Combining Multimedia Collaboration and Workflow Management

Michael Weber[1], Siegfried Höck[1], Gerhard Partsch[1],
Astrid Scheller-Huoy[2], Georg Schneider[3], Jean Schweitzer[2]
[1]University of Ulm, [2]Siemens Corporate Research,
[3]German Research Center for Artificial Intelligence

Abstract

Two different areas of distributed group work are supported by workflow management systems and real-time collaboration systems. Workflow management systems support work being structured in steps such that each step can be handled with the results of former steps and the expertise of the person working on that step. On the other hand, multimedia collaboration systems are best suited for unstructured group activities. Audiovisual connectivity and shared documents enable flexible group processes. All coordination tasks are left to the conference participants. This paper introduces an integration concept and prototype system which combines the advantages of both types of CSCW-systems.

1 Introduction and Motivation

In computer-supported cooperative work one approach being followed is business process engineering succeeded by workflow management to coordinate work processes. A second approach is to provide real-time multimedia collaboration tools in order to facilitate conferencing between geographically dispersed teams. The tasks being assisted by the two approaches diverge significantly. On the one hand a formalized sequence of separate activities is supported. The system coordinates the work of the different workflow participants who work in succession and independent of each other at a point in time. To work this way is also called asynchronous collaboration. On the other hand formally unstructured conferences are supported. All participants work jointly together at the same time on a task. This style of working is called synchronous collaboration.

An integrated telecooperation platform consisting of multimedia collaboration (MMC) tools and workflow management systems can support both problem classes at once. Embedding synchronous teamwork as part of the workflow means a change in the workflow paradigm being limited to „one person - one task - one application [1]. An activity can now incorporate as well more than one person as multiple applications. A smooth integration of workflow management and MMC system enables a continuous stream of tasks and activities in which fast, informal, ad-hoc actions can be taken through conferences within the usual formal workflow. This integration has to go beyond the simple provision of both systems on the same desktop [2]. The systems have to be coupled in order to interoperate between workflow and conferences.

2 Embedding MMC Conferences in Workflows

Since workflows have a longer duration than conferences, we propose to model conferences into workflows. We distinguish between process and conference activity. This formalism is used to characterize conferences with regard to their integration into workflow management systems. The term process activity has been defined by the Workflow Management Coalition [3] as follows:

„A process activity is a logical step or description of a piece of work that contributes towards the accomplishment of a process. A process activity may be a manual process activity and/or an

automated workflow process activity.“

In contrast to this we call an activity which takes place during an MMC conference a conference activity. Here several persons are working simultaneously. If conference and process activities have a 1:1 relationship consequentially the conference’s organization is structured because a workflow determines its proceeding. In this case the purpose of an MMC system is only to eliminate the „one person“ constraint. Considering a 1:n relationship a conference looks like a single process activity during which many conference activities will be accomplished. In this case the control of the process activity has to be passed to the MMC system. The inflexible coordination mechanisms of workflow management are replaced by flexible mechanisms for conferences.

Modeling Time. We distinct two instants in time when a conference is being modeled. Conferences that are already planned at the time of workflow development are called pre-scheduled conferences. In contrast to these ad-hoc conferences are not foreseeable at the time when the workflow model is specified. The initiation of such a conference depends on the current situation.

Conference Coordination. In a structured conference all conference activities are fixed. Conference and process activities have a 1:1 relationship. The conference activities are fully controllable by the workflow system. In unstructured conferences the control of task execution is passed to the MMC system. The entire conference is regarded as one process activity comprising several conference activities (1:n relationship). The workflow management system cannot influence the course of the conference. It only knows that there is a conference.

The combinations of modeling time and coordination lead to conference profiles.

