

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems
(AMCIS)

2005

Knowledge Engineering in Agent Oriented Business Process Management

Minhong Wang

City University of Hong Kong, magwang@cityu.edu.hk

Huaiqing Wang

City University of Hong Kong, iswang@cityu.edu.hk

Haijing Jiang

City University of Hong Kong, h_j_jiang@nuaa.edu.cn

Follow this and additional works at: <http://aisel.aisnet.org/amcis2005>

Recommended Citation

Wang, Minhong; Wang, Huaiqing; and Jiang, Haijing, "Knowledge Engineering in Agent Oriented Business Process Management" (2005). *AMCIS 2005 Proceedings*. 281.

<http://aisel.aisnet.org/amcis2005/281>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Knowledge Engineering in Agent-oriented Business Process Management

Minhong Wang

Department of Information Systems
City University of Hong Kong
magwang@cityu.edu.hk

Huaiqing Wang

Department of Information Systems
City University of Hong Kong
iswang@cityu.edu.hk

Haijing Jiang

Department of Mathematics
Nanjing University of Aeronautics and Astronautics, China
h_j_jiang@nuaa.edu.cn

ABSTRACT

The challenge of dynamic environment requires managing business processes with the ability to adapt to changes and to collaborate in activities. As a promising technology to process management, agent technology with its flexible, distributed and intelligent features has been studied in numerous studies. However, most existing approaches are special and ad-hoc. They have not looked much into the nature and characteristic of agents and their rational behaviours in process management. This paper intends to investigate the mechanism how to build intelligent agents in dynamic process management from the view of knowledge engineering. An agent-oriented approach to dynamic process management with its knowledge engineering is discussed, and a three-layer knowledge model of intelligent agents is proposed. By exploiting the knowledge involved in dynamic process management and transforming it into a computational model, this work provides an essential support of developing agent-oriented approaches to business process management.

Keywords

Knowledge engineering, intelligent agent, business process management, workflow.

INTRODUCTION

The challenge of dynamic environment requires managing business processes with the ability to adapt to changes and to collaborate in activities. As an extension and alternative to process management, the application of agent technology with its flexible, distributed and intelligent features has been studied in a flurry of research. However, most approaches are special and ad-hoc. They do not look much into the nature and characteristic of agents and their rational behaviours in process management. This paper intends to investigate the mechanism how to build intelligent agents in dynamic process management from the view of knowledge engineering. With a close relationship to agent-oriented software engineering, knowledge engineering is regarded as a basis of developing autonomous behaviours of agents in business applications. In this study, an agent-oriented approach to dynamic process management is discussed with its knowledge engineering method. Based on this, a three-layer knowledge model of intelligent agents in process management is proposed. A case study is demonstrated to illustrate how agent-oriented technology is applied to process management from the aspect of knowledge engineering.

BUSINESS PROCESS MANAGEMENT (BPM)

‘A business process is a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer (Hammer and Champy, 1993). According to (Davenport, 1993) processes are defined as “structured, measured sets of activities designed to produce a specified output for a particular customer or market”. Business processes often display complexity because of interactions of their internal components and interaction of the process with its environment. Rather than using machine metaphor to describe processes as an assembly of interchangeable components, researchers have addressed the complex, dynamic and interactive features of business processes (Melão and Pidd, 2000). Nowadays, management of business processes has been a fundamental aspect in most business applications. Research of business process management (BPM) can be seen in the fields of business management, information technology and others. From

management view, BPM can be seen as a collection of methodologies, techniques and tools supporting the analysis and improvement of business processes (Melão et al., 2000). From technology view, BPM is to support business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents, and other sources of information (Weske, Aalst and Verbeek, 2004).

