Implementation of a Multiplatform P2P System

Nobuhiro Nakamura*, Souichirou Takahama*, Leonard Barolli*, Jianhua Ma**, Kaoru Sugita*

*Faculty of Information Engineering, Fukuoka Institute of Technology (FIT)
E-mail: {s01a1036, s01a1027}@ws.ipc.fit.ac.jp
E-mail: {barolli, sugita}@fit.ac.jp
**Faculty of Computer and Information Sciences, Hosei University
jianhua@k.hosei.ac.jp

Abstract

Peer-to-Peer (P2P) computing offers many attractive features, such as collaboration, self-organization, load balancing, availability, fault tolerance and anonymity. However, it also faces many serious challenges. In our previous work, we implemented a synchronous P2P collaboration platform called TOMSCOP. The TOMSCOP provides four types of services: synchronous message transportation, peer room administration, peer communication support and application space management. By using these services, different kinds of shared applications for various specific purposes can be relatively easily developed and associated collaborative cyber spaces or communities can be quickly built across the JXTA virtual network overlaid on top of the existing physical networks. However, the TOMSCOP was implemented only in Windows XP OS. In this paper, we extend our previous work and present the implementation of a Multi-Platform P2P System (MPPS). We show two implemented applications of MPPS: Joint Draw Pad and Shared Web Browser. The proposed system operates very smoothly in UNIX Solaris 9 OS, LINUX Suse 9.1 OS, Mac OSX, and Windows XP. In the future study, we would like to evaluate the implemented P2P platform and to deal with security problems.

マルチプラットフォーム P2P システムの実装

中村暢大*, 高濱聡一郎*, バロリ・レオナルド*, 馬　建華**, 杉田薫*

*福岡工業大学情報工学部
E-mail: {s01a1036, s01a1027}@ws.ipc.fit.ac.jp
E-mail: {barolli, sugita}@fit.ac.jp
**法政大学情報科学部
jianhua@k.hosei.ac.jp

Abstract

P2P コンピューティングは、コラボレーションや自己組織化、負荷分散、可用性、耐障害性、匿名などのたくさんの魅力的な特徴を提供する。これまでの研究では、TOMSCOP と呼ばれるシンクロナス P2P コラボレーションプラットフォームが提案されている。 TOMSCOP は synchronous message transportation、peer room administration、peer communication support and application space management という４つのタイプのサービスを提供している。これらの４つのサービスを使って、P2P ネットワーク上で協調的に作業を行うためのアプリケーションが比較的簡単に開発できる。しかし、TOMSCOP は、Windows XP OS でしか実装されていない。本研究では、Multi-Platform P2P System (MPPS)を実装し、MPPS の二つの応用について報告する。

1. Introduction

Peer-to-Peer (P2P) computing offers many attractive features, such as collaboration, self-organization, load balancing, availability, fault tolerance and anonymity. However, it also faces many serious challenges. Collaborative computing, usually known as groupware or computer supported collaborative work (CSCW), refers technologies and systems that support a group of people engaged in a common task or goal and that provide an interface to a shared environment. Grudin [1] in 1994 defined a time/location matrix to generally categorize collaborative systems as four types, among which there is one called a distributed *synchronous collaborative system* that can support a group of people in different locations to conduct a common task or activity at the same time. A necessary and fundamental element in a synchronous collaborative system is the *shared application* that multi-users can synchronously view and manipulate with following the mode of what you see is what I see (WYSIWIS) [2].

The shared applications fall into two categories, screen-copy system and event-aware system [3]. The former allows many existing single-user applications to be used by multi-users in cooperative fashion via capturing an application window and sending it as image data similar as a video camera. Typical collaboration-transparent systems are Sun ShowMe, Microsoft NetMeeting, and Intel ProShare. In the latter system, only events related to an application are captured and sent out. The event-aware systems make more efficient use of networks, and can support more advanced groupware functions.

There are generally three types of connection and message passing topologies between multiple users' computers/devices used for their collaborations. One is called a *centralized topology* in which there is no direct connection between computers and all messages are mediated by an intermediator generally known as a group server. VCR [4], Habanero [5], Worlds [6], TANGO [7], and TeamWave [8] adopted this topology. However, systems built on the above platforms suffer common problems that a communication bottleneck may arise since all messages must first go to and then get out from the server, and the whole system may be down when the server has some troubles. Actually such the connection topology follows the ordinary client/server model. The P2P model is used for the other two, *hybrid topology* and *decentralized topology* [9,10]. The hybrid topology is one in which a peer needs to connect to both a group server and other peers. In this connection, some group administration messages are passed via the server and other messages are sent directly to others. The decentralized topology is one in which every peer is able to directly connect to all other peers and messages are sent without intermediation via a server. Groove [11] and Endeavors' Magi [12] have adopted the hybrid topology. Even they overcome some drawbacks of client/server-based systems, a peer has yet to go to the server and strictly follow the procedures defined by a particular system. Peers have no enough flexibility to quickly find each other and easily form a group under the full control by themselves. Furthermore, the two systems only work in Microsoft Windows environments.

