AGENT-BASED PROCESS MONITORING: APPLYING EVENT MANAGEMENT PARADIGMS TO A STRATEGIC PLANNING PROCESS

Stefan Winkler
Roland Zimmermann
Bernd Vögele
Freimut Bodendorf
Department of Information Systems
University of Erlangen-Nuremberg
Lange Gasse 20, 90403 Nuremberg, Germany
{Stefan.Winkler; Roland.Zimmermann; Freimut.Bodendorf}@wiso.uni-erlangen.de

KEYWORDS

Process Monitoring, Agents, Event Management

ABSTRACT

The monitoring and coordination of planning processes requires a very flexible support with information systems. Due to the high amount of activities conducted in parallel as well as numerous interdependencies between inputs and outputs of every process step, present coordination concepts result in information deficits during process execution. In cooperation with an industry partner an agent-based information logistics concept is developed to reduce these deficits. The approach builds upon event management paradigms which have shown significant improvements in supply chain management. A prototype is realized and serves as the basis for evaluation.

PROBLEM AND OBJECTIVE

Sales planning processes in large companies encompass development of strategic guidelines, operational planning of sales quantity, turnover to be achieved, and future sales activities. All planning results have to be agreed upon by different organizational units and must then be aggregated before being approved by top management.

Further analysis considers specific sales planning processes of an industry partner. These processes are conducted according to a predefined process model by multiple actors in many different countries. The main problem is coordination of these activities which are often dependent upon each other but are not well synchronized. In many cases results of a preceding activity are required as a precondition for some succeeding activity. These results arrive too late and information on available results is not communicated. Thus, an information deficit exists among the actors who are responsible for the planning processes. A high degree of complexity of the coordination problem results from the fundamental characteristics of the processes:

- A variety of dependencies exists between individual activities and activity results. To initiate a single activity a set of inputs such as e.g. local sales quantities are required. These figures in turn are the output of preceding process steps.
- Outputs (and partial results) of certain activities within the planning process differ considerably, ranging from only slightly structured (e.g. texts or guidelines) to highly structured outputs (e.g. tables or spreadsheets).
- The planning process frequently consists of decision making activities rather than purely administrative functions. Employees have to make decisions based on results of previous process steps as well as additional information available to them. Resulting planning values are communicated to other actors.
- Temporal dependencies between sub-processes are of considerable significance. Changes to the time schedule are common during the course of the planning process.

Shortcomings in the Area of Information Logistics

Current process management at the industry partner is restricted to modelling processes and a large number of the resulting documents with a graphical notation. The process models represent the process steps in a clearly structured manner and are available for reference purposes to all actors involved in sales planning. However, from an information logistics perspective, no coordination support is provided and the identified information deficit is not tackled. Some characteristic shortcomings are:

- Existing enterprise applications are characterized by low levels of data and functional integration.

- A global data model as well as management of access rights and different document versions has not (yet) been implemented with respect to relevant planning documents.

- Existing coordination mechanisms take insufficient account of temporal changes and dependencies during the planning process.

- Automated feedback among actors in the planning process is not possible. All communication regarding process results has to be initiated manually.

Consequently, it is assumed that parties involved in the planning process do not have all information relevant to them and known to the enterprise at the time of the decision. Obtaining this information is associated with higher costs than necessary. Finally, effects of disruptions (e.g. delays) on subsequent process steps are not apparent to affected actors.

Requirements and Objectives

The overall objective of the suggested concept is to bridge the gap between simple process models and automated support of decentralized execution of the processes in question. As described above, process models are usually generated during the first phase of process management using fully developed modelling tools. To support the transition from process models to a distributed support system with a focus on the requirements of a flexible information logistics architecture, agent technology is employed.

To relieve actors of the burden of obtaining relevant information as well as getting the current status of related process steps from other actors, software agents are introduced which are able to act autonomously. According to the different needs of actors in the process steps, software agents proactively request information from various resources and communicate their status to the agents of all related activities. Users are notified every time the status of related process activities has changed (e.g. all input data is available) or certain deadlines during the course of the process are being exceeded.

