Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2009 Proceedings

Americas Conference on Information Systems (AMCIS)

2009

Enhancing Supply Chain Reliability through Agent-Based Supply Chain Event Management

Earl Kurbel
European University Viadrina, kurbel.bi@euv-frankfurt-o.de

Kamil Nowakowski
European University Viadrina, wi-sek@euv-frankfurt-o.de

Denny Schreber
European University Viadrina, mbi@euv.frankfurt-o.de

Recommended Citation

Kurbel, Earl; Nowakowski, Kamil; and Schreber, Denny, "Enhancing Supply Chain Reliability through Agent-Based Supply Chain Event Management" (2009). *AMCIS 2009 Proceedings*. 226.
http://aisel.aisnet.org/amcis2009/226

Enhancing Supply Chain Reliability through Agent-Based Supply Chain Event Management

Karl Kurbel
European University Viadrina,
kurbel.bi@euv-frankfurt-o.de

Kamil Nowakowski
European University Viadrina,
wi-sek@euv-frankfurt-o.de

Denny Schreber
European University Viadrina
mbi@euv-frankfurt-o.de

ABSTRACT

Supply Chain Event Management (SCEM) is an approach to the monitoring of supply chains. It observes specific events and exceptions in real-time and then alerts managers if problems occur. This paper presents an architecture for an SCEM system based on intelligent software agents, Auto-ID technologies and mobile user interfaces. The motivation for this approach is to enhance existing SCEM solutions by exploiting up-to-date technologies. It delegates the task of automated problem solving when disruptions in supply chains occur to software agents.

Keywords

Supply Chain Event Management, SCEM, software agents, tracking & tracing simulation

INTRODUCTION

Supply Chain Event Management (SCEM) systems focus on real-time event monitoring and on detecting and analyzing supply chain disruptions. They support real-time exception management by notifying the responsible persons and recommending actions that can be taken to resolve unexpected problems. SCEM systems aim to improve collaboration between internal and external supply chain partners. They provide the partners with up-to-date information about supply chain processes, workflows and events, and facilitate decision making. SCEM systems offer functionalities that are beyond the scope of systems aiming to improve business processes, such as Supply Chain Planning (SCP), Supply Chain Execution (SCE), Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP).

The main challenge for an SCEM system is to provide accurate, up-to-the-minute information about the states of crucial business processes and assets to all members of a supply network. Supply chain visibility is essential for the ability to respond quickly and effectively to supply chain disruptions (Wieser and Lauterbach, 2001). SCEM systems provide supply chain monitoring in real-time, or close to real-time. A good SCEM system is able to react automatically to disruptions that match known, established patterns. Today's SCEM solutions are usually based on tracking, tracing and monitoring systems that use barcodes, RFID chips or sensors. They are often supported by mobile technologies which transmit data between the system and its users. Oracle's Business Activity Monitoring (Oracle Corp., 2005) and SAP Event Management (SAP AG, 2007) are examples of such systems.

Our research concentrated on improving existing Supply Chain Event Management solutions through the use of state-of-the-art technologies. In order to examine the benefits and drawbacks of these technologies for SCEM, we followed the "construction and building" research approach as described in (Backlund 2005). The goals of our research were to enable semi-automatic problem solving in SCEM, increase supply chain visibility, enable intra-organization exchange of information and decrease response times when supply chain disruptions occur. A prototype was developed using technologies that promise significant improvements of SCEM, in particular intelligent agents in a multi-agent system (MAS), Auto-ID technologies, web services and mobile computing.

This paper consists of six sections and is organized as follows: In the next section, the impact of supply chain disruptions on the partners is discussed. The third section briefly characterizes existing SCEM solutions and their limitations. In the fourth section, the solution developed in our research project is described. The focus in this section is on the concept of agent-based SCEM, the architecture of the system and the prototypical implementation. The fifth section presents an evaluation of the solution with the help of discrete-event simulation. Finally, open issues for further research and development are outlined.

