

December 2003

Active Portals to Support Collaborative Business Processes

I. Hawryszkiewicz
University of Technology, Sydney

Follow this and additional works at: <http://aisel.aisnet.org/amcis2003>

Recommended Citation

Hawryszkiewicz, I., "Active Portals to Support Collaborative Business Processes" (2003). *AMCIS 2003 Proceedings*. 51.
<http://aisel.aisnet.org/amcis2003/51>

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 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

ACTIVE PORTALS TO SUPPORT COLLABORATIVE BUSINESS PROCESSES

I. T. Hawryszkiewicz
University of Technology, Sydney
igorh@it.uts.edu.au

Abstract

The trend to collaborative business processes requires ways to identify and configure collaborative technologies to support such processes. Collaborative business processes are usually open in the sense that the process steps followed cannot be completely predefined and emerge over time. They must, however, be aligned to organizational goals. The paper suggests that modeling concepts used to develop well defined systems are not applicable for designing collaborative systems and alternative concepts are needed to do so. The paper defines collaborative concepts to model collaborative systems and ways to support them using software agents. It defines software requirements and stresses the need to provide software agents to assist system evolution as collaborative processes emerge and an approach to analyzing agent requirements.

Introduction

There are now many business processes where collaboration is essential. Just to name a few, there are distributed project teams, design teams, planning and evaluation teams, client support teams, as well as the need for meetings during various stages of the business process. There is also a substantive growth in applications, which require collaboration to create new knowledge in the sense of new products and services (Grant, 1995). For example, producing a new model car can require well over 200 designers of different components to coordinate their activities and share their knowledge. Many such collaborative business processes are emergent and can be characterized as innovation (Kuczumarski, 1997) processes, or processes that support personalized and varying client needs. Typical industry scenarios that require sharing of knowledge include large organizations and small organizations. Large firms such as for example banks outsource its Information Technology function to a third party. Small firms are increasingly setting up similar but smaller relationships. Such relationships are becoming common as some industries restructure as for example found in the Telecommunications industry (Reyes, et.al. 2002). Another example is an academic community setting up an expert group to address a goal – for instance, a course committee to design a university course. New collaborative tasks may emerge as the committee identifies new avenues that must be followed or evaluated.

Collaborative processes possess characteristics that require different design approaches than found in traditional systems. One characteristic is that such processes are emergent and cannot predefined. They include intensive interaction where experts combine tacit and explicit knowledge to create and exchange ideas and identify the next process steps. The evolving ideas and changes must be recorded in documents and directed to those members who possess the necessary tacit knowledge to interpret them and add further knowledge. Such change can create new documents and change document flows. Lack of ability to vary dynamic document flows restricts such interaction and consequently the user abilities to monitor tasks, raise and discuss issues within complex situations. Effective goal oriented work in such communities must go beyond centering on the loose exchange of information (Cummings, 2002) but provide ways to maintain intense relationships to develop new knowledge. Such intense work carried out by virtual distributed organizations is often constrained by the ability to manage document flows and structures within emerging processes.

Support for emergent processes requires ways to maintain awareness while allowing processes to emerge. Our proposal is to provide such support through software agents. Hattori and others (1999) have first identified the use of software agents for communities that work towards common goals. They suggested personal workspaces connected to community workspaces and supported by agents. Hoffman and others (2000) have described how such communities can evolve and capture knowledge as part of their everyday work. Our work goes beyond software agents that support loosely coupled communities but to extend them

to support focused and goal oriented work. This in turn requires flexible governance structures (Jones, et.al. 1997) and awareness mechanisms to ensure that distributed workspace activities converge to a common aim.

Development of such software support systems require a set of concepts that provide a systematic way to define collaboration, process emergence and facilitate agent support. The paper proposes a way to define collaborative processes. The goal is to provide collaborative semantics to define collaborative structures, specify process emergence and then to use the same semantics to define agents. The paper first describes such concepts. It then considers implementations that directly support such semantics and their integrated agents. The paper proposes digital workspaces to support such collaborative processes and ways to augment these with agent systems to facilitate the use of these workspaces

The Collaborative Metamodel

We have developed a set of metamodel concepts to describe such dynamics and directly support them with technology. A subset of these are shown in Figure 1. The metamodel provides generic concepts for collaborative processes (Hawryszkiewicz, 2002).

