

# DESIGNING COLLABORATIVE BUSINESS SYSTEMS

Igor Hawryszkiewicz

*University of Technology, Sydney, e-mail: igorh@it.uts.edu.au*

**Abstract:** The trend to collaborative business processes requires ways to identify collaborative technologies to support such systems. The business processes are usually open in the sense that the steps followed are often not predefined but must achieve organizational goals. Computer support must thus emphasize the provision of business services that can be dynamically integrated as the process emerges. The paper describes a way of designing such processes by defining related workspaces that focus on organizational goals. Each individual workspace provides the business services needed to achieve its goal, but which can be easily customized within the workspace to allow collaboration to emerge. The paper describes a method for identifying such workspaces and identifies technologies that provide ways for processes to emerge within them.

**Key words:** Collaboration, eBusiness, Methodologies

## 1. 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]. Many such business processes are emergent and can be characterized as innovation (Kuczmarski, 1997) processes, or processes that support personalized and varying client needs.

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35604-4\\_20](https://doi.org/10.1007/978-0-387-35604-4_20)

There are now many systems that support small collaborative groups. However, collaborative business applications must go beyond the interaction of small local groups and use the expertise of a distributed workforce and require individuals to participate in many teams and to easily move between the teams. 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.

A business process often includes a number of different kinds of collaborative activities. Typical activities include jointly working on a document, making a decision or conducting meetings. The activities are all integrated into the business process. Often each such activity follows a particular intuitive approach and requires a different collaborative technology. Thus the way users perceive a meeting is different from the way they perceive the working of a project team. It is now widely recognized that a new computer system cannot be built for each specific collaborative activity nor to integrate activities into a business process. What is needed are software systems that are customizable to a large variety of collaborative activities and ways to integrate these activities into processes.

Consequently design methods must produce models that show both how collaboration fits into the business process and the kind of collaborative activities. The activities defined in the models must then be converted to components, which can be dynamically configured as collaborative processes evolve over time. The question is how to provide models that show how such components fit together in a business process. Modeling methods in this case must show the broad structure of each component and how it fits into the business system. Just as others [Sparling, 2000], we have found that modeling methods such as structured systems analysis or object analysis identify requirements in their most detailed form before development commences. Collaborative processes on the other hand need requirements analysis to identify components in their broad form and leave it up to the users to customize their detailed needs. Hence there is a mismatch between the modeling method and the nature of the process. As a result, the lack of tools to model collaboration within the business process results in most systems development proceeding in a relatively ad-hoc manner. What is needed is a design process that defines the general goal of the system and how the components will fit into the enterprise.

The paper describes a design process, which is based on a metamodel that provides the terms needed to describe collaborative applications. The metamodel is used in the design method to identify and describe the major collaborative activities. The model is then implemented directly using a workspace system that directly supports such terms.

## **2. DOMAIN CHARACTERISTICS**

Collaborative systems must support closely linked communities that can work together towards common goals. In our terminology each community is supported by a workspace that allows its members to interact and carry out their work. In any enterprise there may be any number of such communities all working towards a common goal.

A general view of workspace structures within a business framework for a knowledge-based enterprise is shown in Figure 1. Here there are the common processes supported using an ERP system. These are presented generally through an enterprise workspace, usually the IntraNet home page. Additional workspaces are then established to support particular areas and projects within the enterprise. People can set up their own private spaces that link to the organizational project spaces. They can also set up joint workspaces for their groups. Examples may include sales teams, design teams or communities that exchange other kinds of knowledge. In education the project spaces are subjects. The personal spaces are student spaces customized to the subjects that they take.

The design objective then becomes to identify the needed organizational workspace structures for the kind of networked organization shown in Figure 1. Any process evolution must be user driven and any collaborative tools should allow the users themselves to initiate such change and modify their processes. Thus it should be possible at any time for an activity participant to initiate some new actions and define events and milestones which may need follow-up actions. It should also be possible for each participant to personalize their place of work.

Generally it is found that collaborative processes are opportunistic in nature (Dourish, 1998), result in islands of disconnected work, which nevertheless must be coordinated towards some common goal. Furthermore, they evolve dynamically and sometimes rapidly change as new situations arise. Evolution can include changes in participants, goals, and methods of interaction as people begin to understand the collaborative tools and learn how to collaborate electronically. The evolution must be user driven and any collaborative tools should allow the users themselves to initiate such change and modify their processes. Thus it should be possible at any time for an activity participant to initiate some new actions and define events and milestones which may need follow-up actions.

