Design Methodologies for Collaborative Systems: Supporting the Services Industry

Igor Hawryszkiewycz
University of Technology, Sydney
igorh@it.uts.edu.au

Abstract

The growth of service industries has resulted in increasing demands for systems that support collaboration in the work context. Such systems must support processes that are adaptable to changing situations and whose workers are often mobile in nature. Most system development methodologies, however, are oriented towards defining processes that are integrated into the context but are fixed both in the process steps and often location of workers. This paper proposes a way to develop collaborative support services and methodologies for workers in service industries. The proposed system architecture will be able to capture changes in user work processes and provide users with customizable services to support collaborative context sensitive work.

Keywords:

Services, modeling, composite objects, engagements

INTRODUCTION

Modeling methods used in methodologies can be viewed as providing an ontology that can be used to describe systems. The ontologies of most existing system development methodologies are useful in describing predefined processes. Ontology concepts include data flows, entities, relationships or objects depending on the modeling method used. The ontology rules define basic process steps together with information flows between them. The process function usually defines a relatively well-defined function whereas the information flow, or communication, is often a self-contained simple message. Such methodologies are traditionally suitable for environments with well defined transaction flows such as bank transactions. A typical process may be preparing a submission in response to a client request, making a financial plan, identifying a strategic direction, or rescheduling some project activity.

Service oriented processes are different as they are mostly collaborative in nature and usually deal with more intense exchange of often complex information. Hence modeling methods where communication flows are usually simple messages can result in complex models. Furthermore, especially in asynchronous work, the implementation is to use e-mail to exchange these simple messages. In this case collaborators must themselves group the messages into meaningful collections to maintain awareness of the work context. Earlier research here (Cummings, 2002) has shown that there are limits to the kind of collaboration that can be supported in this way.

We propose an alternative approach that supports complex communication patterns and provides users with higher level messaging to maintain collaboration. Each of these higher level process steps achieves a higher level collaborative goal, as for example, agree on a document change, or finalize a budget.

The difference from current practice is illustrated in Figure 1. Currently the approach is to provide services for the exchange of simple messages. In this case collaborators must themselves group the messages into meaningful collections to maintain awareness of the work context. The alternate is to provide communication services that are oriented towards some limited collaborative goal. Here simple messages are grouped to support some limited goal, called engagement for the

purposes of this paper. This may be to get ideas, or prepare a proposal, or review the proposal Each engagement can include many messages but these messages all take place within the business context. The steps are based on customizable services that can both be easily integrated into the work context and adapted by their human users to work situation changes. Here simple messages are grouped to support some process step goal, called for want of a better term, engagement. A work process in a service environment is usually made up of many engagements. This may be to get ideas, then prepare a proposal, then review the proposal Each engagement can include many messages but these messages all take place within a context. The modeling goal is to go beyond the ontologies used in earlier methodologies and extend or replace them by ontologies suitable for describing the engagements in collaborative work.

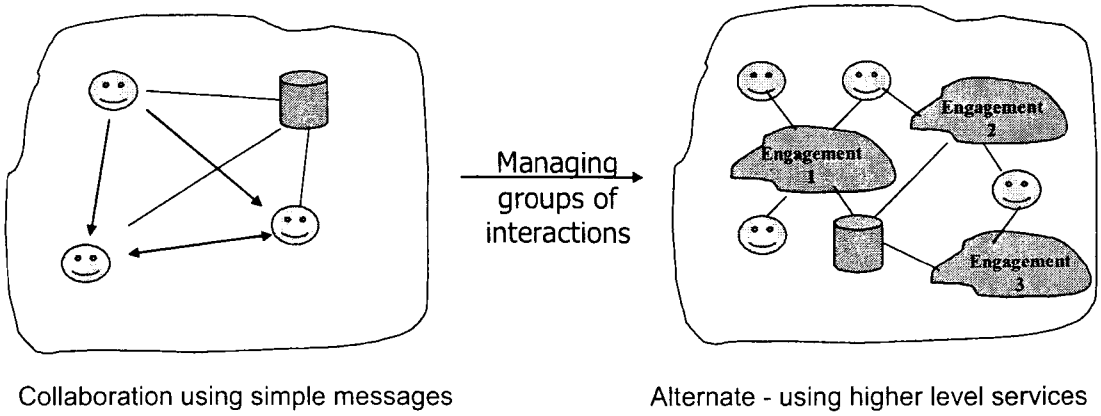

Figure 1 – Identifying patterns for collaboration in context

In summary, such service based processes are characterized by four major criteria, namely:

- they are user driven in that decisions are made dynamically by users,
- they are composed of complex steps requiring collaboration between users,
- the process itself can be changed dynamically by its users,
- users must be able to work anytime from anywhere using the available technology.

