Decentralized Execution of P2P Collaborative Processes

Ustun Yildiz, Thomas Tamisier, Hala Skaf-Molli, Fernand Feltz

Decentralized Execution of P2P Collaborative Processes

Ustun Yildiz[1], Thomas Tamisier[2], Hala Skaf-Molli[1], and Fernand Feltz[2]

[1] University of Nancy1- LORIA
Campus Scientifique,BP 239, F-54506 Vandœuvre-lès-Nancy, France,
{yildiz,skaf}@loria.fr,
[2] CRP - Gabriel Lippmann, 41, rue du Brill
L-4422 Belvaux Luxembourg,
{tamisier,feltz}@lippmann.lu

Abstract. The growing importance of exchanges and collaborations in all business areas calls for fast, efficient, and flexible models of computer-supported cooperation. As the traditionnal client-server paradigm is hampered by its structural limitations, the interest of the Information System Community is aroused by the promises of alternatives known as peer-to-peer approaches. This paper introduces a generic architecture, designed for the execution of collaborative business processes. Basic motivations of the model are discussed, as well as major contributions notably in terms of service-oriented routing and failure handling.

1 Introduction

The success of Napster system draws the attention of the information systems community to the importance of the P2P paradigm, as a distributed model to make computer agents cooperate. In a P2P architecture, the agents, called peers, are all equal partners. They

perform the same tasks, acting both as servers and clients, and connecting to each other directly.

On the contrary, in the traditional client-server model, only the participants and the tasks of a business process are distributed, whereas coordination and data localization mechanisms are centralized. The existence of a central entity introduces many weaknesses. Among them, the most important is certainly the fact that a correct execution of the processes relies entirely on a single coordinator. Therefore, unavailability of the later automatically provokes an interruption of the services depending on its activities. Other limitations of this centralized model relate to availability, scalability, performance, and ownership[2].

The basic motivation of P2P architectures is to offer a true alternative to the client-server paradigm. A P2P execution of a business process aims to eliminate any risk of global breakdown, by ensuring that in spite of the unavailability of a participant, the process is able to continue the execution.

This decentralization of the resources and the services leads to many technical advantages pushing back the limits of the centralized model. New software developed on P2P [9] infrastructures provide advanced functionalities exceeding the traditional utilities of file sharing such as replication, synchronization, or process management.

Furthermore, the P2P model deeply influences the whole life cycle of information systems and business processes. Current trends in service-oriented computing offer a flexible way to implement a P2P

based infrastructure. Particularly, dedicated services allow to access data and/or applications through web accessible services that can be invoked by common web protocols (SOAP over HTTP).

This paper presents P2P-Coop, a new P2P middleware infrastructure for distributed execution of business processes based on service-orientation principles. The rest of the paper is organized as follows. The next section is devoted to some recalls concerning the domain of collaborative processes. Section 3 introduces the P2P-COOP architecture in a whole. Section 4 describes in detail the execution of concurrent processes according to the P2P model. Last, we conclude by summarizing the actual status and perspectives of our researches.

2 P2P Systems for Collaborative Processes

This section discusses the basics motivations and requirements concerning the P2P approach in the contect of business activities. The first point is the definition of an effective cooperation between the processes within the distributed framework. Then, we concentrate on actual solutions for coordinating the processes.

2.1 Cooperation in Peer-to-peer systems

We see cooperation 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.

A quick look at the literature reveals a considerable number of different definitions of P2P systems. In [2], 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[8], Gnutella[5] and Kazaa[7].

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.

Basically, when people work together, they need to share data and stay informed about the progress of the project [6]. In the context of a distributed P2P system, these requirements are formalized through the following aspects.

Data sharing 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[1] 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 coordination. Other services are also important but they are out of the scope of this paper.

2.2 Coordination in Peer-to-peer systems

Coordination is an integral part of teamwork. Every organized human activity needs coordination. It is well known that there are two complementary ways [6] 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*. Much of the research efforts on *task coordination* are mainly studied in Workflow domain[14]. Unfortunately, we cannot apply a workflow approach for two reasons. On one hand, the relevant research literature [13, 4, 10] confirms that workflow is not fully adequate to express cooperative activities. Empirical research results[3] 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[12]. On the other hand, most WfMS are based on centralized

architecture. Recent research works[14] proposes distributed workflow systems, if we examine the given architecture; there is still a central server that manages data and processes.

Existing Solutions for 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[11] 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[5]. 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.

3 Some Requirement Analysis

We use classical definition of the process. A process corresponds to an ordered set of activities. The order of these activities defines the sequence of service calls since activities are directly mapped to existing services (which again can be processes).

3.1 Coordination Issues

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 dierent 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 peertopeer network, since they are dynamically assigned and, if they fail, the network will automatically take action to replace them with others.

3.2 Coordination

Obviously, coordination depends on the underlying architecture that we want to adapt. In hybrid decentralized 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.

4 Conclusion

This paper describes motivations and requirements analysis to support interorganizational 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. i.e. P2P inter-organizational cooperation. Optimistic data replication management and coordination are the main required services. Both of these services must be decentralized.

References

1. *The Structuring of Organizations.* Prentice Hall, 1979.

2. Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.
3. Cleidson R. B. de Souza, David Redmiles, and 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*, pages 105–114, New York, NY, USA, 2003. ACM Press.
4. 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*, pages 52–60, New York, NY, USA, 2001. ACM Press.
5. Gnutella. The gnutella project web site. *http://www.gnutella.com*, 2006.
6. Claude Godart, Pascal Molli, Gérâld Oster, Olivier Perrin, Hala Skaf-Molli, Pradeep Ray, and Fethi Rabhi. The toxicfarm integrated cooperation framework for virtual teams. *Distributed and Parallel Databases*, 15(1):67–88, 2004.
7. Kazaa. The kazza project web site. *http://www.kazaa.com*, 2006.
8. Napster. The napster project web site. *http://www.napster.com*, 2006.
9. Gérald Oster, Pascal Urso, Pascal Molli, and Abdessamad Imine. Data consistency for p2p collaborative editing. In *CSCW*, 2006.
10. Karsten A. Schulz and Maria E. Orlowska. Facilitating cross-organisational workflows with a workflow view approach. *Data Knowl. Eng.*, 51(1):109–147, 2004.
11. SetiAtHome. The seti@home project web site. *http://setiathome.ssl.berkeley.edu*, 2003.
12. Diane M. Strong and Steven M. Miller. Exceptions and exception handling in computerized information processes. *ACM Trans. Inf. Syst.*, 13(2):206–233, 1995.
13. W. M. P. van der Aalst and 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*, pages 42–51, New York, NY, USA, 2001. ACM Press.
14. Wil M. P. van der Aalst, Boualem Benatallah, Fabio Casati, and Francisco Curbera, editors. *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005, Proceedings*, volume LNCS 3649, 2005.