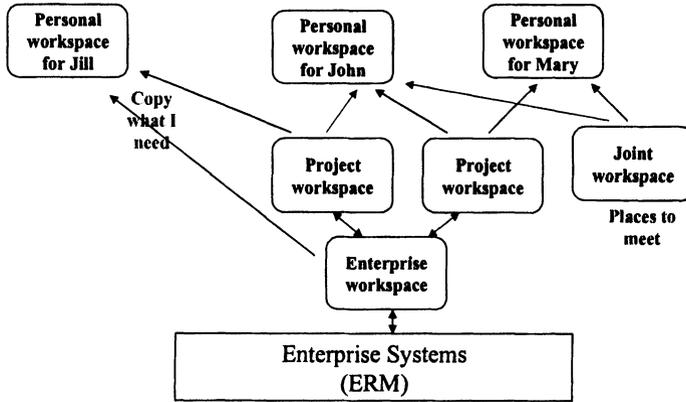


Figure 1: Generic Places

This paper proposes that one way to describe such evolving activities requires a paradigm that naturally expresses changing relationships within a dynamic environment composed of a large number of interrelated activities. Each activity is defined as a set of roles, which are responsible for taking actions within the activity and can view or change documents. The activities are all going on at the same time and ways to coordinate and maintain continuous awareness between them are needed. In addition information is shared between the activities and messages can be passed between them.

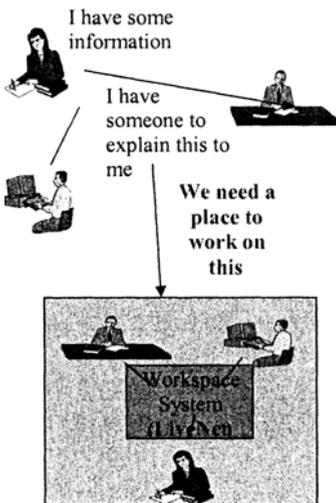


Figure 2: Organizing the teams.

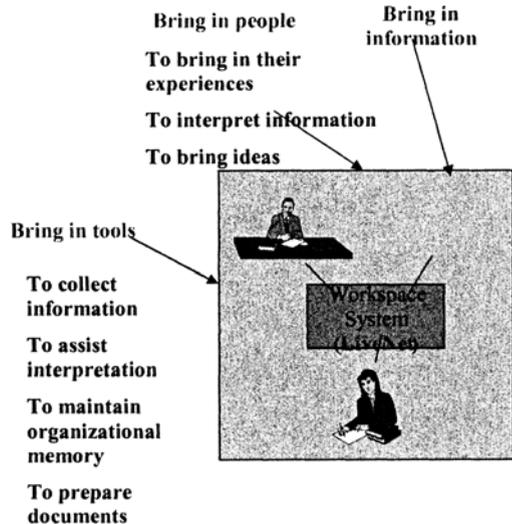


Figure 3: Setting up the workspace

## **2.1 System dynamics**

The general first step in most collaborative processes is to set up workspaces for different teams. The way to do this is illustrated informally in Figure 2. Here a manager may set up any number of groups, in this case the marketing and production planning groups. The manager will set the goals for these groups and provide them with initial information. This will be made available through the workspaces set up for the groups. The way workspaces are set up is illustrated informally in Figures 2 and 3. The steps involved are:

A number of people get together to address a long term issue,  
These people make arrangements to meet and work together,  
They bring in the information, tools to work in this area,

The goal of the remainder of this paper is to describe the semantics for such workspace evolution and a modeling process based on these semantics. The objective is for such semantics to be directly convertible to components within workspaces.

## **3. THE METAMODEL**

We have developed a set of metamodel concepts to describe such dynamics and directly support them with technology. These concepts are illustrated in Figure 4 and can be used to dynamically build work environments following the evolutionary path suggested in Figures 2 and 3. Each ellipse in Figure 4 represents one metamodel concept and links between represent the relationships between the concepts. We have also found that presentation of the concepts to users is critical and have developed alternate presentation methods for the different classes of user roles within an activity. The metamodel centers on generic concepts for collaborative processes (Hawryszkiewicz, 2001) that include roles, activities, artifacts and actions. The metamodel centers on shared workspaces that have any number of roles that are occupied by people. Recently we expanded the metamodel to include the idea of workgroups and thus support the ability of groups to evolve independently and later combine into more complex coordinated structures. It also supports scalability in the sense that independent workgroups can exist in the same system but gradually merge or intersect if needed.

