Business Process Reengineering and Software Agents Development

Malgorzata Pankowska[1], Henryk Sroka[2]

[1] University of Economics, Information Systems Department, Bogucicka 3 street, 40 226 Katowice, Poland
pank@ae.katowice.pl

[2] University of Economics, Information Systems Department, Bogucicka 3 street, 40 226 Katowice, Poland
sroka@ae.katowice.pl

Abstract. In the paper it is assumed that Business Process Reengineering (BPR) is the prerequisite and consequence of new software and Information Technology (IT) solutions implementation. Process reengineering includes process modelling as well as roles, rules and information modelling. In the paper the main question is if agent methodologies are BPR-oriented. Analysis includes the current multiagent methodologies like AAII, Gaia, MaSE, Comparative Information Agent Design, ADEPT project, MAS-CommonKADS, CoMoMAS, DESIRE and Cassiopeia.

1 The Move to Business Process Reengineering (BPR)

Organisational theorists propose that the organisation of the future will be networked across functions and designed around business processes rather than functional hierarchies [22], [14], [5]. BPR is now being offered as a paradigm of organisational change necessary in order to achieve the required flexibility and competitiveness of the networked organisation [20]. BPR is a strategy-driven organisational initiative to fundamentally re-examine and redesign business processes with the objectives of achieving competitive breakthrough in quality, responsiveness, cost, satisfaction and other critical process performance measures [36].

While business processes can be reworked without Information Technology (IT), recent technological advances have placed greater importance on IT as an enabler of BPR. Increasingly, BPR is being deployed in tandem with the use of IT to revamp or overhaul existing business processes that limit effectiveness. Technologies such as Local Area Network (LANs), client-server architecture, Electronic Data Interchange (EDI), Executive Information Systems (EIS) and Decision Support Systems (DSSs) are some examples of IT which now allow firms to achieve performance gains in the communications dimensions of business processes. The emerging technologies of video/teleconferencing, GroupWare, image processing and workflow technologies are

enabled by BPR and by reducing or replacing manual tasks and improving communication. Legacy systems already in place must be selectively destroyed and replaced by cross-functional systems that allow many departments to share a single data warehouse or utilize data simultaneously received form different sources.

BPR has been defined as the restructuring of organizational processes through the innovative use of information systems and technology. Reengineering is about rejecting the conventional wisdom and received assumptions of the past. Reengineering is the search for new models of organization work [15], [7]. BPR requires changing the basic assumptions and principles of management, re-examining the nature of processes and creating changes of magnitude through innovation.

2 Processes and Role-oriented Methodologies for BPR

Business processes are considered as goal-directed, socially coordinated activities, established through social actions. A process as a flow of activities is to produce an output of value to recipient. Each process can be viewed as an exchange of services embedded in a recipient-supplier relationship. The recipient (customer, client, patient) requests the desired output whereas the supplier performs the process. The process's activities are the means to transform process input (materials, information) into the desired output. During its performance the process changes states, which are triggered by events. The process flow is determined by business rules (conditions). Although a process can be viewed on different levels of abstraction, depending on the domain of an analysing person, each process follows these outlined principles.

Most manufacturing and some office processes are treated as simple and mechanistic. They are modelled as linear, sequential and repeatable series of activities. These processes are interpreted as manipulation of physical objects and controlled flows of tasks e.g. inventory control, manufacturing scheduling, office automation, transaction processing. However upper and top management processes that include knowledge management and lead through idiosyncratic decisions made individually or in groups to unique values, are much more complex and requiring new IT tools and software agents for information searching and processing. For example managers take up normatively regulated interactions and they seek to reach understanding about the problem or situation in order to coordinate their actions by way of cooperating processes of interpretation, argumentation and agreement e.g. problem identification, diagnosing, solving management and strategic problems by boards, groups or communities, defining organizational mission, goals, policies and strategies.

