Policy-Based Management for Multimedia Collaborative Services

Hamid Harroud[1], Mouhsine Lakhdissi[1], Ahmed Karmouch[1], and Cliff Grossner[2]

[1] Multimedia & Mobile Agent Research Laboratory, School of Information Technology & Engineering (SITE), University of Ottawa,
161 Louis Pasteur St. Ottawa, ON, Canada K1N 6N5, Canada
{hharroud, mlakhdis, karmouch}@site.uottawa.ca
http://deneb.genie.uottawa.ca
[2] Nortel Networks, Ottawa, Canada

Abstract. Virtual team issue has emerged as an important new teamwork model, distinguished from the conventional way in which people work by its ability to transcend distance, time and organizational boundaries. A virtual team consists of a dynamic collection of individuals, a set of collaborative services and network facilities that ensure a flexible and secure coordinated resource sharing. In the Multimedia and Mobile Agent Research Laboratory we have developed the V-Team system, an agent-based multimedia collaborative environment to support virtual teams. V-Team aims to provide a set of team services for better collaboration between a virtual team participants, facilities for managing virtual teams with fully customized team services, and a simpler interface to network services. In this paper, we describe the main components of V-Team, the principles of context customization and the system monitoring approach via policies.

1 Introduction

With recent advances in network infrastructures and computing capabilities, groupware technology is becoming an important issue to offer an appropriate collaborative environment for distributed teams that transcend distance, time zones, and organizational boundary, known as *virtual teams* [1].

A virtual team consists of a dynamic collection of individuals, a set of collaborative services and network facilities that ensure a flexible and secure coordinated resource sharing. It requires agile and flexible groupware system that overcomes technical and organizational difficulties, without sacrificing the ability of its easy use and management from participants' point of view.

In the Multimedia and Mobile Agent Research Laboratory we are exploring different aspects of agent and groupware technologies in order to provide virtual teams with a system capable of supporting such requirements. The goal of the **V-Team** system is to develop a collaborative environment that would not only provide a set of team

E.S. Al-Shaer and G. Pacifici (Eds.): MMNS 2001, LNCS 2216, pp. 285–298, 2001.

services for better collaboration between a virtual team participants, but also facilities for managing virtual teams attributes and the context customization of their collaboration.

Our previous research experience has shown that agent paradigm combined with the use of policies is a promising approach for the development of flexible distributed systems [2][3]. Agents have the necessary autonomy to act on behalf of users or programs and to migrate from host to host on a network while performing their tasks. The use of policies carries out the ability to dynamically change the behavior of the system without altering its code. These characteristics highly reduce the system complexity, while permitting an efficient control of virtual team services at different levels.

Thus, we have designed an agent-based system that offers virtual teams an appropriate collaborative environment, with fully customized team services and flexible agent-based monitoring. Team services include multimedia conferencing, distance group meeting facilities and virtual teams' context management service. The context management service includes participants' attributes, their capabilities, the selection of applications and resources, and network mechanisms to be used.

The organization of the paper is as follows. The following section introduces the main components of V-Team. The basic principles of context customization are then described, with the policy-based approach and the system monitoring being presented in subsections. Then, basic team services that offer virtual team participants mechanisms to "meet" without being in face to face situation are presented. The benefits of our approach and related work are then discussed. Finally, we draw our conclusion and future work.

2 Agent-Based System for Collaborative Environment

In this section we first present fundamental requirements on an environment to permit collaborative work within a virtual team. Based on the identified requirements, we present a generic conceptual framework that supports synchronous collaboration between virtual team members. Then, we describe agents that compose the system architecture, the agent platform that supports them, and the inter-agent communication protocol.

2.1 System Requirements

In developing V-Team, we considered various requirements that could enable an efficient relationship among a recognizable team of people, a set of collaborative services, a set of network capabilities and a collection of rules binding these elements together such they behave in a consistent manner to satisfy team needs. For an environment supporting the integration of such entities, we have identified the following specific requirements:

- Ability to create and manage a virtual team context that enables a full customization of services.