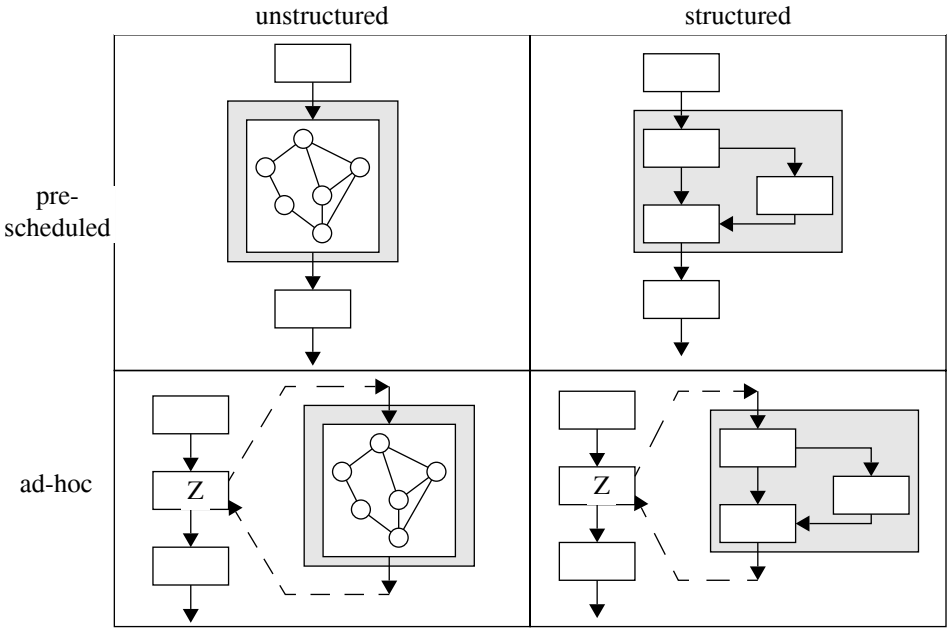

fig. 1: Conference Profiles

Unstructured Pre-scheduled Conference. This conference as such is already part of the workflow at modeling time. The conference activities however will be fixed before or even during the conference (fig. 1, upper left corner).

Structured Pre-scheduled Conference. Structured pre-scheduled conferences are modeled as part of the workflow (fig. 1, upper right corner). They are workflows themselves. The workflow management system only uses an MMC system as a special resource to support this subworkflow. The course of the conference is coordinated and controlled by the workflow management system.

Unstructured Ad-hoc Conference. Ad-hoc conferences are not foreseen and therefore cannot

be part of the workflow from its beginning (fig. 1, lower left). In such cases a conference will be invoked while the process activity „Z“ is suspended. After finishing all conference activities the workflow will resume with activity „Z“.

Structured Ad-hoc Conference. Structured ad-hoc conferences are also not part of the workflow at modeling time. However, they describe a standard situation. When a conference of this type is invoked the participants have to follow a specific procedure (lower right corner, fig. 1).

2.1 Integration concept

The described conference profiles define the requirements for six integration phases.

Phase 1. In Phase 1 we have to look at the interface parameters required to start up conferences. Inspecting various MMC systems [4, 5, 6] revealed similar interfaces to operate such systems. The conference parameters mainly contain data which is necessary to establish a conference: title, participants, chairperson, topic, conference attachments, date, agenda.

Phase 2. The parameters are used to establish the conference. The system invites the participants and shares the required applications and documents. After this the participants can jointly work on the conference activities.

Phase 3. Conference controlling depends on whether the conference type is structured or unstructured. Structured conferences are controlled by the workflow. If a conference activity is completed the next conference activity will be sequenced automatically. Unstructured conferences are controlled by the participants. A conference assistant supports such conferences.

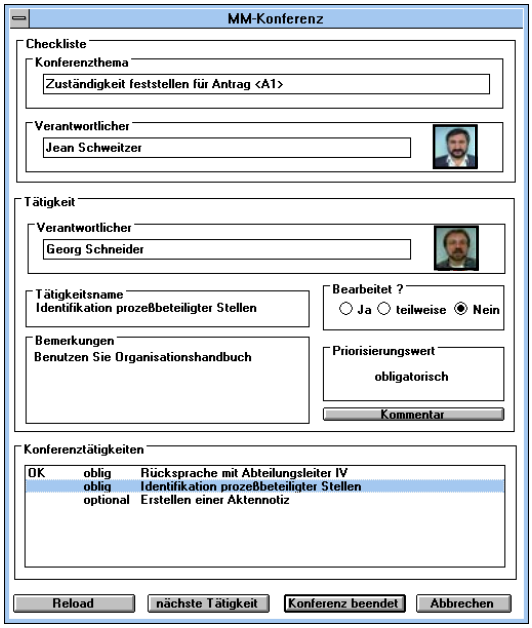

fig. 2: A Conference Assistant's Checklist

Phase 4. Controlling the progress of conferences is essential for the integration of conference results into a running workflow. This requires monitoring the conference to record performed activities. Before a conference is closed this record is evaluated to deduce the conference results.

Phase 5. When the last conference activity is finished the workflow will terminate the conference for structured conferences. Unstructured conferences are terminated by the participants.

Phase 6. Choosing the next task after a conference depends on its result. If all conference activities are completed a decision can be taken. For structured conferences this can be done by the workflow management system. For unstructured conferences the results must be delivered to the

workflow management system in an adequate way. For this we also use the conference assistant.