An area of research closely related to BPM is workflow technology. It is a promising approach aiming at modelling, executing, and monitoring business processes. Workflow technology usually refers to the development of Workflow Management System (WFMS), in which the overall process logic is explicitly represented in an executable process or workflow definition. A fundamental assumption in most WFMSs is that workflow schema are predefined. Hence, a significant weakness of predefined workflow schemas is their lack of flexible mechanisms to adequately cope with ever-changing and complex environments, e.g. complex process control and collaborative tasks (Jennings, Faratin, Norman, O'Brien, and Odgers, 2000; Aalst, 2003). To meet the challenge, numerous new approaches and technologies have been addressed and under research. Among them, the agent-based technology is a fundamental new approach by giving a novel conceptualization of process management to overcome the limitation of traditional workflow approaches (Jennings et al., 2000).

INTELLIGENT AGENTS WITH BPM

The concept of agent has generated many excitements for many years. The term agent is used to denote an encapsulated software-based computer system that enjoys the properties of autonomy, social ability, reactivity, and pro-activity (Wooldridge, 2002). An agent is situated in some environment; it has a set of goals, certain capabilities to perform actions, and some knowledge (or beliefs) about its environment. Agent-oriented software engineering provides designers and developers with a new way of structuring an application around autonomous, communicative elements, and leads to construction of software tools and infrastructure to support the design metaphor. A key point is that agent-oriented software is well suited for complex – open, networked, large and heterogeneous – application domains. In complex systems, it is impossible to foresee all interactions at design time. By modularizing a complex problem in terms of multiple autonomous components that can act and interact in flexible ways, agent-oriented techniques are well appropriate for complex, dynamic, and distributed software systems (Jennings, 2001).

As an extension and alternative to process management with flexible, distributed, and intelligent features, the application of agent technology to process management has been studied in a flurry of research (Zhao, Nunamaker and Briggs, 2002; Kuo, 2004). Agents in workflow management systems may act as personal assistants, performing actions on behalf of the workflow participants, continuing watching for information and responding to it when it meets certain specified criteria. Agent technology is proposed to automate process operation integrating the different business systems to reduce the complexity. An agent-based workflow system usually consists of multiple agents, each responsible for specific work items. The management of the whole process is performed by one routing agent through its mediating with other agents. In other cases, the whole process is broken into pieces and embedded into each agent.

KNOWLEDGE ENGINEERING OF INTELLIGENT AGENTS IN BPM

Though agent technology has been used in number of applications of business process or workflow management, the approaches involved in most applications are somewhat special and ad-hoc. They do not look much into the nature and characteristics of agents and their rational behaviours in process management. More efforts are needed to investigate the mechanisms how to employ agent-oriented technology into business process management, especially focusing on solving the problem through the essences of agents (Wang and Wang, 2005). This section is to investigate the mechanism how to build intelligent agents in dynamic process management from the view of knowledge engineering.

Knowledge Engineering and Agent-oriented Engineering

Knowledge engineering has evolved from the late 1970s onwards, from the art of building expert systems, knowledge-based systems, and knowledge intensive information systems. They are used to aid in human problem solving with the benefits of faster decision-making, increased productivity, and increased quality of decision-making. By its very nature, knowledge engineering has a very close relationship with the engineering of agents and multi-agent systems (more specially, of knowledge bases used by agents) (Weiß, 2001). For this reason it appears to be promising to think about development approach based on established knowledge engineering methods.

In a multi-agent system, software agents are proposed to perform tasks autonomously on users' behalf, which means that they can act independently of humans. It is essential to design a set of autonomous type of behaviour for the agent class, including reactive, proactive, and cooperative behaviour. Such kind of autonomous behaviour relies on the knowledge or perception of

the environment and the knowledge of problem solving. Various kinds of intelligence are supported by this kind of knowledge. In order to fulfill the organizational goals and objectives, business policies or rules to conduct business activities play an essential role in governing peoples' behaviour. These business rules may form an important part of knowledge base for software agents to perform delegated tasks on the users' behalf. Once the rules are understood, captured and represented in form of logic, they will serve as a basis for building intelligent agents to perform rational activities (Wang, Wang, Xu, Wan and Vogel, 2004).