- Automatic authentication of team members, and configuration of collaborative sessions without requiring their direct intervention.
- Enabling a seamless physical resource allocation of logical resources requested when creating the team.
- A separation between team services (applications) and network services (e.g. mobility location, CoS/QoS, multi-party, peer to peer, multimedia session control), so that a team service may dynamically be adapted to network conditions and team's context.
- Distributed control and management of dynamic changes that may occur during the team life cycle, either in its structure, its activity and planning, or in its members' locations.

2.2 Conceptual Approach

We have designed the V-Team environment as agent architecture. Agents show special promise in enabling the rapid construction of robust and reusable software [4]. Agents cooperate and communicate with each other, and have the ability to communicate with and directly control explicit applications. To monitor the behavior of an agent and its decision making, we attach to the agent a set of policies. A policy is a set of conditions that prohibit or permit an agent (called the subject) to perform actions on target entities. Conditions may concern the subject, the target or the state of the system. A policy may trigger events for performing some actions when conditions are applicable. The use of policies, in V-Team, highly reduces the system complexity, while permitting an efficient control of virtual team activities.

The key component of V-Team system is that of V-Team Context Agent (VTC). VTC agent manages information about one virtual team participants' attributes, their capabilities and roles, team services to be used by all or certain team members, logical resources requested when creating the team, network services and different forms of underlying transport mechanisms. By gathering information about a virtual team and its context, the VTC agent generates a set of policies that monitor the behavior of agents representing a virtual team's participants.

V-Team also features the network service agent (NSA) and the team control agent (TCA).

NSA provides a simple interface to network services, such as mobility management, transport classes-of-service, privacy of transport and multimedia session control. It exposes these, and other, services to teams and their team services in a manner that makes the services easy to access and use. NSA also permits the underlying mechanism to change dynamically according to the network conditions and the environment context as needed.

TCA supports collaborative session and controls interactions between active virtual team members. It manages the distributed events' serialization and dispatches them to each active participant. Management includes the reordering and the filtering of events.

These three agents establish a relationship between participants, team services or applications to collaborate with and the network links that enable participants' interactions (figure 1).

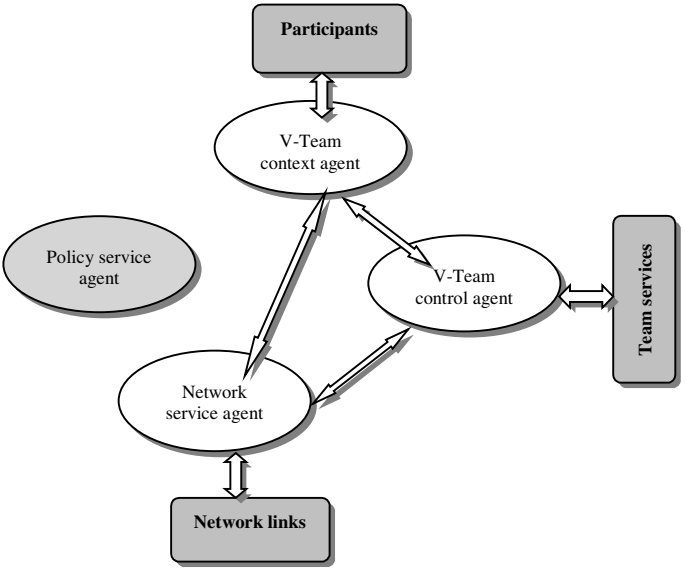

Fig. 1. V-Team system basic model

The Policy Service Agent (PSA) manages policies that govern the utilization of the system and network resources at each participant location. For instance, PSA may prohibit authorizations to a participant for obtaining a particular team service or resources at a specific location. Policies include the monitoring of the overall system configuration at different levels.

When an authorized authority (i.e. team leader) creates a virtual team, the system assigns a VTC agent to manage its specific environment and a NSA agent to support its network services. A TCA agent will then be associated to each collaborative session that would be activated by team participants.

2.3 System Components

The architecture of the system is depicted in figure 2. It is composed of several cooperating agents that work together to provide a collaborative environment for virtual teams.