Supporting BPR practitioners requires models that can be used to describe the processes to be redesigned. In BPR functionality specification has originally come from Yourdon [37], Coad and Yourdon [4], and Rumbaugh et al. [27] analyses. The business models should be aimed at depicting cross-functional business processes. The models are to give a basis for simulation of the existing processes and proposed re-engineered processes [12]. The purpose of any software requirements technique is to understand

and document the needs of the user and the external behaviour of the solution. Software requirements engineering includes two primary groups of activities: problem analysis and requirements specification. Problem analysis is the activity of learning all aspects of the problem domain so as to determine how best to solve a specific set of user needs. Problem analysis is essential for any system development for which the problem being solved is not totally self-evident. Requirements specification is the activity of documenting in a software requirements specification (SRS) the expected external behaviour and external characteristics of a system that solves the problem [8]. "Expected external behaviour" means a description of all inputs to the system all outputs from the system and all the possible mapping relationships between the inputs and outputs [8]. Although requirements engineering is for software design it is difficult to distinguish the activities undertaken in the name of BPR from systems' implementation. This was due to the origins of many early BPR enthusiasts. Information technology was regarded as the "enabler" for BPR [24]. BPR efforts were launched to prepare the ground for more effective software implementation. The supporters of IT-based BPR often take one view, that the "systems is the solution" [24]. However, it is simply not good enough to spend money on new technologies and then to use it in the same old way. Therefore the solution is in the problem analysis.

According to Butler [2] a model for BPR should include development of vision and objectives, identification of processes for redesign, understanding and measuring existing processes, pilot/trial new process, developing supportive solutions, making new processes operational and continuous process improvement. So researchers like Butler, Davenport underlies the need of analysis of legacy processes, simply audit of them. Next they opt for modelling a new activities [7]. This analysis is believed as necessary although the important goal of reengineering is to radically free information systems from the past organizational structures.

Many researchers emphasize that BPR methodologies should cover not only activities (what?) modelling but also roles (who acts?). According to Hawryszkiewicz model of business processes includes artefacts, roles, actors, environments and tasks specifications [16]. Artefacts are representation of organization's information base including files, reports, and designs. The artefact includes data values. Roles are abstract entities that represent system decisions. An actor is assigned to each role to make the decisions. Actors are often positions that can be derived using organizational rules. Roles are not permanently associated with positions but are dynamically assigned using organizational rules. A person can have thus many roles and there is nothing to prevent a person from transferring knowledge between roles as often happens in organizations. Environments provide the support structures for workgroups. Environments can be of prolonged duration and support a variety of processes each composed of tasks and roles. These processes can combine the tasks objects and roles into workgroup procedures. Environments can include other environments, as well as tasks, roles and data. An environment must include at least one role and usually covers one workgroup. Tasks represent a complex transactions or simple work tasks such as editing a document.

Similarly Walford wrote about the need of rules, roles, dialogs, workflows specification for business process redesign and implementation [31]. Business rules are state-

ments that articulate an operating principle by defining or constraining some aspect of the business. The strength of a rule is the rigor with which the rule needs to be followed. In that regard rules can be classified into requirements, standards, guidelines. Business rules as applied to processes are of particular interest in the context of the process life cycle. So business rules are constraints on the business processes or their relations. Business rules also can be applied to different aspects of a process and its implementation, including the flow control of the process, the functionality required by the process, the data used in the process, and the assignment of human performers to the process.

According to Walford a role is an unordered set of cohesive activities. It is defined by the collection activities assigned to it. By that definition, a role is passive and because of that, there must be some mechanisms for the animation of a role. In this case animation means causing the activities defining a role to occur. The mechanism for animation is through the use of a role performer. Walford introduced to process modelling the concept of dialogs, which are interpreted as a portion of a process. Dialogs define a workflow that is implemented using reusable components and a suitable control mechanism.

The necessity of activities, rules and roles modelling for BPR is emphasized in Systematic Technique for Role and Interaction Modelling (STRIM) approach [26]. Similarly Curtis et al. specify agents and roles in their method of process [6]. According to them agent is a human or machine who performs a process element what we call an actor. Role is defined as a coherent set of process elements to be assigned to an agent as a unit of functional responsibility.

Proposed by Sun and Liu Method for InteracTive Articulation of Information Space (MITAIS) consists of two major phases: domain analysis and modelling and articulation of decision space and configuration of information space [29]. Domain analysis and modelling involves process analysis, agency structuring, role/activity analysis and finally information analysis. The process analysis focuses on the process aspects at the enterprise level. Agency structuring conducted at the same time as the process analysis for their interdependency, identifies human actors (i.e. agents) their roles and responsibilities within the organizational structure. Role/activity analysis further identifies the activities involved in by each role. Information analysis determines the information requirements for each role and activity. The phase produces process models from the process analysis. These models represent the work processes of the problem domain. Process analysis in the MITAIS method adopts Unified Modelling Language (UML), specifically the techniques of use case and interaction diagramming.