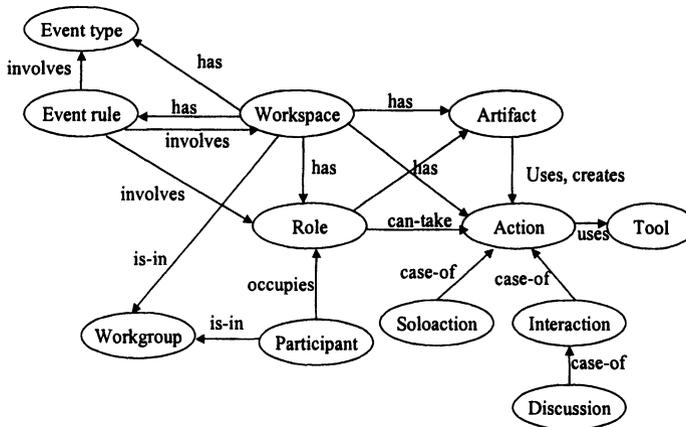


Figure 4: The metamodel for community evolution

The semantics are that a workspace represents an organizational recognized activity, whose goal is to produce an organizationally recognized artifact. Any number of people may be involved in the activity. The people are assigned roles, which can take a variety of actions, both individually or in collaboration with others. Usually the actions are combined into tasks that may be needed to produce intermediate outcomes prior to the production of the activity artifact. Each workspace can include a number of events that can lead to messages being sent to other workspaces, thus supporting emerging workflows.

Workgroups differ from workspaces in two major ways – separation of communities and management of growth. The general effect is that people in a workgroup can only participate in workspaces in that workgroup but not in workspaces of other workgroups. This is needed in situations where there is geographic separation with responsibilities vested in geographic teams, or in situations where one might be managing different client groups that require maintenance of confidentiality between them. With workgroups such confidentiality is maintained to the extent that members of one workgroup are not even aware of who the members of other workgroups are. It is of course possible to have people common to more than workgroup that can transfer knowledge between the workgroups or move people between them.

### 3.1 An Example

Figure 5 illustrates an example how the metamodel concepts can be used to model collaborative processes. Figure 5 illustrates the main terms of our

metamodel. It uses a notation based on rich pictures, which are derived from soft systems methodologies [Hawryszkiewicz, 2000]. Here there is a model of a system whose activities are shown as clouded shapes. Artifacts are here represented by rectangular boxes, and roles are represented by their names. Figure 5 illustrates a simple example that uses the notation to describe activities in organizing course presentation. A more detailed example of applying the method to model global processes can be found in [Hawryszkiewicz, 2000]. The example in Figure 5 shows three activities, which are shown as clouded shapes. Artifacts used in these activities are represented by rectangular boxes, and roles are represented by their names. There is a link between roles and the activities in which the roles participate. The activities in Figure 5 are:

- A *course development* activity to produce training courses that involves the service manager and experts,
- A *course selection* activity, where clients contact the service manager to select courses, and
- A *provide course* activity where a trainer, with possible assistance from an expert, provides a course to the client.

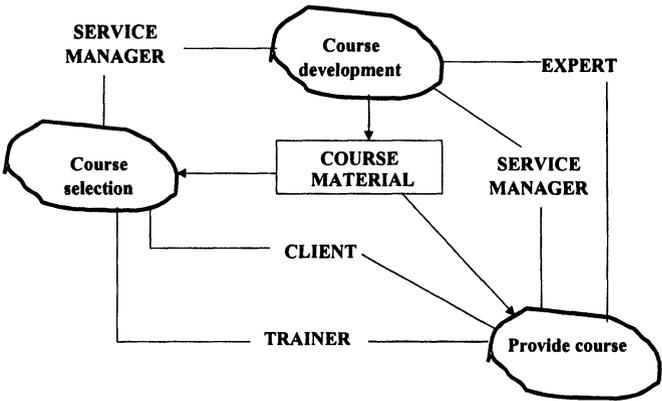


Figure 5: A rich picture for course presentation

Computer support for such activities centers on adapting to the work practices within each activity. It must also maintain awareness across activities by facilitating milestone formation rather than strict workflows.

