

ADVANCED COLLABORATIVE SERVICES PROVISIONING IN NEXT GENERATION NETWORKS

Miguel Gómez¹, Tomás P. de Miguel², and Fermín Galán¹

¹*Agora Systems S. A., C/ Aravaca 12 1ª, 28040 Madrid, Spain;* ²*Universidad Politécnica de Madrid, E.T.S.I. de Telecomunicación, Ciudad Universitaria s/n, 28040 Madrid, Spain.*

Abstract: The 3GPP IMS subsystem, adopted also by the ITU-T/ETSI NGN initiative, proves to be a powerful framework for the provision of conferencing services. Collaborative Services, although based on multipoint conferencing, present their own set of requirements and distinguishing features. This paper tries to present an overview of the conferencing framework available in IMS-based networks, in order to analyse next the requirements and necessary extensions for advanced collaborative service provisioning in this environment. Finally, an architecture proposal for an IMS-compatible collaborative server is presented, and the undertaken validation experiments and prototypes tackled.

Keywords: collaborative services; CSCW; IMS; next generation networks

1. INTRODUCTION

Collaborative Services attempt to bring people together by offering a diverse set of communication and information sharing features that enable the effective cooperation of geographically distributed teams. Advanced Collaborative Services go one step further by trying to emulate real-life cooperation environments (e.g. conferences, meetings, classes, business presentations, etc.) and make them available to a distributed audience by providing an enhanced multipoint conferencing framework. These services have been traditionally relegated to large corporations or educational institutions due to the low availability of both the network and terminal requirements necessary for their operation.

The adoption of packet-switched core networks by mobile operators and the deployment of SIP-based multimedia provisioning platforms, like the UMTS IP Multimedia Subsystem (IMS) standardised by 3GPP, promises to change this environment in the next few years. Besides, the adoption of the IMS standard by the ITU-T/ETSI NGN initiative will make this framework available also in fixed networks, extending the scope of this platform and making even more interesting to design services that can be deployed on it.

This paper tries to provide an overview at the IMS multimedia framework and multipoint conferencing model, in order to analyse how it could be extended to support the provisioning of IMS-compatible Advanced Collaborative Services.

2. CONFERENCING IN IMS-BASED NETWORKS

One of the basic features provided by the 3GPP IMS is to allow users to create, modify and terminate SIP-based multimedia sessions [Rosenberg et al., 2002]. Although these basic mechanisms already enable users to setup multipoint conferences on their own, the IMS network architecture comprises a set of dedicated elements and functions designed to allow more advanced conferencing scenarios and to facilitate an easier and more efficient conference setup. This section will provide an overview to the IMS architecture, focusing on the nodes relevant to multipoint conferencing services, as well as to the centralised conference model adopted in NGN and the set of protocols which is currently being designed at IETF for its control and management.

2.1 Subsystem Architecture

The IMS [IMS, 2005] is a subsystem included in the UMTS Packet-Switched Core-Network since Release 5 for the provision of IP-based multimedia services.

Figure 1 summarises the IMS elements relevant to conferencing [CN, 2005], as well as the interfaces between them. As we can see, all the native IMS element groups need to be combined for advanced conferencing services provisioning. The involved nodes are:

- **Serving Call Signalling Control Function (S-CSCF):** The S-CSCF is the node in charge of session management in the IMS network. It behaves as a stateful SIP proxy providing registrar, authentication and session control features. The S-CSCF is always located in the operator's home network, and it constitutes the central control point for operator-provided services by applying the filter criteria for the execution of SIP-based IMS services. Filter criteria are just a set of rules defined per user which enable to trigger services upon information contained in the SIP header fields.

- SIP Application Server (SIP AS): SIP Application Servers are the key elements for the provision of native IMS services, taking the roles of SIP proxies, UAs, or B2BUAs. The S-CSCF, upon a correct match of a service filter criterion, will forward the SIP request to appropriate SIP AS, which will in turn perform the associated service logic.
- Multimedia Resource Function (MRF): The MRF is the IMS element in charge of providing the media-related functions for the provision of advanced services, comprising features such as announcement playback or media mixing for multipoint conferencing. It is divided into two logical functions:
 - Multimedia Resource Function Processor (MRFP): The MRFP is the actual media processing function, in charge of manipulating media streams in order to perform tasks like media transcoding, mixing, filtering, etc.
 - Multimedia Resource Function Controller (MRFC): This is the entity in charge of MRF control. It behaves as a SIP User Agent, accepting SIP requests coming from the S-CSCF or a dedicated SIP Application Server. It controls the MRFP via the MEGACO-based Mp interface. The border between the AS and the MRFC is quite diffuse, since services may be provided just by an AS controlling directly the MRFP, by the MRFC acting as SIP AS and receiving directly requests from the S-CSCF, and by a combination of both, with a SIP AS behaving as SIP B2BUA or proxy and the MRFC acting as SIP UA.