IMPACT OF DISRUPTIONS ON SUPPLY NETWORKS

Depending on the origin of a supply chain disruption, three types of problems can be distinguished (Jüttner, Peck and Christopher, 2003):

- External network disruptions – such disruptions cannot be prevented or influenced from within the supply network. Adequate countermeasures include, for example, a Supply Chain Risk Management policy with a predefined set of remedies for disruptions (Gaonkar and Viswanadham, 2004).
- Internal network disruptions – these disruptions usually occur at the nodes of the network (partner companies) or at the links between the nodes (e.g. transport routes). In these cases, SCEM is an adequate approach for the detection and resolution of problems.
- Disruptions due to the network – if a disruption occurs due to the supply network itself, a strategic redesign of parts or of the entire network is necessary. Our approach helps to identify network problems, e.g. unreliable suppliers.

Although many enterprises apply SCEM systems today, research on supply chain reliability has shown that it is still necessary to improve supply chain visibility and to reduce uncertainty (Gaonkar and Viswanadham, 2004). SCEM strongly supports Supply Chain Risk Management. Even though SCEM is not the "silver bullet" for managing disruptions, it can significantly enhance existing SCM solutions.

Not reacting properly to detected supply network problems can be a cause of significant follow-up costs. Hendricks and Singhal (2003) analyzed 512 glitches that occurred in supply networks between 1989 and 2000. In their sample, the five most frequent glitches were shortages of parts, order changes by customers, ramp-up and roll-out problems, as well as quality and production problems. Direct consequences of such glitches include the following (Hendricks and Singhal, 2008):

- Lower customer satisfaction
- Lower sales and profits
- Deteriorating supplier-customer relationships (reputation)
- Shrinking shareholder value

Supply chain disruptions can have severe economic consequences. However, the cost of resolving a disruption may be higher than the potential loss resulting from this disruption (e.g. loss of a customer). Therefore, decision makers have to weigh the cost of resolving the disruption against the loss incurred by it.

RELATED WORK

Many existing agent-based approaches for SCEM focus on monitoring the flow of goods with the help of tracking and tracing (T&T) systems. Technologies such as RFID chips, barcodes and sensors are employed to determine the locations and/or conditions of the monitored resources and to increase the visibility of the network.

Approaches addressing basic SCEM problems include ECTL-Monitor (Hofmann, Deschner et al., 1999), PAMAS (Zimmermann and Paschke, 2003) and Dialog (Kärkkainen, Främling and Ala-Risku, 2004). They mainly attempt to detect disruptions and exceptions in supply chains and deliver this information to appropriate recipients. However, these approaches do not provide semi-automatic or automatic problem solving regarding the detected problems.

More advanced systems, with software agents accomplishing actions that are usually performed by humans, include PROVE (Szirbik, Wortmann et al., 2000), CoagenS (Dangelmeier, Pape and Rüther, 2004), Agent.Enterprise (Frey, Mönch et al., 2003), and Speyerer and Zeller's (Speyerer and Zeller, 2004) approach based on web services. These solutions try to automate the problem-solving process. Some of them address intra-enterprise communication between different departments.

In our approach, intelligent agents are used to perform routine tasks, analyze unplanned events and elaborate on standardized solutions for them. With the developed algorithms, diverse types of disruptions can be detected. Calls for action may be delegated to local agents. (A local agent is an agent within the local information system of a participant of the supply chain.)

In such cases, agents have to migrate from the multi-agent system (MAS) that detected the disruption to another multi-agent system that can establish a solution.

AGENT-BASED SUPPLY CHAIN EVENT MANAGEMENT

Architecture of the Solution

The architecture of the Mobile Agent-based SCEM System (MASS) is outlined in figure 1. This system was developed as a prototype, encompassing four functional layers.

1) The first layer, the *data layer*, consists of three databases. These databases are needed for the underlying ERP system Compiere (Compiere, 2008), the multi-agent system used in the project and the tracking and tracing system.

