Coordination Support for Scalable Cooperative Work

Henrique João L. Domingos, J. Legatheaux Martins, Nuno M. Preguiça, Jorge F. Simão

{hj,jalm,nmp,jsimao}@di.fct.unl.pt

Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

This paper will appear in the proceedings of the DEXA 98 Workshop, Vienna, Austria, August 1998.

Abstract

This paper discusses the coordination and awareness control supports in the context of large-scale collaborative work. It presents the foundations and design principles of an integrated, generic and flexible distributed system architecture as a support platform for the requirements of scalable groupware. The platform is structured in three fundamental layers: the base communication services, the cooperation support and the coordination facilities. The paper is more focused on the characteristics and implementation issues of the coordination layer. This layer is composed by a set of components and system services, reused by different groupware applications as base building blocks.

1. Introduction

Designing and implementing a generic platform for Computer Supported Cooperative Work (CSCW), based on large scale distributed system supports, is our fundamental work direction. We believe that in the long run this approach has interesting advantages to build flexible support infrastructures for large scale cooperative computing. Requirements imposed by the "*large-scale factor*" emphasize and justify this perspective in order to provide: *reliable group-oriented communication facilities*; *fault-tolerance characteristics*; *dynamic reconfiguration facilities for collaborative and shared workspaces; tailorability* and *adaptability properties;* and *high* availability criteria for applications and services.

The research on new solutions for those requirements is an interesting trend, considering some limitations of today's CSCW technology. Most of current groupware applications are largely incompatible with one another, implementing limited and specific interoperability, resource-sharing and coordination control criteria from scratch. This causes important limitations in terms of large-scale settings.

In this paper we will describe the generic support issues involved in the materialization of a distributed system platform called DAgora. DAgora is an object-oriented groupware platform materializing a generic framework for the requirements of different collaborative applications running in large-scale settings[1].

2. Building an integrated platform for large scale CSCW

Three fundamental layers compose the platform: the communication layer, the cooperation layer and the coordination layer.

The communication layer implements a set of base services and reliable group-oriented communication protocols. These protocols implement a flexible and complementary support for different modes of collaborative work interactions, namely: synchronous, asynchronous and multi-synchronous

The cooperation layer is basically a distributed system infrastructure for object replication with scalable characteristics. This infrastructure materializes the basis to support the notion of group-oriented collaborative workspaces. It provides a complementary support for peer-synchronous models of consistent and active object replication (useful for synchronous groupware requirements) and lazily replicated objects with persistent characteristics (useful for the requirements of high-

[1] More references about the project are available: http://dagora.di.fct.unl.pt

available asynchronous groupware). To implement the cooperation support, we adopt the base communication services. Peer-synchronous replication models adopt a stackable suite of reliable multicasting protocols (with flexible semantics and reliability warranties). Lazy object replication is provided by an object-storage system used as a large-scale replicated data repository for asynchronous cooperative-oriented applications.

The coordination layer implements a set of coordination components and services that can be reused in different contexts of specific groupware applications. These components are used as base abstractions to manage dependencies and to detect and solve conflicts during collaborative work processes. In the rest of the paper we will focus on the specific foundations and design issues involved in the coordination support.

The coordination perspective of the platform

We will use the term coordination as introduced by Malone & Crostown [Malone90], [Malone94], [Holt88]: the act of managing interdependencies between activities. In our case this definition relies with computer supported collaborative work activities. These activities are workgroup processes modeled and materialized by different specific groupware applications. In this perspective, the activities are collaborative tasks developed in a certain period of time by a group of users to achieve a common goal. The coordination support must ease the achievement of that goal, warranting adequate levels of effectiveness and productivity and maintaining the cooperation principles of different workgroup tasks [Wino 86].

At computational level the coordination support is related with different issues and support mechanisms to manage the occurrence of potential interdependencies and conflicts, when users interact with different groupware applications in the shared workspace. This includes things like:

- Concurrency and consistency control mechanisms for object-sharing;
- work facilitation mechanisms like dynamic resource allocation as well as planning, scheduling and announcing of events and work sessions;
- awareness control facilities based on the notification of potential events occurring in the shared workspace;
- collaborative workgroups management and users' participation control;
- configuration of users' roles and definition of access control rights related with those roles.

In collaborative workgroup sessions, the coordination issues related with the above supports are motivated by dynamic facts and work circumstances, not necessarily known in the beginning of the collaboration process. This relates with a set of specific properties of coordination, such as:

- Coordination activities are highly event-driven by nature. This happens because in despite that in certain moments the collaboration goal can be well devised, the workgroup methodology warranting the best productivity can be possibly unclear or unknown. This causes the need of unpredictable informal interactions among all the participants, that cooperate in the coordination process itself.

