P2P InterCoop: P2P-based Decentralised Inter-organizational Cooperation - Motivations and Requirements

Hala Skaf-Molli

P2P InterCoop: P2P-based Decentralised Inter-organizational Cooperation - Motivations and Requirements

Hala Skaf-Molli

Abstract:
As the global marketplace becomes more and more competitive, corporations often need to cooperate in order to utilize the best of their resources for achieving their common business goals. This kind of collaboration gives a corporation a possibility to use the capabilities of their business partners that they do not currently have. In addition, mergers, acquisitions, alliances, and market demands are some of the reasons why companies have distributed work across geographically separated sites. Although P2P systems are emerging as a new form of distributed architecture, the centralized architecture of the existing collaborative environments has major drawbacks. This paper presents a new architecture called *P2P InterCoop* for supporting distributed cooperative work; it identifies basic requirements, advantages and limitations.

1 Introduction

As the global marketplace becomes more and more competitive, corporations often need to cooperate in order to utilize the best of their resources for achieving their common business goals. This kind of collaboration gives a corporation the possibility to use the capabilities of their business partners that they do not currently have. In addition, mergers, acquisitions, alliances, and market demands are some of the reasons why companies have distributed work across geographically separated sites[GHP99].

The existing collaborative environments, such as SourceForge [Sou06], BCSCW [BSC06, RWU$^+$97] or Zope/Plone [Zop06], provide most of functionalities or services needed for cooperative work [GMO$^+$04]. They provide *communication services* (i.e. forum, mail, notification) as well as content management services (i.e. publication, wiki, file management), *file sharing services* (i.e. configuration management, workspace management), and finally, *coordination services* (task management, trackers). One common point of these environments is their centralized architecture. This means that a single site manages all the required services. Consequently, cooperative work supported by such architectures suffers from several limitations.

- One major problem is service *availability*. Since the server is the single point of failure, if it is shutdown, it is not possible to continue to cooperate. Thus, service provision is not guaranteed. Actually, if SourceForge.net breaks down, the access to 80000 projects that it stores can be unavailable. In addition, the high cost of the central server and the overhead of its administration and maintenance are also important problems.

- Another important issue in a centralized management is the *scalability* and *performance*. As the server cannot have the same *performance* as the number of the partners increase, it becomes a bottleneck node.
- The last problem is the *ownership* of the server. If the server is installed inside one corporation, this makes other corporations vulnerable to *censorship*. The ownership corporation can limit access to data and limit the actions of other corporations on this data. One possible solution to this problem is to host the server in a trusted third party. This will introduce extra costs and do not overcome other drawbacks of the centralized management.

Obviously, centralized management of inter-organizational cooperative activity is a major drawback. It suffers from availability, scalability, performance and resistance to censorship and centralized control. A fully distributed infrastructure is needed to support inter-organizational cooperation.

Typically, Peer-to-peer architectures have as inherent characteristics scalability, resistance to censorship and centralized control, and increased access to resources [ATS04]. Administration, maintenance, responsibility for the operation, and even the notion of ownership of peer-to-peer systems are also distributed among the users, instead of being handled by a single company, institution or person. These architectures allow organizations to continue to use their individual resources to achieve the common goal.

In this paper, we propose to use a P2P infrastructure and related paradigms to support inter-organizational cooperative work. We call our system *P2P InterCoop*. The remainder of this paper is organized as follows: Section2 introduces inter-organizational cooperative work and P2P systems. Section 3 identifies services needed for inter-organizational cooperative work : data sharing, communication and coordination services. Section 4 investigates the state of the art of data management in existing cooperative environments and P2P systems. It looks at the limitations of current approaches to manage shared data in P2P Inter-organizational cooperation and points some ideas about managing shared data in *P2P InterCoop*. Section 5 investigates the state of the art of coordination in existing cooperative environments and P2P systems. It looks at the limitations of current approaches to coordinate activities in P2P Inter-organizational cooperation and points some ideas for coordination in *P2P InterCoop*. The last section concludes the paper and outlines future work.

