Specifying collaborative systems requirements in terms of service platforms

Return to Published Papers

I.T. Hawryszkiewycz
Faculty of Information Technology
University of Technology, Sydney
e-mail: igorh@it.uts.edu.au

Abstract

Increasing collaboration in business processes has resulted in increased demand for information systems that support collaboration. Many system development methodologies, however, are oriented towards defining processes that are fixed both in their process steps and process functions. Collaborative systems however are more uncertain in nature and processes usually evolve to accomplish the goal. Hence requirements cannot be specified in detail. This paper proposes a way to identify collaborative requirements in higher level terms and provide platforms of services that support collaborative process evolution while at the same time ensuring that system goals are met.

1. Introduction

Requirements analysis is seen as a process of elicitation, modelling and specification development. The elicitation primarily covers identifying the issues faced by an organization and identifying the actions that must be taken and setting priorities. The modelling assists the process by clarifying system structures, whereas the specification defines precisely the decisions taken and what must be done. It is increasingly recognized [1] that the wide range of processes means that requirements modelling and the specification format has to be adaptive to the kind of process. Hence different kinds of requirements engineering is needed to cater for all kinds of processes. This is particularly the case with the growth of virtual organization [2] with increasing collaborative requirements.

Figure 1 shows the process range in two dimensions. The horizontal dimension is the complexity of a process, ranging from predefined to emergent. The vertical dimension defines the level at which a process is specified. At the top level this may be a generic description such as CRM, the midlevel defines the generic work practice, and specific level the typical tasks. The design method depends on the position of the application in the spectrum. Design methods used in methodologies can be viewed as providing the ontology for describing systems in terms of a model. The model must cater for the kind of process structure. For example in predefined processes the emphasis is on tasks and flows of information between tasks. Ontology concepts to describe such processes include data flows, entities, relationships or objects depending on the methodology used.

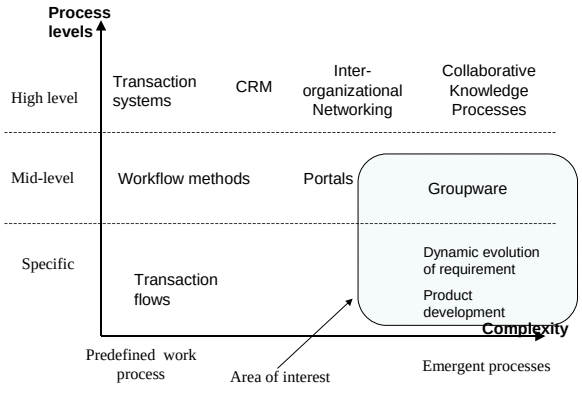


Figure 1 – Range of Processes

Collaborative systems on the other place more emphasis on people responsibilities and their collaborative needs. A typical process may be preparing a submission in response to a client

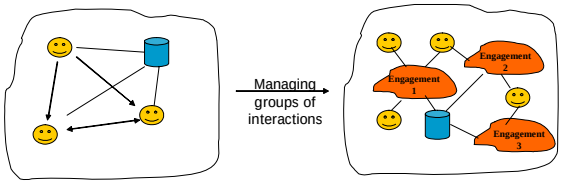
request, making a financial plan, identifying strategic direction, or rescheduling some project activity. The concepts here concern people responsibilities, interactions between people and changes to such interactions as a collaborative situation evolves. In such cases there is considerable uncertainty on the exact processes that will be followed. The difference of collaborative processes from predefined processes is illustrated in Figure 2. Predefined processes provide services for the exchange of simple messages. In collaborative processes, communication services are often oriented towards a combination of limited collaborative goals. Hence 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. A collaborative work process is usually made up of many such engagements.

Requirements analysis should thus specify the kinds of engagements to be supported, together with the platform that provides the services that support the engagements and allow users to dynamically configure them in applications. This platform is sometimes called the Cyberinfrastructure [3].