Part of the V-Team context agent is the automatic authentication of participants, via their Personal Assistants (PAs). PAs are agents that represent participants and act on their behalf. The VTC agent provides a user interface that allows an authorized authority (i.e. team manager) to create a virtual team by specifying its members with

their privileges, logical resources assigned to the team, and the QoS assigned to different team services and media. The Team Participant Agent (TPA) customizes the user interface according to each user's privileges and the context. To setup a V-Team session, a negotiation process among PAs may then be engaged under the control of the VTC agent, which plays the role of a mediator.

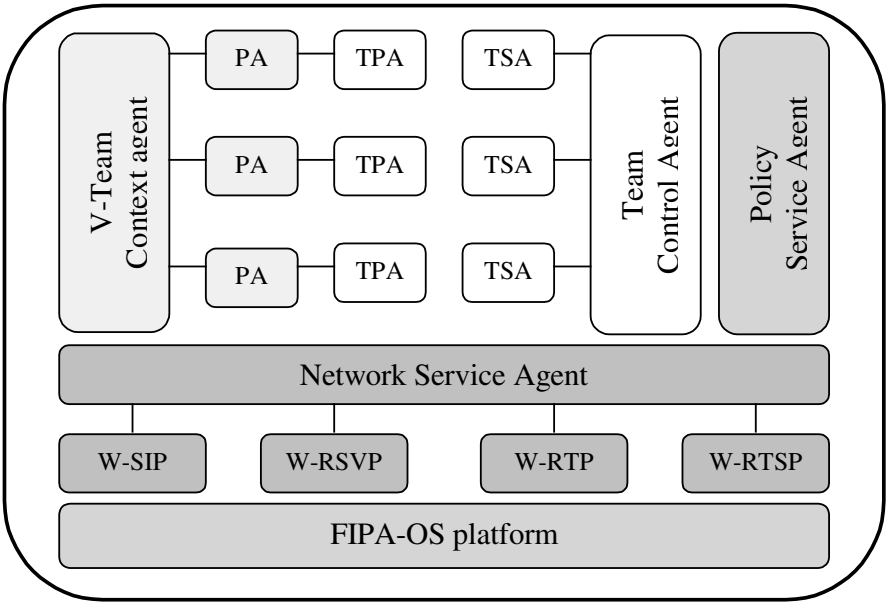

Fig. 2. Generic V-Team agent-based architecture

At the time of starting a V-Team session (e.g. audio conferencing), the NSA agent is required to locate the team members wherever they are and to control the session, using W-SIP and W-RTP agents that wrap SIP protocol [5] and Real-time Transport Protocol respectively.

Different network services become, in effect, integrated to the system by wrapping them through agents.

V-Team is intended to provide a basic set of team services that could be either a collaborative aware or unaware services. Any team service becomes "collaborative" by defining its corresponding Team Service Agent (TSA). TSA agents facilitate interaction among participants under the management of the team control agent (TCA).

V-Team is designed to be highly flexible and dynamically manageable through multi-level policies. Policies are maintained and distributed over V-Team agents through the PSA.

V-Team agents are executed under the control of FIPA-OS platform [6], which provides a framework for agent communication and management, including Agent

Communication Channel (ACC) using Agent Communication Language (ACL), Agent Management Service (AMS), Directory Facilitator (DF), and mobility services.