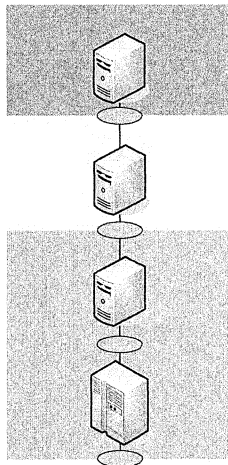


Figure 1. IMS elements for multipoint conferencing.

2.2 Conference Model

Once we have described the functional entities for conferencing in the IMS, we may take a closer look to the supported conference model [Rosenberg, 2005] [Barnes, 2005].

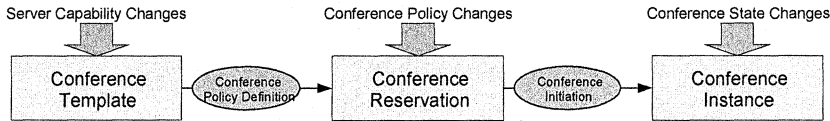


Figure 2. The Conference life cycle.

Figure 2 depicts the life cycle of a generic conference. First of all, every conferencing server (MRFC/AS + MRFP combination) must provide a set of conference templates describing the available service features, so that conferencing clients (UEs) may define a conference by filling the appropriate data or choosing from the different options provided in the template. Since conference templates are tightly coupled with the conferencing features available at the server, a change in the server capabilities must comprise a subsequent change in the set of offered templates (or the addition of a new template to the set) so that users may take advantage of these new capabilities.

Users create a conference reservation by filling the appropriate template in order to define the policy for a particular conference. We may describe Conference Policy as the set of parameters and rules that govern the conference behaviour. Conference Policy may be changed as many times as desired during the conference lifetime, even once the conference is active. A conference reservation is a policy definition for a conference that has not started yet.

Once a conference starts, the conference reservation becomes a conference instance. Conference initiation also comprises the creation a SIP signalling user agent, named conference focus, associated to the SIP URI which univocally identifies the conference instance. The SIP focus allows the exchange of call control signalling between the conferencing server and the conferencing clients. Only for the fact of being instantiated, a conference also acquires a certain Conference State, which may be defined as the set of dynamic information that describes a conference instance in progress. Conference State, due to its dynamic nature, may change many times during a conference:

- Due to 1st party call control signalling (e.g. a new participant dials into the conference).
- Due to 3rd party call control signalling (e.g. a user requests the conference focus to invite a new participant using SIP 3PCC – e.g. REFER – mechanisms).

- Due to policy changes (e.g. a conference policy update bans users belonging to domain “unwanted.com”, so “user@unwanted.com” is expelled from the conference).
- Due to direct conference state changes (e.g. a user requests the conferencing server to invite a new user to the conference using the conference state control features provided).

Conference state is maintained at the conferencing server, but clients are notified periodically of conference state changes so that they can keep a consistent local copy of the conference state parameters relevant to them.

An additional optional feature that may be provided by conferencing servers is Floor Control. A Floor may be defined as a set of resources belonging to a particular conference instance to which a user is granted access or modification rights, either exclusive or not. A Floor example might be the right to speak in a full-mesh audio-conference. Floors and Floor policy are part of Conference Policy, but Floor Control provides the mechanisms for managing floor requesting, releasing, granting and revoking within an active conference [Camarillo et al., 2005].

The conference model standardised by the 3GPP in UMTS Release 6 does not consider direct conference control features, and therefore conference policy is either implicit or controlled via proprietary mechanisms and all conference state changes are triggered through call control signalling. However, conference control is currently an active research topic within the IETF Centralised Conferencing (XCON) Working Group, and the 3GPP has expressed the necessity of this kind of features and its intention to include them in future releases of the UMTS specifications.

