

© 2002 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Assisting the Design of Virtual Work Processes via On-line Reverse Engineering

Robert P. Biuk-Aghai  
University of Macau  
Faculty of Science and Technology  
P.O. Box 3001  
Macau S.A.R.  
China  
fst.robert@umac.mo

Simeon J. Simoff  
University of Technology, Sydney  
Faculty of Information Technology  
P.O. Box 123  
Broadway, NSW 2007  
Australia  
simeon@it.uts.edu.au

## Abstract

*The design of virtual workplaces that can support virtual work processes has traditionally been either ad-hoc, or has been influenced by the virtual architecture or requirements engineering disciplines. The problem with these approaches is the difficulty in obtaining, and subsequently retaining and reusing, ready-made configurations of collaborative work processes. Such configurations naturally occur during the actual use of CVEs for conducting projects. Can we predict some elements of the evolution of a new collaborative process, based on similarities and analogies with processes formalised and supported before? Can we capture and utilise the evolutionary component in the workspace design process, so that we can provide better support to the developers of collaborative workspaces? The paper presents a new approach for supporting design and redesign of virtual workspaces, based on combining data mining techniques for refining lower level models with a reverse engineering cycle to create upper level models.*

## 1. Introduction

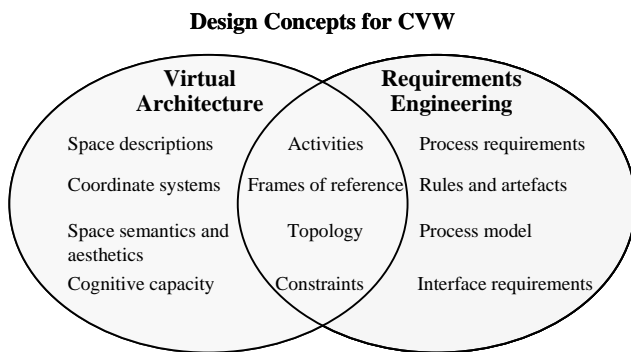
Supporting collaboration among a distributed group of users can be complex and time-consuming. One of the key components in setting up virtual collaboration is the design of the *virtual workspace*—the information environment that unites the networked computers in a coherent medium to support the activities involved in collaborative project development. In the early days, in order to develop such collaborative environments, the developer had to be proficient in conceptual modeling, network programming, object management, graphics programming, device handling and user interface design. During the recent years, an alternative approach has been to *design collaborative virtual workplaces* (CVWs), using

and integrating existing underlying groupware technologies. These technologies provide virtual environments, workspaces, or virtual places for collaboration, and can be populated with people, items and tools required for collaboration. We refer to the various kinds of such groupware technologies collectively as collaborative virtual environments (CVEs). The design of virtual workplaces in this case is focused on the “arrangement” of the workplace in a way that will support computer-mediated collaboration between geographically dispersed participants. The goal of the design of a workplace is to meet some needs (requirements) of the collaborators, whether this be an educational, research or business collaboration. Such requirements are usually expressed in terms of activities (see [12, 13] for examples of how the notion of activities is used in design, and how a design ontology can be refined, respectively) and their attributes (e.g. people who are executing those activities, objects involved in the activities, etc). Thus, the design can be viewed as an ordering and definition of a semantic information space and types of objects that inhabit that space. Virtual architecture and requirements engineering (see Figure 1) are the two most common approaches to systematic design of collaborative virtual workplaces.

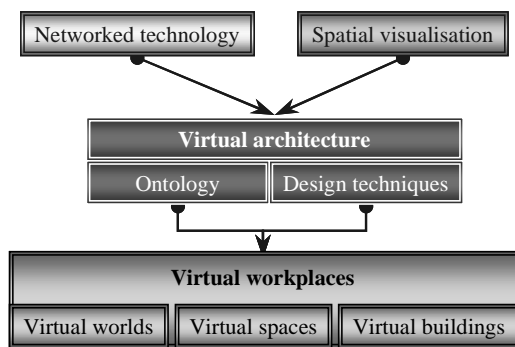
### 1.1. Virtual architecture approach

In virtual architecture, the left half of Figure 1, the emphasis in the design is in reflecting some social and cultural needs and values in particular forms [1]. The establishment of a sense of place is achieved by creating a communication location for groups who share a common interest or purpose. The idea of the virtual architecture approach is summarised in Figure 2. An established method in virtual architecture is to start with the development of an ontology of the virtual place. The ontology usually operates with terms that are familiar from the architecture. For example, the ontology of a building

as a collection of rooms and their content is popular among the designers of virtual environments independently of the underlying technology [5, 14]. The ontology provides guidelines for the agreement on the semantics of the forms used in the workplace. An example of the application of virtual architecture to the design of virtual design workplaces is presented in [9]. While virtual architecture is concerned with the definition of functionality, it also pays significant attention to the semantics of form, geometry and the spatial layout.



**Figure 1. Design concepts in virtual architecture and requirements engineering**



**Figure 2. Virtual architecture approach**

Procedurally, virtual architecture begins with the analysis of the design brief, followed by a conceptual design, and then, by the detailed design of the workplace. In general, this is a top-down process, which in practice has a number of loops between the different stages.