- During a complex collaborative session, different sub-tasks (with different interaction modes) need to be scheduled in a dynamic way, adapting the work circumstances to adequate methodologies or to the manifestations of possible interdependencies and potential conflicts. Thus, the coordination support must be adaptable and based on dynamically tailorable components and services. This characteristic is essentially different than the case of classic workflow systems for example.

- The coordination perspective of collaborative processes must be centered in the facilitation of user's interoperability, not restricting (unnecessary) their roles, not restricting specific methodologies but on contrary: promoting common backgrounds and motivating individual contributions [Dourish 92].

- In collaborative processes, the awareness control mechanisms represent a fundamental coordination context of understanding different group activities as a way to provide self-coordination contexts in each individual task. At the same time this ensures that the individual contributions can be more relevant to the group's activities as a whole. Furthermore awareness control information warrants implicit evaluations of the relevance of each individual contribution with respect to the common goals and work progress.

Workspace organization

Fig. 1 represents the workspace organization and topology for large scale cooperation in the case of our platform (DÁgora system) [Dom 97],[Simão 97]. We use this topology as the architectural model implemented by the DÁgora system.

In this workspace topology we consider different domains of cooperation and coordination:

LLC - Low Latency Clusters - are typically cooperation domains supported by LANs (Local Area Networks) or MANs (Metropolitan Area Networks) or internetworking infrastructures based on LAN-TO-LAN connectivity solutions. In each LLC a set of workstations is used by different users to interact, possibly using the "same time/different place" paradigm. An LLC offers a realistic environment for a cooperative workspace to support synchronous "soft-real-time" interactions among strongly-coupled workgroups. The fundamental criteria to define an LLC is only its potential responsiveness characteristic, low latency and good quality of the communication service infrastructure. We consider 1 sec. as the maximum round-trip-time involved between two participants in an ideal LLC.

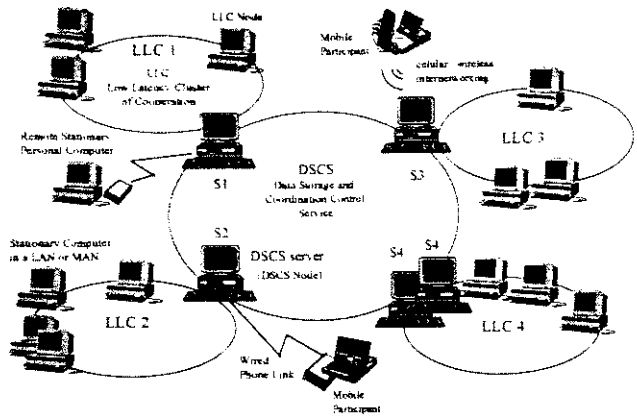

Fig.1 Large scale workspace topology and organization

The **DSCS** - **D**ata **S**torage and **C**oordination **S**ervice – implements the replicated object-storage service rekated with the cooperation layer. This support is provided by a distributed group of replicated servers (ex: S1, S2, S3, S4 or S4') managing a persistent global data-repository service acting in each case as the group collaborative (persistent) memory. This service also stores special coordination objects per session and the coordination and management information configured for different collaborative work sessions under development. Each server has also support for asynchronous user's interactions. It also provides the support for dynamic (re)integration of mobile and partially connected collaborative work. Each DSCS server also provides administration and configuration tools and a session-manager. The DSCS infrastructure offers high availability and is the essential support for dynamic reconfigurability and scalability criteria for large scale loosely-coupled cooperation

In the shared workspace, mobile users can work asynchronously and disconnected. In the most part of the time, they develop disconnected work. Periodically they submit their contributions to the DSCS server. Each DSCS server provides the necessary support for asynchronous-group dissemination. With this support, even disconnected, each mobile user can belong virtually to a well-coordinated group and session.

The DSCS is an high-available and flexible infrastructure providing inter-group communication and participant-to-group communication (communication between each participant and the other group members). With this infrastructure, users can interact asynchronously storing individual contributions in a persistent way. Objects stored by the DSCS servers are replicated in a weak consistency way using optimistic replication protocols based on epidemic algorithms, implementing a lazy replication strategy [Preguiça 98].

3. Implementing the coordination support

There are two main issues directly related with the coordination support: (1) the definition of a collaborative-session model, and (2) the components and services implementing the base system coordination support.