Agency models are created based on the process analysis and analysis of agent structure (i.e. agency structuring). This is a conceptual model that identifies agents, their roles and relationships. Decision question repository is a collection of decision question related to the problem domain. This repository is a result of the role/activity analysis, which examine agents in each role and their management and decision-making activities involved. This repository contains all the possible questions from the agents in their roles, which will be used to facilitate an articulation process in the second phase. The last product of this phase - Information category repository resulting from the information analysis takes input from the agency models.

The second phase consists of two analyses. The articulation of decision space takes input from the agency model and the decision question repository. It enables a user to express the question concerned and facilitates the user to specify the question in a more articulated manner. The user will have opportunities to interact during the process and to confirm the representation of the question in terms of dimensions of the decision space. Developers of MITAIS have understood that the decision-making process involves monitoring the business process.

The major drawback of many process representations is that they are static, fixed, and stable. Models describe the current state of a process more or less faithfully, but without taking into account any possible changes or improvements. The BPR methodologies deal with routine human-computer interaction, but knowledge management requires task flexibility. Originally the process models lack role interaction and roles' integrity modelling [28]. Later the process designers have tried to establish processes as much natural as only they can be. Mostly BPR methodologies see actors only in the sequences of action to process goal achievements. Agent system methodologies emphasize actors and consider roles in social structure and organizations as structure where relations are among the roles. Some people compare BPR to restructuring (redesign organizational structure) that is changing the structural elements and connections among them. However according to the origins of BPR business process redesign is more important.

3 Agent Methodologies Review

Agent technology has received a great attention in the last years, so the industries are interested in using this technology to develop its own software products. In spite of the different developed agent theories, languages, architectures rather comparatively less work are done for BPR oriented software agent methodologies development. The important review of agent-oriented methodologies has been delivered by Iglesias et al. [18]. Presented by them methodologies like Burmeister's Agent-Oriented Analysis and Design [1], Agent Modelling Technique for Systems of BDI agents [23], Multi-Agent Scenario-Based Method [25] emphasize the roles modelling. Modelling technique (IDEF) widely used for BPR diagramming has been also applied in agent-oriented methodology for enterprise modelling [21].

Particular attention on multiagent methodologies development has been paid by Wooldridge [35]. For him a multiagent system is one that consists of a number of agents, which interact one with another, typically by exchanging messages through some computer network architecture. So the first problem is that of agent design, and the second is that of agents' society (or structure) design. Literature studies enable analysis a lot of quite well developed multiagent methodologies like AAII, Gaia, MaSE, Comparative Information Agent Design, ADEPT Project, MAS-Common KADS, Co-MoMAS, DESIRE, Cassiopeia.

3.1 The AAII Methodology

The AAII (Australian AI iNstitute) methodology for agent-oriented analysis and design was developed as a result of experience gained with these major applications [35]. It draws primarily upon object-oriented methodologies, and enhances them with some agent-based concepts. The methodology is aimed at the construction of a set of models which, when fully elaborated, define an agent system specification. The AAII methodology's external model presents a system-level view. The main components visible in this model are agents themselves. The external model is thus primarily concerned with agents and the relationships between them. It is not concerned with the internals of agents: how they are constructed and what they do. In contrast the internal model is entirely concerned with the internals of agents: their beliefs, desires, and intentions. The external model is intended to define inheritance relationships between agent classes, and to identify the instances of these classes that will appear at runtime. It is further composed of two further models: the agent model and the interaction model. The agent model is divided into an agent class model and an agent instance model. These two models define the agents and agent classes that can appear, and relate these classes to one another via inheritance, aggregation, and instantiation relations [35].

3.2 The Gaia Methodology

Wooldridge, Jennings and Kinny have presented the Gaia methodology for agent-oriented analysis and design [33], [34]. Gaia is a general methodology that supports both the micro-level (agent structure) and macro-level (agent society and organization structure) of agent development, it requires that inter-agent relationships (organization) and agent abilities are static at run-time.

The first step in the Gaia analysis process is to find the roles in the system, and the second is to model interactions between the roles found. Roles consist of four attributes: responsibilities, permissions, activities and protocols. Responsibilities are of two types: liveness properties - the role has to add something good to the system, and safety properties - prevent and disallow that something bad happens to the system. Permissions represents what the role is allowed to do, in particular, which information it is allowed to access. Activities are tasks that a role performs without interacting with other roles. Protocols are the specific patterns of interaction, e.g. a seller role can support different auction protocols, e.g. ``English auction''. Gaia has formal operators and templates for representing roles and their belonging attributes, it also has schemas that can be used for the representation of interactions. In the Gaia design process, the first step is to map roles into agent types, and then to create the right number of agent instances of each type. The second step is to determine the services model needed to fulfill a role in one or several agents, and the final step is to create the acquaintance model for the representation of communication between the agents.