The second important point to remember is that processes within such activities continually evolve – something not prevalent in transaction based systems. Even within enterprises they can start as a totally open community with an idea that it later pursues in some organized way. Alternatively existing workgroups can initiate new tasks and create structured workgroups

to follow them up. This then requires use of components that can be used by people within the activities to dynamically change activity processes

The paper now describes a way to reengineer the process to meet new business goals. The business goal is to change the process to provide specialized courses for clients.

## **4. DESIGNING THE NEW SYSTEM**

The design process (Hawryszkiewicz, 2001) is primarily made up of two steps, namely, to specify the requirements and then develop the design model.

### **4.1 Specifying Requirements**

Requirements definition proceeds in two stages. The first is to clearly define the business objectives, their impact on the existing process, expected benefits and how existing activities are affected. The second stage is to propose the business innovations needed to implement the objectives and their effect on work practices in the existing activities. These requirements can now be expressed in terms based on the metamodel. The requirements of the first stage are described in Table 1. Here there are three business objectives. These are to:

- Go beyond simple provision of course but define services that may include a course but are more closely integrated with the clients specialized needs,
- Reduce the cost of delivery of the service, and
- Provide after training service.

Table 1 also illustrates the kind of innovation needed to realize the objective and its expected effect. A similar table is then constructed to broadly define the new activities.

<b>Business objective</b>	<b>Business Innovation</b>	<b>Expected benefit</b>	<b>Effectuated activities</b>	<b>Proposed change</b>
Specialize service to client	Extend course preparation to include client	Better client retention through more specialized service	All activities	Rearrange client contact by specialized activity for each client. Create support role for each service
Reduce cost of delivery	Enable electronic delivery	Improve productivity of trainers	Training delivery	
Provide after training service	Provide electronic after sales service	Retention of clients and increased business with each client	Introduce after sales service	Extend activity to include after sales service

Table 1: Specifying business requirements in terms of the model

### 4.2 The Design Model

The challenge now is to define the new activities and describe them by a model. The design model itself is described by rich pictures and business scenarios. The major proposal here is to take the activity that concerns a client selecting from available services to one where the service is specially designed for a client. The process is predominantly creative requiring designers to implement the business goals within the business social context. It can include redefinition of roles as well as the creation of new activities. The rich picture of the new design of this example is shown in Figure 6.

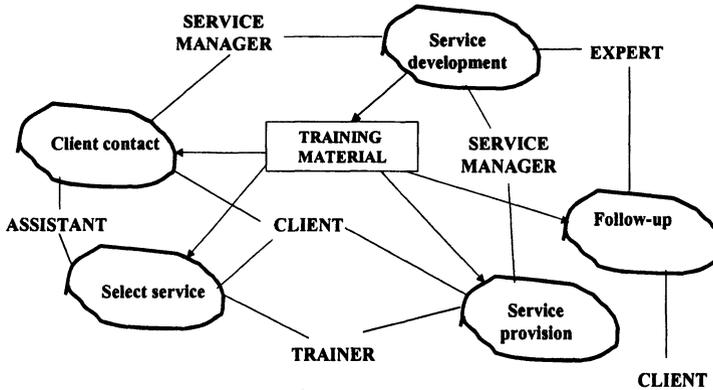


Figure 6: New design

We have also found it necessary to describe the scenarios or use cases in such processes. A typical scenario here may be:

1. The services manager looks at a client profile and develops a draft proposal based on existing courses.
2. The service manager contacts the client and discusses the proposal. Further changes can be made here to fit in with the client needs.
3. The service manager then assigns a trainer to the course.

The scenarios can then be used to identify the collaboration process and the tools needed to support it.

## 5. IMPLEMENTATION

We have developed a system to directly implement semantics of our metamodel. This implementation enables the design model to be directly mapped onto the system. The system to illustrate these ideas is known as LiveNet, which has been used to support teaching and meetings. The system contains 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. The interface is shown in Figure 7. It provides the following basic functional commands:

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.

A people component to construct workgroups groups and teams.

A component to include awareness features into the workspace, including news, terminology and milestones.

A component to support communications within the workgroup.

A workflow component to support coordination and exchange of information between workspaces.