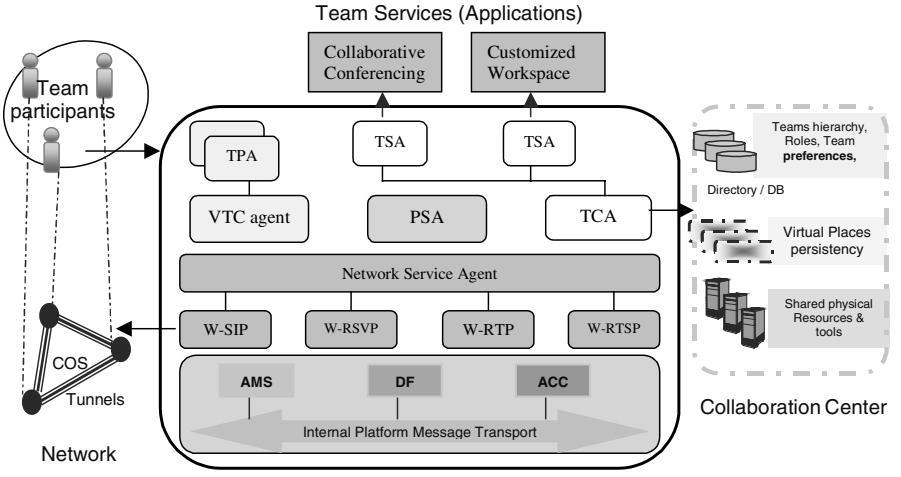

Fig. 3. V-Team System View

Figure 3 shows an example of V-Team system view. TSA provides linkage and execution environment necessary to control Collaborative Conferencing and Customized Workspace services. The NSA provides the control and resource reservation necessary to enable Class-of-Service and privacy in underlying transport mechanisms. Team participants are in different locations and may move from one location to another. Automatic locating of team participants is then required. Using SIP proxy server and location server, tunnels may be automatically established between participants. Participants "meet" at virtual places. The collaboration center permits places storage and playback, resource sharing, and information management.

3 Policy-Based Context Customization and Monitoring

A system architecture like that of the V-Team system described in the previous section requires specific mechanisms to collect and maintain virtual teams' context, and to dynamically manage the overall behavior of the system. Those are the VTC and the PSA agents' roles that are described in this section.

3.1 V-Team Context Agent

The VTC is a context-aware agent that uses context information to adapt the corresponding virtual team behavior to the current situation of use. Context information

includes any information that characterizes the team operating-environment, such as member's identity, the role within the team, location, time, and availability of resources (e.g. bandwidth and device capabilities) to name but a few.

The VTC features are the following:

- To collect the entire context about a particular virtual team, by gathering information from the system entities such as the team manager, participants via their PAs, the network service agent and the PSA.
- To interpret context information by translating it from different representation formats to a set of context policies that could then be distributed to specific agents via the PSA (e.g. TPA, TCA) or used by the VTC agent itself.
- To evaluate context-triggered policies and to execute enabled actions accordingly when in a certain context. This includes the selection of information and services to be customized to the virtual team members. An example of customization would be that when a participant joins a collaborative session on a phone, he would use audio only and other materiel of the ongoing session (e.g. shared document) would be e-mailed to him or shown on a nearby terminal (e.g. fax machine). The VTC agent is also able to satisfy a participant specific needs by enabling a third party application (e.g. MS-Word), combined with a selection of network capabilities, as a customization of an authorized team service (e.g. join editing document).

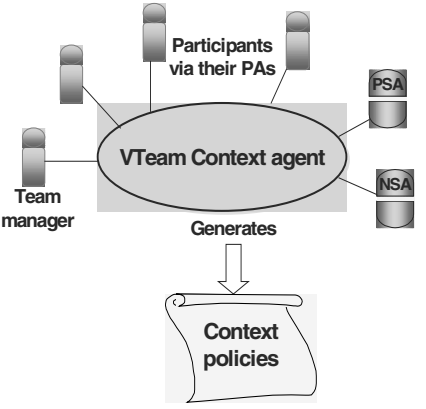

Fig. 4. VTC context information

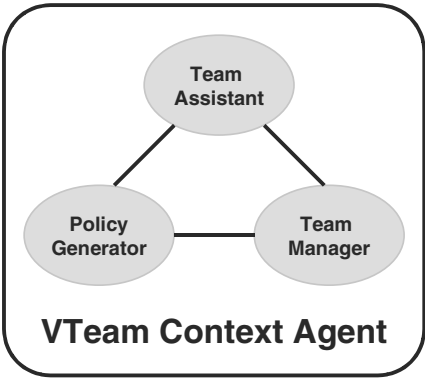

Fig. 5. VTC components

As shown in figure 5, the VTC agent is composed of three components: policy generator, team manager, and team assistant.

- The *Policy Generator* implements and interprets context information. The interpretation consists of translating received and caught notifications from different representation formats (e.g. user interface, ACL messages and events) to a set of policies, called **context policies**. Context policies maintain different types of information. These types are virtual team's attributes (members and roles), team's activities (team services), scheduling (time, day) and network services (location,

class of services, QoS). These policies are represented and stored in same way as the PSA does.