3.3 Multiagent Systems Engineering (MaSE) Methodology

The methodology to solve complex problems agent working cooperatively with other agents in heterogeneous environment is called Multiagent Systems Engineering (MaSE) [32]. MaSE is similar to Gaia with respect to generality and the application domain supported, but in addition MaSE goes further regarding support for automatic code creation through MaSEtool.

The MaSE methodology is divided into seven sections (phases) in a logical sequence. Capturing goals as the first phase transforms the initial system specification into a structured hierarchy of system goals. This is done by first identifying goals based on the initial system specification requirements, and then ordering the goals according to importance in a structured as topically ordered hierarchy. Applying use cases, the second phase, creates use cases and sequence diagrams based on the initial system specification. Use case presents the logical interaction paths between various roles in and the system itself. Sequence diagrams are used to determine the minimum number of messages that have to be passed between roles in the system. The third phase creates roles that are responsible for the goals defined in the phase one. In general each goal is represented by one role, but a set of related goals may map to one role too. Together with the roles a set of tasks are created, the tasks define how to solve goals related to the role. Tasks are defined as state diagrams. The fourth phase, called creating agent classes, maps roles to agent classes in an agent class diagram. This diagram resembles object class diagrams, but the semantic of relationships is highlevel conversation as opposed to the object class diagrams' inheritance of structure. The fifth phase, known as constructing conversations, defines a coordination protocol in the form of state diagrams that define the conversation state for interacting agents. In the sixth phase, called assembling agent classes, the internal functionalities of agent classes are created. Selected functionality is based on five different types of agent architectures: Belief-Desire-Intention (BDI), reactive, planning, knowledge based and user-defined architecture. The final phase, defined as system design, creates actual agent instances based on the agent classes, the final results are presented in a deployment diagram. Visions of the future for MaSE is to provide completely automatic code generation based on the deployment diagram.

This methodology guides a multiagent system designer through the entire software development lifecycle, beginning from a textual system representation and proceeding in a structured manner toward a working implementation. The methodology combines several preexisting models (e.g. use cases models, sequence diagrams, state diagrams) in to a single structural methodology.

3.4 The Cooperative Information Agent Design

The Cooperative Information Agents Design is the methodology proposed by Verharen [18]. The methodology includes four models:

- Authorisation model: describes the authorised communication and obligations between the organization and the environment and the internal communications using authorisation diagrams. After identifying the current situation, it is redesigned for improving the efficiency of the business processes.
- Communication model: refines the previous model describing in detail the contracts between the agents, using Petri nets. The transactions between the agents are modelled using transaction diagrams that describe the relationship between speech-acts and goals
- Task model: specifies the task decomposition using task diagrams
- Universe of Discourse model: concerns the modelling of the content of the messages exchanged between the agents, using object-oriented techniques.

Verharen has proposed this agent system methodology from a business process perspective expressing it in four equally important models.

3.5 The ADEPT Methodology

The ADEPT (Advanced Decision Environment for Process Tasks) project is presented as a novel solution to the problem of software agent inter-operation in domains such as business process management [19]. The architecture can model structure of hierarchical or flat organization, or a mixture of the two, through the concepts of agents and agencies. In coordinating the actions of agents within a multiagent architecture it is important to fund a balance between the autonomy of agents within the system and the communication overheads involved in coordinating action. The ADEPT architecture can be viewed at two levels: the architecture of the multi-agent system in which an agents acts and the internal architecture of a single agent.

In ADEPT project the considerations are in two areas - business process management systems and agent-based system. There needs to be a definition of the business processes. It describes in some specification language, the activities that need to be performed, the participants who could or should perform them, and the interdependencies that exist between them. Specification languages vary in their details, but at the conceptual level they are broadly similar - they must provide a set of concepts useful for describing processes, their tasks, the dependencies between the tasks, and the required roles that can perform the specified tasks. The activities in the process description may be automated, or involve human interacting with computers. A software system needs to be devised that is capable of ensuring the process description is realised in practice. This system must: allow the human and the manual activities to be assigned appropriately: provide access to the software tools (e.g. database. spreadsheets, design software) required to complete the tasks, and ensure the dependencies between the tasks are satisfied.