To ensure usability of the agent-based system the approach has to cope with the various restrictions regarding the highly distributed execution of the planning process. Synchronization of tasks realized in parallel, information exchange using different means of representation as well as access to heterogeneous information resources such as business applications or database systems are just a few examples of further requirements of the process monitoring system.

Finally, changes in the planning process model (e.g. as a result of business process reengineering) shall be reflected in the agent-based information logistics system automatically to support integrated process monitoring. Therefore, all participating software agents implement a configuration component which allows them to adapt easily to changes concerning the process flow or the access to different information resources.

RELATED WORK

Common standard software products for the execution of business processes are Workflow Management Systems (WfMS). These systems are used to coordinate business processes. The integration of process modeling and workflow instantiation through translation of process models into workflow configurations is realized by Business Process Management Systems (BPMS). However, WfMS and BPMS are primarily applied to highly structured and frequently executed processes. In general, neither of these systems provides decision relevant information to the user in a proactive manner. This deficit is especially evident, if information is distributed among multiple applications and has to be collected and evaluated individually.

Service oriented support systems based on web services are increasingly important (e.g. BEXEE, http://bexee.sourceforge.net/). In particular, Service-oriented Architectures (SoA) offer flexible integration of different information sources as well as orchestration of web services which is for instance based on BPEL4WS or WS-BPEL. However, web services still lack the proactivity and autonomy necessary to provide information logistics services in processes that are not highly structured.

Software agents offer the ability to imitate human coordination and cooperation mechanisms due to their properties such as autonomy, reactivity, proactivity and social behavior (Jennings et al. 1998). Hence, software agents can offer information logistics support similar to that which human actors can provide.

Agent software has reached a level of sophistication that allows it to be used to design and implement industrial strength applications. Within a prototype at Daimler-Chrysler production of cylinders is coordinated with the aid of agent-based negotiation mechanisms (Bussmann and Schild 2000). Agents are used in electronic market places where they buy and sell goods. One provider of such solutions is e.g. Whitestein Technologies (http://www.whitestein.com). Further applications in particular for C2C markets are presented e.g. by Eymann (Eymann 2003).

Information logistics solutions are provided within the scope of Supply Chain Event Management (SCEM). Well-engineered concepts exist for inter-organizational monitoring of orders (Bodendorf and Zimmermann 2005, Zimmermann et al. 2004). To improve acceptance of agent based applications in the industrial context, different approaches for adjustable autonomy are discussed by e.g. Hexmoore (Hexmoore et al. 2003). Further research relevant to the information logistics problem has been conducted in the fields of robustness of agent systems as well as security and cryptography.

AGENT-BASED CONCEPT

Suitability

In general agent systems are particularly suitable for complex, decentralized systems because they effectively support the principles of decomposition, abstraction and flexible organization which are required for such a system (Jennings 2004).

The advantages of using agent technology as opposed to other options in the context of the process management task are due to agent characteristics:

- *Proactivity*: Agents actively request information from other agents on demand and communicate with their human users.
- *Reactivity*: Status inquiries are answered immediately whereas new status information is selectively broadcasted.
- *Social ability*: Agents are able to communicate with other agents and human users.
- *Autonomy*: Warnings are autonomously escalated to the responsible agents or users, if an input factor reaches a critical value.

The agent based approach adopted here supports a decentralized coordination. Agents are (partly) autonomous software components that represent autonomous units such as decision makers or parts of an organization responsible for decision making.

Agent Types

In the context of process monitoring agents support process activities as well as business roles. The system design is based on the assumption that there can be several decisions for each role (n:1-Relationship). But there should be only one person who is responsible for the decision (1:1-Relationship). This restriction also reflects the target state of the process. Hierarchical relationships between decisions ("aggregating decisions") are possible as well. This allows the system to support multiple process levels in hierarchical process models.

A so called activity agent is assigned to every activity which is relevant for a decision. This agent permanently monitors the state of the activity. The user (role) accesses the activity agents using another type of agent, the so called visualization agent. Possible access methods are status requests or the input of parameter values. The visualization agent is responsible for the representation of several decisions of a specific role. If necessary the representation is customized according to the role of the current user. The configuration is shown in Figure 1.