2) The *service layer* represents the core of the system. It encapsulates components that allow data retrieval and operations on the stored data (data access), it reflects the underlying process model and its subjects in the form of business assets (business logic), and it offers access to services through standardized web interfaces (web services). A web services façade was implemented that offers the advantages of a service-oriented architecture (SOA): platform-independent access to the underlying system's functionality and easy, ad-hoc integration with other systems. Two points to note are:

- For the Compiere ERP system, data access and business logic are merged into a persistence layer. Data access to the multi-agent system is performed with the help of wrapper agents. Business logic is encapsulated in diverse types of agents (resource, monitoring, gatekeeper and user agents, cf. next subsection). Gateway agents communicate with the web services layer.
- The tracking & tracing system is simulated with the help of a set of Java classes. A web services façade (service interface) is built on top of these classes.

3) The *presentation logic layer* interprets the results obtained from the service layer and generates appropriate outputs depending on the detected device type. Within the presentation logic layer, user agents play a crucial role. User agents may reside on the server (for thin clients) or on mobile devices (for smart clients). The MASS Servlet acts as an intermediary between the user agent and the client's device if the client has a browser-based interface.

4) The *presentation layer* comprises browsers and software on various device types including notebooks, Java-enabled PDAs and smart phones.

Agents on the Platform

The multi-agent system in the middle of figure 1 has the following types of software agents: User agents, monitoring agents, wrapper agents, gateway agents, a gatekeeper agent and resource agents. Their tasks are as follows:

- A *user agent* provides an interface between the human user and other agents and thus minimizes the required interaction between human users and the multi-agent system. The user agent mediates between the agents and the human user.
- A *monitoring agent* supports the investigation of the flow of goods in the entire network. In order to monitor the entire network, this 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 & tracing system, the monitoring agent instructs wrapper agents to query those systems. The monitoring agent investigates the incoming data for irregularities.
- A *wrapper agent* is responsible for accessing the database of the multi-agent system. This database contains user profiles and preferences.
- A *gateway agent* is a specific kind of wrapper agent. This type of agent communicates with the primary data suppliers – ERP and T&T systems. An ERP system, for example, discloses information such as open orders and order details, while T&T systems deliver data on tracked order items.
- A *gatekeeper agent* authorizes the human user (and possibly agents from other supply-network nodes that wish to communicate with agents of the platform or to migrate onto the platform).
- A *resource agent* is a practical reasoning agent based on the BDI (believe, desire, intention) model. This agent is capable of reasoning about a given problem and deciding on a possible action to solve the problem. Resource agents search their knowledge bases or external applications, e.g. an ERP system, to find data pertinent to the problem (i.e. disruption) they are trying to solve.