- The *Team Manager* provides the VTC agent with access to context information. It is responsible for context acquisition, which is achieved either by using user interfaces (e.g. team creation and personal preferences) or by interacting with agents that are involved in the virtual team activities.
- The *Team Assistant* uses context policies to trigger events that apply to specific agents according to their state and the context of their use. At setup phase of a collaborative session, the team assistant manages a negotiation process, that guides PAs agents to persuade each other for making decisions on users' behalf (e.g. when to "meet", duration, frequency, ...etc). The team assistant monitors the negotiation and helps agents to achieve a join agreement in conformance to enabled policies.

3.2 Policy Service Agent

3.2.1 Policy Definition

A policy is a set of rules reflecting an overall strategy or objective, affecting the behavior of agents and thus designed to help control and administer a system. A policy rule is a set of actions to be performed by a subject agent on a target agent providing some conditions are satisfied and/or some events are triggered.

The conditions, events and actions are all related to one or more agents of the system. Conditions are typically based on the state of an agent or a resource in the system. The events are triggered either by a time-period condition, a change in the state of an agent or as a result of an action (i.e. before/after events).

Policies could be applied through several levels of a system. From the low network level where policies monitor network devices to the organizational level where they define the role of members of a team. Policies are also useful for resource management and service monitoring.

3.2.2 Policy Management System (PMS)

In the architecture of the PMS (Fig.4), two agents are of particular interest, the Policy Management Agent (PMA) and the Policy Service Agent (PSA). These two agents are interacting and cooperating through the Policy Information Base.

- **Policy Management Agent** (PMA): the purpose of the PMA is to assist the administrator of the application through the policy editing process. PMA is also responsible for detecting static conflict between policies that may occur during the editing process. Policies are stored in the Policy Information Base under the management of the PMA. The PMA is application-independent. Information about the application is stored in the application profile within the PIB. Application profile is a set of attributes (e.g. application structure and components) used by the PMA to communicate with the applications.

- **Policy Service Agents** (PSA): each application (V-Team in our case) is associated with a Policy Service Agent that is responsible for locating the events and conditions likely to trigger a policy through the Event Listener and Constraint Manager. To enforce a particular policy, the PSA invokes a set of actions to be executed. For each triggered policy, the PSA keeps track, in a log file, of the result of policy evaluation and enforcement.
- **Policy Information Base** (PIB): a database of information about the application storing information about agents in the application, the services they provide and the resources they manage along with the policies monitoring their behavior.

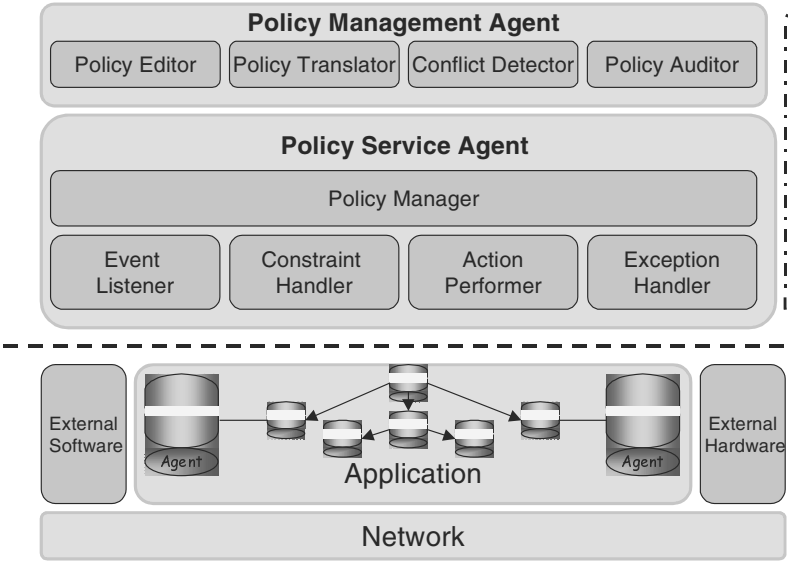

Fig. 6. Policy Management System Global Architecture

4 Basic Team Services Implementation

We have implemented V-Team components as Java-based agents on the top of FIPA-OS platform, and we have defined appropriate interfaces for linking and setting up existing basic services. Such services consist of SIP proxy and SIP User Agent (SIP-UA) developed by Columbia University [8] and RAT audio conferencing and WBD shared workspace tools developed by the Networked Multimedia Research Group at University College London [9].

