process with that supplier's selling agent to expeditiously resolve the shortage.

3.2 Multi-Agent Systems (MASs)

Agents are autonomous, problem-solving computational entities that can effectively operate in dynamic and open environments (Luck et al., 2004). Agent-based systems containing several interacting agents are called MAS. The standards for agents and agent-based systems are issued by the Foundation for Intelligent Physical Agents (FIPA) (FIPA, 2005). According to the agent management reference model developed by FIPA, the following components are crucial to MAS frameworks (cf. (FIPA, 2004) for an extended overview).

- *Agents*. Agents provide one or more services and communicate using the Agent Communication Language (ACL). Agents can be distinguished and contacted by their unique identifiers.
- *Agent Platform (AP)*. An AP is the basic agents' infrastructure comprising the following components:
    - *Agent Management System (AMS)*. The AMS regulates access to and use of the AP by allocating/assigning unique IDs to the agents and by offering a white pages service to other agents. Only one AMS can exist for each AP.
    - *Directory Facilitator (DF)*. The DF is an optional component that acts as a yellow-pages service; agents can register their services in the DF and search for available services from other agents.
    - *Message Transport System (MTS)*. The MTS provides methods to allow inter-platform communication.

A specific architecture depends on the chosen framework. One of the well-established FIPA-compliant frameworks is JADE (Java Agent Development Framework). JADE builds upon standard Java technologies. MAS have to be integrated into existing architectures, e.g., into a SOA, to enable access to services provided by agents from other components and to offer agents the possibility to use the functionality available in different systems.

The advantage of using an existing AP is that the developers can focus on the main tasks of agent design and agent service implementation, since lower-level features like agent communication and migration have already been solved by the AP.

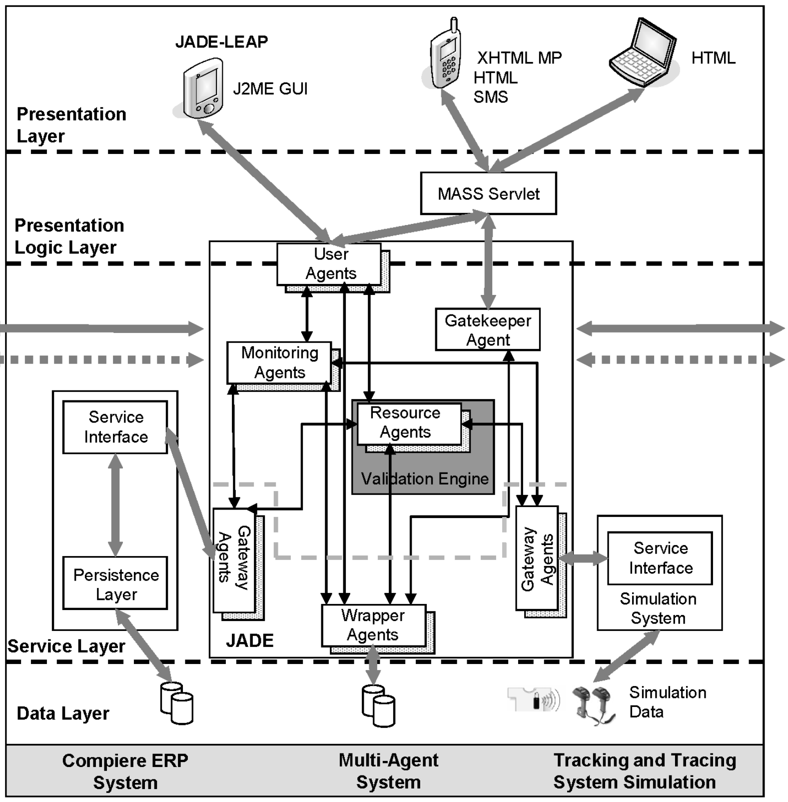
3.3 Sensor-based tracking and tracing systems

Modern sensor-based tracking and tracing systems in the form of EPC networks are able to identify goods and to obtain information about their status (e.g., location, state). They can help to increase supply chain visibility (Floerkemeier, 2005). Auto-ID technology is a precondition for real-time SCEM. RFID is the most common Auto-ID technology in use today. The chip on an RFID tag contains at least an Electronic Product Code (EPC) that serves as a unique identifier like a barcode.