Methodologies for such processes must be based on semantics that can describe these process characteristics. They must identify the more complex process steps and provide services to support them.

OUR APPROACH

The general structure of the methodology is illustrated in Figure 2. It is based on an ontology that provides the concepts to describe collaboration in generic terms. The ontology is defined empirically from identifying commonalities in existing modeling methods as well as the study of collaborative applications. This ontology is then used for three purposes. These are:

1. To define a set of generic software services expressed in conceptual terms, and
2. Model applications in terms of these conceptual terms, and
3. Implement the conceptual model by matching the application semantics to service semantics.

The application model can then be used to select the higher level services using the generic concepts as search terms to discover and customize the services to the application.

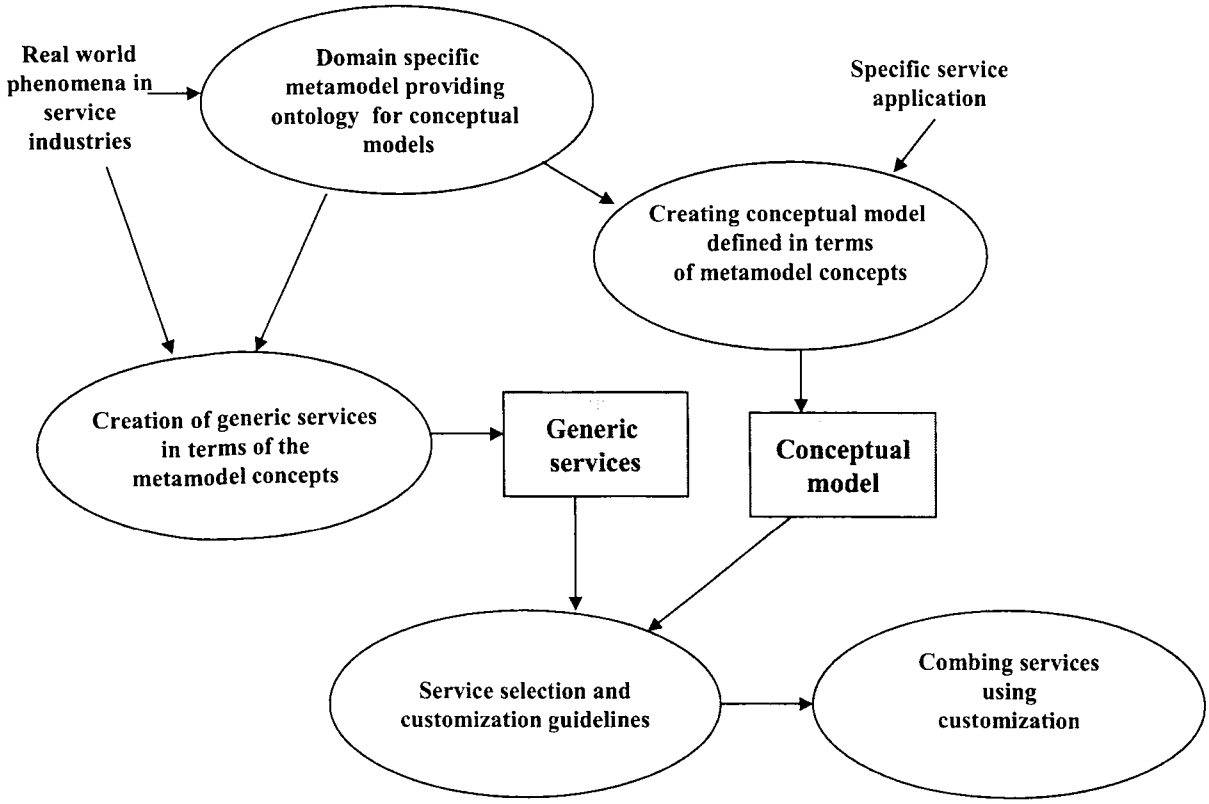

Figure 2 – Methodology philosophy

This paper proposes ways to identify such service communication patterns and support them within an integrated work context. From a modeling perspective each such general communication patterns can be viewed as composite objects (Shanks, Lansley, Weber, 2004) that can be represented in terms of more basic modeling concepts. We define an ontology for describing collaborative processes and suggest a number of composite objects suitable for communication in these processes. We then describe such objects and ways to implement them. We then show how such patterns can be integrated into design methodologies.