1.1 The proposed approach

The requirements are this made up of two parts, namely:

- The generic services needed by an application, and
- The combination of these services into applications.



Collaboration using simple messages

Alternate - using using higher level services

Figure 2 – Identifying patterns for collaboration in context

These services must support the collaborative environments, which:

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

The outcome then is a specification of services required to maintain collaborative activity and the ability of this activity to evolve. Users will be guided to choose services at execution time depending on the situation at the time. The goal is to develop what are commonly known as lightweight systems that can be easily understood and driven by users. The method used is shown in Figure 3. It starts with a high level requirement of what kind of collaboration is to be supported and then expands it into detailed services that support collaborative engagements. The process begins with a high definition of collaborative activity and then reduces to more detailed requirements in terms of *engagements*, finally defining requirements in terms of groupware communication services. Such engagements are used here as a general term to describe interactions and artifacts involved in the interactions.

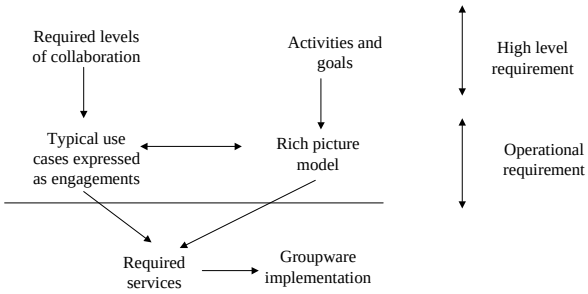


Figure 3 – Requirements development process for applications

The kind of engagement will depend on the collaboration level supported. The paper first defines levels of collaboration. The paper then identifies generic engagements and describes where they are applicable and how they support collaboration in work processes.

The paper also defines an ontology to define people's roles and the engagements with others. The goal of this paper is develop a lightweight method to identify requirements of collaborative systems.

2. Top-level requirements

The collaboration levels [4] proposed in this paper are shown in Table 1. These levels provide a way to gradually increase levels of sophistication as the interaction between people increases with each level. Table 1 describes each level and its characteristic, the technology needed as well as the knowledge needed to realize the level. The levels are:

Event notification, where roles are informed of any changes that effect the roles.

Document sharing, where documents are distributed between responsible roles,

Work process support, which often defines monitoring levels of activity and sending reminders to collaborators,

Joint work, where users work together in a synchronous manner, and

Joint goal setting, where people jointly decide how they will work together.

Usually collaboration starts with simply notifying people of changes that can impact on their work. Then document sharing is added to ensure that people are provided with information needed to carry out their responsibilities. Subsequent levels are more complex as they require more intense interaction to coordinate activities. Work process support, requires a precise definition of the way a collaborative process takes place. It includes the definition of responsibilities of identified roles. For example the process may define a way to propose a response to a customer request. The specific rules may define the expertise needed to define a solution, the risk assessment, budgetary evaluation, legal aspects and so on. Joint work is an extension of level 3 by providing ways to carry out synchronously thus reducing completion time. Defining the processes to be followed also requires collaboration and agreement on the ways people will work to achieve organizational goals. This process must be clearly defined and clearly understood and followed. Joint planning requires involved units together plan and agree on their work processes. This level often requires support for asynchronous work as goal setting often includes resolving many imprecisely defined alternatives.

Table 1 – Levels of Collaboration

Level	Characteristics of collaboration levels	Knowledge requirements
Collaboration level 1 Event Notification	Informing people about events related to their roles. Presenting the functional situation globally.	People responsibilities in the organization and their location or contact.
Collaboration level 2 Document Sharing	Sharing explicit information. Presenting to roles responsible for functional units. Obtaining comments on information.	Role responsibilities and information and documents that they need.
Collaboration level 3 Work process support	Explicit definition of work activities and responsibilities. Definition of relationships between tasks. Group meetings to resolve issues.	Optimum team structures for identified situations. Location of experts. Ways to assign responsibilities.
Collaboration level 4 Joint work	Jointly create and develop artifacts.	Location and responsibilities of people involved in a task.
Collaboration level 5 Joint planning	Developing shared plans. Devise coordination strategies between functional units. Develop and agree on work processes.	Organizational strategy and mission. Responsibilities of different organizational units.