The EPC is usually transmitted to a server. The server (e.g., Savant) filters and bundles data from readers to query the Object Naming Service (ONS). The ONS returns a Uniform Resource Locator (URL) that references an EPC Information Service. The EPC Information Service delivers product-related data (e.g., product name, delivery source and destination) from the objects via Physical Markup Language (PML) messages. PML messages can also include dynamic data provided by the underlying sensor technology, such as environmental changes (e.g., thermal fluctuations).

Due to the globally unique identification provided by the EPC, agents are able to monitor and control the flow of goods in the whole supply network. In our architecture (cf. Figure 1), agents query tracking and tracing systems by means of web services. If the tracking and tracing systems support basic analysis of the tracked data, an agent can be notified in critical situations. A web service on the agent's side that initiates or informs a responsible monitoring agent can provide this functionality.

Figure 1 Architecture for mobile SCEM



3.4 Mobile and web-based user interfaces

Today's corporate communication and information infrastructures are expected to offer users round the clock access to enterprise information systems from any location so that, for example, decision makers can react quickly to environmental changes. Mobile computing technology in connection with agent-based frameworks offers an excellent opportunity to design and develop systems that fulfil these requirements.

The capabilities of mobile devices are constrained by relatively small user interfaces, limited storage, and low processing power. The user cannot be in the 'always connected' mode due to limited battery life and disruptions in wireless networks (Varshney et al., 2004). Therefore, APs running on mobile devices have to be lightweight, hosting only a small amount of the business logic. Alternatively, the business logic may remain on a separate application server, while the mobile device is only responsible for the presentation.

Existing APs for mobile devices may be classified into three categories:

- portal platforms
- embedded platforms
- surrogate platforms (Carabelea and Boissier, 2003).

In portal platforms, agents are running on an application server and the mobile device provides only an interface to the server. In the case of embedded platforms, agents are executed on mobile devices, but their functionalities are usually limited in scope compared to APs designed for regular computers. Surrogate platforms split the AP between a server and the mobile device.

The MobiAgent (Mahmoud, 2001) platform is an example of a portal platform. It consists of a mobile device with a configuration midlet, an Agent Gateway, and the network resources. The user can configure an agent by setting appropriate parameters in the J2ME application, a mobile device sends these parameters via HTTP to the gateway, and the gateway performs the requested task by executing agents. The results are then sent back to the midlet through the same mechanism.

kSaci (Albuquerque et al., 2001) is an implementation of a surrogate platform for mobile devices running the kVM (Topley, 2002). The mobile device is equipped with an agent that possesses a mailbox to exchange messages. This agent is, however, unable to communicate directly with other agents; it must route HTTP requests to a proxy running on a computer. This proxy communicates with the agents and returns the results to the agent located on a mobile device.

A very popular embedded platform for mobile devices is JADE-LEAP (LEAP = Lightweight Extensible Agent Platform) (Adorni et al., 2001). JADE-LEAP is a FIPA platform that can be used to deploy systems spread across a heterogeneous network of mobile and wired devices. JADE agents can run on small devices without any modifications, providing the devices offer sufficient resources and processing power. LEAP runs on desktop PCs, PDAs, and Java-enabled mobile phones. It is able to use wireless networks such as TCP/IP over GSM and IEEE 802.11 Wireless LAN, and it can be deployed on several operating systems.

LEAP comes in two different flavours: as an embedded platform and as a surrogate platform. If it is executed in a stand-alone mode the complete container is run on the wireless device. In the so-called split execution mode, the container is divided into a front-end on a mobile device and a back-end executed on a J2SE (Java 2 Standard Edition) host. The front-end communicates with the back-end through the HTTP protocol. This mode is particularly suited for devices with limited resources. The back-end conceals the mobile devices' IP addresses and possible changes to these addresses. The front-end can detect a lost connection and can try to re-establish the session with the back-end. This allows messages to be queued or buffered for later delivery when temporary service interruptions prevent immediate transmission.