SIP is a protocol for initiating interactive communication multimedia sessions between users. It provides mechanisms so that user agents and proxy servers can determine the location of participant and perform call setup including SDP for media description. We have developed a wrapper agent (W-SIP) that interfaces SIP-UA to the

NSA for automatic team participant's availability and invitation. An example of SIP-UA call invitation is shown in figure 7. *Karm* team member is invited, via the proxy server (i.e. *spica.uottawa.ca*), to participate in Advanced Networks team meeting (i.e. *subject*), with audio conferencing and shared workspace (i.e. *Media*). W-SIP sets up SIP-UA parameters and waits for the invitation status and SDP description that are then sent to the NSA using ACL messages.

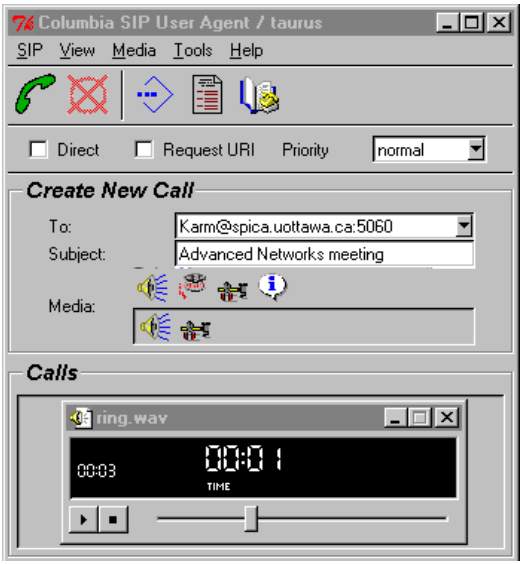

Fig. 7. An example of SIP-UA call setup

In the following, we provide a scenario that describes the V-Team approach and, in particular, highlights features that have already been implemented as a prototype.
Since the behavior of the system is aimed at being completely monitored by policies, the administrator, using the PMA, defines policies that determine who can create virtual teams and what logical resources are available to those team leaders.

The virtual team creation is customized according to the team leader privileges (e.g. resources assigned to the team, media and QoS to be assigned). The team leader plays the role of the team controller, but can assign other team members as team controllers. The team controller has access to the TCA agent, which provides collaborative session facilities.

When creating the team, the team leader can either set up scheduled team services (i.e. team meeting) or delay these to be done later by a team controller. Team management is performed by assigning policies to the system agents, as shown in figure 8.

In figure 9 one can see an example of customized TPA user interface (i.e. *Allen* participant). It displays, on the left, two virtual places (*Advanced Networks and Architecture Engineering*), since *Allen* participates in two teams. *Allen* can collaborate in

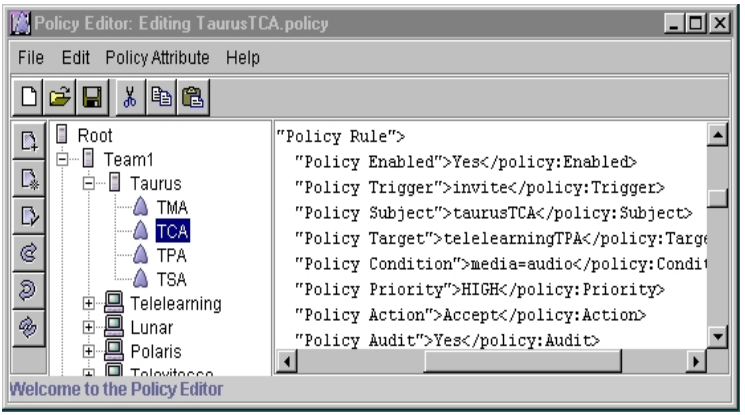

Fig. 8. A sample of editing TCA policies

one or another by selecting the corresponding virtual place. A virtual place corresponds to the pair virtual team and session. It represents a place where virtual team members can "meet" and work on a certain topic.