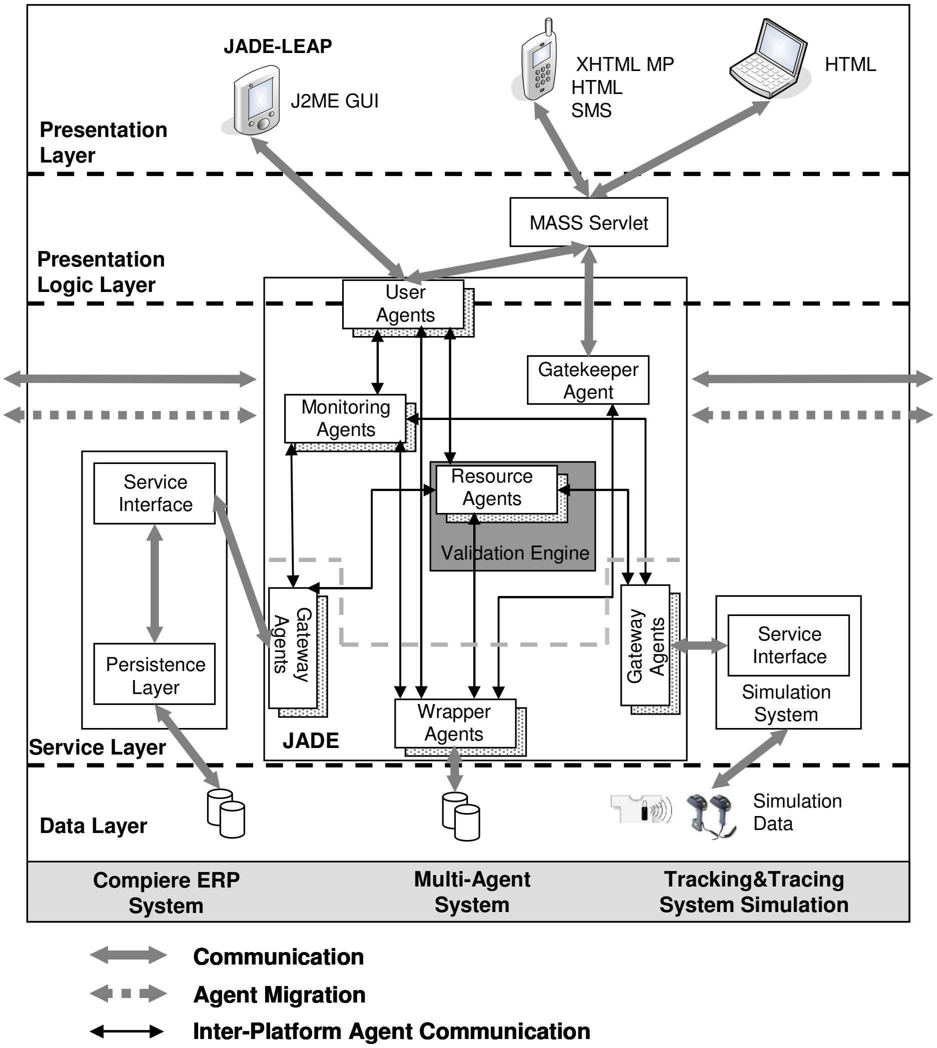

Figure 1. Architecture of the Mobile Agent-based SCEM System (MASS)

The interaction among these types of agents is as follows: Users log onto the system. Authorization is performed by the gatekeeper agent. After successful authorization, the user can examine the released orders and decide which orders should be monitored. For each selected order, a monitoring agent is initialized. This agent investigates real-time flows of resources when an order is being completed. Monitoring agents need advanced algorithms to be able to identify events and undertake appropriate actions. In order to do their work, they also need information from the databases. In the architecture outlined in figure 1, monitoring agents will ask gatekeeper agents to communicate with the ERP and T&T systems and retrieve this information. (In our prototypical implementation, we used a simulation of tracking and tracing instead of a real T&T system.)

One monitoring agent is assigned to exactly one order in the ERP system. This agent analyzes incoming data. Tracking and tracing data are obtained from the web services façade that resides on top of the simulated T&T system. Additional information is retrieved by wrapper agents residing on the same platform or obtained from agents on remote platforms upon request. A monitoring agent can analyze data and detect problems in the following ways:

- By comparing the current state with the target state with the help of suitable algorithms.
- By searching for irregularities in the tracking data.

In the latter case, the goal is to identify patterns that cannot be detected by plain actual-target comparisons. Stream clustering algorithms are particularly helpful to detect follow-up problems (Gianella, Han et al., 2004). When required, these algorithms can be activated to analyze data from sensor-enhanced T&T systems in order to derive information about irregularities. For example, they can compare currently tracked data with previously tracked data and use previous analysis results to point out possible upcoming problems.

Examples of typical problems that can occur on a supply network's links are shown in table 1. Links between the network's nodes are transport routes. (Other problem areas include problems at the network's nodes, e.g. unreliable suppliers or problems with the customer.) For example, a vehicle might be re-routed if the traffic-jam monitor indicates heavy traffic on the original route. Whenever such a situation is detected, the multi-agent system can immediately notify the process owner to allow for fast reaction, preventing possible damage. The monitoring agent would forward the exception to the user agent that interacts with the user.

For problems that can be solved without (or with minimal) human intervention, a resource agent is triggered to find a solution to the problem. Resource agents access the ERP systems of the companies involved with the help of web services and perform necessary actions. If automatic problem solving is not possible, the resource agent notifies the user agents who pass this information on to the responsible persons.

Resource agents follow a rule-based approach, attempting to remove exceptions with the help of a predefined set of rules that implement the last two columns of table 1. For security reasons, this agent was designed as a mobile agent capable of migrating to other platforms and negotiating directly with agents on the target platform. Afterwards the agent returns to its home platform, together with the results of the negotiation. In this way, it is not necessary to send internal company data over unsecured networks.