### 1.2. Requirements engineering approach

The right-hand part of Figure 1 presents a more rigorously formalised approach towards the design of virtual workplaces based on the requirements engineering methodologies. This approach assumes that sufficient knowledge about the collaboration process is available to make it possible to model it. Processes can be classified as either deterministic or non-deterministic. In deterministic

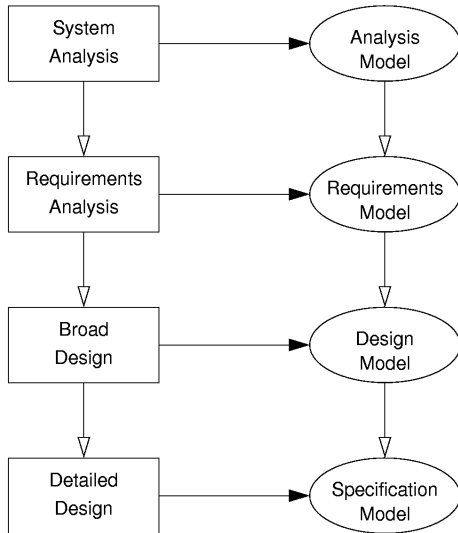
processes, the steps within the process are well defined, thus the process can be modeled with the workflow methodologies, and is referred to as a workflow process. In non-deterministic processes, not all steps can be planned ahead. While workflow processes have received much attention in the literature, and are supported by a number of modeling methods, few techniques exist for modeling partially planned or emergent collaboration processes. Such processes are common in knowledge-intensive activities such as product innovation or collaborative design, and usually follow only general process structures, with details of the process emerging during execution. Processes of this type are not well supported by workflow technology, which requires entire processes to be defined in advance, and then enacted according to this definition. Instead, such collaboration processes need a greater degree of flexibility. Environments that are based on the notion of collaboration spaces (a set of virtual workspaces), incorporating features of document management, inter-personal and group communication, notification, and a configurable governance structure provide a more adequate form of support [2].

Similar to virtual architecture, requirements engineering for the design of collaborative workplaces that support a particular process have to deal with the human factor. One approach is to base the modeling in this case on the soft systems methodology [11]. An instance of such a modeling methodology that addresses the requirements of collaboration processes, and that is tailored to the use of collaboration spaces, has been proposed in [8]. The methodology consists of four consecutive modeling steps (see Figure 3):

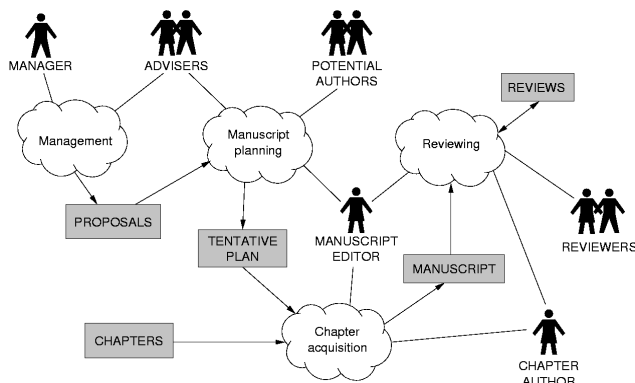
1. System analysis: develop an understanding of the current system, which is documented in an analysis model using a modified form of rich pictures, accompanied by so-called transition diagrams.
2. Requirements analysis: develop a requirements model, describing required changes to the existing system.
3. Broad design: prepare a design model, which describes the modified collaboration process, incorporating the requirements identified in the previous step, and which is again represented by a modified rich picture notation and transition diagrams.
4. Detailed design: produce a specification model which shows the detailed setup of collaboration spaces needed to support the new design, using a notation called MOO diagrams.

As an overall approach of conceptual modeling, this is an *activity-centered* approach. We illustrate the main modeling steps and notations by applying them to the formalisation of a manuscript preparation process. Figure 4 shows a rich picture that conveys high-level properties

of the process. The figure reveals the main *activities* (shown as clouds), the *roles*, or main *actors* (shown as stick figures) that are engaged in these activities, and the main *artefacts* (shown as boxes) that are used and produced by these activities. This amounts to the main features that may be known in advance about a partially planned process.



**Figure 3. Analysis and design method for virtual collaboration (adapted from [8])**



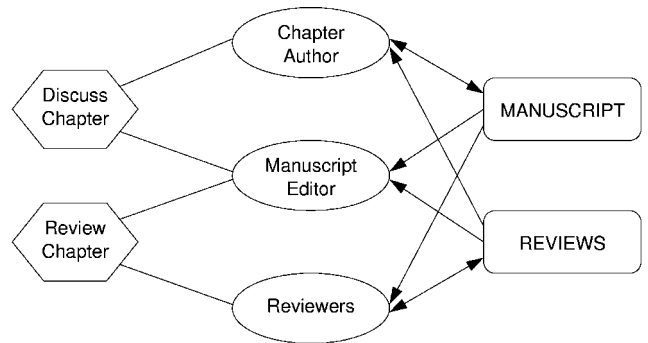
**Figure 4. Rich picture of a manuscript preparation process**



**Figure 5. Transition diagram of a manuscript preparation process**

The transition diagram in Figure 5 shows the sequence in which the activities of this process are carried out. It can be seen that two of the activities, Chapter acquisition and Reviewing, may be performed iteratively.