The TOMSCOP system [13] is based on JXTA framework that consists of the virtual JXTA network and basic peer group services [14~16]. The JXTA virtual network is based on the five abstractions: universal IDs, peer-groups, advertisements, resolvers and pipes, and a set of standard protocols overlaid on top the existing physical network infrastructure. It allows a peer to exchange messages with any other peers independently of its network location (firewalls, NATs or non-IP networks). JXTA is a general framework potentially able to support a broad range of P2P computing such as distributed computation, storage, agent, content distribution, system test, etc. On the JXTA open source web site [17] there are currently about 90 JXTA based projects.

On the top of JXTA framework, TOMSCOP provides four types of services: synchronous message transportation, peer room administration, peer communication support and application space management, to ease developments of shared applications and creations of collaborative communities. However, the TOMSCOP was implemented only in Windows XP OS. In this paper, we extend our previous work and present the implementation of a Multi-Platform P2P System (MPPS). The proposed system operates very smoothly in UNIX Solaris 9 OS, LINUX Suse 9.1 OS, Mac OSX, and Windows XP.

This paper is organized as follows. In the next Section, we will explain TOMSCOP. In Section 3, we introduce our proposed multiplatform P2P system. In Section 4, we present some functions of the proposed system. Finally, we conclude the paper in Section 5.

2. TOMSCOP Overview

The TOMSCOP is a event-aware system but with a complete different connection topology and other special features.

As we mentioned previously, the JXTA framework is aimed at a set of general virtual network protocols and basic peergroup services to support a broad range of P2P distributed applications. Therefore, the same work may be repetitively done for developers to implement similar functions in a relatively large application domain like the distributed computation, storage or agent. So does the domain of synchronous collaboration. To provide necessary services used in this domain, the TOMSCOP platform is developed as a bridge between the JXTA framework and possible shared applications as well as collaborative communities as shown in Fig. 1.
TOMSCOP is designed using the metaphor of center-room-facility. Users or peers gather in a virtual community center to meet each other, enter some rooms corresponding to specific groups of interests, and work together via using available facilities, i.e., shared applications. As shown in Fig. 2, the platform provides four kinds of services:

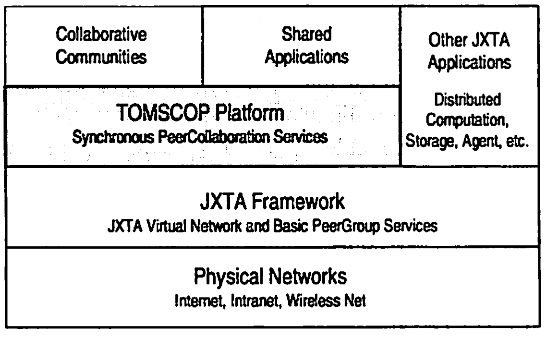

Figure 1. TOMSCOP platform.

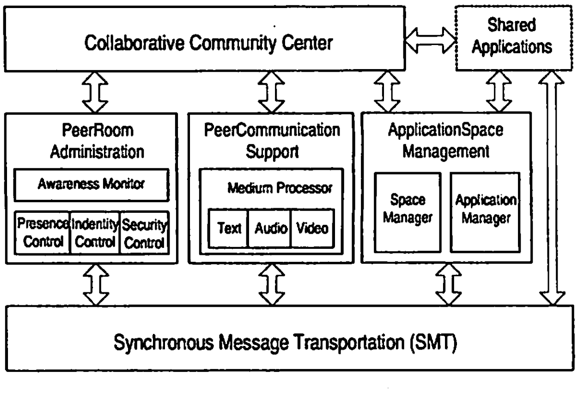

Figure 2. TOMSCOP architecture.

- *Synchronous Message Transportation* to transport all messages between peers in the center and inside rooms based on the JXTA pipe service.
- *PeerRoom Administration* to administrate peers and rooms in the center, and promptly shows the core awareness information of peers and rooms via using presence control and identity control.
- *PeerCommunication Support* to provide a set of communication channels in different media among all peers in the center, among peers in rooms and between any pair of peers. The JMF (Java Media Framework) technology is also used for audio and video communications.
- *ApplicationSpace Management* to manage usages of virtual center spaces and common shared applications to maintain good harmonization in collaborations among multi-users in a virtual room but physically in different places.