Resulting System Architecture

The architecture for implementing systems is shown in Figure 3. It integrates three components – the process structure, the work context and the technology. The architecture must provide the flexibility to adapt the service to the context and technology. Technologies can range from e-mail to workspaces provided through the Intranet. The software services will present the work context to collaborating mobile users. Thus mobile users will not only be able to communicate from anywhere but such communication will enable them to use a wide range of technologies available at their current site.

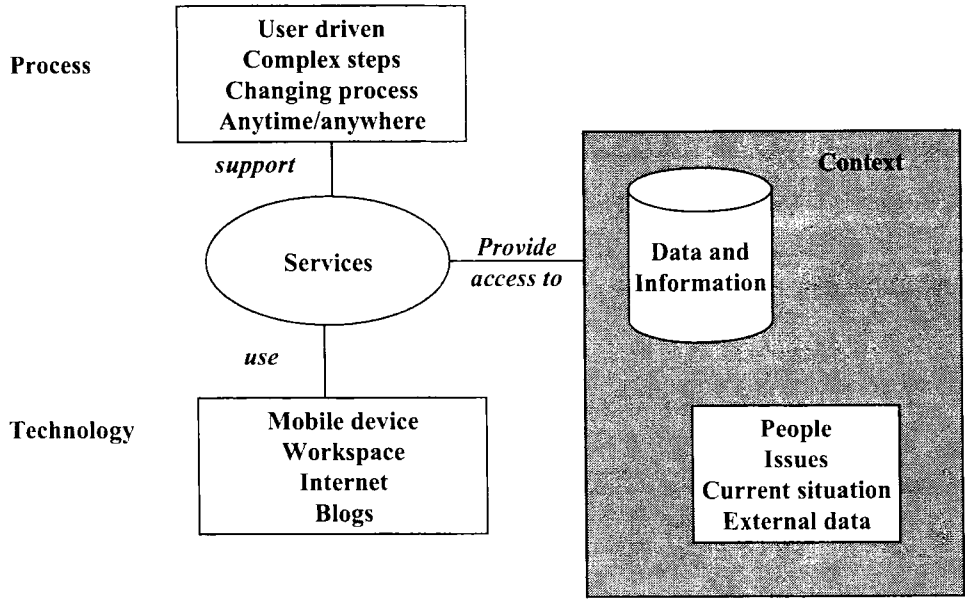

Figure 3 - Service architecture

The services are specified in general terms that can be adapted to a range of applications. These generic terms are provided by our ontological framework, which is described following.

AN ONTOLOGICAL FRAMEWORK FOR MODELING EMERGENT PROCESSES

The metamodel is briefly described in Figure 4 and more details can be found in (Hawryszkiewycz, 2005). In Figure 4, the rectangular shapes represent concepts whereas lines between the oval shapes are relationships between the concepts. The metamodel has evolved over a number of years. It includes concepts from earlier systems such as Conversation Builder (Kaplan, 1992) or Oval (Malone, 1992) and has been verified through a variety of applications that include business networking (Hawryszkiewycz, 1996), strategic planning (Hawryszkiewycz, 1997). Organization computational theory (Carley, Gasser, 1999) provides a further foundation for the metamodel. Figure 3 also groups the concepts into three parts, namely:

- The organizational concepts center on activities and work-actions. These actions usually need to access artifacts to refer them and to change some. An activity can include many work-actions, which in turn can use many artifacts. Responsibilities for such actions are assigned to designed roles.
- The social aspects center on people organized into groups. The groups can then assume roles with defined responsibilities in organizational activities. It provides ways to combine work-actions into activities with members of groups assigned responsibilities through roles for those work-items. Any participant can be part of a number of groups, and each group can have any number of participants.
- Workflows are supported by associating events with roles. People associated with these roles can initiate completion events, which in turn trigger initiation events that notify roles to carry out their tasks.