2.3 Conference Control framework

In order to implement the conference model described in the previous section, a set of different protocols is being object of study and standardisation both by IETF and 3GPP. In this section we will try to summarise the most relevant aspects of conference control, as well as to present the set of protocols and signalling mechanisms that are being designed in order to fulfil the previously presented service picture.

2.3.1 Call Control Signalling

As explained in section 2.2, each conference is univocally represented by a SIP URI. Conference initiation comprises the creation of a SIP signalling User Agent, named the conference focus, associated with that URI. The conference focus is in charge of the SIP signalling exchange between the conferencing server and the clients in order to implement the SIP-based conferencing framework defined in [Rosenberg, 2005]. As described in the conference model, clients may influence the

conference state either using SIP first-party or third-party call control, and the new conference state is notified to clients using a specific event package [Rosenberg et al., 2005] of the SIP notification framework [Roach, 2002]. These SIP-based conferencing call control features are defined in [Johnston and Levin, 2005] and, as we have previously described, they are the only conference control features offered by the IMS multipoint conferencing framework.

A key element for this SIP-based conferencing framework is the ability of Conference-Aware clients to identify that their interlocutor is a conference focus and behave in consequence. The “isfocus” feature tag defined in the SIP User Agent capabilities discovery framework [Rosenberg et al., 2004] is used to indicate that a SIP URI belongs to a focus entity and, therefore, the SIP dialog belongs to a conferencing session. The mechanism is fairly simple; it consists on appending the “isfocus” tag to the standard contact address header field contained in the SIP requests and responses sent by the focus (e.g. “Contact: <sip:3402934234@example.com>;isfocus”). A Conference-Aware user agent may be defined as a SIP User Agent which recognizes the “isfocus” feature parameter, is able to process Conference State notifications, and can perform and handle SIP third-party call control signalling.

2.3.2 Conference Policy Management

The Conference Policy Control Protocol (CPCP) [Khartabil et al., 2004] has been designed to enable conferencing clients to define and manipulate Conference Policy. It defines an XML Schema that enumerates the conference policy data elements that enable a user to define the settings and behaviour for a particular conference.

Users may create a conference reservation by placing a new conference policy document at the server. Depending on server policy and user privileges, the server may accept or reject the reservation. Once a reservation has been made, it can be modified as many times as required, even once the conference has started. A conference can be deleted permanently by removing the conference policy document from the server, which consequently frees the allocated resources.

In many cases, the creator of the conference policy will be the only user with access rights to the conference policy file. This implies that other users might take part in the conference, but only the conference creator will be able to read and manipulate the conference policy. Although this is the default behaviour, in many other cases it may be convenient to allow other users to read and/or modify different parts of the conference policy document. The Conference Policy Privileges document format defined in [Khartabil and Niemi, 2004] provides the means to set and regulate those privileges. A certain conference creator may override the default policy rules by uploading a conference policy privileges document along with the conference policy file.

The CPCP protocol specification is transport-independent; therefore the conference policy and conference policy privileges documents may be uploaded, read, modified and deleted from the server using any mechanism. However, a key feature in a conferencing framework like the one presented above is to provide an efficient and reliable mechanism for performing those actions. The protocol of choice for CPCP transport has been The XML Configuration Access Protocol (XCAP) [Rosenberg, 2005].

XCAP is a protocol intended to allow the manipulation of XML data resident on a server. By extending the XPath notation to HTTP URLs, XCAP allows to identify univocally with an URL any element from any XML file located on the server, from the whole document itself to any element or attribute on it. The invocation of standard HTTP methods allows clients to read (GET), add or modify (PUT) and delete (DELETE) any possible XML resource identified with the URL syntax presented above.

Since XCAP is a generic protocol for remote XML manipulation, each particular application requires the definition of the application-specific conventions that will apply. These conventions normally consist on the XML schema that defines the structure and constraints of the data to be managed. The XCAP usage for the manipulation of Conference Policy and Conference Policy Privileges is defined in [Khartabil, 2004].