Descriptive information about the workspace and roles within the workspace.

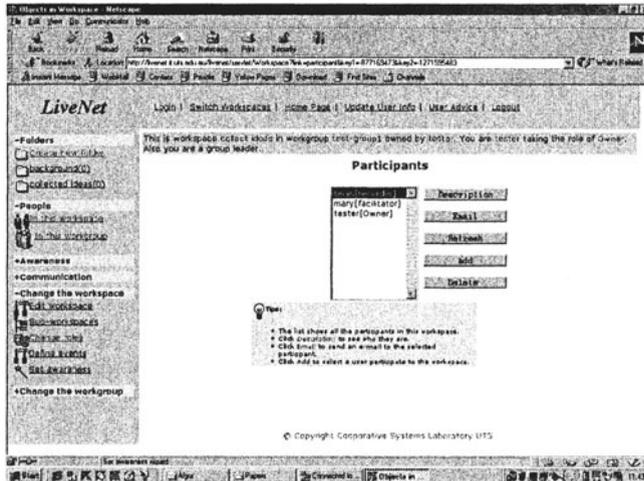


Figure7: A LiveNet Interface

The basic idea of customizing LiveNet applications is to follow a process that:

Starts by creating a community or workgroup

Then organize the work in the community into a number of activities, and define a LiveNet workspace for each activity,

Finally defines the precise practices followed by each activity by configuring each workspace, and if necessary organizes the activities into a business process.

The customization usually follows these steps:

Create a new group and designate leaders. A workgroup has at least one leader and a master workspace. A workgroup leader can add new users to the workgroup.

Create new workspaces within workgroups and designate their owner.

The workspace owner can create workspace roles and define their governance structures. Roles can be empowered in a range of ways, including ability to change workspace structures, the actions in the

workspaces, and access to documents. Personalization results in each participant only seeing those objects needed in their task.

Create folders with the explicit knowledge needed in the workspace.

Set up discussion or other ways that people within workspaces can interact.

Define workflow messages and notification schemes between workspaces.

Set up awareness features.

A workspace has one owner, who is the user, who created the workspace, and any number of participants. A workspace belongs to one workgroup. All participants in a workspace must be members of the workspace workgroup. Participants are invited into the workspace by the workspace owner, who assigns them to a workspace role. The workspace owner also defines what each role can do in the workspace. There can be any number of participants taking the same role, but each participant takes only one role.

Although the interfaces shown in Figure 7 provide the semantics for workgroup formation, we have found that many users face a cognitive gap in applying them. Two cases in point are use in teaching with group projects and in setting up meetings. In both cases workgroups needed to be set up.

Using high level semantics places the additional burden of users having to map their intuitive perception of their work onto workgroup semantics. Thus we have a student group, or a meeting. In either case we identify the workgroup involved as the group or the meeting. Its implementation using generalized semantics requires the user to first of all see their problem in terms of the generalized semantics and express it at the interface in terms of these semantics. Thus although the system is generalized in the functional sense and can be easily customized to a particular application, the customization still requires users to think in ways to map their problem semantics to generalized semantics when using the system, thus placing barriers in its use. Thus instead of selecting a button called meeting, they have to select a workspace. To invite a person they have to first define a role and then invite a person to that role.

Our current work is to find ways to provide technologies that can customize both the functionality and the interface terms. In that case it will be possible to quickly define interfaces that match the users intuitive perception to the problem. Currently to do this requires the development of a Java servlet that can take considerable time. Our work is to provide a way to define the interface terms in terms of the general semantics using a definition language.

Commands are expressed in relatively low level terms such as create a role, assign it to a workspace. This often means that in the cognitive sense a user

has to be translating their intuitive action in general semantic commands.

## 6. INTERFACE CUSTOMIZATION

Users of the general interface must express their commands in metamodel terms. Although these terms provide the ability to customize many different kinds of collaborative activities, many users face a cognitive gap in applying them and require training to do so. Using high level semantics thus places an additional burden on these users to map their intuitive perception of their work onto workgroup semantics. Often busy users prefer to use interfaces that directly represent the semantics of a specific collaborative activity. One of these is meetings. One goal then is to be able to follow the kind of collaboration pattern shown in Figure 3 by instantiating collaborative activities as needed while integrating them within the context of a foundation place. As an example Figure 8 shows the interface customization for meetings. Here there is an interface that keeps track of meeting currently between a group of people but ultimately to be integrated within a foundation place. The MeetingSpace provides ways to follow up issues within a particular context.