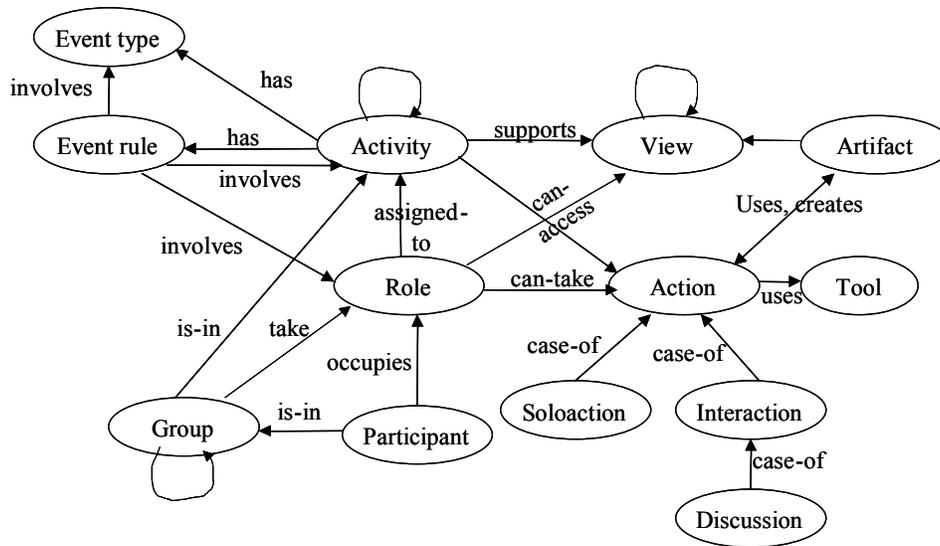


Figure 2. The Metamodel for Workspace Evolution

The main concepts are:

- Role** — defines responsibilities in system
- Participant** — a specific person assigned to a role
- Group** — a collection of participants
- Artifact** — data objects such as documents
- View** — a collection of artifacts
- Activity** — produces a well defined outputs and usually require many work-items, actions and interactions to do so (eg. Produce a planning document)
- Workspace** — an interface that supports an activity
- Work-item** — a set of actions and interactions needed to produce intermediate outcomes that eventually produce an activity output (eg. Review part of a planning document - which may include a number of actions). Can be:
 - Action** — a specific unit of work carried out by a role as part of a work-item (eg. Change an artifact, send an artifact)
 - Interaction** — the basic exchanges between people when they collaborate in the activities. An interaction may not produce an explicit output although it may change people’s knowledge
- Process** — describes timing relationships between activities, work-items and interactions
- Event** — the completion of some action in a workspace

Each of these concepts can become an object class in UML. Agents can then be associated with these object classes as suggested by Bauer (2001).

Some details are:

- The metamodel centers on activities, which can be made up of a number of sub-activities as indicated by the looping arrow.
- A person, here called a participant, 'is-in' group, which can evolve independently, and contain subgroups. Workgroups support scalability as independent workgroups can exist in the same system but gradually merge or intersect if needed.
- An activity 'is-in' a group, 'has' any number of roles and 'contains' any number of views, which can be made up of other artifacts or define groups of artifacts. An activity 'is-in' a group.
- The roles define access abilities to artifacts and 'can-take' the actions.
- Each role is 'occupied' by participants, who have the role abilities.
- Actions 'use or create' artifacts. They can be solo actions, which are taken by individuals, or interactions, such as discussions, which can include more than one participant actions 'use' tools.
- An activity can include a number of event types, which are assigned to roles. Event instances of particular types lead to message being sent to other activities.

The model also includes command to create the collaborative objects and relationships between them. These commands then will become software agent actions.