3.1 The model for collaborative sessions

Logically, a *session* embraces all tasks, procedures, data and associated users, applications and dependency relations established in the process of reaching a common goal - the session's purpose (or the session collaborative goal). Coordination among users involved in a session is used to guarantee that each participant's efforts are positive contributions to the common goal. Different types of sessions (for instance, with different objectives and different number of participants) require different coordination degrees. Moreover, in the same session, different subtasks also need different types of coordination. Consequently, flexibility and tailorability of coordination control is required. There are *open sessions* and *closed sessions* depending when the participation is free (users just binds to the session without previous authentication) or when requires the previous registration of participants (users must authenticate when join to the session).

The session represents a *coordination unity* related with a collaboration domain in which all the activities inherent to each workgroup collaborative process take place.

In the framework, several properties are associated with this notion of a session: *general information* - comprising a set of attributes such as the session title, description of the session goal and associated notification services; a

workgroup of participants - including a registry service with individual users information (as names and roles in session's context if there are roles assigned); *tasks and interdependencies* - including the information and bindings related with the tasks that will be performed (dynamically) in the context of the session.

Session's tasks have two different types: *intermediate and final*. Intermediate tasks are defined through the definition of a new *collaborative sub-session* (the session is therefore a recursive concept). Final tasks clearly define their goals and procedures to achieve them. There are *synchronized and non-synchronized* tasks. Associated with each synchronized task it is defined (besides general task description) a set of data: its participants and associated roles if assigned (correspondent to minimal and maximal subsets of sessions participants); scheduling time; applications to be used in each task (if applicable); a reference to data produced (filled in the end of the task, by its coordinator or by some participant). Associated with each non-synchronized task it is defined a deadline for its execution and to obtain the respective data, participants and associated permissions (for manipulating the data).

The *collaborative session manager application* - CSm - controls the coordination information associated with a *collaborative session*, including all tasks executed in its context. Each participant may *jump* to some task through simple interface mechanisms provided by the session manager - applications associated with the task are started and the binding process is automatically executed (if any - used primarily in synchronoized tasks).
Although the same concepts are apparently used in both WFm (workflow management programs) and CSm (collaborative session management programs), fundamental differences exist. A workflow is used as a systematic description of well known (possible repetitive) processes to reach some goal, while the CSm is used to create and evolve dynamically and collaboratively possible unknown processes to reach the session goal.
For the above reason, flexibility and tailorability are key-properties in the context of the CSm. Not only new tasks can be created within the session, but also new types of tasks. For instance, new synchronized tasks may be defined with new different associated tools. Moreover, sessions coordinators must have the ability to modify dynamically some previously task, define it as concluded, or trace dependencies among them. These activities can be done starting from previous negotiation processes, established by previous tasks.
Although, the process to reach the session goal can be unknown "a priori", some fundamental tasks may have been identified and reused later in the context of the current or new collaborative sessions. Thus, templates for sessions exist, allowing the definition of some initial tasks. However, the overall process is unknown, and must be defined as a result of session interaction - therefore, we are not in presence of a traditional workflow. This ability to evolve session templates for collaborations done with success, reusable in the context of future activities seems to be a fundamental aspect of capturing the memory of collaborative methodologies, contributing for the widely acceptance of CSCW systems in the perspective of organizational structures.

Summarizing, the tailorability support at coordination level relates with the anticipation of coordination behaviors and the functionality aggregated in the CSm. In fact, the CSm acts as a tailoring tool to compose coordination components: the combination (binding facilities) for simple tools (Java classes implementing collaborative tasks) or complex multi-synchronous collaboration-aware applications; facilities to dynamically change the session customizations, dynamic configurations of existent tools and applications; and ways to control session-awareness notifications. All these facilities are provided as end-user level adaptations in the perspective of individual users, workgroups and organizational coordination. The system also provides facilities for reusing coordination contexts applicable to other future similar activities.

3.2 Session coordination components and services

The session services implementing the coordination support level are: the session binding service, the user's registration service, the tasks scheduler and its multi-tool binding components and the session notification service.

All these services are accessible from the *CSm* environment conceived an application supported in base components interacting with all the above services.

We will describe briefly each one of those services explaining in more detail the implementation of the awareness support provided by the session notification service.

Session binding. The session binding service is responsible by providing browsing and binding facilities to all the sessions supported in each moment in the data repository. Each session has an unique name (similar to an usual web URL address, ex: http://dagora.di.fct.unl.pt/SessionID).

The sessionID is an unique identifier. Each session maps on a volume as explained above when we described the data-storage component. Sessions can be registered in a session name service, with the set of coordination attributes. A user binds to a session just by downloading initially a CSm which provides the adequate functions to access to the coordination context of the session (coordination attributes related with the session model explained above and facilities for all the other session services).