For each activity in the rich picture, a separate MOO diagram shows details of required support from a collaboration space. The idea is illustrated in Figure 6, the MOO diagram for the Reviewing activity. It shows which roles (ovals) have which kind of access (directionality of arrows) to which artefacts (boxes with rounded corners), and which discussion forums (hexagons) they are assigned to. The example illustrates the double blind review process where communications between authors and reviewers are mediated through a separate entity, here the manuscript editor.

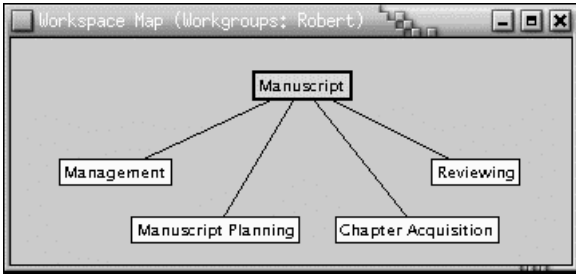


**Figure 6. MOO diagram of the reviewing activity**

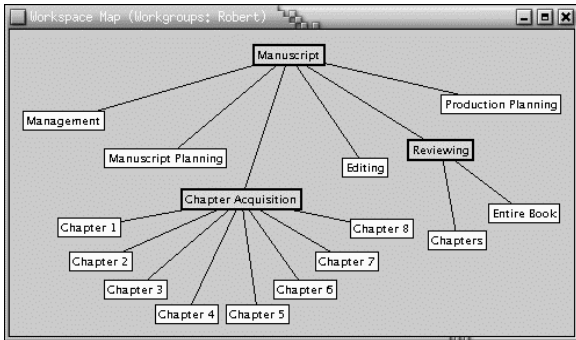
For each MOO diagram, a collaboration space with the corresponding features needs to be created. This is where the requirements engineering top-down approach ends. The example in Figure 7a illustrates this idea—the formalised process is the actual design.

However, the initial configuration of collaboration spaces and their features are meant to represent only a general structure, a starting point for the collaboration process. During the process, the initial configuration will be modified and tailored by the collaborators according to the evolving needs of the collaboration. The example in Figure 7b illustrates this idea—it constitutes an evolution of the original process and thus contains more workspaces than were initially identified and created.

Can we predict some elements of the evolution of a new collaborative process, based on similarities and analogies with processes formalised and supported before? Can we capture and utilise in the workspace design process the evolutionary component, so that we can provide better support to the developers of collaborative workspaces? The rest of the paper presents a new approach for supporting design and redesign of virtual environments, based on combining data mining techniques for refining the lower level models with a reverse engineering cycle to create upper level models. The approach allows comparison at all levels of the initial model of a process built based on the top-down requirements engineering approach with the model built from the actual workspaces by the proposed reverse engineering approach.



a. Network of collaborative workspaces at the end of a requirements engineering cycle



b. Network of collaborative workspaces that has evolved as a result of the evolution of the process

**Figure 7. Examples of resulting workspaces in the requirements engineering approach**

## 2. Principles of the approach towards reverse engineering of processes

Our approach is based on the following principles:

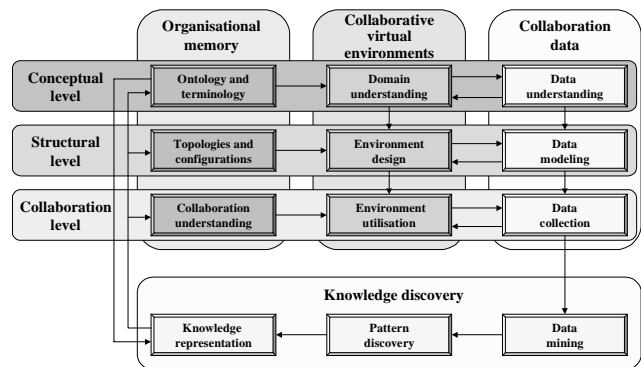
- The *availability of a schema* is a prerequisite for the development of effective reverse engineering methods in CVEs. If a well-defined methodology is used to design the virtual workplace (e.g. similar to the one presented in Section 1.2), then some kind of schema (ontology), which preserves the design semantics, is applied during the workplace design process.
- *Collaboration data*, collected during the evolution of the virtual workplace, is the source for the reverse engineering discoveries.
- *Data mining and knowledge discovery methods* applied to the collaboration data need to take into account the schema (ontology) of the design methodology used for the initial development of the workplace.

Reverse engineering of processes from collaborative virtual environments is possible through the analysis of collaboration data. Generally, such data is of two kinds: structural and behavioural data. Structural data captures static aspects of collaboration, such as the setup of a collaboration space. For example, structural data can

capture the variety of roles and artifacts in each workspace, and the links between the workspaces. Behavioural data captures dynamic aspects of collaboration, such as the actions performed by a virtual team in a collaboration space, and the dynamics of discussion threads and discussion content. The assumption is that such data reflects the types of activities supported in the environment, the corresponding topology of the collaboration space and the corresponding underlying technological representation.