These three kinds of concepts are needed are essential for modeling business applications. Most processes follow a workflow, they involve organizational elements and they require social interactions to share knowledge.

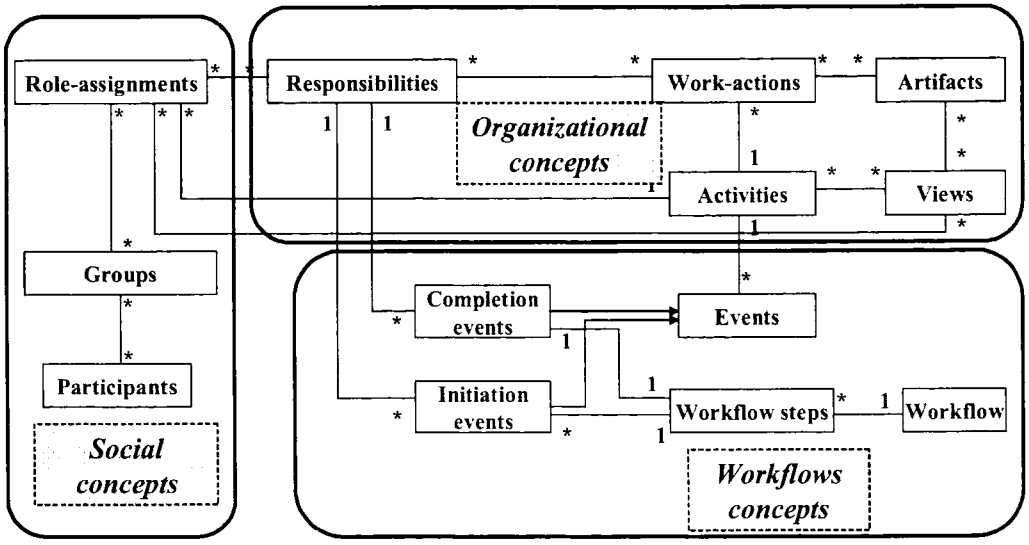

Figure 4 – A Metamodel for Defining Process Communication Patterns

The model includes a variety of commands that can be used up by agents to setup and change systems specified in terms of the model. These include creating new groups, activities and work-items and their associated views. They also include crating workflow events and issuing notifications. The concepts are used to describe systems by developing a conceptual model. In these terms the service identified in Figure 4 will be expressed in parameters that correspond to the model terms. A typical definition would be:

Artifacts: define the artefacts needed in a service and permissions given to roles;

Roles: define the roles in a service:

Actions: define the actions and roles permissions for these actions.

Identifying collaboration patterns

Communication patterns have been under study for many years. One of our goals is to identify patterns that can be applicable across many applications. The idea here is to group the metamodel concepts into larger composite objects (Shanks, et.al. 2004) that can be converted to services, which are used in collaborative applications. An example of two such possible composite objects, which are formed by groups of concepts, is shown on Figure 5. The two objects identified in Figure 5 are:

- Working on an artefact by a number of people. It is called an e-portfolio here and implies a collection of artefacts developed by a number of people. Different responsibilities are assigned in the e-portfolio. Examples include – education with teacher and student responsibilities. Strategic documents with planning and expert responsibilities or paper preparation with author and reviewer responsibilities,

- Arranging work actions associated with an activity – called a workflow. Here a workflow is defined in terms of events, which are assigned to roles. A completion event initiated by one role can result in a initiation event for some other role. The process can change dynamically by adding new events dynamically.