For thin mobile clients equipped with a browser displaying XHTML MP (Mobile Profile) or HTML, several J2EE solutions are possible. Since JADE is written in Java, JADE agents can be integrated with J2EE technologies such as Servlets, JavaServer Pages (JSP), or JSP tag libraries on an application server (Goodwill, 2000). Servlets and JSPs can directly access agent classes implemented in Java and invoke appropriate methods.

The JADE functionality is also implemented as tag libraries. Tag libraries are reusable modules that can build and access programming language objects and influence the output stream. They usually encapsulate frequent tasks and can be used across applications, increasing the speed and quality of development. Tag libraries have access to all objects available to JavaServer Pages. They can communicate with each other and can be nested, allowing for complex interactions within a page. In order to deliver the appropriate content, the application server should be aware of the device context. The device context can be retrieved dynamically from CC/PP profiles (Composite Capabilities/Preference Profiles) (W3C, 2000), HTTP standard Accept headers (W3C, 2002), or may be stored together with users' preferences in a separate database.

4 Current state of the architecture and implementation for agent-based mobile SCEM

4.1 Overview of the architecture

The overall architecture of our Mobile Agent-based SCEM System (MASS) is presented in Figure 1. The architecture is divided into four layers although some components do not lend themselves easily to an unambiguous assignment due to technical constraints.

The first layer, the data layer, is represented by the databases of the ERP system and MAS, and by simulation data for RFID tags. The ERP system used in our implementation is Compiere, an open-source system that is continuously evolving (Compiere, 2005).

The second layer is a service layer, following Microsoft's architectural naming conventions (Travis, 2003). This layer encapsulates three further layers which place services at the user's disposal: the data access layer, the business logic layer, and the web services layer. For the Compiere ERP system, the data access layer and the business logic layer are merged into a so-called persistence layer (cf. Section 4.2). A module located on the service layer (cf. Section 4.2) simulates the tracking and tracing system.

Wrapper agents running within the JADE-based MAS provide access to the data layer. The business logic layer of the MAS is represented by resource, monitoring, user, and gatekeeper agents. Gateway agents represent the web services layer.

The Servlet in the third layer acts as a mediator between thin clients and the presentation logic incorporated in user agents. Fat clients using JADE-LEAP are also partially responsible for the presentation logic.

The fourth layer, the presentation layer, consists of different mobile and stationary devices like XHTML MP enabled cellular phones, PDAs, Palmtops, Pocket PCs, and regular computers with their respective browsers and GUIs.

This architecture considers and applies the requirements of mobile users identified by (Andreou et al., 2005) where practicable and feasible. For example, users want a quick and convenient way to connect to the system and also fast system response. This is achieved using a two-way possibility to connect to the system with personalisation capabilities through the user agent.

Furthermore, reliability is improved using APs. Since agent execution space is isolated, the failure of a single agent does not destabilise the AP itself – other users can continue their work unimpeded. The AP also holds the agents' states, which allows users to resume interrupted work following connection problems.

To illustrate the layers and the underlying ideas, a scenario describing the interaction between components follows. Suppose a user accesses the system through a mobile device, for example querying the ERP system for new orders or capacities of resources. If the user is interested in monitoring an order he or she may assign a new monitoring agent to that order. The newly initiated monitoring agent queries the tracking and tracing system for tracking data isochronously and examines the data received immediately. If the monitoring agent detects any irregularities it notifies one or more user agents with a description of the disruption.

User agents represent the human users in the system and interact with those users. Each user agent registered with the monitoring agent generates a notification and sends an e-mail or SMS to its owner. The kind of notification depends on the options that the user selected in his or her settings. If desired, the user can search for more information, e.g., for pertinent data from the ERP system, like the customers affected by the disruption.

If a user prefers manual problem solving then the system is used mainly for information gathering. Nevertheless, a resource agent can still be initiated. This agent should have at least semi-automatic problem-solving capabilities so that it is able, for example, to negotiate with other agents from potential partners in the supply network. A typical example is the loss of a supplier. A different partner has to be found then so that the production process can continue.

The constituents of the architecture outlined in Figure 1, the design decisions, and the underlying ideas are discussed in greater detail in the subsequent sections.