3. Proposed MPPS

We have implemented MPPS in our lab in four OS. The P2P environment includes two Workstations Sun Blade 1500 (OS: Solaris 9; CPU: 1.062 GHz UltraSPARC IIIi, HD: 80GB, Memory: 512 MB), three note book PCs (OS: Windows XP; CPU: Pentium M 1.5GHz, HD: 40GB; Memory: 768 MB), one note book PowerBook G4 PC (OS: MacOSX Ver.10.3.4; CPU: PowerPC G4 867 MHz; HD: 40GB; Memory: 256 MB) and two Desktop computers (OS: SUSE Linux 9.1; CPU: Pentium 4 2.60GHz; HD: 80GB; Memory: 1GB).
The MPSS the same as TOMSCOP uses the Synchronous Message Transportation (SMT), which consists of two modules: a message sender and a message receiver, as shown in Fig. 3.

The message sender includes three main functional components: a data collector, a message encoder and a message pusher. The message receiver includes three correspondent components, a data distributor, a message decoder and a message listener. The pipes are abstractions of JXTA data transmission route on the JXTA virtual network. There are three basic pipes: insecure unicast types, secure unicast type, and propagate type. The propagate pipe is used in SMT due to its multicasting ability necessary for group data transportations. In addition to the pipes used for group communication inside rooms, a peer is able to exchange messages with any individual peer and all peers in a collaborative community, which is a collection of peers and rooms. Therefore SMT provides three categorized transmission modes:

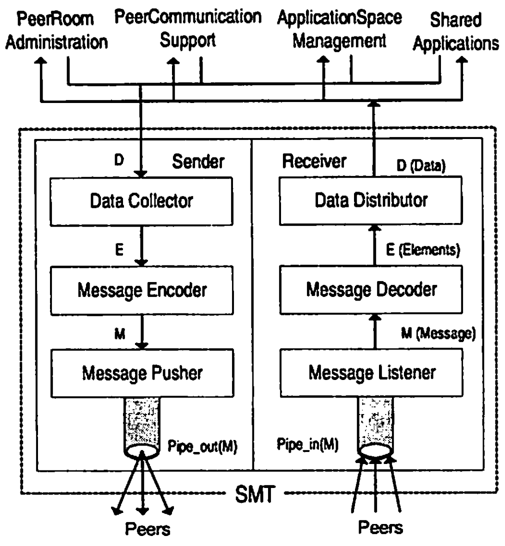

Figure 3. Message sender and receiver.

Table 1: Message formats.

Field Name	Value		
Transmission Mode	Room	Community	One-to-One
Destination	Room/Peer	" Community "	Peer
Source	Peer	Peer	Peer
Coding ID (CID)	Integer	——	——
Application ID (AID)	Integer	Integer	Integer
Object ID (OID)	Integer	Integer	Integer
Element	String, integer, byte	String, integer, byte	String, integer, byte

- Room mode: for group message multicasting among shared applications in a room
- Community mode: for message broadcasting to all peers in a community
- One-to-One mode: for one-to-one private message exchange between any two peers in a community

Thus we have built three kinds of propagate pipes corresponding to the above modes as shown in Table 1.

Due to no server in the system, each peer is able to administrate groups (i.e., rooms) by using the services of Peer Room Administration (PRA), which is responsible for room creation, publication and searching under supported by the JXTA. Any room created by a peer must have a unique name and ID number, and its associated group advertisement should be generated and published on the JXTA network so that other peers can find the room by searching the room advertisement.

The presence control is implemented to show the peer's presence information. It is difficult to administrate peers' presence in real-time only using advertisements provided by JXTA core service. For example, others may not be aware of the peer's leaving by only using advertisements. One solution is that when a peer enters/leaves a community and room, an associated announcement message will be sent to all related peers. However, the announcing mechanism is not yet enough to handle some abnormal leave in the case that a peer can't send the message due to a computer or network trouble. To provide correct presence information in any possible practical situations, the ping-pong detection approach is used to manage peers' presences approximately in real-time.

A peer in a room of community can be assigned as some role such as a group leader or member for synchronous collaborative work. Peers' roles in a room are managed by the identity control. Generally, members with different identities play different roles to conduct meeting, conference and other group work smoothly. In the same way, we can cast these roles to all peers in a room to do some specific collaborative work. We defined three identities in the room of community as follows.