When analysing such data, the ontology of the environment where it has been collected provides most of the semantic information needed for understanding and designing the data collection. A framework that embeds knowledge discovery in the design and use of CVEs has been presented in [4], and is shown in Figure 8. This framework suggests (1) how to obtain process knowledge from CVEs, and (2) how to feed discovered process knowledge back into the ongoing use of CVEs.

The framework consists of four inter-woven components: collaborative virtual environments, collaboration data, knowledge discovery and organizational memory. Moreover, the three components appearing in the upper part of the figure consist of three parts, at different levels of abstraction: conceptual, structural and collaboration levels. Collaborative virtual environments are seen as central in this framework. This is where processes supporting collaboration are designed and later enacted. This occurs in three steps: initially, understanding of the concepts of the domain to be supported is achieved, followed by the design of the collaboration environment according to requirements of the process to be supported (e.g. following the methodology discussed in Section 1.2), and lastly the utilization of the environment to carry out the process.



**Figure 8. Framework for extracting and feeding back process knowledge in collaborative virtual environments**

To make it possible to extract knowledge from the environment, a suitable set of collaboration data needs to be created. Thus the activities in the area of collaborative

virtual environments are paralleled by mutually complementary activities in the area of collaboration data: domain understanding feeds the understanding of aspects of the collaboration data needed, and when this understanding is built up and refined, it feeds back into the further development of domain understanding by bringing out facets of the concepts developed. Data understanding eventually leads to data modeling, where details of the data and their inter-relationships are established, and this activity likewise goes hand-in-hand with the corresponding activity of environment design. Finally, when the environment is utilized, data starts to be collected.

In the next component, knowledge discovery, this data is mined in order to obtain useful knowledge from it. In this framework there is a slight departure from the classical schema [6] in that the selection and data pre-processing stages are implicitly embedded in the data design. Therefore, collected data is expected to be ready for data mining. Data mining seeks to discover aspects of processes from the collaboration data. This leads to the discovery of patterns, which are subsequently represented in terms of the ontology of the CVE.

Finally, representations of discovered knowledge are fed into organizational memory, or more specifically, its *procedural memory* component (for a discussion of the role of both declarative and procedural memory in improvisational work processes, see [10]). Such knowledge may reside at different levels of abstraction. On the collaboration level, this knowledge relates to an understanding of the collaboration. For example, it can identify the main types of activities carried out by a virtual team in particular collaboration environments, how these activities were carried out over time, or what differences exist in the activity of different people. This knowledge can feed into the use of the environment itself, for example by adapting it to facilitate the execution of predominant process activities. On the structural level, representations of micro and macro aspects of process are maintained as topologies and process patterns. These patterns may later be consulted when setting out on similar work, and thus feed back into the design of future environments. The following section will discuss how such process patterns can be obtained through the analysis of collaboration data. Finally, use of collaborative virtual environments may, over time, also lead to the emergence of new concepts, or an application of existing concepts in ways that were not previously anticipated. These are deposited on the conceptual level as modifications to the underlying ontology, and feed into the ongoing development of a CVE.

### 3. Method for reverse engineering of processes

The reverse engineering method presented here aims to recover, or discover, the design of a collaborative process, and express it using the modeling notations introduced in the introduction, i.e. rich pictures, transition diagrams, and MOO diagrams. Rich pictures are used for representing entire processes, transition diagrams for showing task sequences, and MOO diagrams for showing individual task detail. The method proceeds in the reverse order of Hawryszkiewicz's methodology: first individual task models are obtained, then these are combined to a process model, and finally a model of task sequences is obtained, as illustrated in Figure 9.

#### 3.1. Task analysis

Individual collaboration spaces are seen as being equivalent to individual tasks (or *activities* in Hawryszkiewicz's terminology). Analysing a task aims to produce a task model, represented in the form of a MOO diagram. Depending on the CVE system in which the collaboration was carried out, this may be a straightforward mapping that can be fully automated, or it may require a manual process of identifying and mapping modeling elements. MOO diagrams contain mainly three modeling elements, namely roles, artefacts, and discussion forums, which may be related through certain defined types of relationships.

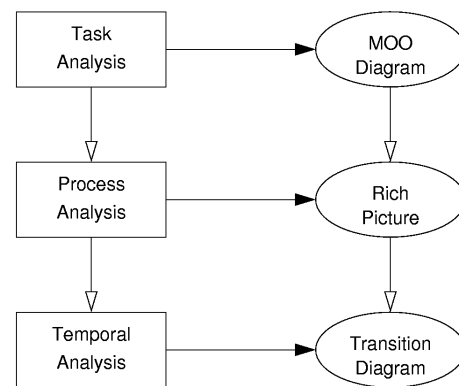


Figure 9. Method for reverse engineering of processes

#### 3.2. Process analysis

Once task models have been produced, relationships between tasks need to be analysed in order to discover which tasks belong to the same process. A number of methods are available to aid in this analysis.

One method is to analyse shared task elements, such as artefacts, discussion forums, roles, users, etc. The higher

the proportion of shared elements between a pair of collaboration spaces, the greater the likelihood that the tasks in the two spaces are related and are part of the same process.

Another method of analysis is to examine traversal patterns between collaboration spaces. This can reveal a network of spaces among which their users traverse back and forth. Such networks are a good indication of related tasks that are part of the same process.