4.2 Web services façade for ERP and tracking and tracing system

In order to make integration of enterprise information systems possible, ERP system vendors have begun to offer access to their systems through web services. However, hardly any system exposes its complete functionality in this way. The open-source ERP system we are using, Compiere (2005), provided no such interface. Therefore, we decided to implement a web services façade for selected Compiere functionalities. This implementation is available at http://sourceforge.net/projects/scmws. Unfortunately, Compiere possesses 'distributed' business logic: some parts are found in database stored procedures, some within code modules, and some within the client's graphical user interface. Therefore the implemented web services façade cannot rely on typical business objects such as customers, products, and orders but is based mainly on Compiere's proprietary/custom persistence layer which largely mirrors the underlying database structure. Direct calls to the database are made only where necessary, for example, when the necessary business logic is found only in a stored procedure.

Furthermore, tracking data are required so that agents can investigate the flow of goods and other resources in a supply network. Instead of using a real tracking and tracing system, the required functionality is simulated by stochastically generating tracking data representing disruptions. These data, which would normally be obtained from RFID or barcode readers, are provided to the agents by our module that simulates tracking and tracing. For an in-depth analysis, agents access tracking data via web services that provide data such as the last item scanned or on all scanning data within a specific period.

4.3 Design and implementation of the Multi-Agent System (MAS)

Several agents have to interact with each other to investigate the supply network and to notify and support the user in case of problems. In the design of the MAS we followed design patterns from Hayden et al. (1999) and Woods and Barbacci (1999). The subsequent types are used in that approach.

User agent

The user agent is a member of the mediator family (Woods and Barbacci, 1999). It mediates between the agents in the platform and the human user who, when oversimplified, can be seen as an agent that exists at a remote location. This is achieved through a web-based or mobile interface or through APs on mobile devices. The user agent provides an interface between the human user and other agents and thus minimises the required interaction between human users and the MAS (Chen et al., 2005).

Monitoring agent

The monitoring agent is a member of the monitor family (Woods and Barbacci, 1999) and the broker family (Woods and Barbacci, 1999). The monitoring agent supports investigation of the flow of goods in the entire network. In order to monitor the entire network, the agent needs access to the ERP system to identify the assigned orders and to track manual changes by human users.

To access the ERP system or the tracking and tracing system, the monitoring agent instructs wrapper agents to query those systems. The monitoring agent investigates the incoming data for irregularities, e.g., by comparing them with key performance indicators. Furthermore, the monitoring agent acts as a broker between the user agent and the wrapper agent. If order related information is needed by the human user, the monitoring agent uses the services provided by the wrapper agent to supply the human user with the necessary information.

Wrapper agent

The wrapper agent is a member of the wrapper (Woods and Barbacci, 1999) and the embassy family (Woods and Barbacci, 1999). In general, wrapper agents allow using the functionality of the underlying systems. In our approach the wrapper agent is responsible for accessing the MAS database. That database contains user profiles and preferences.

Gateway agent

The gateway agent is a specific kind of wrapper agent. This agent accesses underlying systems, e.g., ERP systems, the tracking and tracing system, or legacy applications by means of web services. Gateway agents are based on the Web Service Integration Gateway (WSIG) (Greenwood and Calisti, 2004). The main components of a WSIG are

- the Gateway Registry
- the ServiceDescription Translator
- the StubCreator
- the WebServiceInvocation component
- the AgentServiceInvocation component
- the MessageTranslator.

The GatewayRegistry is in charge of processing incoming directory operations. ACL-encoded service descriptions received from agents are stored in the internal DF registry and are automatically translated by the ServiceDescriptionTranslator into their WSDL equivalents and stored in the internal UDDI registry. The reverse case is also true for WSDL encoded service descriptions received from web service owners. In this manner the WSIG maintains a mapping of each service description in both ACL and WSDL.

Web services are exposed as service endpoints (stubs) which are called remote invocations. When a new agent service is registered with the WSIG, the GatewayRegistry invokes the StubCreator that produces a stub to be exposed via the AgentServiceInvocation component. When an agent invokes a web service via the WebServiceInvocation component and a response is expected from the web service, a temporary stub is created by the StubCreator.