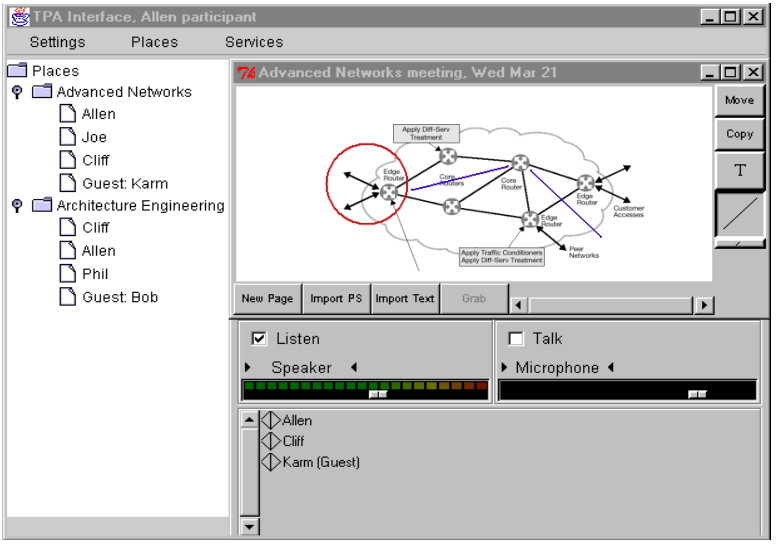

Fig. 9. TPA interface with white board and audio tools

The right part of figure 9 shows an on going session among *Advanced Networks* virtual team members (*Allen*, *Joe*, *Cliff* and *Karm*). In this example, members collaborate using integrated shared-workspace and audio-conferencing tools that are WBD and RAT respectively, however, there is one member (i.e. *Joe*) that has been invited but presently is not participating in the audio conferencing (e.g. Joe's terminal capabilities, authorization policies).

TPA interface provides also other options, top of figure 9. A member uses these options to specify personalized preferences (*Settings* option), create new virtual places or enter existing ones (*Places* option) and access authorized team services such as virtual team creation, scheduling and support for joining or leaving an on going collaborative session (*Services* option).

5 Related Work

Many collaborative environments providing a shared team workspace with common tool-set and shared applications have been developed. These systems generally attempt to single-handedly provide necessary services to serve a large broad user base, and tend to use the network only for transport. In contrast, V-Team is strongly motivated by providing different users with different levels of reliability and awareness, and by enabling the integration of a variety of network services to satisfy the unique needs of diverse classes of virtual teams.

V-Team has certain similarities to mSTAR environment [10]. mSTAR features an agent-based architecture and separates application agents from the network agent, which makes the collaborative application more adaptable to existing network conditions. Despite these characteristics, mSTAR is primarily a framework that supports developers in creating distributed and real-time multimedia applications.

NCSA Habanero [11] is also an approach that supports the development of collaborative environments. Habanero lets developers create and convert Java applets into multi-user applications, known as “Hablets”. DISCIPLE [12] lies mainly in using Java event delegation model in conjunction with JavaBeans applications. DISCIPLE, like Java Collaborative Environment (JCE) [13], extends the Java-AWT to intercept events. These approaches have the cost of altering the application code in order to make it collaborative. The use of policies, combined with context-aware agents, enables V-Team to dynamically combine a collection of third party applications with a selection of network capabilities, and off-loads the need of dealing with these capabilities directly.

6 Conclusion and Future Work

In this paper, we have presented an agent-based multimedia collaborative environment to control and coordinate resource and service sharing in dynamic virtual teams. The proposed approach aims at more of a teamware approach, where the team context is a key factor that extracts, interprets and uses context information.

Since the structure and context of a virtual team may change significantly across time and space during its life cycle, the use of policies in managing dynamically the team behavior at different levels is of the highest relative value. We have introduced a policy management as a service that guides a team manager in the process of defining, enforcing and auditing policies.