Describing Emergence

The system commands can be used to change the structure and process flows. Emergence is often specified in terms relevant to an application and then need to be converted to computer specific terms. In this paper, activity change can be classified as follows:

- *Local emergence* or change to the internal structure of an activity, usually through the addition of new components to the workspace. As for example, "we need to create a work-item to further comment on topic X.". This can be implemented by creating a new folder that includes information relevant to topic x and providing access to users to contribute to this topic.
- *Extension emergence* of a new but predefined activity simply adding a new prespecified activity when needed. An example here may be to create a new meeting, which will require creating a new workspace.
- *Specialized emergence* of a variant of a predefined type of activity where the work of an existing activity can be customized to a particular need. An example here may be to create follow-ups each of which may be of a different type, and
- *Global emergence* or creation of an activity whose type was not defined earlier. In this case we have something like a new project will be needed to progress further.
- Change workflows by adding new workflow steps,

The metamodel provides ways to add a number of components to the workspace or create workspaces. These can include commands to:

- Create new objects such as roles or folders, to bring together the explicit information needed in the system and to include discussion systems to allow users to exchange their interpretations of this data.
- Create new groups within a space to carry out a subtask. One example here has been setting up student groups for a particular learning activity,
- Configure each workspace, and if necessary organizes the activities into a business process,
- Set up new discussion or other ways that people within workspaces can interact.
- Define changes to workflow flows and notification schemes between workspaces.

An Example

Figure 2 illustrates an example how the metamodel concepts can be used to model collaborative processes. Figure 2 illustrates an example of modeling using our metamodel concept to model a research process within a research institution. Workspaces as

shown as clouded shapes, and roles as facial icons. Figure 2 shows three workspaces – one for applicants, another for facilitation the process, and a third for reviewers. Documents are represented by the rectangular boxes outside the workspace, whereas boxes within the workspaces represent actions that are taken by roles and events by circles. An arrow from an event to a role signifies that the event notifies the role. An arrow in the opposite direction means that the role has completed a task. The round circles represent events in the process with events assigned to roles within workspaces. For example, an applicant can select the event ‘submit application’, paste an application from the application folder, and activate the event, ‘submit application’. The event will then cause the event ‘record application’ to notify the secretary to record the application. Arrows between events describe the document flow.