The gateway agent publishes the available services in the DF so that other agents can search for services provided by the gateway agent. This gateway agent accepts requests from many agents and communicates with one information system. To access a necessary service, an agent contacts the gateway agent for this service. The gateway agent translates incoming ACL messages into SOAP requests and invokes the requested services for local agents and for authorised foreign agents, respectively.

Afterwards, when an answer is received, the agent translates the SOAP response back to an ACL message to inform the requesting agents. The agent should be capable of bundling requests, i.e., the agent has to know how individual requests can be aggregated and translated into one request to match the available web services. Likewise, the agent should be able to disaggregate agent requests into multiple web service invocations when the web services are more fine-grained.

Gatekeeper agent

The gatekeeper agent is a member of the embassy family (Woods and Barbacci, 1999). It authorises the human user and agents from other platforms that communicate with the agents or migrate onto the platform. If the authorisation was successful, i.e., the wrapper agent confirmed the user's rights, the gatekeeper agent informs the Servlet in the presentation-logic layer that the user may be connected with his or her user agent if such an agent exists. Otherwise the Servlet creates a new user agent.

The resource agent and the validation engine are still in their infancy as significant research effort is still needed to reach automatic problem solving. The design is as follows

- *Resource agent*. The resource agent is a practical reasoning agent based on the BDI (believe, desire, intention) model as described by Wooldridge and Jennings (1995). This agent is the 'intelligent' component capable of reasoning about a given problem and deciding on a possible action to solve the problem. Jadex, an extension for JADE, provides such a reasoning-agent framework based on the BDI model. A short overview can be found in Bordini et al. (2006). The resource agent will search its knowledge base or external applications, e.g., the ERP system, to find data pertinent to the disruption. This agent is a mobile agent able to migrate to other APs in the network.
- *Validation engine*. The validation engine is an extension for the resource agent. While the resource agent should be able to learn and modify its knowledge, the validation engine can be seen as a body of rules and regulations, i.e., immutable strategies that cannot be modified by the agents. A solution proposed by the resource agent can subsequently be simulated and checked by the validation engine to see if it meets all conditions or requirements. This step improves security and performance by validating the solution of a migrating agent upon its return to its originating system. The validation engine must be protected from foreign resource agents to ensure the secrecy of a company's strategic goals.

4.4 Applied approach for user interfaces

Mobile and wireless applications are typically deployed with three-tier or multi-tier architectures. In a three-tier architecture the business logic can be entirely or partially contained in a middleware component running in an application server. If the business logic is on the application server and clients are only responsible for the presentation layer, then such clients are called 'thin'. If significant portions of the business logic are on the client, then it is considered a 'fat' client.

Wireless appliances with mobile browsers are examples of thin clients. They provide suitable user interfaces without knowing about the application logic and they cannot connect to different applications. Browsers on thin mobile clients are able to display various media formats. Currently most mobile browsers support XHTML Mobile Profile (XHTML MP), HTML, and WML for older devices (W3C, 1999, 2003; WAP Forum, 2002). Personal Digital Assistants (PDAs), Palmtops, and J2ME-enabled phones are examples of fat clients. They can perform tasks and manage data without the help of a server.

In our approach, the presentation of content for mobile devices is generated by placing the responses retrieved from agents into appropriate tags and setting the correct MIME type for a particular device. The task of wrapping the content into tags is performed by user agents, which use presentation logic to format content to be delivered to thin clients. The generation of markup elements (e.g., tables, buttons, paragraphs, etc.) is based on special Java template classes. User agents, with the help of wrapper agents, communicate with the MAS database where user preference information is stored. Depending on the information retrieved, these agents render the content in a specific markup language. The MASS Servlet then sends the content to the end device.

On fat clients, the JADE-LEAP platform is used, which allows fat clients to directly communicate with the AP. They are responsible for both the presentation and the presentation logic.

5 Managerial implications of agent-based mobile SCEM

Based on the preceding description of an architecture for mobile SCEM some managerial implications can be outlined. Mobile technologies, in combination with intelligent agents, make proactive SCEM possible. When the flow of goods and the status of resources are permanently tracked and this information can be obtained from every node in the supply network, agents are able to identify problems, notify decision makers, and initiate subsequent problem-solving processes. In this way, transparency in a supply network can be achieved, avoiding information asymmetries between the partners (Hanebeck and Tracey, 2003).