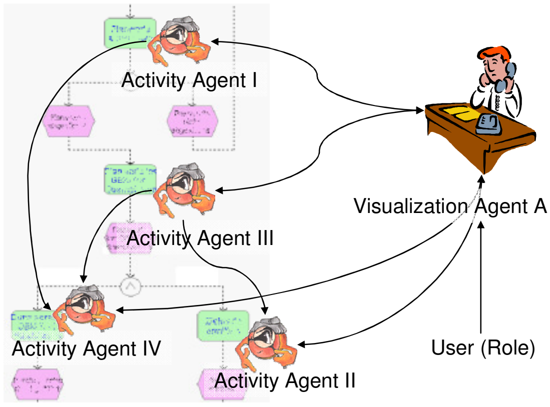

Figure 1: Agent society

Due to the characteristics of activity agents and visualization agents, both types can be easily adapted to a certain process model. To reduce implementation and maintenance of the activity agents their functionality is limited to support only one decision. Moreover, central management of all activity agents for one role is easier to handle and ensures a higher degree of availability.

The drawback of the use of different types of agents is the increasing complexity of the overall system which results from the extensive agent communication. Furthermore, companywide use would require an authorization management which accomplishes mapping between roles and associated activities.

Agent Behavior

The individual parties involved largely decide independently and with the aid of highly heterogeneous application systems (e.g. MS-Office products, data bases). Results of a decision are communicated and visualized by agents. Consequently, the level and quality of information available to all actors involved in a decision are improved.

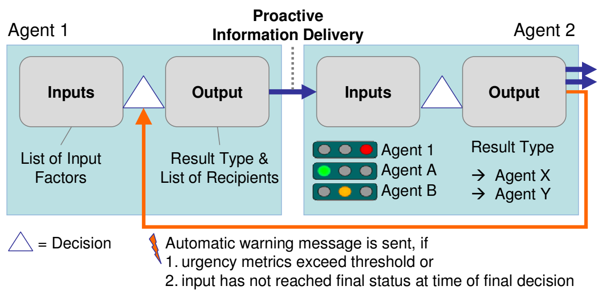

Figure 2: Agent-based information logistics

The basic principle of the information logistics concept used is the proactive delivery of information to and from the software agents. Changes in status and interim results are communicated by the responsible agent to all agents who are defined as parties interested in the result.

These in turn send warning messages if temporal or logical restrictions are violated (see Figure 2). The user interface is continuously updated with the current status of a decision.

Visualization uses a traffic light metaphor. If all results required as a precondition for an activity are available in time, the traffic light is set to green. If the due date of an activity is reached and e.g. a single input is still missing, the traffic light is set to yellow. Finally, a red traffic light notifies the user that a given deadline has been exceeded and necessary results of an activity are still missing.

Rule-based Communication

The agent system communicates with two different types of recipients according to a set of communication rules: On the one hand, there are agents and other software components. On the other hand, agents are able to communicate with human actors, the users. Hence, two categories of communication rules are supported:

- *Communication rules between agents:* These rules specify when an agent *A* has to send a warning or a message to an agent *B*. Furthermore, the type of the message is determined.
- *Notification rules between agent and user:* The notification rules are used by an agent *A* to decide under which circumstances the user has to be notified.

In the case of communication between agents the current state of the result is exchanged after acknowledgement by the user. Notification of a user is triggered by values of different variables. The *state* of a result type indicates its condition for a given user at the current time. The *urgency* is a numeric value that shows how urgent an input for a given decision needs to be delivered. Finally the *time* represents the system time of the agent system. In relation to a deadline for an activity, *time* influences *urgency* of follow-up messages.

On the communication level between software agents and human users notification rules determine the proactive behavior of the agent system. According to these rules the agents continuously evaluate the variables as described above and inform the user if e.g. a given threshold is exceeded. A configuration interface allows the user to adjust both types of communication rules.