The main advantage of rule-based problem solving is that a solution can be found which was not originally envisioned by the process owner but can still help to remedy a disruption. Nevertheless, the rule-based approach has a number of limitations. Negotiations between agents and automated therapies performed by the agents do not always lead to acceptable solutions. Tomlin (2006) showed that no best practice exists per se, i.e. it all depends on network settings such as supplier reliability, the firm's inventory size or the general strategy being applied.

Event	Coordinates	Timestamp	State of Sensors	Detection	Possible Countermeasures
Deviation	Equal distances	Equal time intervals	OK	Actual/target comparison: route	Re-routing, notification of vehicle driver
Traffic jam	Lower distances	Equal time intervals	OK	Actual/target comparison: distance	Re-routing
Damaged goods	Equal distances	Equal time intervals	Not OK	Actual/target comparison: state of goods	New transportation order, notification of vehicle driver
Driver injury or vehicle problem	Zero distance	Equal time intervals	OK	Actual/target comparison: distance & state of goods	Request alternative driver or new transportation order
Fatal accident/ crash	Zero distance	Equal time intervals	Not OK	Actual/target comparison: distance & state of goods	Vehicle recovery and new transportation order
Tracking error	Different distances	Different time intervals	OK or not OK	Actual/target comparison: distance & time	Check tracking systems

Table 1: Possible events and appropriate countermeasures (isochronous tracking)

Another disadvantage of the rule-based approach is its lack of adaptivity. Once the agents are created, they do not have learning abilities that would enable them to examine human problem-solving behavior and learn from it. This limitation could potentially be eliminated by machine learning approaches. Consider, for example, a manufacturing process that always needs to be restarted in order to fix a recurring problem. If the resumption of activities after the restart is successful and an agent could recognize this pattern, then the agent could apply the pattern automatically to similar problems without involving a human decision maker in the problem-solving process. However, the current state of machine learning does not support such scenarios yet.

Prototypical Implementation

For the MASS prototype, a supply network of companies from the mountain bike industry was modeled. All use Compiere as their ERP system. Data of the participants of the supply network were created and stored in Compiere ERP installations. The mountain bike industry was chosen because it has a transparent structure regarding the production and sourcing of bicycle components. Market participants are typically small and medium-size enterprises.

The underlying database is Oracle 10g. JBoss 4.0.1SP1 is used as an application server. To make the main functionalities of the Compiere system accessible from other applications, a web services façade was implemented. Users can access the MASS prototype through a web browser on a mobile device or by launching a Java application. JADE, an open-source FIPA (FIPA, 2005) compliant framework written in Java, serves as the underlying multi-agent system.

In order to obtain tracking and tracing data, a T&T system was simulated and implemented. It provides the monitoring agent with randomly generated tracking data. Through a graphical user interface, additional manipulations and events in the transportation processes can be created, including accidents, medical problems, speed variations, traffic jams and navigation problems as outlined in table 1.

Resource agents were implemented in Jadex, a reasoning engine for software agents (Pokahr, Braubach and Lamersdorf, 2005).

EVALUATION OF THE SOLUTION

In order to evaluate the solution implemented in the above prototype, discrete-event simulation was applied. The criterion examined in the simulation is the *time needed to react* to a problem in the supply network. In addition, the *quality of the diagnosis* given by the agents is discussed below. Both time and quality are very important criteria. When it takes longer to detect a problem, more follow-up problems are likely to occur further down the supply chain. For a good quality diagnosis, error-free detection of unforeseen events is important.

Therefore we examined if statistical tests, in addition to target-state comparisons, could further enhance the quality of a diagnosis. On the one hand, there are optimal routes from the warehouses to the delivery destinations, stored in the ERP systems. In reality, however, road conditions may require re-routing, and inaccurate data from mobile tracking systems may show deviations from the optimal routes. The differences between actual and optimal routes were simulated.

Statistical tests can be used to assess the significance of the deviations. A chi-square test was chosen to ensure that the deviations from the optimal route are normally distributed. If this is not the case, the conclusion is that the deviations are significant. This information can then be used to define thresholds when to notify the process owner.