Another implication for management is that problem-solving knowledge assembled by the agents is available anywhere in the network and it is accessible from mobile devices. Whenever a problem is detected, human decision makers can react to this problem independently of their location, no matter whether they are in their wired offices or in the field with only a mobile device available.

Likewise, knowledge workers are enabled to accelerate problem-solving processes. This can be critical for the timely resolution of supply chain disruptions. Knowledge workers may continue their accruing field work and at the same time respond to exceptions without interrupting their normal work schedule. Specialised agents could search for pertinent data and prepare it for the knowledge workers. These agents need to distinguish between useless information and information that is contextually relevant to the problem at hand.

A crucial management challenge is to ensure the existence of trust in the network. Collaboration between members is only possible if they provide reliable information to each other without any counterfeiting. Information also has to be secured against external counterfeiting attempts.

New security issues are emerging, necessitating decisions and guidelines from IT management. With technologies like RFID, for example, spoofing RFID tags or corrupting tag data have been observed, yet countermeasures to these problems are already available (Shih et al., 2005).

Mobile agents pose yet another security threat. Just as members of a network must trust new mobile agents entering their platform, mobile agents must trust the platform onto which they migrate. To overcome the risk in the latter case, (Tsaur and Ho, 2005) suggested a solution using pairing-based cryptosystems. This approach could also be applied in the other direction, regarding the risk that new agents may represent to a platform.

Finally, it should be noted that our architecture provides a possible solution to the problem of Enterprise Application Integration (EAI). We have shown how heterogeneous systems can be integrated via web services. This can be an effective approach to EAI when such different systems like an ERP system, a tracking and tracing system, and a Multi Agent System (MAS) have to work together in one information system.

6 Conclusions and future work

Automatic detection and management of disruptions and other irregularities in supply networks are important issues that led to the emergence of SCEM. The architecture presented in this paper is intended to support process owners and eliminate supply chain disruptions before they lead to losses in profitability or in partners' trust. Intelligent agents investigate unexpected events on their behalf and give decision support, at least in standard cases. The main focus of our work is the assistance of mobile workers. Through the use of mobile devices and mobile communication networks, relevant business information and functionality can be made accessible, regardless of the location and time of disruption.

One of the most important requirements for SCEM is the ability to exchange real-time information between supply chain partners. Software agents in combination with SOA can be very helpful in this case. web services enable platform-independent and cost-effective access to the underlying information systems and can be easily integrated with MAS. It should be considered, however, that many enterprises are not ready to disclose their internal data because they fear attack from potential competitors. Moreover, many firms are reluctant to introduce costly new solutions before their existing systems are fully amortised. The lack of common ontologies for standardised data exchange between heterogeneous systems presents further problems.

Our current work focuses on the design and implementation of suitable ontologies for an application scenario in the bicycle industry and on the mechanisms for resolving complex supply network problems using intelligent agents.

Acknowledgements

This work is part of the joint research project 'Mobile Internet Business' (http://mib.uni-ffo.de) funded by the German Federal Ministry of Education and Research. The authors wish to thank Brandon Ulrich for contributing ideas and implementing the web services Façade for the Compiere ERP system.

References

Adorni, G., Bergenti, F., Poggi, A. and Rimassa, G. (2001) 'Enabling FIPA agents on small devices', *Proceedings of the 5th International Workshop on Cooperative Information Agents*, Modena, pp.248–257.

Albuquerque, R.L., Hübner, J.F., de Paula, G.E., Sichman, J.S. and Ramalho, G.L. (2001) 'KSACI: a handheld device infrastructure for agents communication', in Meyer, J-J. and Milind, T. (Eds.): *Pre-proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages*, pp.327–337.

Andreou, A.S., Leonidou, C., Chrysostomou, C., Pitsillides, A., Samaras, G., Schizas, C.N. and Mavromoustakos, S.M. (2005) 'Key issues for the design and development of mobile commerce services and applications', *International Journal of Mobile Communications*, Vol. 3, No. 3, pp.303–323.

Backlund, P. (2005) 'On the research approaches employed at recent European Conferences on Information Systems (ECIS 2002 – ECIS 2004)', *Proceedings of 13th European Conference on Information Systems (ECIS 2005), Information Systems in a Rapidly Changing Economy*, Regensburg.