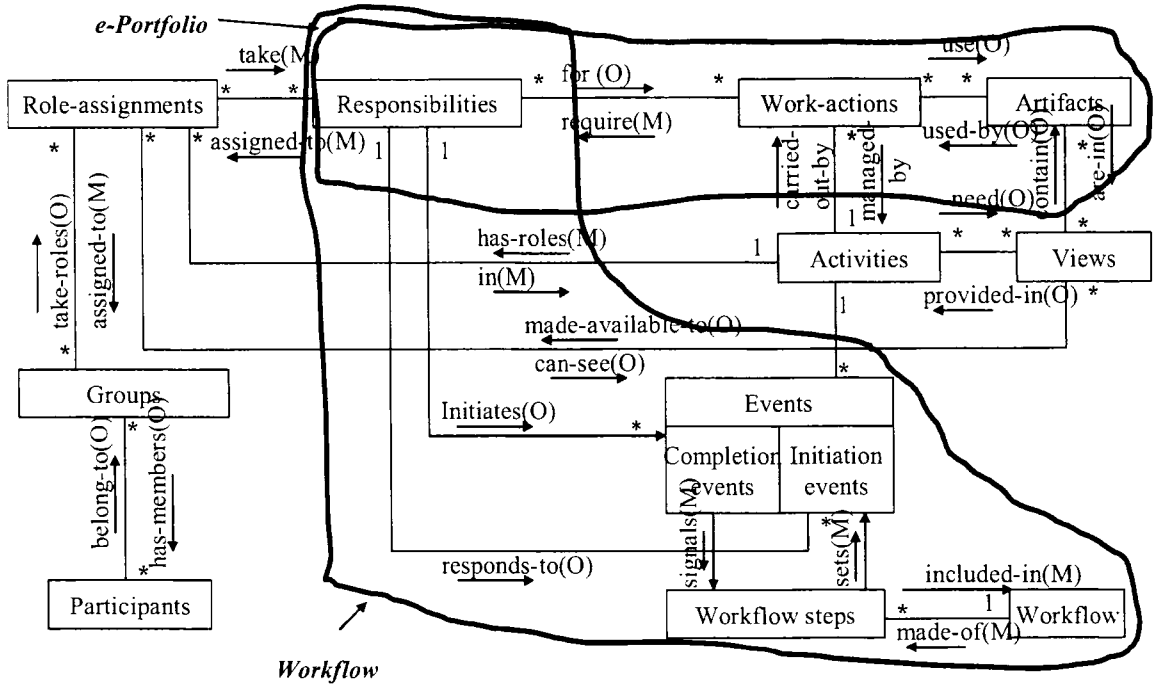

Figure 5 – Identifying generic services

A typical definition using XML-like structures is shown in Table 1.

Table 1 – Example of concept definitions

Concept	Abstract definition	Instantiation
Service	<service>: e-portfolio; <what-must-be-complete > work-action; <document-to-be-prepared> artefact; <tasks needed> responsibility, 'identifies tasks make up the work action'; <work-distribution> require, 'identify the tasks needed to complete the work-action'; <document assignment> used-by, 'defines allocation of tasks to roles'; Each has a tag, which describes why the component is needed and identifies a concept that must be created to realize the component. Both the service and its components must be defined when an actual application service is to be constructed	<e-portfolio> make-proposal-A , <what-must-be-completed> construct-document; <what-must-be-completed>: approve document; <document-to-be-prepared>: proposal-A; <task needed> update-document; <task needed> comment-on-document; <task needed> review-document; <work-distribution> construct document to update document; <document-assignment> proposal-A : construct-document; In the actual definition, the actual names are matched to concepts in the generic definition using the tags. Thus author is a role, update-document is a responsibility, and so on. The particular values and components are instantiated when the service is defined.

Concept	Abstract definition	Instantiation
Work-action The way of working specifies just that. How the work will proceed including reporting criteria, and levels of collaboration	<work-action> 'identifies the way of working': <reporting-method>: <value> <needed expertise>: <value>; <collaboration-level>: <value> <process type> {emergent, predefined}; };	<work-action> prepare-budget-part: <reporting-method>: 'notify on completion' <needed expertise>: 'budgeting'; <collaboration-level>: 'document exchange' <process type> emergent; };
Artifact	<artifact> 'identifies the document': <type>: <value> <location>: where the document is located;	<artifact> proposal-A: <type>: word-file <location>: main-office;

The example only includes part of the definition. A complete definition would include a number of work-actions each being responsible for a different part of the document.

Building Application Models

Thus the emphasis of the model is not to focus on identifying data objects and processes as is the case with many methodologies. On the other hand, the goal is to identify organizational entities and their agencies. This approach also supports the notions of (Rehfelldt, 2000) of organizations supported by interacting agencies.

The concepts can be used to build application models and identify generic services. Figure 6 shows a top level diagram of a model of making an insurance claim. There are two main activities – making a claim and assessing the claim. We can expand to show the detailed work in each of these activities.