- *Chair* to control the identities of other peers and also play a coordinative role in a room
- *Player* to be able to control shared spaces and manipulate shared applications, such as a game player, meeting presenter, etc., in the real world.
- *Observer* to only watch the shared space and applications but have no right to manipulate the space and applications.

As in TOMSCOP the PeerRoom Administration (PCS) provides built-in communication tools needed for collaborative processes on administrating rooms and using shared applications. The communication tools are built based on both SMT and medium processors for processing text, audio and video. The SMT offers three transmission service modes: community, room and one-to-one to send a message to all peers, room peers, and an individual peer, respectively. Each mode uses its own pipe corresponding to the related pipe advertisement. An example in Fig. 4 shows three parallel channels of text-based chats, the community chat, room chat and Instance Message (IS) between peers outside and inside rooms in a collaborative community.

All peers who login into JXTA network belong to community. The community chat is for all peers, while the room chat is for peers only inside a room. Any two peers can also simultaneously do one-to-one chat using IS in any time when needed.

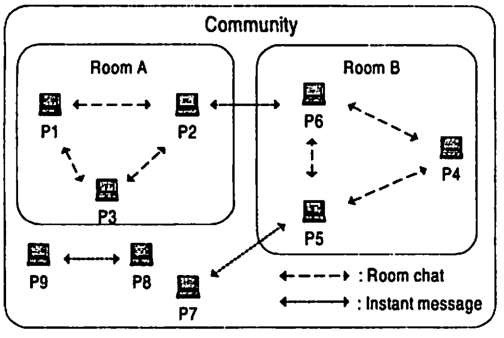

Figure 4. Communication channels.

Each of three communication channels uses its own format and pipe as given in Tab. 1. Two peers in different rooms can't communicate with each other via their room channels that use different pipes, but they can send messages to each other via IS or the community channel.

Peers in a room of community share a virtual space and related virtual facilities by using ApplicationSpace Management (ASM). The ASM consists of a space manager to control the shared spaces, and an application manager to control operations on shared applications. Both managers are related to peers' identities (chair, player and observer) in a room. A presenter peer in a room has initiative to operate a virtual space and others can only see what he/she has done. A chair can change other peers' identities according to actual situations.

4. Applications of MPPS

In this section, we will give two applications of the proposed MPPS called Join Draw Pad and Shared Web Browser.

The Joint Draw Pad is a tool for making joint figures or designs. The users may be in different locations but they can draw or make a design in the same pad. We will show in the following figures the display for four environments (Windows XP, Linux, Mac and UNIX). In Fig. 5 (Windows XP) is drawn the circle and this circle is shown at the same time in three other environments. The heart is drawn in Linux, the rectangular shape is drawn in Mac, and the triangular shape is drawn in UNIX. The display captures for Linux, Mac and UNIX are shown in Figs. 6, 7, and 8, respectively. This tool also can be used for collaboration research from different sites.

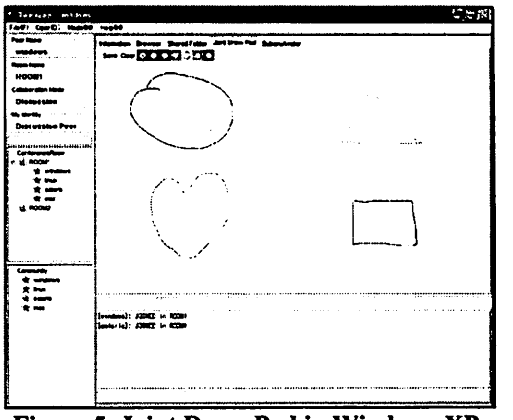

Figure 5. Joint Draw Pad in Windows XP.

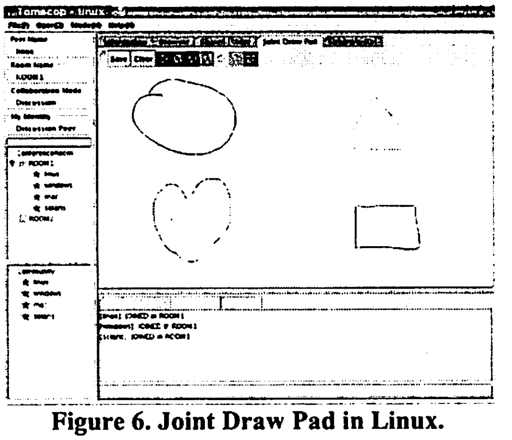

Figure 6. Joint Draw Pad in Linux.