The simulation was performed on the fictitious supply network mentioned above. Each partner in the network is represented by an instance of the Compiere ERP system. In the simulation, agents automatically change transaction data through pertinent web services. A tracking simulation is assigned to each running transportation order. The simulation system generates a starting point and an endpoint for the order. Then it calculates the route and simulates the goods transportation. Gateway agents access tracking data and relevant ERP data so that the monitoring and resource agents can monitor and analyze the orders.

The following conditions were set in order to allow comparisons of the simulation results:

- Each agent monitors the same order from the ERP system.
- A fixed data set of 258 tracking points and a fixed route with 5 nodes are provided.
- All agents run on one computer. The ERP system and the T&T system are hosted on a server and are accessed by web services.
- Within one simulation run, all agents use the same algorithms for monitoring and analysis.
- Agents do not negotiate, i.e. only "request – inform" speech acts are allowed.

Since many transport orders are active at the same time, the MASS system has to monitor all orders concurrently and still provide the user with up-to-date information. The larger the amount of orders being processed, the more agents simultaneously monitoring the orders there are. This means the system's performance is likely to decrease. The focus of the simulation was therefore on how the system performs with an increasing number of monitoring agents. The time between the start and the end of the diagnosis process was measured.

The results of four simulation runs are summarized in figure 2. Average response times are plotted against the numbers of agents involved. (The term "response time" refers to the real-time delay the MASS takes for identifying and diagnosing an event, not to the time a user has to wait for a response at his or her computer.) The four curves correspond to the following scenarios:

- Interval 15 means that an agent updates its data every 15 seconds. This curve shows a steady increase in response time from 9 agents on. When 25 agents access the underlying systems simultaneously, the process requires more than one minute, which is too long in many cases. The limiting factor is access to the ERP system, in particular database calls to retrieve order details such as quantity and delivery dates.
- Interval 15/10: ERP data do not change as rapidly as tracking data. Therefore, better performance can be achieved by limiting the number of calls to the ERP system. This effect is shown in the Interval 15/10 curve. ERP data are checked only every tenth time tracking data are read. 88 agents can now be active before the response time exceeds one minute.
- Interval 60/10: Here the T&T system is checked every 60 seconds and the ERP system is accessed every tenth time the tracking system is checked. The curve shows that 152 simultaneous monitoring requests can be processed before the response time exceeds one minute. The increase is linear from the 84th agent on.
- Interval 15/10/C: Finally, the effect of simultaneously applying statistical tests (chi-square tests) in addition to the target-state comparisons was measured. The results are shown in the Interval 15/10/C curve. Obviously the additional computing effort is minimal compared to the Interval 15/10 curve. This means that it is worth trying to enhance the results by chi-square tests.

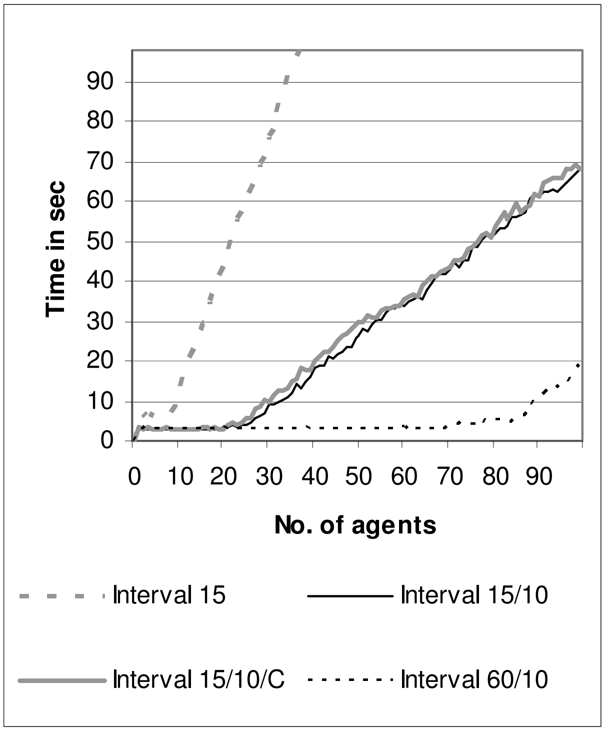

Figure 2. Response time vs. number of agents

The *quality of a diagnosis* is good if the initial problem is correctly determined. Since we try to detect problems automatically, the question of which types of problems can be found and treated in this way arises. We limited our investigation to problems that can be solved automatically, i.e. without intervention of a human user. Automation is possible for routine problems which can be treated through routine actions based on logical considerations.