2.3.3 Conference State Management

The Centralised Conference Control Protocol (CCCP) [Levin et al., 2005] allows conference participants or any other authorised entity to directly manipulate the state of an active conference. CCCP is based on a set of transactions issued over a reliable transport protocol. Each transaction consists of a request carrying the required information in an XML body and a corresponding response or set of responses (any number of provisional responses plus a final one) carrying an appropriate XML body. Opposed to CPCP-based Conference Policy manipulation through XCAP, CCCP-based conference state manipulation does not imply direct modifications on the conference state XML file. Clients issue CCCP requests containing the desired conference state after the change is applied, and the server will process these requests according to its own policy and the role of the requesting agent. When a request is processed, its result is notified to the requester, and all conference participants are informed of the new conference state by the notification mechanism defined in section 2.3.1.

CCCP syntax is based on the conference information schema type and its subtypes described in [Rosenberg et al., 2005], plus the necessary extensions to support the CCCP request primitives: conference state retrieval, addition, modification and removal, user retrieval, addition, modification and removal, etc.

Although no preferred transport mechanism is yet chosen for CCCP, transaction-based object-oriented approaches such as SOAP, XML-RPC, or similar are the most likely options for its implementation.

3. MODEL EXTENSION FOR COLLABORATIVE SERVICES PROVISIONING

Computer-Supported Cooperative Work (CSCW) is a vast research field comprising disciplines as heterogeneous as computer-science, telecommunications, psychology, and sociology. Its main goal is to develop tools and techniques that enable to ease and facilitate collaboration for workgroups thanks to computer networking.

Groupware is one of the physical concretions of CSCW, as it may be defined as the software designed to enable the collaboration of physically distributed teams by means of network communications. Groupware may provide many different collaboration models, from asynchronous communication via messaging and web-based repositories to multimedia-based real-time collaboration. The cooperation facilities offered to users by the combination of groupware and networking features may be defined as Collaborative Services. Advanced Collaborative Services are usually based on multipoint videoconferencing features, but they are habitually oriented to the provision of professional or academic services such as distributed lectures or classes, business meetings or presentations, etc. Therefore, they present the following particularities:

- **Media rich conferencing:** The lack of face-to-face interaction is minimised through high-quality and media extensive communication.
- **Complex moderation mechanisms:** The lack of social mechanisms to regulate interventions needs to be overcome by moderation mechanisms such as user roles, turn management, floor control, etc.
- **Information sharing features:** Opposed to conferencing, Collaborative Services most of the times place the value not on the conversation itself but on the work resulting from it (e.g. a document has been produced). Additionally, the shortcomings of remote collaboration need to be compensated through additional reinforcement mechanisms such as drawings, schemas, slides, etc. Therefore, information-sharing features such as file distribution, application sharing, collaborative document edition, shared whiteboards, etc. are key components in this kind of services.

In this section we will try to outline the requirements for Collaborative Services provisioning in IMS-based networks, as well as the necessary modifications and extensions to the conferencing model previously presented in order to meet these requirements.

3.1 Requirements for Collaborative Services Provisioning in Next Generation Networks

As already described in this section's introduction, one of the key elements of Advanced Collaborative Services is media-rich interaction. If we want to substitute efficiently a face-to-face meeting we must provide the multimedia mechanisms to make communication fluent and meaningful, imitating when possible the interaction mechanisms found in real life collaboration.

In order to provide this kind of interaction, we cannot constrain media exchange to the symmetric model habitually found in conferencing environments, in which the central mixing element (MRFP) processes (i.e. mixes) media so that each participant just receives a downstream flow per each of his tributary upstreams. Advanced Collaborative Services require a more complex behaviour of the MRFP, providing different asymmetric media combinations at different moments of the session (e.g. in a lecture, participants will just receive the professor's video and audio during the presentation, but during the questions they will receive in addition the video and audio of the alumnus issuing the question).

Additionally, users should be able to provide descriptive information about themselves (e.g. name, institution, etc.) to be distributed among participants, enabling thus easier identification between people that have never met in person before.

Another key element to reflect real life interactions as close as possible is the definition of interaction modes and enabling fast interaction mode switching within the collaboration session. We may define an interaction mode as the particular combination of some of the available sub-elements of the collaborative service in order to offer the user a certain interaction experience. Example interaction modes would be the ones previously presented for the lecture scenario: the collaboration activity would be a distributed class, and the interaction modes would be the distributed presentation mode, which involves the professor giving the lecture through audio, video and maybe a whiteboard or a slides presentation, and the questions mode, which would involve the audio and video of both the professor and one alumnus at a time.