A further method is to look for so-called “handover points”, where objects are passed from one collaboration space to another. Such handovers occur when an object, such as an artefact, is produced by one task as its output, and is received by another task as its input. A handover point usually is a good indicator that two tasks are part of the same process.

To produce the final set of tasks belonging to the same process, each of the above methods is applied to every pair of collaboration spaces, producing an *individual process predictor* value. Next, all of these values are summed together, to yield the *total process predictor* value. The tasks are considered to belong to the same process if their total process predictor value exceeds a given threshold, which is empirically defined. Pairs of tasks are linked together into a task network in such a way that each pair of connected nodes in the network is represented in the set of pairs of tasks remaining from the previous elimination step. The final output of this step is a process model, expressed as a rich picture.

### 3.3. Temporal analysis

Once a process model has been obtained, further analysis can be performed to derive a task sequence model. This analysis takes the temporal relationship of actions in different collaboration spaces into account. Actions that occur in different spaces can be related to each other in time in different ways. Looking at all the actions occurring in a collaboration space in their entirety, fundamentally there are only two temporal relationship types: either actions in one space precede actions in another space, or actions in two spaces occur in parallel. Usually a combination of these relationship types exists in a given pair of collaboration spaces, e.g. partially overlapping actions, interleaved actions, etc.

To determine task sequences, an analysis of temporal action relationships is performed on a pair of tasks taken from the process model. This analysis is based on *action levels*, which refers to the temporal clustering of actions in a given task, i.e. task intensity. For each collaboration space, action levels over the entire recorded history of the space are obtained, broken down per unit of time (e.g. day, week). Next, based on the observed distribution of action levels, a threshold is established above which activity in the collaboration space is considered to represent task

activity. Using this threshold, a temporal sequencing of actions in collaboration spaces, and thus of corresponding tasks, is now possible. It also makes it possible to identify parallel or interleaved tasks, where after the handover from one task to another, the previous task resumes activity. When this is followed by a switch back to the successor task, an iteration, or loop, is identified. Once all task sequences have been identified, a task sequence model can be produced, represented in the form of a transition diagram.

At the end of the reverse engineering cycle, a set of models is available which reflect certain essential process features, expressed in terms of the ontology of the CVE from which they were obtained. These can be deposited in an organizational memory as expressions of how collaboration has occurred, i.e. as procedural memory, complementing other information on the outcomes of the collaboration. Such process models thus become available for future retrieval and reuse, adding to the tool chest of the designers of collaboration environments.

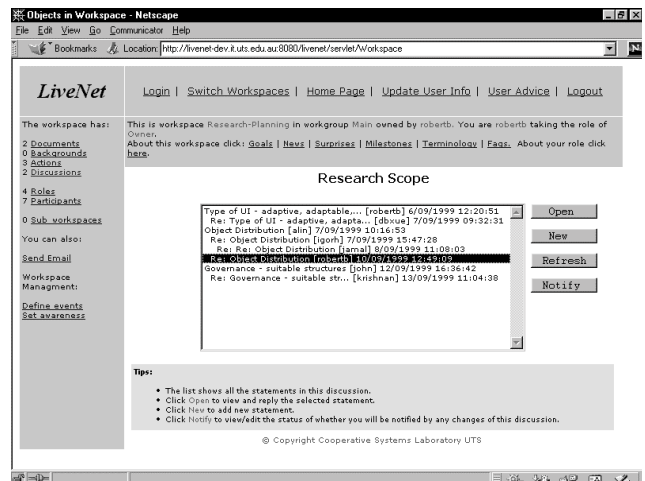


Figure 10. Web interface to LiveNet

## 4. Example of reverse engineering of processes in a CVE system

In this section we present an example of the application of our methodology for reverse engineering of processes. The CVE system in our example is LiveNet, a virtual collaboration system prototype developed at the University of Technology, Sydney [7]. It supports mainly asynchronous collaboration of distributed groups of people, i.e. different-time, different-place interactions, although its design does not limit it to only this mode of collaboration. A central server is accessed across the network through one of several client interfaces, most commonly through a Web interface (as illustrated in Figure 10). The environment is built around a particular



ontology, which is one of the premises for the application of our method.

#### 4.1. The ontology of the LiveNet CVE system

LiveNet provides virtual workspaces, which bring together people, artefacts (e.g. documents), communication channels, awareness facilities, and a collection of tools, all tied together through a configurable governance structure. A simplified ontology of the LiveNet CVE is shown in Figure 11. In terms of the ontology, workspaces contain roles, occupied by participants (i.e. actual people), who perform actions. Some actions may operate on document artefacts, others may be interactions with other workspace participants through discussions. Most workspace elements such as documents, discussions and participants may be shared between workspaces. Thus LiveNet workspaces are not just stand-alone entities like in some collaboration systems, but nodes in a network of inter-connected collaboration spaces. Neither are structures of workspaces in LiveNet static—once created, a workspace can be dynamically adapted to evolve together with the collaboration carried out in it, while likewise entire “ecologies” of inter-connected workspaces can co-evolve.