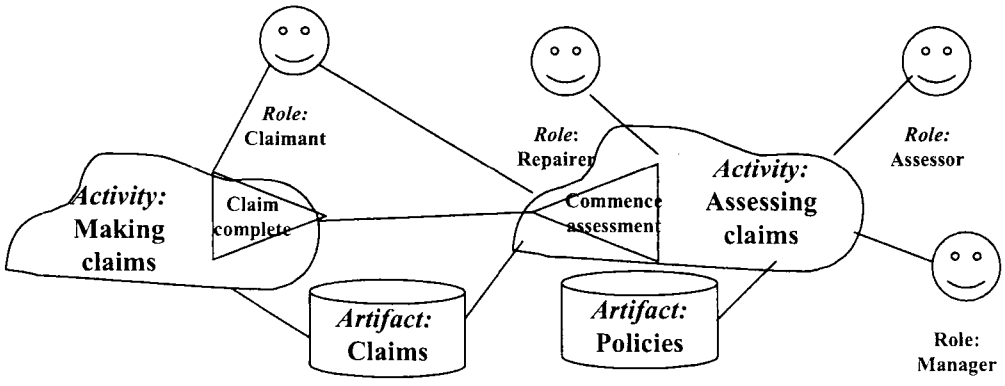

Figure 6 – A conceptual model of the application

Figure 7 is a detailed lower model of activities for assessing claims. The work activities in the lower level model describe the higher level collaborative activities in more detail. It specifically defines the more detailed responsibilities of the different roles. Thus for example the manager is responsible for making the final decision on a claim. The assessor on the other hand gathers the quotes from different repairers.

Completing the Design Methodology

Our proposed methodology is to use the work items identified during analysis to select generic services. The activity identifies the context needed by the work-item. This context can go beyond the specific artifacts needed by the work-item and also include more general artifacts that are often provide the guidelines needed in decision making. The general mapping is:

Work-item→ Service

Activity → Context needed by the service

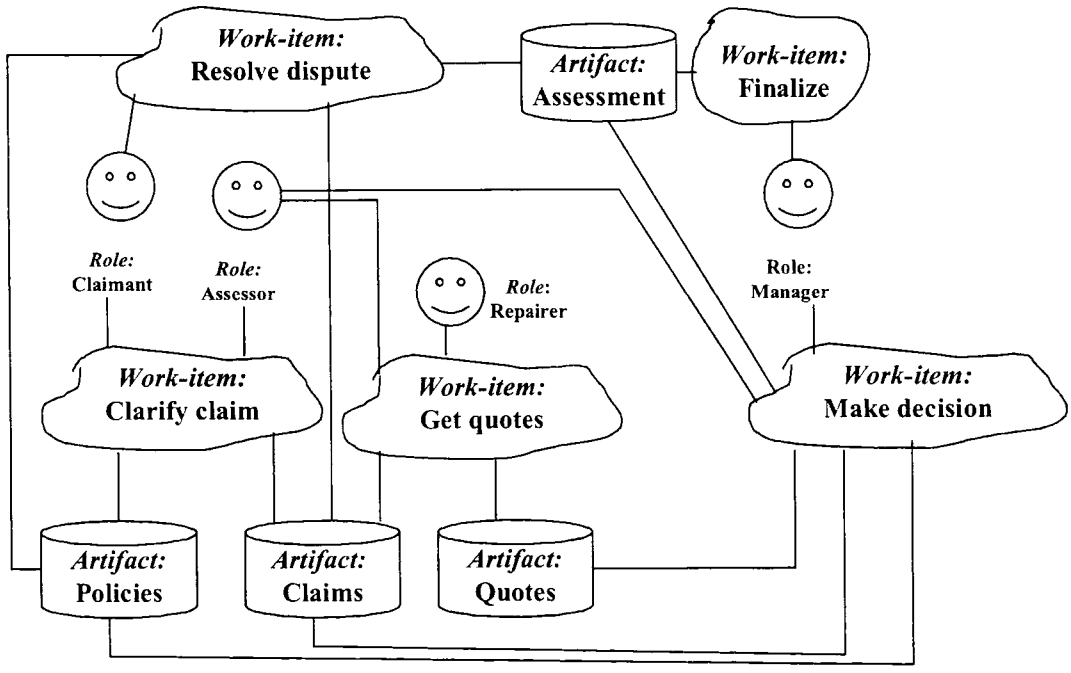

Figure 7 – Detailed model of the 'assessing claims' activity

Thus there can be any number of e-portfolios in the system. Get quotes can be an e-portfolio as can clarify claims. One of these can simply include the claims and policies and include the claimant and assessor for completing the claim. Thus in Figure 7 work items such as clarify claim or make decision can all be realized as an e-portfolio.