The collaboration levels provide a way to introduce technology with increasing levels of sophistication. Usually this starts with awareness, which simply concerns notifying people of changes that can impact on their work. Then document sharing is added. Subsequent steps are more complex as they require more intense interaction to coordinate activities. This paper concentrates on ways to support higher levels of collaboration using software agents.

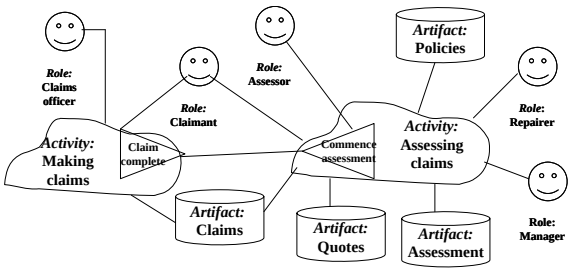


Table 2 – Collaboration Capability Levels

Collaboration Capability	Description	Expected benefits
Capability Level 0 Ad-hoc	Use of current technologies in ad-hoc ways. Messages sent and documents exchanged in unpredicted ways depending on user preferences.	Available technologies such as e-mail or portals facilitate exchange of information.
Capability Level 1 Functional coordination.	Policies exist for maintaining awareness in local functional areas and for updating documents. Usually requires level 1 and 2 levels of collaboration. Can apply to individual business units or across business units.	Consistency of produced documents and less duplication and unnecessary work. Improved awareness of what is going on in different tasks.
Capability Level 2 Process coordination.	Work across different units is formally coordinated. Requires collaboration level 3.	People across the organization can respond quickly to changes. Ability to provide global responses quickly.
Capability Level 3 Work alignment	Work processes shared across business units. Individuals discuss (usually synchronously) ways documents and processes should be managed before implementing them. Requires collaboration level 4 and higher.	Quicker alignment of business units to business goal. Ability to provide global solutions quicker.

1.2 Collaboration Capability Level

The idea of collaboration capability comes from earlier adoption of the idea of capability maturity model adopted in software engineering. This centered on defining the process requirements that ensure the development of software products. Similarly the idea of collaboration capability is to define processes needed to ensure the effective sharing of knowledge to develop ways to respond to situations. Table 2 defines the collaboration capability levels.

Having decided on the required collaborative capability level the next step is to define the kinds of engagements needed to realize the level.

1.3 Defining high level application requirements

The concepts can be used to build application models and identify generic services. Figure 4 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 4 – A lightweight conceptual application model

At this stage the collaborative levels and goals are also specified. For example, the high level collaborative requirement may be level 1 where level 2 must be supported in making claims and level 3 is needed in assessing claims. Goals can also be specified. For example:

Making claims: Complete a form with assistance of a claims officer, minimizing the time of claims officer assistance while provide complete

information. Requires level 1 collaborative capability.

Assessing claims: Follow a process where the cost of the claim is assessed, culpability is established, and decision made on how to proceed with the claim. Requires level 2 collaborative capability.

The next level must define the requirements needed to support the level collaboration. To do this we use a general ontology of collaboration to identify the generic engagements and include them in the specification.

3. Specifying Generic Engagements

From a modeling perspective each such general engagement can be viewed as a composite objects [5] that can be represented in terms of more basic modeling concepts. We thus identify generic engagements based on metamodel of collaboration. The metamodel is briefly described in Figure 5 and more details can be found in [6]. 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 [7] or Oval [8] and has been verified through a variety of applications that include business networking [9], strategic planning [10]. Organization computational theory [11] 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.