Architectural Approach

The implementation approach is based on the *Presentation Abstraction Control* pattern (PAC pattern, see Buschmann 2002). With respect to the requirements of process monitoring and given standards (e.g. FIPA - Foundation for Intelligent Physical Agents, http://www.fipa.org) the architecture will be adapted as appropriate.

Two distinct types of agents and a layered agent model (PAC-Model) result hereby. Decisions which are subordinated in terms of hierarchy are represented as a specific input type and do not require a separate agent type. This serves to reduce the complexity of the system.

The separation of concerns (Dijkstra 1976) inside the system occurs between the activity agents and the visualization agents (see Agent Types). Each activity agent refers to exactly one decision making activity and examines its status. A visualization agent by contrast refers to a certain role and shows the user interfaces of several activity agents. Agents are divided into three separate layers respectively (see Figure 3):

- The *presentation layer* manages the user interaction.
- The *control layer* encapsulates the communication and business logic.
- The *abstraction layer* separates the data access.

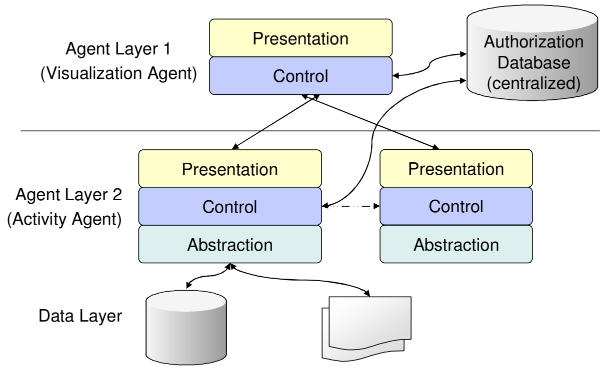

Figure 3: Agent layer model

Agent Interactions

The FIPA reference model specifies interaction patterns which for example enable FIPA-compliant agents to negotiate (see FIPA "contract net" as an example). In addition to such complex and predefined structures new interaction protocols can be defined on the basis of a request-inform-mechanism or other sub-protocols.

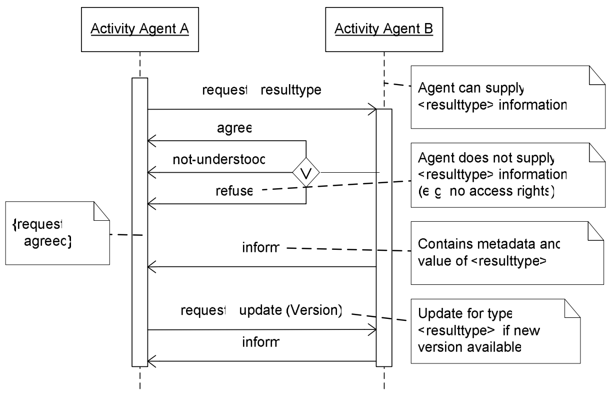

Figure 4: Communication protocol

Figure 4 shows the communication protocol of the agent system used by two agents. An activity agent *A* registers himself with an activity agent *B* to be kept informed about the state of the result type of agent *B*. The graphical notation represents a sequence diagram based on Agent UML (Odell et al. 2001), an extension of the widely used Unified Modeling Language (UML).

PROTOTYPE

The user interface represents a role's local view on all activities the role is responsible for. Each activity is represented by a tab. The basic layout of the user interface corresponds to the direction of the process flow: The relevant inputs are shown on the left. All input fields and additional information for the active decision is placed in the middle. Finally, the recipients of the decision result are shown on the right (see Figure 5). The different types of variables used for inputs or receivers are separated within different branches in the tree representation.

An example process (see Figure 6) is used to present the three main objectives and functionalities of the system: providing the user with easy access to relevant process information, communicating status changes as well as sending warning and feedback messages. Customization functions adapt the behavior with regard to the rules for communication of events and notification of users. Since many similar process instances are usually running in parallel, a time simulation component enables the system to simulate the effects of a particular setup in accelerated time or assess the projected urgency values of missing results at a given date. Therefore, this component may be used to find a suitable setup for communication rules to e.g. prevent information overload, or to help analyze the planning process as such.