Intelligent Agents Supported Business Process Management

Considering the large-scale and distributed settings in today's business environment, business activities can be delegated to a number of autonomous problem-solving agents, each of which plays a specific role in business activities. In addition, in a dynamic and complex business environment, process management is more like a real time dynamic decision-making task. It is a type of task that requires a series of interdependent decisions in a continuously changing environment (Wang et al., 2005; Lerch and Harter, 2001). With a view to real time dynamic tasks in BPM environment, an agent-oriented approach is characterized by the ability to continuously perceive the environment and make real-time decisions of tasks according to underlying business logic (Wang et al., 2005). The mechanism of this kind of approach is elaborated in Figure 1. The evolution of business processes is driven by changes from environment and runtime decision of tasks in current situation. Each agent may orchestrate business activities dynamically at runtime and continue the evaluation of process throughout execution, during which business changes occur and business rules are dynamically bound to the decision of tasks. The changes from environment may activate tasks, and activated tasks may produce new changes into the environment and subsequently invoke the next round of decision making of tasks.

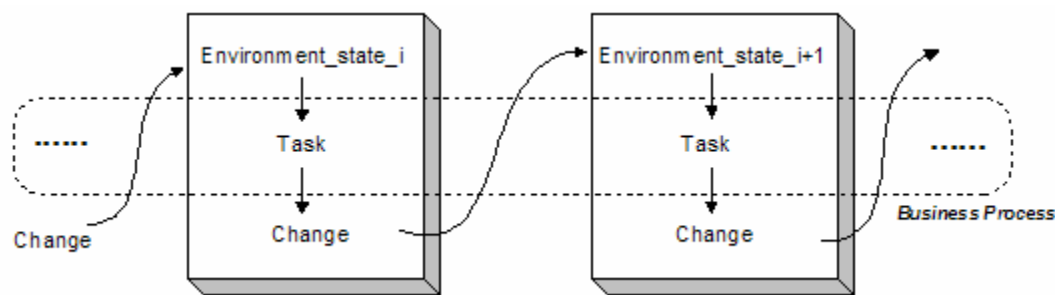


Figure 1. Mechanism of Agent-oriented Business Process Management

Knowledge Engineering of Intelligent Agents in BPM

Based on the above-proposed agent-oriented approach to dynamic process management, the architecture of agents in dynamic BPM is presented in Figure 2. It consists of functionalities, knowledge base, and communication facilities. The functionalities execute different operations in dynamic BPM, such as situation perceiving, process controlling, and tasks execution; they are the action part of an agent. Knowledge is required by each agent to perform its internal and external activities. It may contain environmental information, knowledge of process management, and expertise of particular tasks. The communication facilities envelope each agent, and provide access to it and serve communication with the outside.

As mentioned in before, for an agent-oriented approach, it is essential to think about its development based on the knowledge engineering perspective. In all applications of knowledge engineering, the conceptual modeling of knowledge at different levels of details is a central topic (Schreiber, Akkermans, Anjewierden, deHoog, Shadbolt, VandeVelde and Wielinga, 2000). As presented in Figure 2, we propose the knowledge model of agents in three layers, termed as **situation knowledge**, **process knowledge**, and **problem solving knowledge**. The agent-oriented approach proposed in this paper is characterized by the continuous perception of the environment and real-time control of processes. Therefore, environmental knowledge and process knowledge are concerned. Furthermore, problem solving knowledge used for performing specific tasks is also essential for agents to perform autonomous behaviours. In this model, we separate process knowledge from problem solving knowledge. The separation may enable the run-time system to assist users in coordinating and scheduling the tasks for problem solving, and contribute to simplify and speed up application development. The details of the knowledge model are illustrated as the follows. For easy understanding, some examples are used for illustration. They are written in JESS (Java Expert System Shell) language, which is a rule engine and scripting environment written in Sun's Java language (<http://herzberg.ca.sandia.gov/jess>).