2.2 Conference Assistant

The conference assistant consists of a checklist and an evaluating mechanism. The checklist is an additional document to be shared during a conference. In structured conferences the checklist only serves as an information source to the conference participants, since all conference activities are process activities. Unstructured conferences need the checklist (figure 2) for conference coordination and monitoring. It describes the conference activities like a to-do list. Furthermore the checklist is a tool to record the conference results. To specify a result we distinguish three completion states of a conference activity: „done“, „partially done“, and „to do“.

Furthermore priorities are assigned to conference activities to support the decision on the conference result. The priorities are: „optional“, „deferrable“, and „mandatory“. Deferrable activities have to be worked on but not completed. They can be resumed in a future conference or as a process activity in the workflow. The final conference result is derived from the different activity completion states. We define three possible results. „Successful“: All deferrable and all mandatory conference activities are marked with „done“ in the checklist. „Partially Successful“: All mandatory activities are marked „done“ and at least one deferrable activity is marked „partially done“. „Not Successful“: None of the activities is marked „done“.

3 Prototype System

In our prototype system neither a workflow management system nor a MMC system are modified. A separate conference broker serves as a mediator between the two systems. The broker takes conference descriptions from the workflow management system and conveys the parameters to the MMC system (figure 3). The conference descriptions are assembled in an incremental way. If conferences follow each other, the broker extracts the difference of the two and forwards only changes in the parameters to the MMC system. We call this method Δ-conferencing.

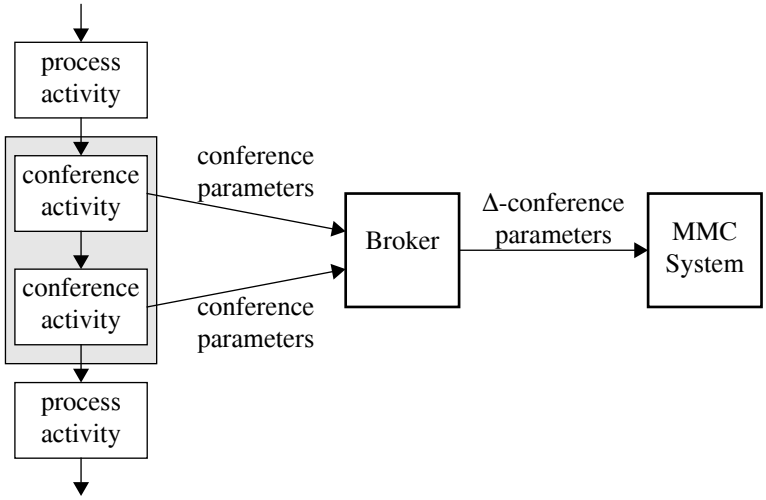

fig. 3: Conference Broker

Since the broker initiates conferences, it also has to be notified about conference states. Thus, the broker takes active part in conferences as a virtual participant. This virtual participant is visible to the human participants to keep them aware of being in a workflow embedded conference.

We use the knowledge query and manipulation language KQML [7] being developed for agent

communication to transfer the conference parameters. The workflow management system subscribes at the broker to the newest information on the set of conference participants and on the conference results. If intermediate conference results arrive the workflow management system can monitor the course of the conference and might interact when structured conferences are performed. The following KQML performatives show how conference information is „told“ and how the workflow management system subscribes to actual information.

```
(tell  :content ( (title              test-conf)
                  (participants       (Michael Siggi Gerhard) )
                  (chair              Michael)
                  (agenda             test.doc)
                  (conference-system  GroupX)
                  (media              (sharing audio video) ) )
       :language     WOTEL-Language
       :ontology     WOTEL
       :reply-with   id1
       :sender       WFMS
       :receiver     WFMS_Module )

(subscribe   :content      (participants chair media conference-result)
             :language     WOTEL-Language
             :ontology     WOTEL
             :reply-with   id1
             :sender       WFMS
             :receiver     WFMS_Module )
```

With the above information the broker is able to initiate a conference using its conference module. Despite the separation and distribution of broker modules, we still have a fixed coupling of a workflow management system and an MMC system. However, since many new MMC systems appear on the market, it should be possible to have a dynamic choice of which system to use during run time. The trading concepts as defined in [8, 9, 10] promise to fulfill these requirements.

3.1 Trading of MMC Systems

A trader comprises an exporter part representing the service and an importer part requesting a service. The trader is able to perform a late selection and binding of the respective interfaces. We include importer functionality into the workflow module and exporter functionality into the conference module of the broker (figure 4). The trader itself is implemented as a distributed service with peer traders distributed over the internet at participant domains.