2 Inter-organizational Cooperative Work and P2P systems

Peer-to-peer systems provide the infrastructure to support inter-organizational distributed work. However, they miss some important functionality. Let us first see the main aspects of these two concepts:

2.1 Inter-organizational Cooperative Work

We see cooperatikon as a group activity of a large number of participants designed to achieve particular purposes or goals. The participants can belong to different organizations, therefore, they are not necessary co-located at the same site and they can be distributed all over the world. We assume that all (or at least most) of people implied in the cooperative work are aware of the common goal. They have reasons to cooperate, communicate and exchange information spontaneously. Typical cooperative activities that we want to consider are co-Editing, co-engineering and collaborative learning. These activities are interactive and creative in nature. Therefore, unlike strictly predefined cooperation, tasks and interactions are defined with regards to the common goal without having an explicit representation. The outcome of cooperative activity is known, but the means for achieving the goal is not explicitly detailed[OO04].

2.2 Peer-to-peer systems

A quick look at the literature reveals a considerable number of different definitions of P2P systems. In [ATS04], the authors propose the following definition: *Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralized server or authority.* Most of the current P2P systems fall within the category of content distribution, which designed for the sharing of digital media and other data between users. Some examples are: Napster[Nap06], Gnutella[Gnu06] and Kazaa[Kaz06].

In this paper, we consider P2P systems as distributed systems consisting of interconnected nodes able to share data and coordinate work for the purpose of allowing participants to work together in order to achieve common objectives without requiring the intermediation or support of a global centralized server or authority. The major advantage of decentralized infrastructure is to deal with traditional limitations of central architectures (failure, scalability, performance and ownership). This means that collaboration does not require any leading organization that stores the shared data.

If organizations want to collaborate without having any privileged organization, the underlying P2P architecture must provide some services. The next section identifies the basic requirements of cooperative work.

3 Inter-organizational Cooperative Work Services

Basically, when people work together, they need to share data and stay informed about the progress of the project [GMO$^+$04].

Shared data People involved in cooperative work need to share information. This information can be files and directories, Internet bookmarks, databases, or more sophisticated tools of knowledge management or a combination of all of these.

Coordination Coordination is an integral part of teamwork. As Mintzberg[Min79] observes: *Every organized human activity- from the making pottery to the placing of a man on the moon- gives rise to two fundaments and opposing requirements: the* ***division of the labor*** *into various tasks to be performed and the* ***coordination*** *of those tasks to accomplish the activity*.

Awareness People involved in cooperative work need to be aware of the current status and the activity in the project.

Communication Team members need to talk to each other, discuss, show results and update them. All groupware tools can be useful if they are well integrated in cooperation support.

In the following sections, we focus on two services data sharing and coordination. Other services are also important but they are out of the scope of this paper. We start by investigating the state of the art of data management in existing cooperative environments. After that, we study shared data management in existing P2P systems, finally, we give some ideas about how to manage data in P2P InterCoop. We follow the same schema for coordination service.

4 Shared data Management

People involved in cooperative work need to share information. This information can be files and directories, Internet bookmarks, databases, or more sophisticated tools of knowledge management or a combination of all of these. In the following, we consider file sharing only.

4.1 Data management in existing cooperative environments

Working on shared data is very common in cooperative work. Configuration Management tools[AFK$^+$95] in Software Engineering define a large spectrum of functionalities needed on shared data.

Objects and Dependencies Tools can manage just files and directories like CVS [Ber90] or support objects dependencies like major configuration management tools [GCCM95].

Concurrency Concurrent access can be managed by explicit locking policies by using copy-modify-merge paradigm like RCS [Tic85] or by using long transaction approach [CGC$^+$98].

Versioning Versions can be use for various purposes: concurrent engineering, experimental development, variants [CW98]

Annotation Tool like BSCW gives functionalities of annotations, ratings for handling review of documents.

Access Control An ACL allows to control the access to shared documents. Only authorized member can read or modify the document.

Notification This allows to keep the member of team informed of all relevant events on shared data. Awareness engines heavily use this information.

The nature of data has an impact on the way team members can shared it. In fact, people do not share files as they do for bookmarks or contact list for example. If the team work with this kind of data, other dedicated tools are more suitable. Domain of knowledge management[AH98] is currently building very interesting tools for this purpose. It is also possible to share data by using databases also. In this case, the underlying database system manages concurrent access with its own mechanism.