IMPLEMENTATION

One simple implementation is to use workspaces that directly support the metamodel concepts. The services oriented approach simplifies the design of such workspaces. This system, like many other workspace systems, supports the concepts of the ontological model. A communication pattern would then be provided as a template that supports the communication structure. It would include the generalized roles, artifacts and Each of the activities in a diagram like that shown in Figure 7 are then mapped onto the pattern and the pattern is then implemented by customizing the template.

The workspace system provides the customization needed to implement the services. Thus any roles, documents and work-items can all be added to the workspace and can be accessed through the workspace. Similar mappings can be made to other technologies but the flexibility provided by workspaces tends to give it some advantage.

As an example the 'get-quotes' activity shown in Figure 7 can be implemented as an e-portfolio. This is presented by one workspace as that shown in Figure 8 using our LiveNet system. It

includes the assessor and repairer roles as well as the artifacts shown in Figure 7. The availability of such generic services can simplify implementations and support change. The workspace shown in Figure 8 is reached from the activity 'assessing claims' which provides the higher level context.

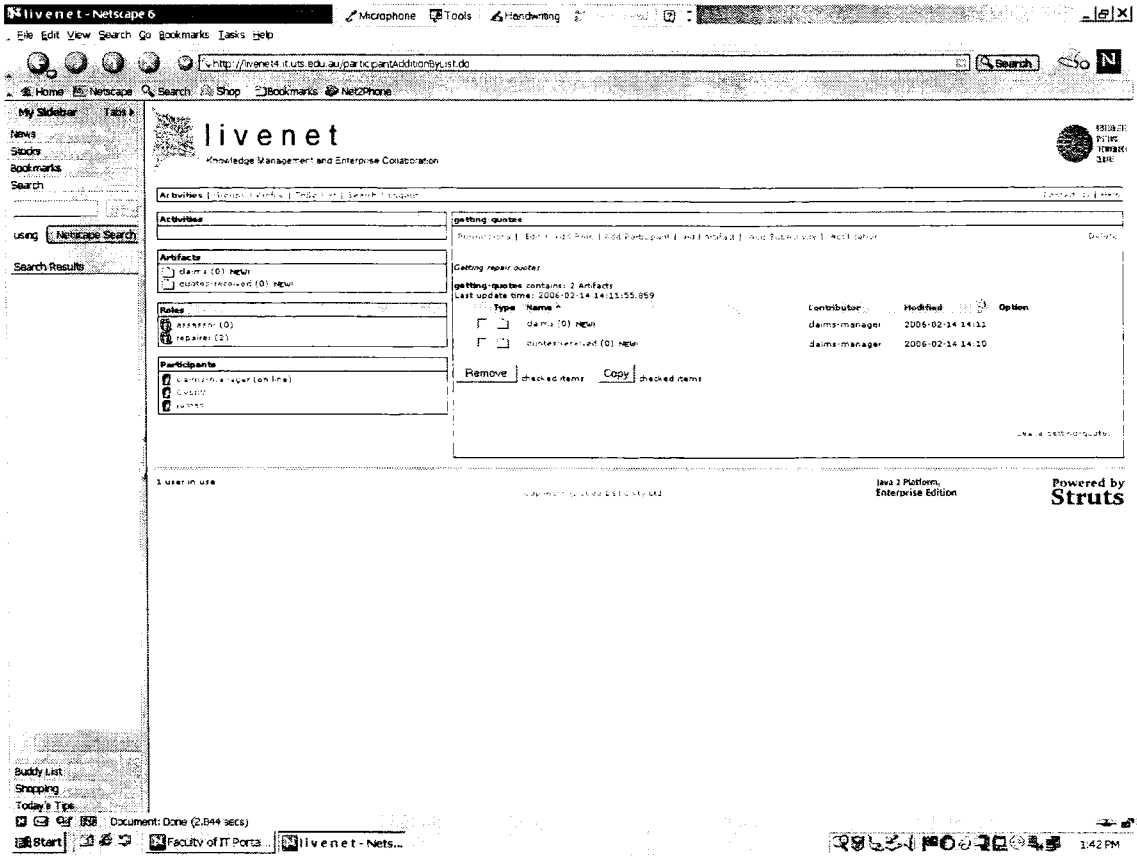

Figure 8 – Implementation of the get-quotes work-item

SUMMARY

This paper developed a design methodology for applications that need to adapt easily to new communication needs. It emphasized the development of generalized communication services that can be customized to many applications. It showed how such services can be identified using a generalized ontology for collaboration. Each generic service can then be implemented as a workspace that can be adapted to a particular application. Our future work will be develop a complete set of services based on the ontological model shown in Figure 4 and to examine mappings to other technologies especially mobile technologies (Hawryszkiewycz, Steele, 2005).

REFERENCES

Belanger, F. (1999): "Communication Patterns in Distributed Work Groups: A network Analysis" IEEE Transactions on Professional Communication, Vol. 42, No. 2, December 1999, pp261-275.Carmel, E. (1999): "Global Software Teams" Prentice-Hall, Upper Saddle River.

Carley, K.M., and Gasser, L. (1999): "Computational Organizational Theory" in Chapter 7 "Computational Organization Theory" by KM Carley & L Gasser in "Multiagent Systems" Gerhard Weiss (Ed) MIT Press -- 1999