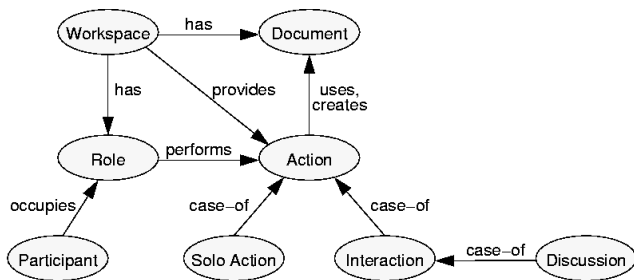


Figure 11. Simplified ontology of LiveNet

#### 4.2. Reverse engineering of processes in the LiveNet CVE system

The presented reverse engineering method for process extraction was applied to data collected from the LiveNet collaboration system. The data originated from 513 student and staff users at the University of Technology, Sydney who used LiveNet for a number of purposes. The collaboration data spans a three-month period in the second half of 2000, during which time a total of 721 workspaces were created. Reverse engineering focused on a set of workspaces that were set up by students learning to use collaboration technology, in this case to support a construction management task. The following describes one instance of reverse engineering of a process designed to support construction management.

**4.2.1. Task analysis.** Initially, information visualization aided the identification of potential candidates for reverse engineering. A specialized tool, the workspace visualizer, developed by us for the visualization of instances of workspaces, was used for this purpose [3]. An example of a so-called *inter-workspace map*, displaying relationships between workspaces, is shown in Figure 12.

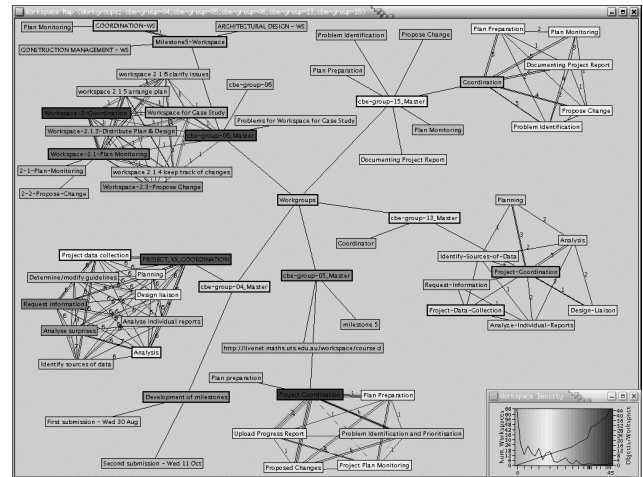
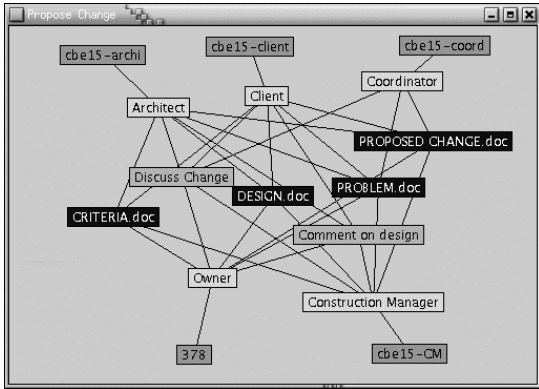


Figure 12. Map of LiveNet workspaces

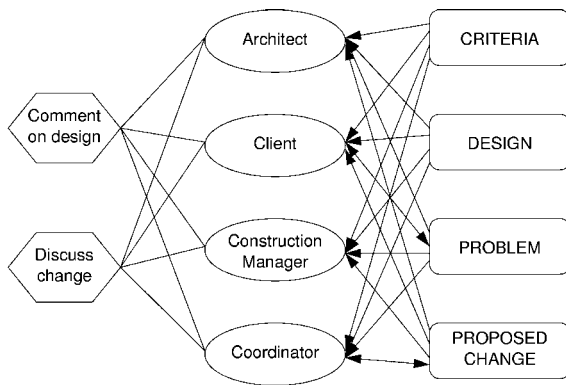
This map reveals a number of *clusters* of workspaces that appear to be closely related and could be part of the same work process. Later, process analysis will show whether this assumption can be supported. Firstly, task analysis is performed for all workspaces. To illustrate this, Figure 13a shows an *intra-workspace map* (also produced by our visualisation tool), displaying the relationships among the elements internal to a workspace, such as roles, documents, and discussion forums. Figure 13b shows the MOO diagram that has been derived from this intra-workspace map. Both figures show that almost all assignments of documents and discussion forums to roles in the workspace are identical. The only differences exist in the creation/modification of the Problem and Proposed Change documents (arrow pointing from the role to the document), which may only be performed by the Client and Coordinator roles, respectively. Coupled with the presence of the discussion forums for commenting on the design and discussing changes, this indicates a participatory work process: all roles may read all documents and join in the discussions, while changes to documents are coordinated by having only one role in charge of making such changes.

Given an ontology of the CVE and a set of rules, the task analysis and derivation of a MOO diagram can be performed automatically. The task analysis is performed for all workspaces under consideration. In the example of the inter-workspace map shown in Figure 12, this is done for 65 workspaces (out of the total of 721).





a. Intra-workspace map of the “Propose Change” workspace



b. Corresponding MOO diagram capturing essential aspects of the “Propose Change” workspace

**Figure 13. Task model derived from a workspace**

**4.2.2. Process analysis.** Following task analysis, process analysis attempts to discover which tasks (i.e. workspaces) are likely part of the same process. This begins by examining shared task elements, traversal patterns, and handover points, as discussed earlier. Table 1 shows an extract from the top of the list of all candidate pairs of workspaces under consideration, together with their individual and total process predictor values. The table shows, for example, that the two workspaces “Plan Preparation” and “Propose Change”, listed at the top, have 10 items in common, were involved in traversals from one

to the other workspace 3 times, and have 1 item that serves as a handover point, i.e. constitutes the outcome of one task and the input of the next, yielding a total process predictor value of 14.