The existing tools such as CVS [Ber90] BCSCW [BSC06, RWU$^+$97], SourceForge [Sou06] and LibreSource [Lib06] are based on centralised architectures.

4.2 Shared Data in existing P2P systems

Most of the current P2P systems fall within the category of content distribution, which designed for the sharing of digital media and other data between users. Some examples are: Napster[Nap06], Gnutella[Gnu06] and Kazaa[Kaz06]. Peer-to-peer content distribution systems rely on the replication of content on more than one node for improving the availability of content, enhancing performance, and resisting censorship attempts[ATS04].

With replication, data are available locally; there is no need to a central server. The geographical distant has no impact on the quality of services.

4.3 Shared Data in P2P InterCoop

Data replication is an important feature of P2P systems. It allows both availability and performance. Generally, P2P systems consider data to be very static or even read-only. Furthermore, replicated data changes are not managed. Data are only propagated from one node to another. In cooperative work, data is subject for frequent concurrent modifications. Data consistency and synchronization issues need to be addressed. If two users generate concurrent operations on the replicas of the same file, the system has to ensure that replicas will converge while preserving effects of concurrent operations. Without this guarantee inconsistency (divergence) between replicas will raise. To illustrate that, let us consider two nodes n_1 and n_2 of a P2P network. These nodes share a common file called f. n_1 has a copy of f called f_1 and n_2 has another copy called f_2. In existing P2P, if f_2 is modified on the node n_2 then there is no way to integrate these modifications in the copy of n_1. This means that different replicas of the same file can have different values. Without a mechanism that ensures data consistency, it is not possible to modify shared data in a cooperative way.

If the consistency of shared data is ensured, this allows for P2P systems to support large-scale collaboration while preserving all the advantage of P2P architecture. In this way, the applications fields of P2P applications become larger and larger. For example, we can use P2P architecture to support collaborative distributed software development, collaborative distributed knowledge management, collaborative eLearning and collaborative editing.

4.4 Discussion

Unfortunately, the existing tools to manage shared data are based on central server and do not ensure replicated data consistency. A lot of work have done been done in the domain of CSCW to ensure shared data [SJZ+98, VCFS00, MOSMI03] consistency.

Recently, the authors in [MOSMI03] have developed a generic synchronizer that ensures, as every optimistic replication algorithm [SS05] eventual consistency i.e. when no updates occurs for a long period of time, eventually all updates will propagate through the system and all the copies will converge towards a same value. In other words, when the system is idle (no operations in pipes), all copies are identical. This algorithm can be used in a distributed environment since no central server is needed [MOSMI03].

We want to integrate this algorithm in P2P InterCoop. If we add to this algorithm a push function based on flooding algorithm then it is possible to propagate updates to concerned nodes. Flooding is an approach to distributing data among a set of network nodes in which each node forwards a message to all of its neighnors until all nodes have acknowledged receipt of the message.

5 Coordination service

Coordination is an integral part of teamwork. Every organized human activity needs coordination.

5.1 Coordination in existing cooperative environments

It is well known that there are two complementary ways [GMO+04] to coordinate cooperative work:

- Task coordination also known as formal coordination: this is based on the hypothesis that it is possible to define a process and enforce this process on working sites.

- Group awareness also known as informal coordination: this is based on the hypothesis that if the right information about what other people do, is sent at the right time to the right people, this information will trigger communication between people that will result in automatic coordination of team.

In this work, we are interested in *task coordination*. It is fair to say that; much of the research efforts on *task coordination* are mainly studied in Workflow domain[vdABCC05]. Unfortunately, we cannot apply a workflow approach in P2P InterCoop for two reasons. On one hand, the relevant research literature [vdAB01, Dou01, SO04] confirms that workflow is not fully adequate to express cooperative activities. Empirical research results[dSRD03] point out that the run-time behavior of a process can be too variable than its model defined prior to execution. If the run-time process is wanted to be handled by the workflow management system (WfMS) despite its unpredictable behavior (characteristic of cooperative activities), the process model is expected to include all possible executions. However, in this case the resulting model can be too sophisticated to define and manage[SM95]. On the other hand, most WfMS are based on centralized architecture. Recent research works[vdABCC05] proposes distributed workflow systems, if we examine the given architecture; there is still a central server that manages data and processes.