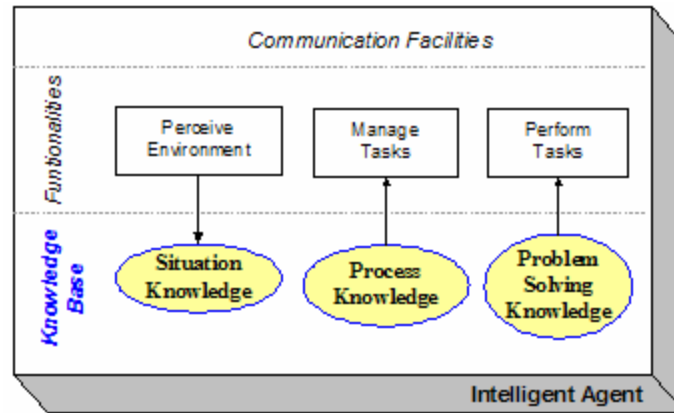


Figure 2. Architecture of Agent in Business Process Management

Situation Knowledge

An agent is situated in the environment by perceiving situation data and in the society by message communicating with other agents. In order to interact with the external world, the agent needs to obtain the information of the process environment, which may include information about various resources, events, states of tasks, etc. The information of events, resources and tasks can be detected by the system, and represented as facts for decision-making. This kind of knowledge forms the agent's dynamic beliefs about its environment. It can be described in form of a collection of patterns that model different classes of facts. For example, the event of receiving an order request No.15421 from a customer can be interpreted as the fact below.

(event (e-type order_request) (ord-no 15421))

Process Knowledge

Agents are able to act with the environment without the intervention of humans or other systems. They have control over their processes. The knowledge for control usually concerns business rules, which are the user's expression of policies followed by agents to manage their tasks. These rules may form an important part of knowledge base for agents to manage delegated tasks on users' behalf. At a high level, the rules could be classified under one or more concerns, such as task routing, operational constraints, process monitoring (Morgan, 2002), etc.

Rules on **process routing** are used for scheduling tasks or activities. Generally, there are four basic types of process routing including sequential routing, conditional routing, parallel routing, and iterative routing. Each type is specified using AND/OR and split/join blocks, and can be mapped into rules to manage the sequence.

Rules about **operational constraints** are used to exercise control of processes, and prohibit unauthorized operations. Such rules can be specified on the basis of business requirements.

In order to keep track of the changes in the process environment to reduce business risks, rules for **exception monitoring** can be used to capture real-time status of business processes and take proactive actions when necessary.

As related before, business rules could be used not only for controlling process but also for reducing business risk, making efficient use of resources and improving customer service. Such type of rules can be regarded as **business strategies**, though most of them are implicit and difficult to translate into explicit rules.

The example below is to reject the cancellation of an order that has been confirmed.

(defrule reject-confirmed-order

(event (e-type order_cancel_request) (ord-no ?no))

(order (ord-no ?no) (confirmation ?cfm))

=> (if (= ?cfm confirmed) then

(assert (task (t-type reject_cancel_order) (ord-no ?no))))

```

else (if (= ?cfm NULL) then
  (assert (task (t-type cancel_order) (ord-no ?no))))))

```

Problem Solving Knowledge

Agents are expected to perform various tasks or solve problems on behalf of humans. Problem solving knowledge of an agent describes its problem solving capability in form of methods and strategies. As an example, the rule below specifies how to capture orders of missing items.

```

(defrule capture-missing-items
(order (ord_no ?o_no) (product_id ?p-id) (quantity ?q) (shipping_time ?s-time))
=> (if (or (eq ?no nil) (eq ?p-id nil) (eq ?q nil) (eq ?s-time nil)) then
  (assert (item_missing ?o_no)))

```

A CASE STUDY

Based on the above agent-oriented approach to BPM with its knowledge engineering method, a case of exception management in securities trading is elaborated in this section. This case illustrates how this approach is used in process management mainly from the aspect of knowledge engineering.