Keeping this assumption in mind, we found that all disruptions entered manually in the T&T module were detected in the simulations. Nevertheless, some problems may be hard to find, e.g. problems that build up when smaller problems have not been recognized, or problems for which sufficient data are not available (as may be the case in a disaster). Beyond the scope of this investigation were problems involving irrational factors such as emotions and intuition.

Since the thresholds for target-state comparisons can be customized arbitrarily, it is possible that deviations from the plan will not be recognized or that unnecessary notifications will be sent to the user. Another problem is the detection of measurement failures. Through the target-state comparisons, it is possible yet cumbersome to eliminate single outliers. Being susceptible to outliers, the chi-square tests are more problematic. Since the chi value used is relatively small and two values (mean, standard deviation) are estimated, the null hypothesis of having a normal distribution is easily rejected. As a conclusion, a major deviation from the planned route may be stated although in reality only a minor problem occurred. Therefore, better statistical tests need to be applied in the future.

Despite these open issues, the presented approach is a step forward for Supply Chain Event Management as the current systems can not automatically find the reason for a disruption. Through the use of software agents, predefined events occurring in the supply chain can be detected reliably. The simulation results show that the response times in detecting a problem are quite good.

Our research results are limited by the fact that we have not been able to actually compare the quality of a diagnosis and the speed at which problems are detected with solutions obtained with the help of other agent-based SCEM approaches. The reason for this is that the known research prototypes for agent-based SCEM are not comparable. In the absence of a reference research question for agent-based SCEM, the methodological approaches underlying the prototypes address disparate research issues, and hence the prototypes support the solution of different types of problems.

OUTLOOK AND FURTHER RESEARCH

This paper presented an approach and a prototypical solution for advanced agent-based SCEM, extending existing approaches. However, there is still work to do in order to put the research concepts for agent-based SCEM into practice. Although the benefits resulting from SCEM adoption are indisputable, many companies do not yet apply SCEM solutions. A survey of German industrial enterprises showed, for example, that only 30% of these enterprises use an SCEM system (Teuteberg and Weberbauer, 2006). Apart from technological barriers and financial issues, managers are hesitant to switch to a completely new technology. It is argued that integration costs are high and prevailing evidence of successful SCEM implementations is lacking, resulting in uncertainty about the future ROI.

Integrating contextual, SCEM-specific information (such as the number of orders monitored, current production capabilities of the participants etc.) into the system would create additional benefits, because it would be easier to find and provide only data which are relevant for the user.

Automated problem solving in supply networks is a difficult task that needs further research. It is an interesting field for AI (Artificial Intelligence) concepts. Approaches following these concepts could help overcome the limitations of the rule-based approach and provide agents with better learning abilities.

Ontologies is another area where further research is required. Within our research project, ontologies were developed to enable agents to work with heterogeneous information systems and to exchange data in a standardized way. These ontologies, however, refer to one industry only (the bicycle industry). Further work to include universal ontologies, e.g. based on standards such as RosettaNet (RosettaNet, 2009), is needed to make the results applicable to industry environments.

REFERENCES

1. 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 (on CD ROM).
2. Compiere (2008) Compiere ERP and CRM Product Overview, http://www.compiere.com/products/ (accessed 18 Feb 2009).
3. Dangelmeier, W., Pape, U. and Rüther, M. (2004) Agentensysteme für das Supply Chain Management: Grundlagen, Konzepte, Anwendungen, Deutscher Universitäts-Verlag, Wiesbaden.
4. FIPA Foundation for Intelligent Physical Agents (2005) FIPA Specification Repository, http://www.fipa.org/repository/index.html (accessed 18 Feb 2009).
5. Frey, D., Mönch, L., Stockheim, T., Woelk, P. and Zimmermann, R. (2003) Agent.Enterprise – Integriertes Supply Chain Management mit hierarchisch vernetzten Multiagenten-Systemen, in Dittrich et al. (Eds.) *Tagungsband der GI-Jahrestagung*, Bonn, 47-63.