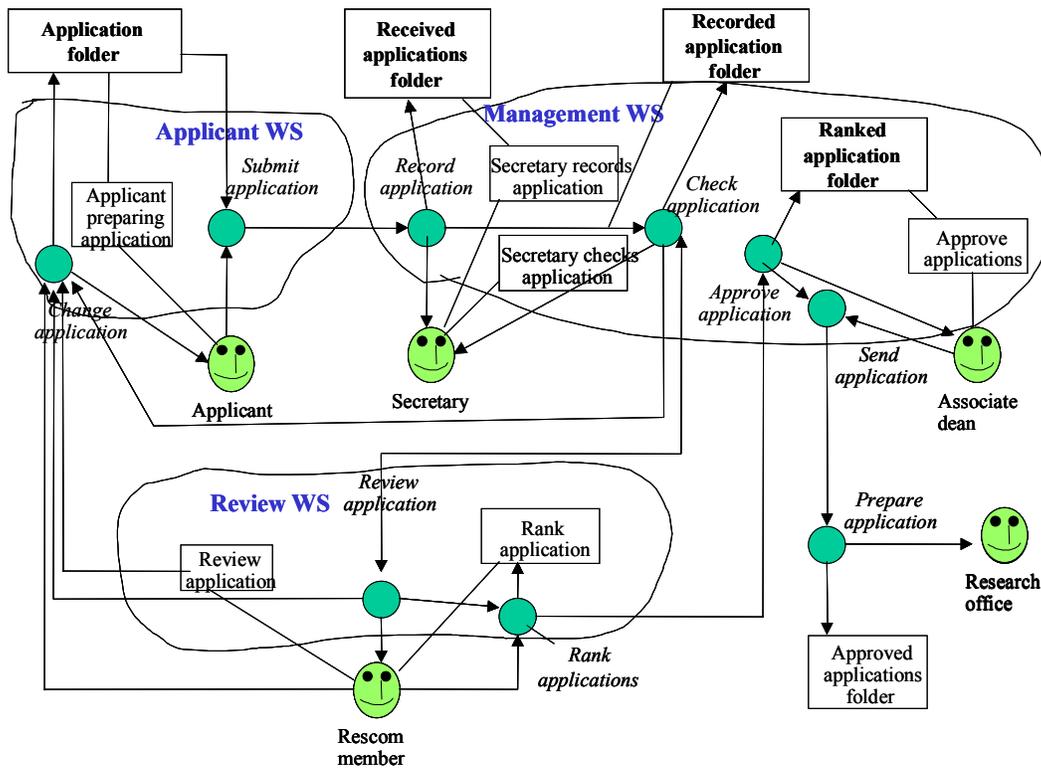


Figure 2. Research Application Process

Emergence can take many forms in this example. One example may be to change the workflow to distribute the proposal to other researchers with the goal of involving them and adding to the strength of the application. Another may be to send to an industry contact with a view of getting some additional support. Each of these will result in new workflow events, changes to workflow event rules, including where a new work-item is to be accommodated in the process.

Implementation

The complex relationships that can arise in emergent processes can make it difficult for participants to keep track of large numbers of related developments. Our proposal as shown in Figure 3 is to use digital workspaces and active portals to support coordination in emergent processes. An active portal goes beyond traditional knowledge portals, which provide content management, search and profile matching to support individual tasks. We extend this by including active elements, in this case software agents, to assist coordination and maintain awareness in workspaces. We thus separate the concept of workspace and knowledge portal, where the workspace primarily supports collaboration within the workspace and the knowledge portal supports coordination between workspaces.

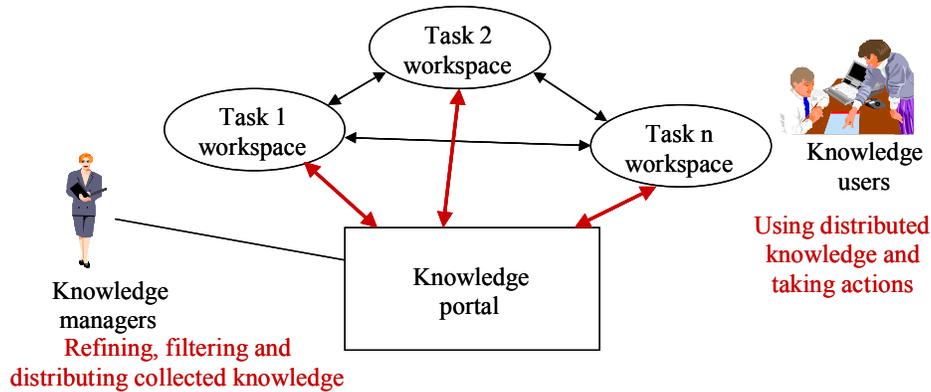


Figure 3. Collaborative Workspaces Supported by a Knowledge Portal

The knowledge portal supports the workspaces. It contains organizational knowledge including available experts as well as knowledge captured during task execution, assists in task coordination by informing users of important events and requesting them to contribute to newly emerging activities. The paper now describes the workspace presentation and follows this with a description of agent architecture.

Workspace Presentation

The portal interface presents the collaborative concepts and provides the commands to easily define workspaces, and their components. These allow users to do things like create roles within activities, add artifacts dynamically whenever they are needed. As an example Figure 4 defines the interface for the research secretary. It shows folders where applications are stored as they progress through the workflow, and the ability to create new events to dynamically change the workflow for one particular application.