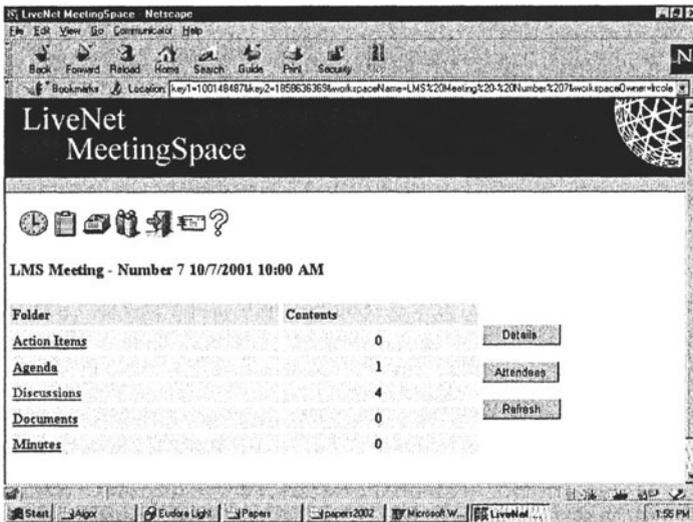


Figure 8: A MeetingSpace

The technology to achieve this requires inbuilt flexibility. Figure 9 illustrates the current implementation methods, which is based on Java. Here the metamodel is maintained on a relational database, with a server that

implements the metamodel semantics. Interfaces are then developed using Servlets using the server metamodel commands.

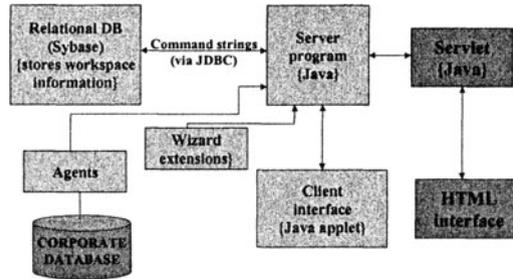


Figure 9: Technical Infrastructure

Currently to provide such an interface requires a special program. In our case, it requires the development of a Java servlet that can take considerable time. Our work is to provide a way to define the interface terms in terms of the general semantics using a definition language.

## 7. SOME EXPERIENCES

We have used the system extensively in teaching both to deliver teaching materials, interact with students and for students to set up cases studies using the system. We have developed a staged approach to the introduction of collaborative technologies. This starts with familiarization and socialization using the system in discussion mode and to distribute materials. In the next stage students set up project group spaces where they share information in their case study where they follow a design process to create a collaborative application. In the final stage they implement their design using the system.

Our current work is to provide technologies that can be customized both in functional and interface terms. It requires both a higher level 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.

## 8. CONCLUSION

This paper showed how design principles used in design methodologies for well-defined processes can be applied to the more emergent collaborative processes. It showed that a set of modeling concepts can be used to describe collaborative processes and that these concepts can be directly implemented using groupware. It then suggested that the design process can be made more effective if the modeling concepts are implemented using terms that intuitive to a collaborative activity. It described the implementation and ways to create special interfaces for different collaborative activities.

## 9. REFERENCES

- Checkland, P. B. and Scholes, J. (1990): *Soft Systems Methodology in Action* Chichester, John Wiley and Sons.
- Dourish, P. (1998): "Using Metalevel Techniques in a Flexible Toolkit for CSCW Applications" *ACM Transactions on Computer-Human Interaction*, Vol. 8, No.2, June 1998, pp. 109-155.
- 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.
- Hawryszkiewicz, I.T. (2001): "Developing the Infrastructure for Knowledge Based Enterprises" *Proceedings of the Third International Conference on Enterprise Information Systems*, Setubal, Spain, July, 2001, pp. 437-444..
- Hawryszkiewicz (2000): "A Framework for Strategic Planning for Communications Support" in Roche, E.M. and Blaine, M.J., "Information Technology in Multinational Enterprises" Edward Alger, Cheltenham, UK..
- 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.
- LiveNet: <http://livenet.it.uts.edu.au>
- eRooms: <http://www.eroom.com>
- Teamwave: <http://www.teamwave.com>