6. Gaonkar, R. and Viswanadham, N. (2004) A Conceptual and Analytical Framework for the Management of Risk in Supply Chains, *Proceedings of the IEEE 2004 International Conference on Robotics and Automation (ICRA '04)*, New Orleans, 2699-2704.
7. Gianella, C., Han, J., Pei, J., Yan, X. and Yu, P.S. (2004) Mining Frequent Patterns in Data Streams at Multiple Time Granularities, in Kargupta, H. et al. (Eds.) *Data Mining: Next Generation Challenges and Future Directions*, Menlo Park, CA, 191-212.
8. Hendricks, K.B. and Singhal, V.R. (2003) The Effect of Supply Chain Glitches on Shareholder Wealth, *Journal of Operations Management,* 21, 5, 501-522.
9. 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, Hawaii, http://ieeexplore.ieee.org/ielx5/6293/16788/00773059.pdf?arnumber=773059 (accessed 18 Feb 2009).
10. Jüttner, U., Peck, H. and Christopher, M. (2003) Supply Chain Risk Management: Outlining an Agenda for Future Research, *International Journal of Logistics*, 6, 4, 197–210.
11. Kärkkainen, M., Främling, K. and Ala-Risku, T. (2004) Integrating Material and Information Flows Using a Distributed Peer-to-Peer Information System, in Jagdev, H.S. et al. (Eds.) *Collaborative Systems for Production Management*, Boston, MA, 305-319.
12. Oracle Corp. (2005) Oracle Business Activity Monitoring – An Oracle Whitepaper, Oracle Corp., Redwood Shores, CA, http://www.oracle.com/technology/products/integration/bam/pdf/BAM_Whitepaper.pdf (accessed 16 Feb 2009).
13. Pokahr, A., Braubach, L. and Lamersdorf, W. (2005) Jadex: A BDI reasoning engine, in Bordini, R.H. et al. (Eds.) *Multi-Agent Programming*, Berlin, 149-174.
14. RosettaNet (2009) RosettaNet Standards, http://www.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/index.html (accessed 16 Feb 2009).
15. SAP AG (2007) SAP Event Management – SAP Solution in Detail – SAP Supply Chain Management, SAP AG, Walldorf, Germany, http://www.sap.com/solutions/business-suite/scm/brochures/index.epx (accessed 16 Feb 2009).
16. 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*, Maoui, Hawaii, CD ROM Edition, IEEE, Los Alamitos.
17. Szirbik, N.B., Wortmann, J.C., Hammer, D.K., Goossenaerts, J.B.M. and Aerts, A.T.M. (2000) Mediating Negotiations in a Virtual Enterprise via Mobile Agents, *Proceedings of the Academia/Industry Working Conference on Research Challenges*, Buffalo, NY, 237-242.
18. Teuteberg, F. and Weberbauer, M. (2006) Mobile Supply Chain Event Management: Eine empirische Studie zum aktuellen Stand in deutschen Unternehmen, *Mobile Internet Business* Nr. 6, ISSN 1861-3926, Frankfurt (Oder), Germany.
19. Tomlin, B. (2006) On the Value of Mitigation and Contingency Strategies for Managing Supply Chain Disruption Risks, *Management Science* 52, 5, 639-657.
20. Wieser, O. and Lauterbach, B. (2001) Supply Chain Event Management mit mySAP SCM (Supply Chain Management), *HMD,* 219, 65-71.
21. Zimmermann, R. and Paschke, A. (2003) PAMAS – an Agent-based Supply Chain Event Management System *Proceedings of the Ninth Americas Conference on Information Systems*, Tampa, Florida, http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1710&context=amcis2003 (accessed 18 Feb 2009).