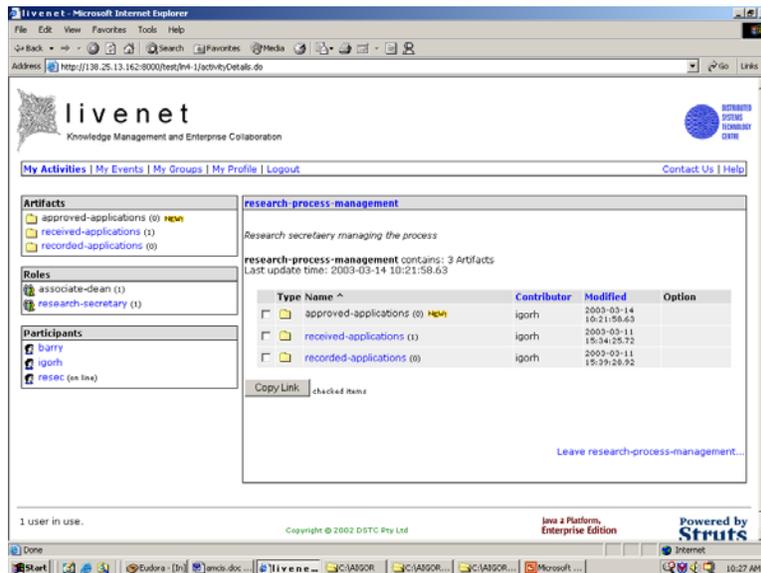


Figure 4. A Workspace for the Research Secretary

As an example, Figure 5 describes changing workflows for a single document. Here new steps can be added or existing events redefined. Notes can be added as the flow proceeds.

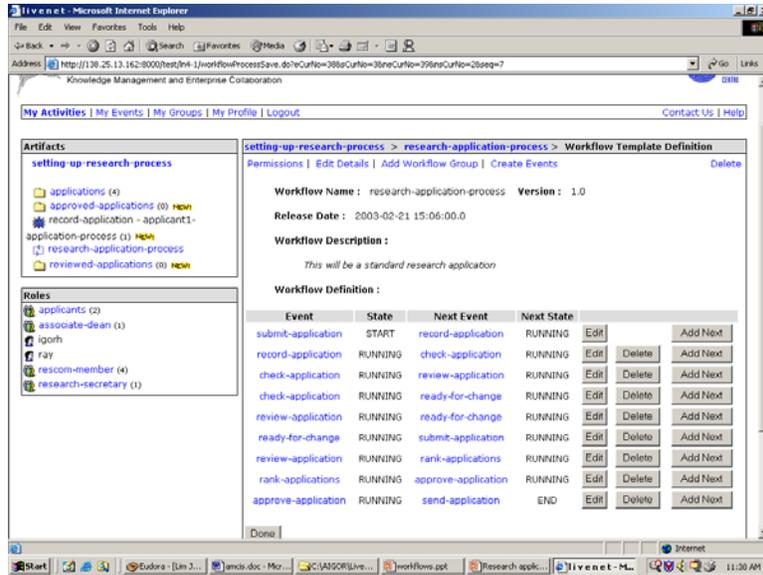


Figure 5. Manually Varying Process Flows

Extending with Agent Support

We now extend the system with agent support. An active portal will include intelligent support to help users to set up and coordinate local processes, while providing the coordinating mechanisms that are appropriate for achieving a common goal. Such a system must combine rich support for free collaboration at the interface with a strong sense of purpose at an underlying level. This follows two steps. One to propose a multi-agent architecture and then to define agent structures. The multi-agent architecture follows the collaborative metamodel. As shown in Figure 6, each object class has its own agent class. The agents in the system communicate with each other given organizational goals. Thus a document agent can notify users who may be interested in the document of potential changes and what they need to do. An activity agent can create a new meeting to resolve issues that arise from such changes.

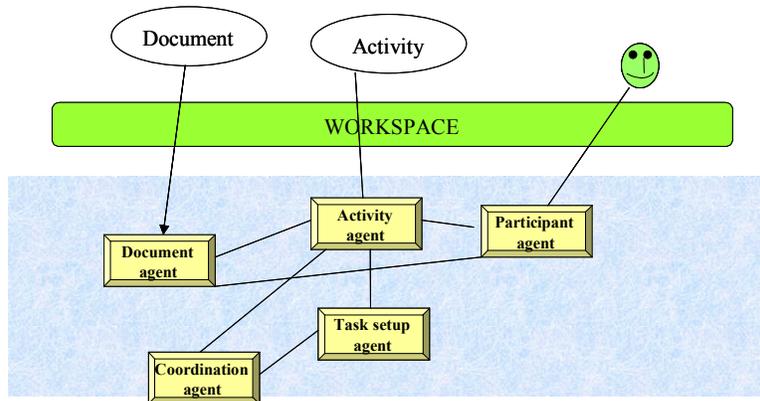


Figure 6. Agent Architecture

The general principle here is that each agent only builds knowledge in its area of responsibility and its relationship to other agents. For example our proposed guidelines are:

- Personal or activity agents do not develop knowledge about coordination plans but only how to use them. They inherit beliefs from a generic personal agent and add their own beliefs agents. They do not develop knowledge about documents

but only how to use them, or interact with the document agents. They concentrate on arranging a person's participation in workspaces.

- Activity agents concentrate on good ways to arrange activities, for example structuring roles or improving discussions, as for example, ways for issue resolution, collaborative document processing.
- Document agents usually more reactive – they follow a plan. But they can have contradictory desires such as “highest quality document to be produced” or “finish as quickly as possible”. Alternatively precepts may be characteristics of file updates and event may be “not updating files – what's going on”.
- Coordination (planning) agents create plans for other agents – in that case other agents look for agents to create a plan for them. Provides standard business processes.

Agent Support for Collaboration Services

Our goal now is to extend the collaborative system with agent support. Each of the collaborative objects will have its own agent, with a multi-agent architecture supporting interaction between the agents. Figure 7 illustrates agent structures, which are based on the widely adopted BDI architecture (Rao and Georgeff, 1991). The agent components are:

- Desire – what the agent sees as its ultimate although abstractly defined achievement,
- Belief – the information that an agent has about its environment. It includes local beliefs that are private to an agent or they can be social beliefs that are shared by a group of agents.
- Precept – what the agent observes in its external environment
- Goal – what needs to be done to react to the event
- Intention – a currently chosen course of action to realize the goal
- Plan – a set of actions to carry out an intention

The general semantic is “Given a set of precepts the agent identifies that it must take some actions and signifies this by an event. The event and precepts can change the agent's local beliefs and also use its beliefs and desires to formulate specific goals. The agent then determines its intentions and develops a plan of actions to implement these intentions”

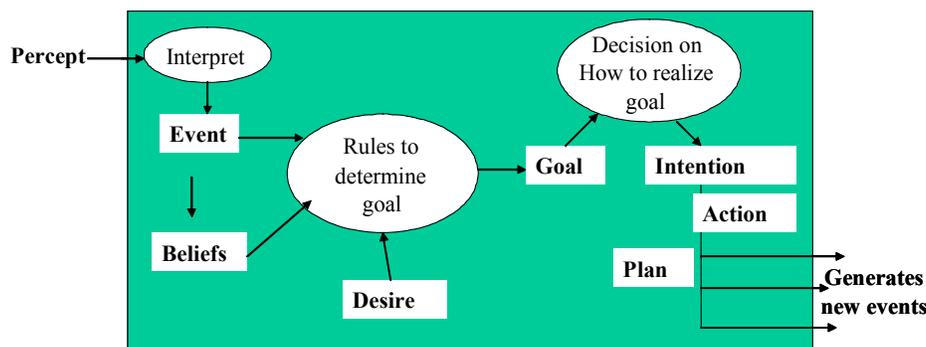


Figure 7. An Agent Structure

Agent designers must then define these components and rules. Table 1 shows a way to carry out analysis and requirements, in this case for the coordination agent. It is not a complete description but provides an idea of what is needed to specify different components.

Our goal is to define such agents in ways to make them reusable through the definition of generic agents and the interactions between them in terms of speech acts as now proposed for object based systems (Bauer, Muller, Odell, 200). These generic agents can then be adapted to many collaborative applications through the use of agent templates.