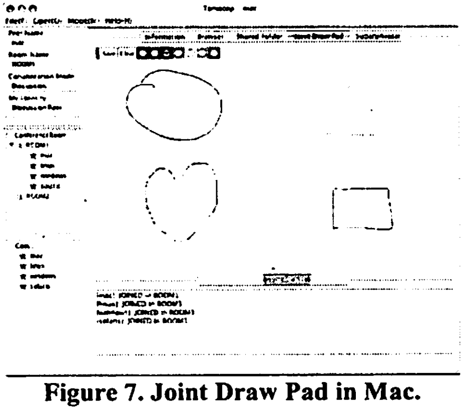

Figure 7. Joint Draw Pad in Mac.

The Shared Web Browser application operates in four environments, but for the sake of space we will show only the display in Linux and Mac OS.

This application can be also used as a collaboration tool. For example while working together with a friend and he wants some information of a homepage, the peer can show directly the home page he is using as shown in Figs. 9 and 10.

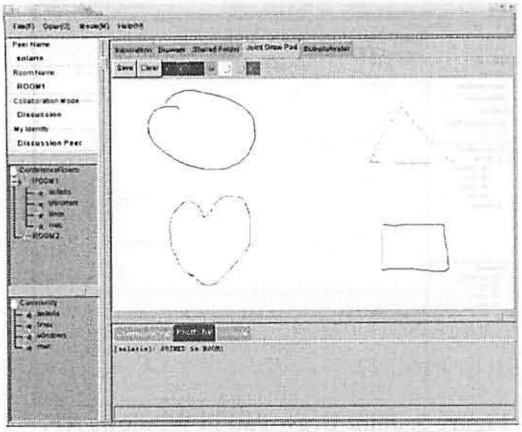

Figure 8. Joint Draw Pad in UNIX.

Figure 9. Shared Web Browser in Linux.

Figure 10. Shared Web Browser in Mac.

5. Conclusions and Future Work

In this paper, we improved our previous platform TOMSCOP and implemented a multiplatform P2P system called MPPS. Different from many other similar platforms, it has adopted a pure P2P architecture and each peer has to administrate collaborative rooms. To demonstrate the platform effectiveness, we showed two applications of the proposed MPPS called Join Draw Pad and Shared Web Browser.

As the future work we will deal with following issues.

- Secure room administration in P2P communication.
- Audio and video communications.
- Efficient pipe advertisement creation to avoid creating duplicated pipe advertisement for a room.
- Development of new shared applications.
- Comparison of MPPS with other P2P systems.

References

1. J. Grudin, *Computer-Supported Cooperative Work*, IEEE Computer, 1994.
2. R. Steinmetz and K Nahrsted, *Multimedia: Computing, Communications and Applications*, Prentice Hall PTR, Upper Saddle River, NJ 07458, 1995.
3. J. B. Begole, *Usability Problems and Causes in Conventional Application-Sharing Systems*, http://simon.cs.vt.edu/begolej/Papers/, 1999.
4. Jinahua Ma, R. Huang and R. Nakatani, *Towards a Natural Internet-Based Collaborative Environment with support of Object Physical and Social Characteristics*, International Journal of Software Engineering and Knowledge Engineering, World Scientific, Vol.11, No. 1, 2001.
5. Chabert, et al, *Java Object-Sharing in Habanero*, Communications of ACM, 41(6), 1998.
6. T. Mansfield, et al., *Evolving Orbit: A Progress Report on Building Locales*, in the proceeding of the Group'97 International Conference, pp.241-250, ACM Press, 1997.
7. L. Beca, etal, *TANGO - a Collaborative Environment for the World-Wide Web*, http://trurl.npac.syr.edu/tango/
8. S. Greenberg and M. Roseman, *Using a Room Metaphor to Ease Transitions in Groupware*, Research report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, January 1998.
9. David Barkai, *Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net*, Intel Press, 2001.
10. Bo Leuf, *Peer To Peer: Collaboration and Sharing over the Internet*, Addison-Wesley, 2002.
11. James Edwards, *Peer-to-Peer Programming on Groove*, Addison-Wesley, 2002.
12. *Magi*, http://www.endeavors.com/
13. T. Kawashima, J. Ma, *TOMSCOP-A Synchronous P2P Collaboration Platform over JXTA*, Proc. of IEEE ICDCS2004/MNSA2004, pp.85-90, 2004.
14. Project JXTA, http://www.jxta.org/
15. J.D. Gradecki, *Mastering JXTA: Building Java Peer-to-Peer Applications*, Wiley Pub., 2002.
16. B. Traversat, et al, *Project JXTA 2.0 Super-Peer Virtual Network*, Sun Microsystems, Inc, 2003.
17. http://www.jxta.org/servlets/DomainProjects