Case Description

With rising trading volumes and increasing risks in securities transactions, the securities industry is making an effort to achieve straight through processing to shorten the trade lifecycle and minimize transaction risks. While attempting to achieve this, exception management is critical to pass trade information within the trade lifecycle in a timely and accurate fashion. However, the major issue of securities exception management has not been adequately addressed, and the mechanism of automating exception management is still under investigation (Guerra, 2002). This project aims to automate the identification and resolution of exceptions to assist securities industry to gain quicker competitive advantage.

Prototype System

Considering the distributed environment and complex process of exception management in securities transactions, the agent-oriented approach is employed for managing exceptions in trading and settlement processes. Based on the analysis of exceptions occurring in securities transactions, a society of intelligent agents is proposed. Trade Detail Monitoring Agent (TDMA) is to capture unusual components in trade details, such as a trade dealt at a price significantly different from the market price, incorrect calculation of trade cash value, missing components, and so on. Trade Status Monitoring Agent (TSMA) is to keep watch on the agreement status of each trade, and detect the trades that have not been matched or agreed by trade parties in a timely manner. Diagnostic Agent (DA) is to investigate the nature of exceptions and offer resolution advice to settle the problems. Resolution Agent (RA) is to carry out actions on exceptions based on the output of the diagnosis.

These agents are deployed in organizations or departments involved in securities transactions, e.g. trading companies, settlement companies, etc. They can communicate with each other through the network. Usually, TDMA is located with the trading system, while TSMA is placed inside the settlement system. They are used to monitor exceptions in trade details and trade agreement status respectively. DA can be set up in relevant management department for assisting in exceptions diagnosing. Based on the report of the DA, RA may help managers carry out actions to resolve problems.

Based on the agent architecture and knowledge model addressed before, the implementation structure of agents is presented in Figure 3. An agent can continuously perceive the environment by receiving internal messages and external information through the communication facility. The messages are interpreted and translated into facts describing the environment. Based on the facts, appropriate tasks are scheduled and then activated. The performed tasks will produce outputs, and result in changes that may activate a next round of decision making of new tasks. The prototype system has been built using Java Web Services Development Package (JWS DP) (<http://java.sun.com>) and JESS (Java Expert System Shell) (<http://herzberg.ca.sandia.gov/jess>). The communication of agents with outside is based on XML language.

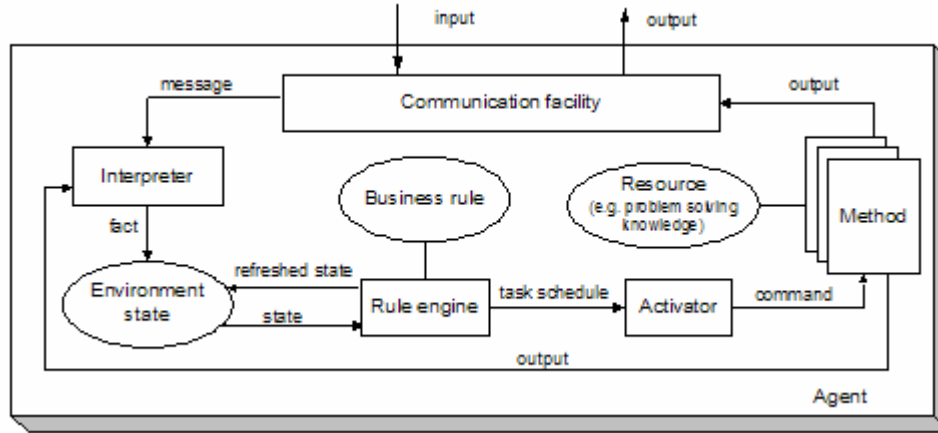


Figure 3. Implementation Structure of Agent

Knowledge Engineering

In the prototype system, each agent is built as a cognitive entity equipped with specific business knowledge, based on which it controls its internal states and activities. Due to the limited space of this paper, we only present the knowledge model of DA (Diagnostic Agent) as the follows.