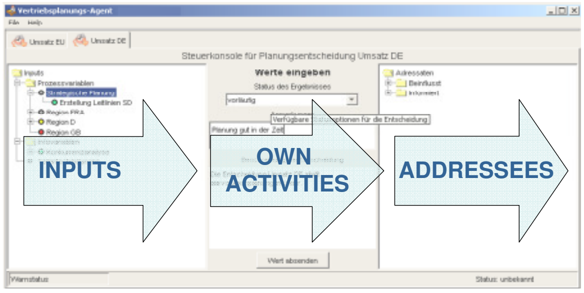

Figure 5: User guidance through GUI

The main window and several information and configuration dialogs offer direct access to the stakeholders. This includes technical process information from the company's process model documentation and contact information from the enterprise user directory. Additionally, the user can administer and update information regarding his own activity (e.g. status or remarks).

With regard to the urgency metric, which is calculated according to a predefined formula, the user can determine the threshold at which events are generated and propagated to the addressees' agents (i.e. customers) or passed on to the input-related agents (i.e. suppliers). During the course of the process the user is presented with a permanently updated view of all parties involved in or affected by each activity's result. This level of transparency is not limited to the immediate predecessors and successors in the process chain. All those other activities from which the user's activity has received warning messages can be viewed as well, thereby enabling the user to locate the origin and development of a disruption while filtering out those parts of the overall process that are irrelevant to him.

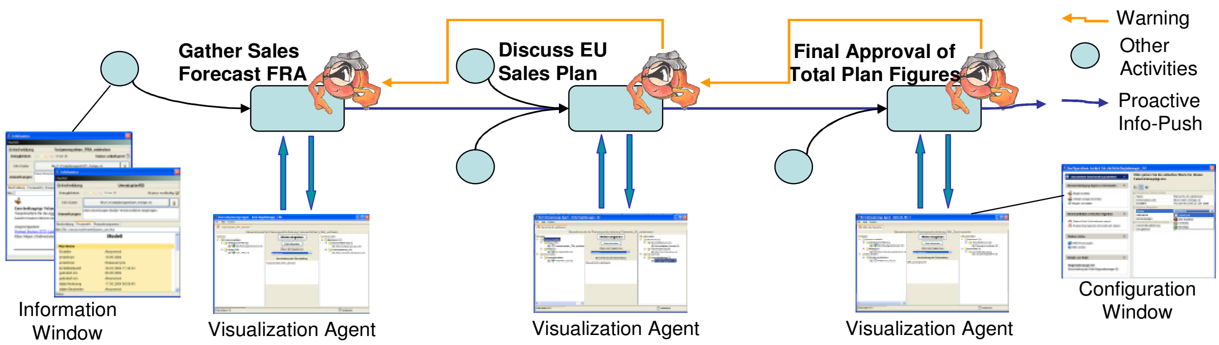

Figure 6: Example process in showcase

EVALUATION

In addition to the general suitability of the agent-based concept, factors concerning the existing system environment, the robustness of the infrastructure and the available know-how are of importance for the operative use in a company. Integration with process support software is generally conceivable, in particular the combination of agent systems with workflow management systems is proposed (Schönfeldt 2001). If strict company-wide IT-guidelines regulate the use of particular products, two approaches are viable: Either try to connect the prototype presented herein with the corresponding products via plug-in interfaces. Or alternatively implement the concept as far as possible using the means of existing products, possibly with a reduction in functionality.

In reviewing agent systems a distinction must be drawn between the technology and the paradigms of the agent based approach. While the use of a dedicated agent platform, as in this work, has its advantages, it is by no means a prerequisite for the development of software agents. In this regard, it should be examined whether the portal or middleware system already introduced in the company can be extended with reasonable efforts to include those features the presented agent system provides.