Table 1. Structure for Analysing Agent Requirements

Agent parameter	Details	Beliefs
Input messages event happens: workflow-event-name; event-late: workflow-event-name; new-work: before doing task x do task y; extra-task: carry out task x sometime in the process; reassign task x;	Interpretation rules about identifying agent-events event-happens(workflow-event-name): get-next-workflow-event; event-late (workflow-event-name): reschedule; extra-task(work-item) and no-task-late: insert work-item; new-work; insert-work-item; reassign-task: find better way to carry out task;	Beliefs about inputs Task-y is always late if it is raining; Task reassignment needs rescheduling; Rescheduling should not increase completion time; Task y (approval) cannot take place before task z (review);
Agent-events get-next-workflow-event; reschedule; insert work-item; add-new-event to workflow;	Rules for determining goals from events on get next-workflow-event : use workflow rule, start-event; on insert work-item: find best workspace;	Beliefs about goals to be addressed A workflow-event usually requires y time if carried out by x; A work-item can be carried out in workspace y; The marketing group is good for reviews; Workflow rules;
Goals find best workspace for work-item; change workflow by adding event; start next event;	Rules on selecting plans of actions to realize goals to find best-workspace, create new-work-item, check-participant availability; assign participant;	Beliefs about what actions are needed Where will an action required by an event be carried out.
Potential actions Find available participant; Create new-work item; Select workspace for work-item;		Beliefs about what actions can achieve Participant must be available to carry out work-item;

Summary

This paper described a way to support emergent processes by technologies that can be customized to user needs. It defined a semantic metamodel and an implementation that allows users to customize how the metamodel objects will appear at the interface. In that case it will be possible to quickly define interfaces that match the users intuitive perception to the problem. The paper suggested that software agents be used to support such evolution and that the semantic concepts provide a basis for a multi-agent architecture.

References

- Bauer, B., Muller, J.P. and Odell, J. (2001): "Agent UML: A Formalism for Specifying Multiagent interaction" in Agent-Oriented Software Engineering, Ciancarini, P., and Wooldrifge, M. (eds), Springer, Berlin, pp. 91-103.
- Cummings, J.N., Butler, B. and Kraut, R. (2002): "The Quality of OnLine Social Relationships" Communications of the ACM, Vol. 45, No. 1, July, 2002, pp. 103-111.

- Grant, R.M. (1996): "Prospering in Dynamically-competitive Environments: Organizational Capability as Knowledge Integration" *Organization Science*, Vol. 7, No. 4, July, 1996, pp. 375-387.
- Hattori, F., Ohguro, T., Yokoo, M., Matsubara, S. and Yoshida, S. (1999): "Socialware: Multiagent Systems for Supporting Network Communities", *Communications of the ACM*, March, 1999, pp. 55-59.
- Hawryszkiewicz, I.T. (2002): "Designing Collaborative Business Systems" in IFIP 17th World Computer Congress, TC8 Stream on Information Systems: The e-Business Challenge, ed. Roland Traunmiller, Montreal, August 2002, Kluwer Academic Publishers, Boston, pp. 131-146.
- Hoffman, M., Loser, K-U, Walter, T. and Herrmann, T. (1999): "A Design Process for Embedding Knowledge Management in Everyday Work" Group 99, Phoenix, Arizona, pp. 296-305.
- Jones, C.T., Hesterly, W.S., and S.P. Borgatti (1997): A General Theory of Network Governance: Exchange Conditions and Social Mechanisms. *Academy of Management Review*, Vol. 22, No. 4, October 1997, pp. 911-945.
- Kuczmariski, T. D. (1997): "*Innovation: Leadership Strategies for the Competitive Edge*" NTC Business Books, Lincolnwood, Illinois.
- Rao, A.S., and Georgeff, M. P.(1991) "Modeling rational agents within a BDI-architecture". Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, KR '91, 473-484, Cambridge, MA, 1991.
- Reyes, P., Raisinhani, S. and Singh, M. (2002): "Global Supply Chain Management in the Telecommunications Industry: The Role of Information Technology in Integration of Supply Chain Entities" *Journal of Global Information Technology Management*, Vol. 5, No. 2, 2002, pp.48-67.
- LiveNet: <http://livenet.it.uts.edu.au>.