The system uses the session initiation protocol as a basis for locating team members, and uses existing multicast-based media applications (i.e. RAT, and WDB) as basic tools for multimedia conferencing service. Future work includes integration of various network services, which will contribute to off-load collaborative services from large transport-related capabilities, resulting in a broader range and potential customization of these services.

Our current focus is on extending the virtual team context with more capabilities in collecting context information and adapting the system behavior accordingly. The use of policy-based agents that allows each agent to monitor its own policies is important in our work as well.

Acknowledgement. This work was supported partly by Nortel Networks and Natural Sciences and Engineering Research Council of Canada.

We would like to thank Allen Eddisford and Stephen Ng from Nortel Networks for their valuable feedback and technical suggestions on this work.

References

1. J. Lipnack and J. Stamps, "Virtual Teams, Reaching Across Space, Time and Organizations with Technology," John Wiley & Sons 1997.
2. T. Magedanz, A. Karmouch, "Mobile Software Agents for Telecommunication Applications," Journal of Computer Communication, Vol. 23, Issue. 8.
3. H. Harroud, M. Ahmed, A. Karmouch, " Agent-based personalized services for mobile users over a VPN," International Conference on Enterprise Information Systems, Setúbal, Portugal, July 2001.
4. M.N. Huhns, "Agent Teams: Building and Implementing Software," IEEE Internet Computing, February 2000.
5. IETF SIP Working Group. SIP: Session Initiation Protocol, Internet Draft. November 2000
6. FIPA-OS V1.3.3 Distribution Notes, Nortel Networks Corporation, 2000, available at http://www.nortelnetworks.com/fipa-os.
7. IETF Policy Framework Working Group, "Policy Core Information Model Specification," Internet Draft, October 2000.
8. sipd - SIP redirect, proxy and registration server, sipc - SIP user agent. Software licensing available at http://www.cs.columbia.edu/~hgs/license/
9. The UCL Networked Multimedia Research Group, Mbone Conferencing Applications, available at http://www-mice.cs.ucl.ac.uk/multimedia/software/
10. P. Parnes, K. Synnes, and D. Schefstrom, "mStar: Enabling Collaborative Applications on the Internet," IEEE Internet Computing, October 2000.
11. A. Chabert et al., "Java Object-Sharing in Habanero," Communications of the ACM, Vol. 41, No. 6, June 1998.
12. I. Marsic, "DISCIPLE: A Framework for Multimodal Collaboration in Heterogeneous Environments," ACM Computing Surveys, Volume 31, No. 2, Article No. 4, June 1999.
13. H. Abdel-Wahab et al, "An Internet Collaborative environment for Sharing Java Applications," IEEE Computer Society Workshop on Future Trends of Disrtibuted Computing Systems, pp. 112-117, October 1997.

14. J. R. Nicol, Y. S. Gutfreund, J. Paschetto, K. S. Rush, and C. Martin, “How the internet Helps Build Collaborative Multimedia Applications,” Communications of the ACM, Vol. 42, No. 1, January 1999.
15. J. Altmann, R. Weinreich, “An Environment for Cooperative Software Development: Realization and Implications,” Proceedings of the Hawaii International Conference on System Sciences, Kona, Hawaii, January 1997.
16. K. Sikkel, “A Group-based Authorization Model for Cooperative Systems,” Proceedings European Conference on Computer-Supported Cooperative Work (ECSCW’97), Lancaster, September 1997.
17. T. Ishaia and L. Macaulay, “ The Role of Trust in Virtual Teams,” Proceedings of the 2nd International VoNet-Workshop, Zurich, September 1999.
18. V. Pham and A. Karmouch, “Mobile Software Agents: An Overview”, IEEE Communications, Special Issue on Mobile Agents and Telecommunications, Vol.36, No.7, pp.26-36, 1998.
19. M. Sloman, J. Labo, E. Lupu, “Policies for Distributed Systems and Networks,” International Workshop, Bristol, UK, Jan. 2001.
20. B.N. Schilit, N.I. Adams, R. Want, “Context-Aware Computing Applications,” Proceedings of the Workshop on Mobile Computing Systems and applications, IEEE Computer Society, Santa Cruz, CA, pp. 85-90, 1994.