User's registration service. In the case of closed sessions, when the session is created and configured, the users that will participate in the tasks of the session must be registered. This is done by means of the user's registration service. To each user is assigned a session role: coordinator, participant or observer. These roles are used by the session manager to provide access control rights in the context of the session. Coordinators can modify the coordination attributes in the context of a session (modifying for instance the roles assigned to the users) and can also schedule new tasks choosing the adequate tools (using the scheduler component of the CSm). Participants act as users without privileges to modify the coordination context but they can use the different tools related with the tasks scheduled and announced in the session manager. With these tools they collaborate to achieve the different goals subjacent to the scheduled tasks. When using the different tools they can handle the different object-types managed in the application-specific context. Finally, observers only use the session manager to bind to tasks results (they cannot bind to the applications concerned with those different tasks). A result is a static information accessible by a link managed in the context of the session manager and represents a snapshot or annotation of a coordinator user (reporting the state or the result of a previous task).

Tasks scheduling and multi-tool binding. This service provides the way to schedule and announce new tasks in the context of a session. Each announcement has all the necessary binding information to users automatically bind to each task. When a task is created (by an user acting as coordinator) it is chosen the tool that will be used to work cooperatively to develop that task.

Awareness support. The awareness support is provided by a service called Session Events Notification Service. This service provides a generic way to disseminate and notify events occurring in the context of all the tasks developed within a session. We explain the architecture and functionality inherent to this service in the next section.

4. Conclusions

In the paper we analyze the coordination support for collaborative sessions. The main motivation for this analysis is based on the background and experience in materializing a generic, flexible and integrated CSCW platform and framework for the requirements of groupware running in large scale settings.

In such a platform, the main support components are structured as an extensible middleware architecture providing base abstractions and support services at different levels, namely:

- a communication providing group-oriented communication protocols and services;
- a collaboration support implementing a large scale object replication infrastructure as the basis for the materialization of collaborative workspaces;
- the coordination support which is composed by a set of components and services organized in the context of the conceptualization of a collaborative session. A session is basically a coordination unit in which a workgroup can develop different tasks modeled by specific groupware applications materializing flexible models of users' interactions: synchronous, asynchronous and multi-synchronous.

Finally we describe the main components involved in the implementation of the coordination layer: session binding services, user's registration services, tasks scheduling and session awareness support.

Bibliography

[Dom 97] Henrique J. Domingos, J. Legatheaux Martins, Nuno M. Preguiça and Jorge F.Simão, "Support for Coordination and Flexible Sunchronicity in Large Scale CSCW" , in Proc. of the CRIWG'97 - 3rd CYTED RITOS International Workshop on Groupware, pp 81-90, October 1997

[Dourish 92] Paul Dourish, Victoria Belloti, "Awareness and Coordination in Shared Workspaces", in Proc. of the ACM CSCW 92, pp 107-114, November 1992

[Holt 88] A. Holt, "Diplans: A New Langiage for the Study and Implementation of Coordination", ACM Transactions on Office Information Systems, n. 6, vol.2, pp 109-125, 1988

[Malone 90] T. W. Malone, K. Crostown, "What is Coordination Theory and How Can It Helps Design Coooperative Systems", ACM CSCW

90, in Readings in "Groupware and Computer-Supported Cooperative Work". Writen and Edited by Ronald Baecker, pp. 375-388, Morgan Kaufmann Publishers, 1993

[Malone 94] T. W. Malone, K. Crostown, "The interdisciplinary study of coordination", ACM Computing Surbeys, 26, pp 87-119, Mar 1994

[Preguiça 98] N. Preguiça, J. Legatheaux Martins, H. Domingos, J. Simão, "System Support for Large-Scale Collaborative Applications" Technical Report, TR-01-98 DI-FCT-UNL, Dep. of Computer Science, FCT - New University of Lisbon

[Santanu 97] Paul Santana, Edwin Park and Jarir Chaar, "RainMan: A Workflow System for the Internet", in Proc. of the USENIX Symposium on Internet Technologies and Systems, pp. 159-179, December 1997

[Sheth 97] Amit Shet and K. Kochut, "Workflow Applications to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration Systems", Technical Report.

[Simão 97] Jorge F. Simão, N. Preguiça, J. Legatheaux Martins and Henrique J. Domingos, "DÁgora: An Object-Oriented Groupware Platform", Workshop on Groupware Platforms, ECSCW 97,

[Wino 86] T. Winograd, F. Flores, "Understanding Computers and Cognition: A New Foundation for Design", Norwood, NJ Ablex, 1986