5.2 Coordination in existing P2P systems

In some P2P systems, there is a coordination mechanism. The coordination and underlying architecture are tightly coupled.

In *Hybrid Decentralized Architectures* there is a central server facilitating coordination between peers. The server breaks down a computer intensive task into small work units and distributing them to different peer computers, that execute their corresponding work unit and return the results, such as Seti@home[Set03] project. The aim of such system is to take advantage of the available peer computer processing power (CPU cycles). Obviously, in these architectures, there is a single point of failure (the central server). This typically renders them inherently unscalable and vulnerable to censorship or technical failure, and therefore they are not adapted for P2P InterCoop.

In *purely decentralized architectures*, all nodes in the network perform exactly the same tasks, acting both as servers and clients. The nodes of such networks are often termed *servents*(SERVers+clieENTS). There is no central coordination of the activities in the network and users connect to each other directly through a software application that functions both as a client and a server (users are referred to as a servents). Example the Gnutella network[Gnu06]. These architectures provide all the advantages of P2P systems.

In *partially centralized architectures*. The basis is the same as with purely decentralized systems. Some of the nodes, however, assume a more important role. These supernodes do not constitute single points of failure for a peer-to-peer network, since they are dynamically assigned and, if they fail, the network will automatically take action to replace them with others.

5.3 Coordination in P2P InterCoop

Obviously, coordination in *P2P InterCoop* depends on the underlying architecture that we want to adapt.

In hybrid decentralized *P2P InterCoop* where there is a central server. Traditional coordination mechanisms can be used. However, we still have the problem of a single point of failure (the central server), which makes *P2P InterCoop* unscalable and vulnerable to censorship or technical failure.

With a purely decentralized approach, there is no central server at all. In this case, at least in order to join the cooperative work the peers have to know a peer that is already in the community to retrieve the necessary information to join the cooperative work. Once a peer joins the cooperative activity the coordination and data sharing can be enacted in a decentralized fashion. We are currently working in this direction.

6 Conclusion

This paper describes motivations and requirements analysis to support inter-organizational cooperative work. It proposes to use a P2P architecture to support this cooperation. This allows scalability, resistance to censorship and centralized control, and increased access to resources. Administration, maintenance, responsibility for the operation, and even the notion of ownership are also distributed among the users, instead of being handled by a single company, institution or person. These architectures allow organizations to continue to use their individual resources to achieve the common goal. However, they miss some important functionality to enable cooperation. The paper identifies the required services for P2P InterCoop i.e. P2P inter-organizational cooperation. Optimistic data replication management and coordination are the main required services. Both of these services must be decentralized.

Literatur

[AFK+95] Larry Allen, Gary Fernandez, Kenneth Kane, David Leblang, Debra Minard und John Posner. ClearCase MultiSite: Supporting Geographically-Distributed Software Development. In Jacky Estublier, Hrsg., *Software Configuration Management: Selected Papers of the ICSE SCM-4 and SCM-5 Workshops*, number 1005 in Lecture Notes in Computer Science, Seiten 194–214. Springer-Verlag, Oktober 1995.

[AH98] Mark S. Ackerman und Christine Halverson. Considering an Organization's Memory. In *CSCW*, Seiten 39–48, 1998.

[ATS04] Stephanos Androutsellis-Theotokis und Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[Ber90] B. Berliner. CVS II : Parallelizing software development. In *Proceedings of USENIX*, Washigton D. C., 1990.

[BSC06] BSCW. The BSCW project web site. *http://bscw.fit.fraunhofer.de*, 2006.

[CGC+98] Gérôme Canals, Claude Godart, François Charoy, Pascal Molli und Hala Skaf-Molli. COO approach to support cooperation in software developments. *IEE Proceedings - Software*, 145(2-3):79–84, 1998.

[CW98] Reidar Conradi und Bernhard Westfechtel. Version Models for Software Configuration Management. *ACM Comput. Surv.*, 30(2):232–282, 1998.

[Dou01] Paul Dourish. Process descriptions as organisational accounting devices: the dual use of workflow technologies. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, Seiten 52–60, New York, NY, USA, 2001. ACM Press.