One problem naturally associated with interaction modes is the ability to deal with heterogeneous client terminals and networks. A desirable feature in Collaborative Services is the consistent interaction mode model presented above, in which all participants receive the same media and present it in a similar manner. But in an environment as NGN in which we will deal with a huge variety of terminals (mobile phones, PDAs, PCs, etc.) and networks (GPRS, xDSL, WLAN, etc.), we cannot assume that all clients will be able to deal with the same amount of media and present it in a similar way. Therefore, another necessary feature is to provide media with associated semantic information, which would enable clients to decide which media to accept or not when negotiating the session and how to render it

locally. Examples of this kind of attributes would be if video/audio is principal or auxiliary; if video can be resized, suggested video sizes, etc.

Collaborative Services must also overcome the lack of natural moderation mechanisms in distributed collaboration, since the social mechanisms for regulating interventions found in face-to-face conversations are lost in this environment. Therefore, Collaborative Services must compensate this handicap with a complex moderation system. Roles are a key feature in this kind of services, and permissions should be more related to roles than to actual personal identities. Continuing our lecture example, we would have just two roles: Lecturer and Alumnus. It is clear that alumni should not be able to speak or write in the whiteboard while the lecturer is presenting. They should just request the permission to ask a question, which should be granted by the lecturer only. Instead of defining those permissions to the alumni one by one based on their SIP identity, it makes more sense to assign user roles and provide per-role permissions.

The example above shows also that Floor Control is not the only necessary moderation element in advanced collaborative systems. Mechanisms for Conference State Control moderation must also be provided, since not only is important to decide who can or cannot apply certain changes, but also when are those changes to be applied (e.g. Alumni can request to change to questions mode, but this change needs to be accepted by the professor first).

3.2 Required Model Extensions

In order to achieve the advanced media control features and the interaction mode model defined in the previous section, the media control framework for conferencing needs to be enhanced. The current proposal for media control in centralised conferencing is based on Media Policy [Jennings and Rosen, 2005]. This Media Policy framework allows conferencing servers to express their capabilities through a set of media policy templates. Conference creators can then select the media configuration for their conference by filling the desired template through CPCP/XCAP. Conference participants can learn also the media policy through CPCP/XCAP in order to control the media server (MRFP) within the declared limits.

This model assumes that media configuration will be maintained during all session, as it is part of the conference policy. Changing media settings on an ongoing conference would imply to upload a new conference media policy using CPCP/XCAP, which is inefficient and collision-prone if changes are going to be quite common. Besides, the CPCP/XCAP model is pull-based, and therefore clients would not be notified of changes in the media configuration.

In order to support multiple interaction modes within the same conference and fast switching between them, the Media Policy model should be enhanced to enable to define several possible media configurations for the conference, selecting also the

default or initial one. These settings should also be included as part of the Conference State, enabling thus all clients to receive them and keep updated via the Conference State notification service. Interaction mode changes could then be triggered through CCCP, affecting only the conference state but not the pre-defined media policy for the conference. If the interaction mode change implies also a change in the kind or number of media involved in the session, the conference focus should be notified also in order to re-invite the participants accordingly for media re-negotiation.

In order to be able to cope with the variety of terminals and networks available in NGN environments, the conferencing model needs also to be extended in order to add semantic information to media definitions. This meta-information can then be used by clients in order to select which media accept or not when negotiating or re-negotiating collaborative sessions, and decide later how to render the available media locally in order to provide the best possible user experience. The conference media streams section of the CPCP conference policy document model could be enhanced in order to provide this kind of information, which could then be notified to the SIP conference focus upon conference activation in order to be included in the SIP/SDP session negotiation via SDP media-level attributes.

Additionally, the Conference Dial-Out and Refer lists of the CPCP conference policy document could be enhanced to include a more complex user-definition model that enables to convey additional information for well-know users invited to the conference, both of personal (e.g. name, company, etc.) and service-related (e.g. user role) nature. This information could then be mapped into the conference state, so that it gets notified to all users and spontaneous users can fill personal data through CCCP. Role assignment for non-predefined users may be controlled via the Authorisation Rules in the conference policy (e.g. all users from the “university.edu” domain are allowed to join and their role is “Alumnus”). Authorisation rules can then be defined according to user roles when necessary, instead of using individual identities.