Situation knowledge of an agent is attained by interpreting the information of the environment, such as the state of events, resources, and tasks. The following example describes an error report received by DA. It reported that Trade No.12362 was pending, i.e. not been matched in a timely manner.

```
(event (e-type error_report) (err-no 102)(err-type pending) (trd-no 12362) (rep-time 1.009598657E9))
```

As another example, the fact below shows a diagnostic task on Error No.102 has started and not yet completed.

```
(task (t-type diagnose) (err-no 102) (start-time 1.009598657E9) (finish-time NULL))
```

The **process knowledge** usually concerns business rules, which are the expression of business logic followed by the agent to manage its tasks. For example, rule-1 below is to start a diagnostic task when an error is reported.

```
(defrule rule-1 "start diagnosis")
```

```
(event (e-type error_report) (err-no ?e_no))
```

```
=> (assert (task (t-type diagnose) (err-no ?e_no) (start-time (time))))))
```

Rule-2 is to keep watch on diagnostic tasks, and send out reminding messages if they have not been completed within a specified time frame.

```
(defrule rule-2 "remind diagnosis")
```

```
(task (t-type diagnose) (err-no ?e_no) (start-time ?t1) (finish-time NULL))
```

```
(test (> (- (time) ?t1) 1200))
```

```
(not (remind-history (r-type remind_diagnose) (err-no ?e_no)))
```

```
=> (assert (task (t-type remind_diagnose) (err-no ?e_no)))
```

```
(assert (remind-history (r-type remind_diagnose) (err-no ?e_no)))
```

For more attentions to special trades, resolutions to large trades need to be confirmed by experts. This policy is specified in rule-3 below.

```
(defrule rule-3 "confirm resolution advice")
```

```
(event (e-type resolution-advice) (trd-no ?t_no))
```

```
(large-trade (trd-no ?t_no))
```

```
=> (assert (task (t-type confirm_resolution) (err-no ?e_no) (start-time (time))))))
```

Problem solving knowledge of an agent describes its capability to perform specific tasks or solve specific problems. The following rule-4 is an example that represents the knowledge of exception diagnosing. When a trade is detected with pending status, i.e. a trade that has not been matched in a specified time, DA will check if the trade agreement has been replied by the counterparty.

```
(defrule rule-4 "check agreement_reply"
```

```
(error-report (err-no ?e_no) (trd-no ?t_no) (err-type pending)
```

```
(transmission-record (trd-no ?t_no) (send-status ?s_status) (reply-status ?r_status))
```

```
=>
```

```
(if (and (eq ?s_status successful) (eq ?r_status nil)) then
```

```
(assert (diag-report (err-no ?e_no) (trd-no ?t_no) (diag-detail unreplied_agreement))))))
```

Evaluation

A demonstration was conducted to elicit end-user satisfaction regarding the prototype. User satisfaction with the prototype system was measured using the 12-item instrument developed by (Doll and Torkzadeh, 1988) that has subsequently been applied in numerous studies. Targeted for an end user computing environment, their instrument identifies five components of user satisfaction: (1) content; (2) accuracy; (3) format; (4) ease of use; and (5) timeliness. Subjects were asked to rank their satisfaction on a scale from 1 to 5, where 1 represents *Strongly Disagree*, 3 is the *neutral* point and 5 represents *Strongly Agree*. The demonstration was conducted in August 2003 for 36 subjects (14 female and 22 male) who were undergraduate students from the Faculty of Business in a major university in Hong Kong. During the demonstration, the subjects were introduced to prototype capabilities relating to exception management in security trading. Prototype system simulations were demonstrated. At the conclusion of the demonstration, subject feedback was collected. Questionnaires elicited demographic data, basic knowledge on security trading and security trading process, satisfaction and usage intentions.