[dSRD03] Cleidson R. B. de Souza, David Redmiles und Paul Dourish. "Breaking the code", moving between private and public work in collaborative software development. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, Seiten 105–114, New York, NY, USA, 2003. ACM Press.

[GCCM95] Claude Godart, Gérôme Canals, François Charoy und Pascal Molli. About Some Relationships Between Configuration Management, Software Process and Cooperative Work: The COO Environment. In *SCM*, Seiten 173–178, 1995.

[GHP99] Rebecca E. Grinter, James D. Herbsleb und Dewayne E. Perry. The geography of coordination: dealing with distance in R&D work. In *GROUP '99: Proceedings of the international ACM SIGGROUP conference on Supporting group work*, Seiten 306–315, New York, NY, USA, 1999. ACM Press.

[GMO+04] Claude Godart, Pascal Molli, Gérâld Oster, Olivier Perrin, Hala Skaf-Molli, Pradeep Ray und Fethi Rabhi. The ToxicFarm Integrated Cooperation Framework for Virtual Teams. *Distributed and Parallel Databases*, 15(1):67–88, 2004.

[Gnu06] Gnutella. The Gnutella project web site. *http://www.gnutella.com*, 2006.

[Kaz06] Kazaa. The Kazza project web site. *http://www.kazaa.com*, 2006.

[Lib06] LibreSource. The LibreSource project web site. *http://www.libresource.org*, 2006.

[Min79] *The Structuring of Organizations*. Prentice Hall, 1979.

[MOSMI03] Pascal Molli, G?rald Oster, Hala Skaf-Molli und Abdessamad Imine. Using the transformational approach to build a safe and generic data synchronizer. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, Seiten 212–220, New York, NY, USA, 2003. ACM Press.

[Nap06] Napster. The Napster project web site. *http://www.napster.com*, 2006.

[OO04] Andrea Omicini und Sascha Ossowski. Coordination And Collaboration Activities In Cooperative Information Systems. *Int. J. Cooperative Inf. Syst.*, 13(1):1–7, 2004.

[RWU+97] Bentley R., Appelt W., Busbach. U., Hinrichs, E., Kerr, D. Sikkel, S., Trevor J. und Woetzel G. Basic Support for Cooperative Work on the World Wide Web. *International Journal of Human-Computer Studies: Special issue on Innovative Applications of the World Wide Web*, 1997.

[Set03] SetiAtHome. The seti@home project web site. *http://setiathome.ssl.berkeley.edu*, 2003.

[SJZ+98] C. Sun, X. Jia, Y. Zhang, Y. Yang und D. Chen. Achieving Convergence, Causality-preservation and Intention-preservation in real-time Cooperative Editing Systems. In *ACM Transactions on Computer-Human Interactions*, Jgg. 5, Seiten 63–108, 1998.

[SM95] Diane M. Strong und Steven M. Miller. Exceptions and Exception Handling in Computerized Information Processes. *ACM Trans. Inf. Syst.*, 13(2):206–233, 1995.

[SO04] Karsten A. Schulz und Maria E. Orlowska. Facilitating cross-organisational workflows with a workflow view approach. *Data Knowl. Eng.*, 51(1):109–147, 2004.

[Sou06] SourceForeg. The SourceForge project web site. *http://www.sf.net*, 2006.

[SS05] Yasushi Saito und Marc Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37(1):42–81, 2005.

[Tic85] W. F. Tichy. RCS – A system for version control. *Software–Practice and Experience*, 185.

[VCFS00] N. Vidot, M. Cart, J. Ferrie und M. Suleiman. Copies convergence in a distributed real-time collaborative environment. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW-00)*, Philadelphia, Pennsylvania, USA, December 2000. ACM Press.

[vdAB01] W. M. P. van der Aalst und P. J. S. Berens. Beyond workflow management: product-driven case handling. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, Seiten 42–51, New York, NY, USA, 2001. ACM Press.

[vdABCC05] Wil M. P. van der Aalst, Boualem Benatallah, Fabio Casati und Francisco Curbera, Hrsg. *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, Jgg. LNCS 3649, 2005.

[Zop06] Zope. The Zope project web site. *http://www.zope.org*, 2006.