Regarding the fulfillment of ergonomic requirements the prototype was assessed using the criteria specified in the ISO 9241/10 standard (Prümper and Anft 1993):

- *Suitability for the task*: The prototype was specifically developed with the objectives of the use thereof in mind. Manual inputs are further reduced by virtue of the fact that it is connected to existing systems.
- *Self descriptiveness*: Technical terms should be known to the user (this can be assumed to be the case in the target group of expert users). A situation specific help system is still to be implemented.
- *Controllability*: A flexible method of working is possible because the dialogues are not modal and values are updated proactively. However, popup-warnings have to be confirmed.
- *Conformity with user expectations*: The user receives constant reports via the protocol function; due to the principle of agent systems, the response times of other agents cannot be predicted precisely.
- *Error tolerance*: The protocols and dialogues issue error reports, solution tips are still largely absent.
- *Suitability for individualization*: A high level of adaptability is achieved by using a rule-based system. A beginner mode has not been realized (and would only be of limited benefit because the software is used by experts).
- *Suitability for learning*: If the concept is known and understood, the functions are accessible via a graphic user interface and motivate the user to fully avail of the level of functionality available.

CONCLUSIONS

The proposed agent-based approach meets the demands of process monitoring. A showcase for a realistic part of the planning process of the industrial partner is realized and documented. Improved coordination reduces transaction costs and increases the availability of decision-supporting information for human actors. The information deficit is reduced. Furthermore, an initial assessment of the prototype from a user perspective based on an ISO norm questionnaire indicates that users accept this form of automated decision support. Further work focuses on integration of the prototype in a productive environment which requires certain adjustments of the underlying agent-based infrastructure. Finally, development of process monitoring is extended to fulfill the needs of different types of processes in varying application domains.

REFERENCES

Bodendorf, F. and Zimmermann, R. 2005. “Proactive Supply Chain Event Management with Agent Technology”. In *International Journal of Electronic Commerce* 9, Inderscience Enterprises Ltd., 57-90.

Buschmann, F. 2002. *Pattern-Oriented Software Architecture: A System of Patterns*. 2nd Ed. Wiley, Indianapolis.

Bussmann, S. and Schild, K. 2000. “Self-organizing manufacturing control: an industrial application of agent technology”. In *Proceedings 4th Int. Conf. on Multi-agent Systems*, ICMAS'2000. Boston, 87-94.

Dijkstra, E.W. 1976. *A Discipline of programming*. Prentice Hall, Englewood Cliffs, NJ.

Eymann, T. 2003. *Digitale Geschäftsagenten*. Springer, Berlin.

Hexmoore, H.; Castelfranchi, C.; and Falcone, R. 2003. *Agent Autonomy*, Kluwer Academic Publishers.

Jennings, N.; Sycara, K.; and Wooldrige, M. 1998. “A Roadmap of Agent Research and Development”. In *Autonomous Agents and Multi-Agent Systems* 1. Kluwer Academic Publishers, 7-38.

Jennings, N. 2004. “An Agent-based Approach for Building Complex Software Systems”. In *Communications of the ACM* No. 4, Vol. 44 (2004), 35-41.

Odell, J.; Parunak, H.; and Bauer, B. 2001. “Representing Agent Interaction Protocols in UML”. In *UML, Agent-Oriented Software Engineering*. P. Ciancarini and M. Woolridge (Eds.). Springer, Berlin. 121-140.

Prümper, J. and Anft, M. 1993. *Beurteilung von Software auf Grundlage der Internationalen Ergonomie-Norm ISO 9241/10*. FHTW Berlin.

Schönfeldt, B.F. 2001. *Einsatz intelligenter Agenten zur Entscheidungsunterstützung in prozessorientierten Unternehmen.* Dissertation, Universität Mannheim.

Zimmermann, R.; Butscher, R.; Bodendorf, F.; Huber, A.; and Görz, G. 2004. “Unternehmensübergreifendes Event Management mit Softwareagenten”, In *Supply Chain Management in der Automobil- und Zulieferindustrie*. W. Dangelmaier; D. Kaschula; and J. Neumann (Eds.), ALB-HNI, Paderborn, 79-88.