The ADEPT project will assist designers of business process management systems in evaluating the appropriateness and benefits of the agent paradigm, in identifying the potentials, and in offering guidance on how to structure their applications.

3.6 MAS-CommonKADS Methodology

The MAS-CommonKADS is an extension of a methodology designed for the development of knowledge-based systems (KBS) analogous to methods of software engineering [17]. The methodology starts with a conceptualization phase that is an informal phase for collecting the user requirements and obtaining the first description of the system from the user point of view. For this purpose the use case technique is used and the interactions of the use cases are formalized with message sequence charts (MSC) [11]. The methodology defines the set of models: organization model, task model, agent model, communication model, expertise model and design model for analysis and design of the system. For each model the methodology establishes the constituents (entities to be modelled) and the relationships among the constituents.

3.7 The CoMoMAS Methodology

Conceptual Modelling of Multi-Agent Systems (CoMoMAS) is another extension of the CommonKADS methodology of agent system design [13]. This approach for MAS modelling is a compositional one where it constitutes an agent model based on results from five analysis steps (functional for agent tasks' identification, requirements to determine the requirements, competence for identification of the cognitive and reactive competencies, cooperative to identify the cooperation protocols and methods, and social to recognize the organization and architecture of the MAS). Developing a multi-agent system and building agent models can be seen as a process which is composed of different phases: specification, construction of models, their validation, their implementation and finally their test.

The development of MAS needs first a description of the functional behaviour of the system and an identification of the functional and non-functional requirements for agent and system design. This is followed by a description of the competencies necessary for solving these tasks. Based on these results designers are able to identify a first set of agents clustering the competencies for solving particular tasks and respecting the design requirements. The needs for cooperation and communication between agents are determined by a cooperative analysis whereas the MAS organization and needs for reorganization are established by a social analysis step.

3.8 The Design and Specification of Interacting Reasoning Components

The Design and Specification of Interacting REasoning components (DESIRE) is *de facto* modelling language and framework for modelling of agent systems.

DESIRE allows a designer to explicitly and precisely specify both the agent functionality (i.e. the expertise required to perform the domain tasks for which the agent is responsible in terms of the knowledge requirements and the reasoning capabilities) and the inter-agent functionality. All functionality is designed as a series of interacting, task-based, hierarchically structured components. Tasks are characterised in terms of their inputs, their outputs and their relationship to other tasks. Interaction and co-

ordination between components, between components and the external world, and between components and users is specified in terms of information exchange, sequencing information and control dependencies. The components themselves can be of any complexity (from simple functions and procedures up to whole knowledge-based systems) and can perform any domain function (e.g. numerical calculations, information retrieval, optimization).

3.9 The Cassiopeia Methodology

The Cassiopeia as the MAS method is a way to address a type of problem - solving where collective behaviours are put into operation through a set of agents [10].

The first step of Cassiopeia consists in identifying the required level of abstraction so that the behaviours make sense with respect to the collective achievement of the task: the designer specifies the sets of behaviours that achieve proper functionalities, and thus determines the individual roles that the agents can play. The designer is free to define different types of agents based on these roles and can choose to design agents that are either homogeneous (all the agents are provided with the same set of domain-dependent roles) or heterogeneous (some agents are supplied with only a subset of these roles).

The second and third steps consist in analyzing the structure of the organization based on the dependencies between the individual roles being considered. Such dependencies can be functional, when they derive from the dependencies that exist between the behaviours implemented by these roles, or relational, when they take place at the abstraction level of the roles. Various types of dependencies can be considered such as coordination, simultaneous or sequential facilitation of conditioning. The identification of these dependencies leads to defining the dependencies layer of the collective task being considered.

The dependencies between the individual roles are naturally translated into dependencies between the agents that can play these roles. The fourth step consists in defining the potential groups that may arise. The paths and the elementary circuits of the dependencies layer can be used to define the potential groupings of the different individual roles therefore provide a global representation of the structure of the organizations to which the agents can belong when playing these roles.

The fifth and last step addresses the dynamics of the organization. It consists in specifying the organizational roles that will enable the agents to manage the formation and dissolution of the defined groups.