Following the derivation of these process predictor values, those pairs of workspaces for which the value is below the defined threshold are eliminated from further consideration. In this case the threshold was set at 3, below which predictors were insignificant in predicting process membership. This left 13 pairs of workspaces, which next were linked together into a task network according to the established relationship. By adding shared roles and artefacts, this network was augmented to produce a process rich picture. Figure 14b shows the resulting rich picture, corresponding to the cluster of workspaces shown in Figure 14a. Both of these figures reveal the greatly inter-connected nature of the tasks in this process: most of the tasks (i.e. workspaces) share a majority of both artefacts and roles, and every task has some relationship to every other task. This is typical of collaborative and knowledge-intensive work processes, which have been described in the literature as resulting in “disconnected and parallel work that must nevertheless be guided to a common goal” [7].

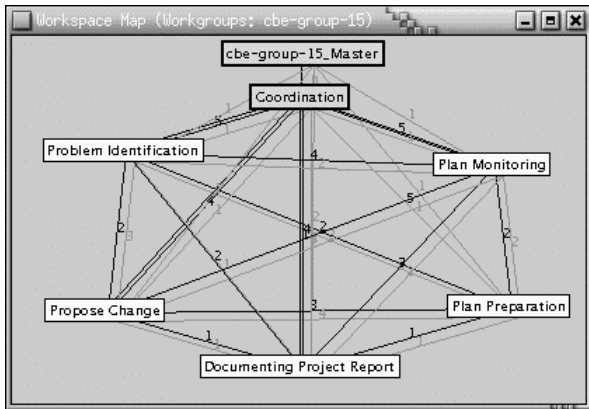
**4.2.3. Temporal analysis.** The final step of reverse engineering consists of performing temporal analysis on the actions in the workspaces of the derived process model in order to obtain a task sequence model. First, the history of actions in the workspaces is broken down into chunks, in this case at the level of days. Based on the distribution of action levels per day, which ranged from a minimum of 1 to a maximum of 120, with the majority of workspaces having action levels in the 10-20 range on most days, a threshold of 5 was set to distinguish tasks. Below this value, an excessive number of task switches resulted, often incurred only for such “tasks” as entering another workspace to look up a document or discussion item.

Temporal analysis then obtained sequences of task switches, which were consolidated into the task sequence model shown in Figure 15. The temporal analysis revealed that tasks in this process were tightly integrated: not only was work interleaved, with frequent switching between tasks, but also was it often parallel. Nonetheless, the transition diagram in Figure 15 does reveal definite

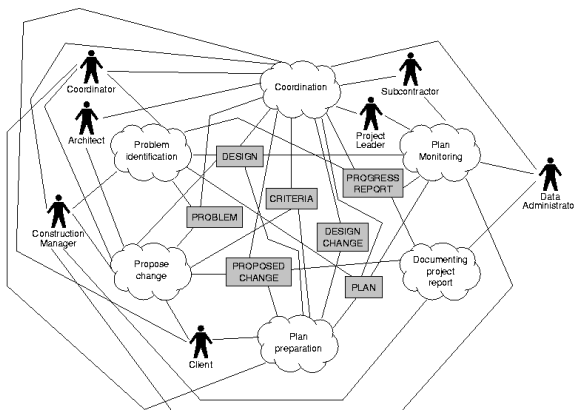
**Table 1. Process predictor values for candidate workspace pairs (extract)**

Workspace 1	Workspace 2	Shared Elements	Traversals	Handover Points	Total Process Predictor Value
Plan Preparation	Propose Change	10	3	1	14
Coordination	Plan Preparation	9	1	2	12
Plan Monitoring	Plan Preparation	5	7	0	12
Plan Monitoring	Problem Identification	7	4	1	12
⋮	⋮	⋮	⋮	⋮	⋮

patterns of task switching. For example, there is only uni-directional switching in five cases (such as from Plan monitoring to Problem identification), and bi-directional switching in four cases (such as between Plan monitoring and Coordination). Certain potential paths don't exist at all (for example, there is no switch between Plan preparation and Problem identification). This indicates to us that even in such relatively poorly structured processes—as compared to workflow processes—certain patterns of work emerge, which are reflected in the collected collaboration data and subsequently the derived process models.



a. Inter-workspace map showing a cluster of workspaces



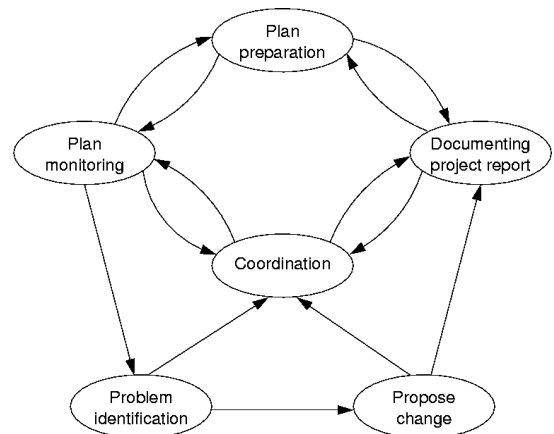
b. Rich picture of the corresponding tasks of the workspace cluster