Finally, in order to achieve the advanced moderation scheme for conference state control required for Collaborative Services provisioning, some kind of moderation protocol for CCCP needs to be implemented. A possibility for implementing this moderation framework would be to define a SIP event package for Conference State moderators, so that they can be notified when a new CCCP request arrives. Petition grating or revocation commands could be implemented over the same transaction-oriented protocol used for CCCP transport (e.g. SOAP). This framework would also need to be integrated into the conference policy framework, so that Conference Policy may regulate which parts of the conference state are moderated and which users or user roles should act as moderators.

4. PROPOSED SYSTEM ARCHITECTURE

Figure 3 depicts the proposed architecture for a Collaboration Server suitable for the conferencing framework to be adopted in IMS networks and alike (e.g. NGN).

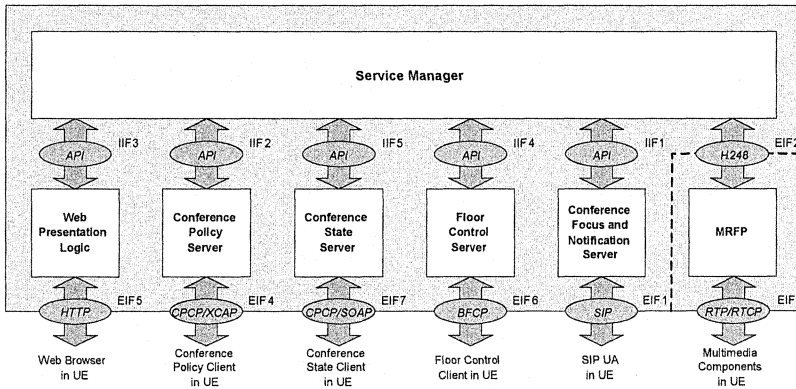


Figure 3. Collaboration Server (MRFC/AS + MRFP) Architecture.

As shown in the figure, the server presents the following sub-components:

- **Multimedia Resource Function Processor (MRFP):** This component performs the required media transformation and processing actions for collaboration support, such as mixing, switching, forking, transcoding, etc. It mainly exchanges RTP-based information with clients, but other formats are possible since they are highly dependant on the components being used in the collaboration session. It communicates with the Service Manager via a MEGACO (i.e. H.248) interface. This architectural decomposition allows to locate this element in a separate node if desired, following thus the MFC decomposition into MRFC and MRFP suggested by 3GPP.
- **Conference Focus and Notification Server:** This element behaves as a SIP signalling UA, allowing to exchange call control signalling with the clients. It also behaves as SIP Notification server for the supported event packages (e.g. Conference event package, conference state moderation package, etc.), accepting subscription requests from the clients and delivering the associated notifications when an event occurs.
- **Conference Policy Server:** This element allows clients to manage Conference Policy through XCAP-transported CPCP transactions.
- **Web Presentation Logic:** This component allows clients to access to the Conference Policy, Conference State, and Floor Control features offered by the server through a web-based interface, and it also presents the information otherwise conveyed through the Notification Service. It is intended to allow standard SIP UAs to take part in the collaboration service by overcoming through this additional interaction interface the lack of capabilities of their local

software or user terminals. It may interact directly with the Service Manager through a dedicated API (shown in Figure 3 as IIF3) or interface through the Conference Policy Server, Conference State Server, Floor Control Server and Notification Server by implementing the appropriate interfaces (EIF1, EIF4, EIF6 and EIF7).

- Floor Control Server: This component allows clients to manage conference floors through TCP-transported BFCP transactions.
- Conference State Server: This element allows to manage Conference State through CCCP transactions transported over an object-based transaction-oriented protocol (e.g. SOAP).
- Service Manager: This element constitutes the highest-level control logic of the server. It handles the conference data objects and ensures a consistent conference state by coordinating the different sub-components and enforcing Conference Policy.

Several types of clients are possible in this architecture, depending on their particular combination of the available framework features: standard SIP UAs, UMTS R6 clients, collaborative clients implementing one or several of the conference control protocols described in section 2.3, etc.

5. EXPERIENCES AND PROTOTYPES

Since developing an Advanced Collaborative Services provisioning framework as the one presented in the previous sections is a major task, a layered incremental approach has been undertaken in order to enable to validate individually all the required elements as soon as they are ready, and to deploy and test increasingly complex services within the available framework.