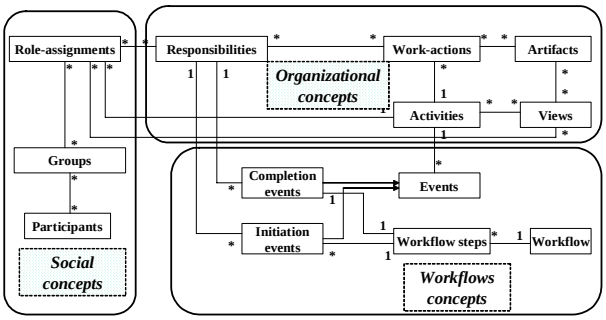


Figure 5 – A Metamodel for Defining Process Communication Patterns

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.

3.1 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 [5] 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

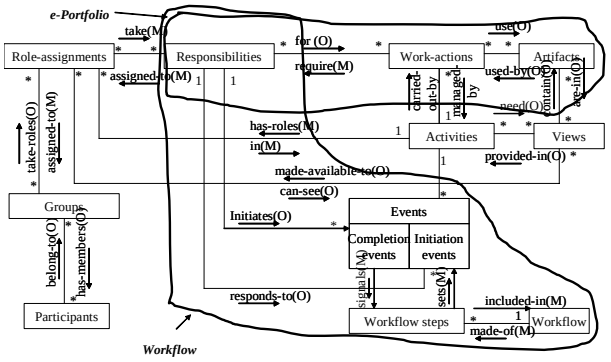


Figure 6 – Identifying generic collaboration patterns

The two objects identified in Figure 6 are:

e-portfolio – Supports working on an artefact by a number of people. It supports 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. The parameters of this engagement will be document names, roles and role responsibilities for each document.

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

There are other engagements not illustrated here that include **group management** or **team formation** or **program and issues boards**. There are a number of advantages of using such higher level concepts in collaborative systems. One is to provide a social construct that can be easily understood. Another is that engagements as particularly suitable as a way of integrating processes. It provides such a basis ranging from predefined processes to emerging processes that include supporting mobility in the workforce. It can be used as the basis for supporting communication beyond the simple exchange of messages to supporting more goal oriented communication that integrates a number of messages into the one engagement. It however sees that support must be provided to manage such engagements and suggests agents as suitable for this purpose. Conceptually it can be viewed as a composite object [5] that can be represented in terms of modeling concepts such as entities or relationships.

Low collaboration levels usually require engagements such as e-portfolio and perhaps group management. Higher levels in particular those supporting work processes will require engagements such as Higher levels of collaboration will require engagements such as team formation or workflow instance.

4. Identifying engagement requirements for applications

Application requirements identify the engagements to be supported and the people and documents involved in each engagement. To do this the high level application conceptual model is expanded into a more detailed form. Figure 7 is a detailed model of the assessing claims activity. 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.

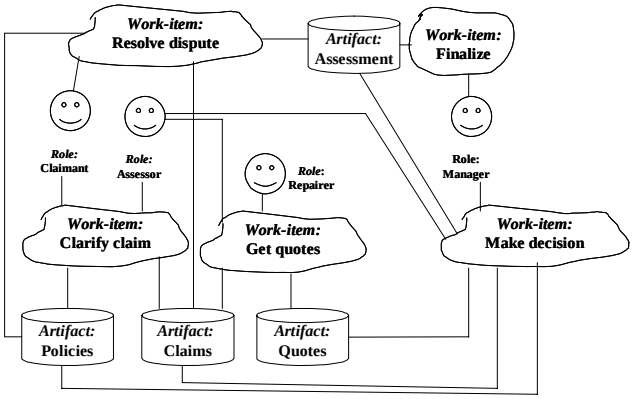


Figure 7 – Detailed model of the 'assessing claims' activity

High level use cases can now be defined in terms of the composite objects, as for example a high level use case for 'making a claim' would take the form:

Create e-portfolio for claim,
Fill in claim form in e-portfolio,
Send e-portfolio to claims officer for comment,
Amend claims form,
Submit e-portfolio for assessment.