The reliability of the 12-item instrument of user satisfaction was 0.89. The reliability (alpha) of each individual construct was above 0.70. Fully 89% and 83% of the subjects reported that they had basic knowledge about security trading and security trading processes, respectively. However, their previous knowledge about security trading and security trading processes did not affect their perceptions regarding satisfaction ($p > 0.5$). The mean of overall satisfaction and the five constructs are all above 3.3. The data collected show positive results of the initial evaluation study (Wang et al., 2004).

CONCLUSION

One basic premise of agent-oriented business process management is that business process must adapt to the constant changes in the business environment. This work results in dynamic process management since it offers high flexibility and knowledge support for conducting business activities. By exploiting the knowledge for dynamic process management and transforming it into a computational model, this work provides an essential support of building agent-oriented systems to business process management. One limitation of this approach arises from the complexity of business logic, such as identification of business logic from business activities, correct representation of rules in a business sense, and maintenance of rules. The future work may refine and extend this research by building the pattern of business logic underlying common business activities, and design user-friendly language support for rule specification and manipulation for process management.

REFERENCES

1. Aalst, W.M.P. van der (2000), Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries, *Information and Management*, 37, 2, 67-75.
2. Davenport, T.H. (1993), *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, Boston, MA, USA.
3. Doll, W.J. and Torkzadeh, G. (1988), The Measurement of End-User Computing Satisfaction, *MIS Quarterly*, 12, 2, 259-274.
4. Guerra, A. (2002), Exception Management: The Safety Net You've Been Looking For? *Wall Street & Technology Online*, URL: <http://www.wallstreetandtech.com>

5. Hammer, M. and Champy, J. (1993), *Reengineering the Corporation: A Manifesto for Business Revolution*, Breaaley, London.
6. Jennings, N.R., Faratin, P., Norman, T.J., O'Brien, P. and Odgers, B. (2000), Autonomous Agents for Business Process Management, *International Journal of Applied Artificial Intelligence*, 14, 2, 145-189.
7. Jennings, N. R. (2001), An agent-based approach for building complex software systems, *Communications of the ACM*, 44, 4, 35-41.
8. Kuo, J. (2004), A document-driven agent-based approach for business processes management, *Information and Software Technology*, 46, 6, 373-382.
9. Lerch, F.J. and Harter, D.E. (2001), Cognitive support for real-time dynamic decision making, *Information systems research*, 12, 1, 63-82.
10. Melão, N. and Pidd, M. (2000), A conceptual framework for understanding business processes and business process modelling, *Information Systems Journal*, 10, 105–129.
11. Morgan T. (2002), *Business rules and information systems*, Addison-Wesley, New York.
12. Schreiber, G., Akkermans, H., Anjewierden, A., deHoog, R., Shadbolt, N., VandeVelde, W. and Wielinga, R. (2000), *Knowledge engineering and management: the CommonKADS methodology*, London, MIT Press.
13. Wang, M., Wang, H., Xu, D., Wan, K.K. and Vogel, D. (2004), A Web-service Agent-based Decision Support system for Securities Exception Management, *Expert Systems with Applications*, 27, 3, 439-450.
14. Wang, M. and Wang, H. (2005), Intelligent Agent Supported Business Process Management, *Proceedings of 38th Hawaii International Conference on System Science (HICSS-38)*.
15. Weiß, G. (2001), Agent orientation in software engineering, *The Knowledge Engineering Review*, 16, 4, 349-373.
16. Weske, M., Aalst, W.M.P. van der and Verbeek, H.M.W. (2004), *Advances in business process management*, *Data & Knowledge Engineering*, 50, 1, 1-8.
17. Wooldridge, M. (2002), *An introduction to multiagent systems*, England, J. Wiley.
18. Zhao, J.L., Nunamaker, J.F. and Briggs, R.O. (2002), Intelligent Workflow Techniques for Distributed Group Facilitation, *Proceedings of the 35th Hawaii International Conference on System Sciences*