Problem with Cassiopeia is that the roles should be fully functional in a single-agent mode as well as in multi-agent modes. Nothing prevents the designer from making irrelevant choices. The only solution so far is to rely on an informal analysis of the domain, while experimenting groupings of elementary behaviours or functions.

4 Conclusions

Taking into account the presented above methodologies it should be noticed that there is comparatively little consensus among them. An important aspect of successful business process management is the optimization of the processes. The first step of this endeavour is to understand the process as it currently operates (this is a typical systems analysis activity that involves interviewing people with expert knowledge about the process and studying relevant system documentation), the second step is to explicitly reconsider and redesign the process. The business processes' modelling would be easier if the processes were classified according to the complexity and human participation. Such Business Process Reengineering is typically carried out in order to increase customer satisfaction, improve the efficiency of business process operations, increase the quality of products, reduce costs, meet new business challenges and opportunities by changing existing services, or introducing new ones. The final step is to encode the revised business process description in the business process specification language.

Although BPR could be treated as the prerequisite of new IT solutions' implementations this process approach is not sufficient for MAS modelling and design. Presented above MAS methodologies are based on tasks, roles and relations among roles modelling, so process approach has been substituted by organization approach. Organization is conceptualized as a structure, a framework for agent interactions through the definitions of roles and authority relations. An organization consists of a group of agents, a set of connections among agents and a set of goals or evaluation criteria by which the combined activities of agents are evaluated. The organization imposes constraints on the ways the agents communicate and coordinate.

References

1. Burmeister, B.: Models and methodology for agent-oriented analysis and design in: Fischer K (ed.) Working Notes of the KI'96 Workshop on Agent -Oriented Programming and Distributed Systems, DFKI Document D-96-06 (1996)
2. Butler, C.: The role of I/T in facilitating BPR: observations from the literature, Business Process Re-engineering: Information Systems Opportunities and Challenges (A-54)/ Glasson B.C.et al. (ed.) Elsevier Science B.V.(North Holland), 147-161 (1994)
3. Brazier, F.M.T., Dunin-Keplicz, B.M., Jennings, N.R., Treur J.: DESIRE: Modelling Multi-Agent Systems In a Compositional Formal Framework, http://www.ecs.soton.ac.uk/~nj/download-files/IJCIS96b.ps
4. Coad , P., Yourdon, E.: Object-Oriented Analysis, Yourdon Press, Englewood Cliffs, New Jersey (1991)
5. Crowe, M., Beeby, R., Gammack, J.: Constructing Systems and Information, A Process View, McGraw-Hill, London (1996)
6. Curtis,W., Kellner, M.I., Over, J.: Process modelling, Communications of the ACM, 35, 9, 75-90 (1992)

7. Davenport, T.H., Short, J.E.: The new industrial engineering: Information technology and business process redesign, Sloan Management Review, 31 (4), 11-27 (1990)
8. Davis, A.M.: Object-oriented Requirements to Object-oriented Design: An easy Transition? J.Systems Software, 30, 151-159 (1995)
9. Douglass, D.P.: The Role of IT in Business Reengineering, I/S Analyzer, Vol.31, No.8, 1-14 (1993)
10. Drogoul, A., Zucker, J.D.: Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences from the RoboCup Challenge http://citeseer.nj.nec.com (1999)
11. Ekkart, R., Grabowski, J., Graubmann, P.: Tutorial on message sequence charts (MSC), Proceedings of FORTE/PSTV'96 Conference, October (1996)
12. Gerrits, H.: Business modelling based on logistics to support business process re-engineering, Business Process Re-engineering: Information Systems Opportunities and Challenges (A-54)/ Glasson B.C. et al. (ed.) Elsevier Science B.V.(North Holland), 279-288 (1994)
13. Glaser, N.: The CoMoMAS Approach: From Conceptual Models to Executable Code, http://citeseer.nj.nec.com/ (1997)
14. Glasson, B.C., Hawryszkiewycz, I.T., Underwood, B.A., Weber, R.A.: Business Process Re-engineering Information Systems Opportunities and Challenges, IFIP Transactions, Elsevier, Amsterdam (1994)
15. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution, London: Nicholas Brealey (1993)
16. Hawryszkiewicz, I.T: Using workgroup modelling as a basis for re-engineering business processes, Business Process Re-engineering: Information Systems Opportunities and Challenges (A-54)/ Glasson B.C. et al. (ed.) Elsevier Science B.V.(North Holland) 289-299 (1994)
17. Iglesias, C.A., Garijo, M., Gonzalez, J.C., Velasci, J.R.: A Methodological Proposal for Multiagent Systems Development Extending CommonKADS http://citeseernj.nec.com/ (1996)
18. Iglesias, C.A., Garijo, M., Gonzalez, J.C.: A Survey of Agent-Oriented Methodologies in: Miller J.P., Singh M.P., Rao, A. (ed.), Intelligent Agents (ATAL'98) LNAI 1555, Springer Verlag, Berlin, Germany, 317- 330, (1999)
19. Jennings, N.R., Norman, T.J., Faratin, P., O'Brien, P., Odgers, B.: Autonomous Agents for Business Process Management, Queen Mary & Westfield College, University of London, London (2002)
20. Jones, M.: Don't emancipate, exaggerate: rhetoric, reality and reengineering, Transforming Organizations with Information Technology (A-49) Baskerville, R., Smithson, S., Ngwenyama, O., DeGross J.L. (ed.) Elsevier Science B.V. (North-Holland) (1994)
21. Kendall, E.A., Malkoun, M.T., Jiang, Ch.: A methodology for developing agent based systems for enterprise integration, in: Luckose D., Zhang C. (ed.) Proceedings of the First Australian Workshop on DAI, Lecture Notes on Artificial Intelligence, Springer-Verlag, Berlin (1996)
22. Kettinger, W.J., James, T.C., Guha S.: Information architectural design in business process reengineering, Journal of Information Technology 11, 27-37 (1996)