Blanchard, R.J. (2004) *SCM Spotlight: RFID and the Adaptive Enterprise: What's Real Now? What's Next?*, http://www.sapinsideronline.com/searchspi/search.htm?page=article&key=37914.

Bodendorf, F. and Zimmermann, R. (2005) 'Proactive supply-chain event management with agent technology', *International Journal of Electronic Commerce*, Vol. 9, No. 4, pp.57–89.

Bordini, R., Braubach, L., Dastani, M., El Fallah Seghrouchni, A., Gomez-Sanz, J., Leite, J., O'Hare, G., Pokahr, A. and Ricci, A. (2006) 'A survey of programming languages and platforms for Multi-Agent Systems', *Informatica Journal*, Vol. 30, No. 1, pp.33–44.

Bretzke, W.R. (2002) 'Rettungsring für krisenmanager – Supply Chain Event Management: mehr als nur ein schlagwort in der logistik', *DVZ*, Vol. 19, p.6.

Britton, C. (2000) *IT Architectures and Middleware: Strategies for Building Large, Integrated Systems*, Addison-Wesley, Boston.

Carabelea, C. and Boissier, O. (2003) 'Multi-agent platform on smart devices: dream or reality?', *Proceedings of the Smart Objects Conference (sOc'2003)*, Grenoble, pp.126–129.

Chen, M., Zhang, D. and Zhou, L. (2005) 'Providing web services to mobile users: the architecture design of an m-service portal', *International Journal of Mobile Communications*, Vol. 3, No. 1, pp.1–18.

Compiere (2005) *Compiere ERP System*, http://www.compiere.org.

Dangelmeier, W., Pape, U. and Rüther, M. (2004) *Agentensysteme für das Supply Chain Management: Grundlagen, Konzepte, Anwendungen*, Deutscher Universitäts-Verlag, Wiesbaden.

Davenport, T.H. and Brooks, J.D. (2004) 'Enterprise systems and the supply chain', *Journal of Enterprise Information Management*, Vol. 17, No. 1, pp.8–19.

Dickmann, A. (1995) 'Two-tier versus three-tier applications', *Informationweek*, Vol. 553, No. 13, pp.74–80.

FIPA (2004) *FIPA Agent Management Specification*, http://www.fipa.org/specs/fipa00023/SC00023K.html.

FIPA (2005) *FIPA and IEEE*, http://www.fipa.org/about/fipa_and_ieee.html.

Floerkemeier, C. (2005) 'EPC-Technologie – vom Auto ID Center zu EPCglobal', *Das Internet der Dinge – Ubiquitous Computing und RFID in der Praxis*, Springer-Verlag, Berlin, pp.87–100.

Frey, D., Stockheim, T., Woelk, P-O. and Zimmermann, R. (2003) 'Integrated multi-agent-based supply chain management', *IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Linz, pp.24–29.

Goodwill, J. (2000) *Pure Java Server Pages*, Sams Publishing, Indianapolis.

Greenwood, D. and Calisti, M. (2004) 'Engineering web service – agent integration', *Proceedings of the IEEE Systems, Cybernetics and Man Conference*, Hague, pp.1918–1925.

Hanebeck, H-C.L. and Tracey, B. (2003) 'The role of location in supply chain management: how mobile communication enables supply chain best-practice and allows companies to move to the next level', *International Journal of Mobile Communications*, Vol. 1, Nos. 1–2, pp.148–166.

Hayden, S.C., Carrick, C. and Yang, Q. (1999) 'Architectural design patterns for multi-agent coordination', *Proceedings of the International Conference on Agent Systems '99*, Seattle.

Hofmann, O., Deschner, D., Reinheimer, S. and Bodendorf, F. (1999) 'Agent-supported information retrieval in the logistic chain', *Proceedings of the 32nd Hawaii International Conference on System Sciences*, Maui.

Kärkkäinen, M., Främling, K. and Ala-Risku, T. (2003) 'Integrating material and information flows using a distributed peer-to-peer information system', Jagdev, H.S., Wortmann, J.C., Pels, H.J. (Eds.): *Collaborative Systems for Production Management*, Kluwer Academic Publishers, Boston, pp.305-319.