**Figure 14. Process model derived from a collection of workspaces**

## 5. Conclusions

While business process (re-)engineering has received much attention in the past years, very little, if any, work has been performed on reverse engineering of work processes, particularly in the realm of virtual collaboration. This paper has presented a novel

methodology for reverse engineering of virtual work processes performed through collaborative virtual environments. It produces design models at micro (task) and macro (process) levels of these processes using notations from a (forward engineering) design methodology intended for virtual collaboration. Through the presented methodology, it not only becomes possible to trace the evolution of processes from an initial design, in the case where such a design has been performed, but also allows the discovery of ad-hoc and emergent processes for which no such initial design was prepared. In both cases, processes obtained through reverse engineering can be retained in a library of reusable process templates.



**Figure 15. Task sequence model derived from observed actions in a network of workspaces**

The presented methodology is independent of the underlying CVE system employed, and only requires knowledge of its schema, i.e. ontology. Only the concrete implementation of the data mining methods used needs to be adapted to the given CVE system so as to capture different CVE elements needed in the calculation of process predictors. Likewise, the interpretation of discovered patterns will need to be framed in the context of the collaboration system utilised.

When combined with the framework presented in Section 2, the proposed approach has the potential to influence the way CVEs are designed or redesigned. Insights obtained through the analysis of collaboration processes can, for example, reveal deficiencies in the levels of support provided by a particular CVE system implementation, leading to a redesign of a future version of the system. In this way, the approach can become the backbone of a new design methodology—design of CVEs by adaptation.

Finally, the illustrated combination of data mining and reverse engineering, and the availability of a rich source of data on actual collaborative practices, can lead to a better understanding of the influence of computer mediation on

collaborative processes. The development of CVEs that support the reverse engineering cycle, and the data mining of the internal structures of such environments, are areas for further research to focus on.

## Acknowledgments

The support by the Australian Research Council, the University of Technology, Sydney, and the University of Macau, which has made this research possible, is gratefully acknowledged.

## References

1. P. Anders, *Envisioning cyberspace: Designing 3-D electronic spaces*. New York: McGraw-Hill, 1999.
2. R. P. Biuk-Aghai, "Virtual Workspaces for Web-Based Emergent Processes," in *Fourth Pacific Asia Conference on Information Systems: Electronic Commerce and Web-Based Information Systems*. Hong Kong, China, 2000, pp. 864-880.
3. R. P. Biuk-Aghai, "Visualization of Web-Based Workspace Structures," in *Proceedings of the 1st International Conference on Web Information Systems Engineering*, vol. 1, Q. Li, Z. M. Ozsoyoglu, R. Wagner, Y. Kambayashi, and Y. Zhang, Eds. Hong Kong, China: IEEE Computer Society, 2000, pp. 302-309.
4. R. P. Biuk-Aghai and S. J. Simoff, "An Integrative Framework for Knowledge Extraction in Collaborative Virtual Environments," in *Proceedings of the ACM 2001 International Conference on Supporting Group Work*. Boulder, CO, USA: ACM Press, 2001.
5. T. L. Fanderclai, "MUDs in education: new environments, new pedagogies," *Computer Mediated Communication Magazine*, vol. 2, 1995.
6. U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: An overview," in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA, USA: AAAI Press/MIT Press, 1996.
7. I. T. Hawryszkiewicz, "Workspace Networks for Knowledge Sharing," in *Proceedings of AusWeb99, the Fifth Australian World Wide Web Conference*, R. Debrency and A. Ellis, Eds. Ballina, Australia, 1999, pp. 219-227.
8. I. T. Hawryszkiewicz, "Analysis for Cooperative Business Processes," in *Proceedings of the Fifth Australian Workshop on Requirements Engineering*, D. Zowghi, Ed. Brisbane, Australia, 2000, pp. 3-11.
9. M. L. Maher, S. Simoff, N. Gu, and K. H. Lau, "Two Approaches to a Virtual Design Office," *International Journal of Design Computing*, vol. 2, 2000.
10. C. Moorman and A. S. Miner, "Organizational Improvisation and Organizational Memory," *Academy of Management Review*, vol. 23, pp. 698-723, 1998.
11. D. Patching, *Practical Soft Systems Analysis*. London: Pitman, 1990.
12. D. Richards and S. J. Simoff, "Design ontology in context - A situated cognition approach to conceptual modeling," *AI in Engineering*, vol. 15, 2001.
13. S. Simoff and M. L. Maher, "Designing with the activity/space ontology," in *Artificial Intelligence in Design 98*, J. S. Gero and F. Sudweeks, Eds. Dordrecht: Kluwer Academic, 1998, pp. 23-44.
14. S. J. Simoff and M. L. Maher, "Loosely-integrated open virtual environments as places," *IEEE Learning Technology*, vol. 3, 2001.