23. Kinny, D., Georgeff, M., Rao, A.: A methodology and modelling technique for systems of BDI agents, in: van der Velde W., Perram J (ed.) Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96 (LNAI Volume 1038), Springer Verlag, Berlin (1996)
24. McHugh, P.M.G., Wheeler III William A.: Beyond Business Process Reengineering Towards the Holonic Enterprise, John Wiley & Sons, Chichester (1995)
25. Moulin, B., Brassard, M.: A scenario-based design method and an environment for the development of multiagent systems, in: Lukose D., Zhang C. (ed.) First Australian Workshop on Distributed Artificial Intelligence, LNAI Volume 1087, Springer-Verlag, Berlin, 216- 231 (1996)
26. Ould, M A.: Business Processes Modelling and Analysis for Re-engineering and improvement, John Wiley & Sons, Chichester (1995)
27. Rumbaugh J., Blaha, M., Eddy F., Premerlani, W., Lorensen, W.: Object Oriented Modelling and Design, Prentice Hall, Englewood Cliffs, New Jersey (1989)
28. Sheer, A-W.: Architecture of Integrated Information Systems, Springer-Verlag, Berlin (1992)
29. Sun, L., Liu, K.: A method for interactive articulation of information requirements for strategic decision support, Information and Software Technology, 43, 247-263 (2001)
30. Wagner, T., Rana, O.: Infrastructure for Agents, Multi-Agent Systems and Scalable Multi-Agent Systems, Springer-Verlag, Berlin (2001)
31. Walford, R.B.: Business Process Implementation for IT Professionals and Managers Artech House, Boston, London (1999)
32. Wood, M.F., DeLoach, S.A.: An Overview of the Multiagent Systems Engineering Methodology, In Agent-Oriented Software Engineering - Proceedings of the First International Workshop on Agent-Oriented Software Engineering, 10th June 2000, Limerick, Ireland Ciancarini, P., Wooldridge, M.: (ed.) Lecture Notes in Computer Science, Vol. 1957 Springer - Verlag, Berlin (2000)
33. Wooldridge, M.J., Jennings N.R., Kenny, D.: A methodology for agent-oriented analysis and design in: Proceedings of the Third International Conference on Autonomous Agents, http:// www.csc. liv.ac.uk/~mjw/pubs/agents99.ps.gz (1999)
34. Wooldridge, M.J., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design, Autonomous Agents and Multi-Agent Systems, 3(3), 285-312, September, http://www.csc.liv.ac.uk/~mjw/jaamas2000b.ps.gz (2000)
35. Wooldridge, M.J.: An Introduction to MultiAgent Systems, JohnWiley & Sons, Chichester (2002)
36. Wysocki R.K., DeMichiell, R.L.: Managing Information Across the Enterprise, John Wiley & Sons, Chichester, (1997)
37. Yourdon, E.: Modern Structured Analysis, Yourdon Press, Englewood Cliffs, New Jersey (1989)