The structure of the e-portfolio would also be part of the requirement. This would require the definition of the specific roles, artifacts and services required in the portfolio. These can be found from the rich picture like that shown in Figure 7.

5. Implementation

One simple implementation is to use workspaces that directly support the metamodel concepts. An e-portfolio is presented by one workspace as that shown in Figure 8.

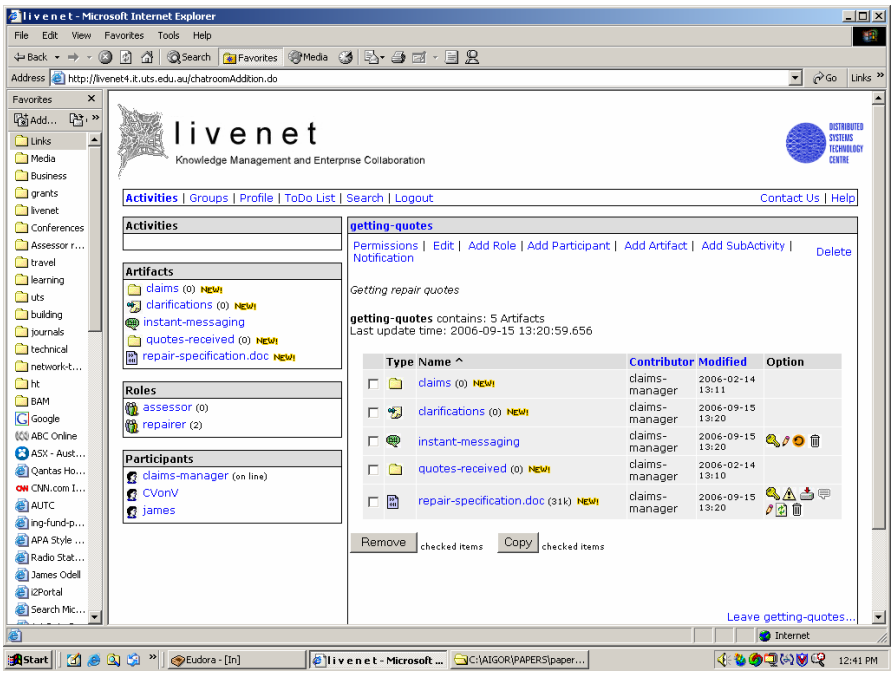


Figure 8 – Implementation of the get-quotes activity

The workspace 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. For example, in this case the roles are repairer and assessor as shown in Figure 7. It also includes a discussion forum for clarifications and support for instant messaging. Similar mappings can be made to other technologies but the flexibility provided by workspaces tends to give it some advantage.

6. Summary

This paper developed a service design methodology. It emphasized the development of generalized services that can be customized to many applications. 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.

7. References

1. Padula, A. (2004): "Requirements Engineering Process Selection at Hewlett-Packard" Proceedings of the 12th International Requirements Engineering Conference.
2. Ott, M. and Nastalsky, L. "Modeling Organizational Forms of Virtual Enterprises" http://fb5www.uni-peterbon.de/winfo2
3. Zimmerman, A., Nardi, B.A. (2006): "Whither of Whether HICI: Requirements Analysis for Multi-Sited, Multi-Iser Cyberinfrastructures" CHI 2006, April 2006, Montreal, Canada.
4. Hawryszkiewycz, I.T. (2006): " A Framework for Raising Collaboration Levels on the Internet" 7th International Conference on Electronic Commerce and Web Technologies, Krakow, Poland, pp.82-91.
5. 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.
6. Hawryszkiewycz, I.T (2005): "A Metamodel for Modeling Collaborative Systems" Journal of Computer Information Systems, Vol. XLV, Number 3, Spring 2005, pp. 63-72.
7. 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.

8. 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.
9. 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.
10. Hawryszkiewycz, I.T. (1997): "A Framework for Strategic Planning for Communications Support" *Proceedings of The inaugural Conference of Informatics in Multinational Enterprises*, Washington, October, 1997, pp. 141-151.
11. 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