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.

Do van Thanh, Jorstad, I. (2005): :A service-oriented Architecture Framework for Mobile Services" Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources conference" July 2005, pp. 65-70.

Ducheneaut, N. and Bellotti, V. (2001): "E-mail as Habitat' Interactions, September-October, 2001, pp. 30-38.

Geurts, G. and Geelhoed, A. (2004): "Business Process Decomposition and Service Identification using Communication Patterns" MSDN Library, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmaj/html/aj1bizproc.asp (last accesses 1 Sept. 2005)

Gloor, P.A., Laubacher, R., Dynes, S.B.C. and Zhao, Y. "Visualization of Communication Patters in Collaborative Innovation Networks: Analysis of some W3C working groups" http://ccs.mit.edu/pgloor%20papers/COIN4CIKM1.pdf (last accessed Sept. 1, 2005)

Hawryszkiewycz, I.T (2005): "A Metamodel for Modeling Collaborative Systems" Journal of Computer Information Systems, Vol. XLV, Number 3, Spring 2005, pp. 63-72.

Hawryszkiewycz, I.T. (June, 1996):" Providing Computer Services For Business Networks" *Proceedings of the Ninth International Conference on EDI-IOS*, ISBN-961-232-000-4, Bled, June, 1996, pp. 398-411.

Hawryszkiewycz, I.T., Steele, R. (2005): "A Framework for Integrating Mobility into Collaborative Business Processes" Proceeding of the Conference on Mobile Business, Sydney, July, 2005, pp. 89-93..

Huhns, M.N., Singh, M.P. (2005): "Service-Oriented Computing: Key Concepts and Principles" IEEE InterNet Computing, January-February, 2005, pp. 75-81.

Kaplan, S.M. , Tolone, W.J., Bogia, D.P. and Bignoli, C. (1992): "Flexible, Active Support for Collaborative Work with ConversationBuilder" Proceedings of the CSCW'92 Conference, November 1992, Toronto, pp. 378-385.

LiveNet: http://livenet4.it.uts.edu.au

Malone, T.W. and Fry, C. (1992): "Experiments with Oval: A radically Tailroable Tool for Collaborative Work" Proceedings of the CSCW'92 Conference, November 1992, Toronto, pp. 289-297.

Ott, M. and Nastalsky, L. "Modeling Organizational Forms of Virtual Enterprises" http://fb5www.uni-peterbon.de/winfo2

Shan, T.C. (2005): "Building a Service-Oriented eBanking Platform" Proceedings of the 2004 IEEE International Conference on Service Computing (SCC04), pp. 237-244.

Shanks, G., Lansley, S., and Weber, R. (2004): "Representing Composites in Conceptual Modeling" Communications of the ACM, Vol. 43, No. 7, July, 2004, pp. 77-80.

Younas, M., Chao, k-m., Laing, C. (2004): "SODA: Service Oriented Design Activities" Proceedings of the 8th. International Conference on Computer Supported Cooperative Work in Design, Vol. 2, May 2004, pp. 419-424.