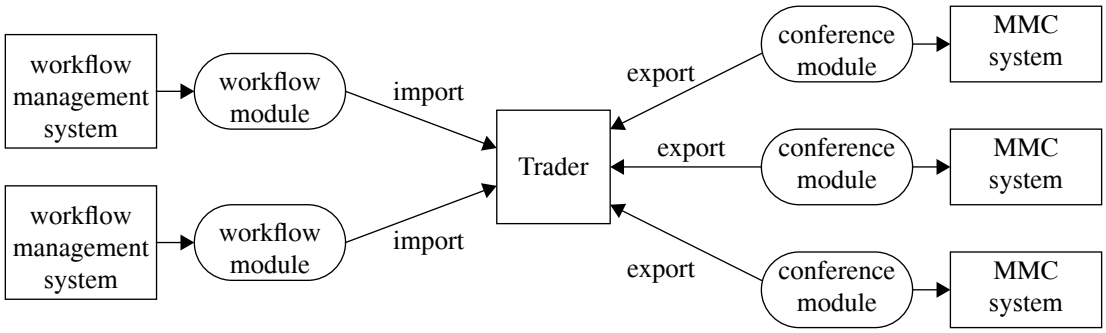

fig. 4: Trader and Broker Modules

The workflow module imports the service „conference“ after the workflow management system passed over KQML performatives. The workflow module then contacts the trader core to receive binding information on the desired conference module. This method of flexible matching of MMC systems has several advantages. First, it is possible to see by the availability of a service offer whether and which MMC system of a person is running. Second, it is possible to specify a person by its role and the matching of the trader resolves the role to a real person. Third, it is possible to choose the most suited MMC system for the current set of participants.

The conference module acts as an exporter. It knows its MMC system type and the system's parameters. Starting the MMC system the conference module will be initialized and the service offer will be exported to the trader. The conference module also measures the actual bandwidth and returns the currently possible quality of service. When the MMC system is shutdown the conference module withdraws its service offer.

4 Conclusion and Future Work

An approach to integrating workflow management systems with real-time conference support has been introduced. A set of conference profiles has been used to incorporate synchronous collaboration into asynchronous workflows. Coordination of conferences is supported through checklists provided by a conference assistant. A broker and trader service is used to mediate between workflow and MMC tools. Our current prototype has been successfully evaluated using real life scenarios taken from a civil construction authority.

In the future we want to support MMC conferences further by using more knowledge available from the workflow model and from workflow execution. The conference preparation is a crucial point for the success of a MMC conference [11]. For this reason we require at least an agenda for all conferences profiles. We will also include short abstracts about the work to do, so the participants will join the conference well prepared. Another topic is resource planning for MMC conferences. Here resources are human resources and time. A schedule must be compiled to fix the time of the conference to assure that all participants are available. Furthermore there will be an integration of MMC systems also into heterogeneous workflow environments where an interface between the workflow management systems has to transport not only workflow relevant data but conference relevant data too. Finally a distributed version will be evaluated to draw conclusions on scalability, performance and failure resilience.

5 Literature

1. H. Stark, L. Lachal: Ovum Evaluates Workflow. Ovum Ltd, London, 1995.
2. K. R. Abbott, S. K. Sarin: Experiences with Workflow Management: Issues for the Next Generation, Conference on Computer-supported Cooperative Work, ACM Press, New York, 1994.
3. Workflow Management Coalition: Reference Model, http://www.aiai.ed.ac.uk/WfMC/, 1994.
4. S. Cronjäger, W. Reinhard, J. Schweitzer: Functional Components for Multimedia Services, International Conference on Communications, 1993.
5. G. Dermler, T. Gutekunst, B. Plattner, E. Ostrowski, F. Ruge, M. Weber: Constructing a Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogeneous Workstation Environments, 4th Workshop on Future Trends in Distributed Computing Systems, 1993.
6. Intel: Intel Proshare Developers Kit, 1995.
7. T. Finin, J. Weber, et al.: Specification of the KQML Agent-Communication Language. DARPA Knowledge Sharing Initiative, 1993.
8. M. Bearman: Trading in Open Distributed Environments. Int. Conference on Open Distributed Processing, 1995.
9. J.-P. Deschreval: The ANSA Model for Trading and Federation. Architecture Report of ANSA Phase III, APM.1005.01, 1993.
10. ODP Trading Function. Final Draft - ISO/IEC DIS 13235, ITU-T Rec.9tr, ISO/IEC JTC1/SC21, 1995.
11. V. Barent, H. Krcmar, H. Lewe, G. Schwabe: Improving Continuous Improvement with CATeam: Lessons from a longitudinal case study, 28th Hawaii International Conference on System Sciences, 1995.