Lee, H.L. (2004) 'The triple-a supply chain', *Harvard Business Review*, Vol. 82, No. 10, pp.103–112.

Luck, M., McBurney, P., Shehory, O. and Willmott, C. (2004) *Agent Technology Roadmap: Overview and Consultation Report*, http://www.agentlink.org/roadmap/roadmapreport.pdf.

Mahmoud, Q.H. (2001) 'MobiAgent: an agent-based approach to wireless information systems', *Proceedings of the Agent-Oriented Information Systems Workshop at Autonomous Agents'2001 Conference*, Montreal.

McGovern, J., Tyagi, S., Stevens, M. and Matthew, S. (2003) *Java Web Services Architecture,* Morgan Kaufmann Publishers, San Francisco.

Nissen, V. (2002) 'Supply Chain Event Management', *Wirtschaftsinformatik*, Vol. 44, No. 5, pp.477–480.

Oracle (2005) 'Oracle business activity monitoring', *Oracle Whitepaper*, http://www.oracle.com/technology/products/integration/bam/pdf/BAM_Whitepaper.pdf.

Risi, W.A. and Rossi, G. (2004) 'An architectural pattern catalogue for mobile web information systems', *International Journal of Mobile Communications*, Vol. 2, No. 3, pp.235–247.

Sharma, S.K., Kitchens, F.L. and Miller, P. (2003) 'A new approach for mobile communication: web services using flexible services architecture', *International Journal of Mobile Communications*, Vol. 1, No. 3, pp.301–311.

Shih, D-H., Lin, C-Y. and Lin, B. (2005) 'RFID tags: privacy and security aspects', *International Journal of Mobile Communications*, Vol. 3, No. 3, pp.214–230.

Speyerer, J.K. and Zeller, A.J. (2004) 'Managing supply networks: symptom recognition and diagnostic analysis with web services', *Proceedings of the 37th Hawaii International Conference on System Sciences*, Big Island.

Szirbik, N.B., Wortmann, J.C., Hammer, D.K., Goossenaerts, J.B.M. and Aerts, A.T.M. (2002) 'Mediating negotiations in a virtual enterprise via mobile agents', *Proceedings of the Academia/Industry Working Conference on Research Challenges*, IEEE Computer Society Eds., Buffalo, NY, pp.237–242.

Topley, K. (2002) *J2ME in a Nutshell*, O'Reilly & Associates, Sebastopol.

Travis, B. (2003) *FoodMovers: Building Distributed Applications using Visual Studio.NET*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnvsent/html/FoodMovers3.asp.

Tsaur, W-J. and Ho, C-H. (2005) 'A mobile agent protected scheme using pairing-based cryptosystems', *Int. J. Mobile Communications*, Vol. 3, No. 2, pp.183–196.

Varshney, U., Malloy, A., Ahluwalia, P. and Jain, R. (2004) 'Wireless in the enterprise: requirements, solutions and research directions', *International Journal of Mobile Communications*, Vol. 2, No. 4, pp.354–367.

W3C (1999) *HTML 4.01 Specification*, http://www.w3.org/TR/html4/.

W3C (2000) *Composite Capabilities/Preference Profiles: Terminology and Abbreviations*, http://www.w3.org/TR/2000/WD-CCPP-ta-20000721/.

W3C (2002) *Delivery Context Overview for Device Independence*, http://www.w3c.org/2001/di/public/dco.

W3C (2003) *XHTML 2.0 The Extensible HyperText Markup Language Specification*, http://www.w3.org/TR/xhtml2/.

WAP Forum (2002) 'Wireless Application Protocol WAP 2.0', *Technical White Paper*, http://www.wapforum.org/what/WAPWhite_Paper1.pdf.

Woods, S.G. and Barbacci, M.R. (1999) 'Architectural evaluation of collaborative agent-based systems', *Technical Report CMU/SEI-99-TR-025*, http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr025.pdf.

Wooldridge, M. and Jennings, R. (1995) 'Intelligent agents: theory and practice', *Knowledge Engineering Review*, Vol. 10, No. 2, pp.115–152.