Therefore, in a first phase a MEGACO-controlled MRFP has been developed following the requirements outlined in section 3.1, implementing a plug-in based architecture that enables the incremental addition of further codecs, media types and transformation functions. In addition, a first attempt of SIP focus and Service Manager has also been developed, allowing the SIP-based conference management interactions for Conference-Unaware UAs described in [Johnston and Levin, 2005]. Several basic adaptive multipoint conferencing services have been tested within this framework, using as clients standard SIP UAs such as Microsoft's Windows Messenger or one Agora's own videophone applications.

In a second phase we plan to enhance this framework by adding CPCP-based policy control features to the server, as well as a web-based interface for non-CPCP enabled clients. The SIP logic will also be enhanced to support Conference-Aware clients, and the SIP notification service will be implemented. We intend to validate this framework with a set of heterogeneous clients, adding to the legacy ones

previously described new Conference-Aware clients for PC, PDA and mobile phone.

In a third and final phase, the server will be enhanced with floor and conference state control features, developing also a compatible client implementing all the necessary reciprocal logic.

6. CONCLUSIONS AND FURTHER WORK

Collaborative Services had until now been adopted by reduced user groups only, since their extensive requirements of bandwidth and system capabilities made them suitable just for large organisations or academic institutions with strong remote collaboration commitment. The development of the Next Generation of SIP-based Collaborative Services and the standardisation of the SIP-based IMS framework by the 3GPP is both a great opportunity for this kind of services, which will finally find widespread the networks and clients required for their operation, but also a great challenge since it is also necessary to adapt service provisioning to this new framework and the available subsystem elements.

The IMS conferencing architecture standardised by the 3GPP for UMTS Release 6 and developed at the IETF SIPPING and XCON working groups proves to be a powerful yet flexible framework for the deployment of multipoint conferencing services. However, although it composes a great base for the development of Advanced Collaborative Services, those services comprise a set of requirements that needs to be taken into consideration in the framework. The scalable and extendable nature of the IMS conferencing framework and its proximity to the Internet world allows to easily include the required enhancements, maintaining compatibility and high interoperability levels with both the set of clients available now and the ones expected in the future.

REFERENCES

- Barnes, M., Boulton, C. and Levin, O. (2005). A Framework and Data Model for Centralized Conferencing, IETF draft-ietf-xcon-framework-01, July 2005, Work in progress.
- Camarillo, G., Ott, J. and Drage, K. (2005). The Binary Floor Control Protocol (BFCP), IETF draft-ietf-xcon-bfcp-05, July 2005, Work in progress.
- CN (2005). 3rd Generation Partnership Project, Technical Specification Group Core Network, Conferencing using the IP Multimedia (IM) Core Network (CN) subsystem, Stage 3 (Release 6), 3GPP TS 24.147 v6.3.0, June 2005.
- IMS. (2005). 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, IP Multimedia Subsystem (IMS), Stage 2 (Release 6), 3GPP TS 23.228 v6.10.0, June 2005.

- Jennings, C. and Rosen, B. (2005). Media Conference Server Control for XCON, IETF draft-jennings-xcon-media-control-03, July 2005, Work in progress.
- Johnston, A. and Levin, O. (2005). Session Initiation Protocol Call Control - Conferencing for User Agents, IETF draft-ietf-sipping-cc-conferencing-07, June 2005, Work in progress.
- Khartabil, H. (2004). An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Conference Policy Manipulation and Conference Policy Privileges Manipulation, IETF draft-ietf-xcon-cpcp-xcap-03, October 2004, Work in progress.
- Khartabil, H. and Niemi, A. (2004). Privileges for Manipulating a Conference Policy, IETF draft-ietf-xcon-conference-policy-privileges-01, October 2004, Work in progress.
- Khartabil, H., Koskelainen, P. and Niemi, A. (2004). The Conference Policy Control Protocol (CPCP), IETF draft-ietf-xcon-cpcp-01, October 2004, Work in progress.
- Levin, O., Even, R. and Hagendorf, P. (2005). Centralized Conference Data Model, IETF draft-levin-xcon-cccp-02, February 2005, Work in progress.
- Roach, A. B. (2002). "Session Initiation Protocol (SIP)-Specific Event Notification," IETF RFC 3265, June 2002.
- Rosenberg, J. (2005). A Framework for Conferencing with the Session Initiation Protocol, IETF draft-ietf-sipping-conferencing-framework-05, May 2005, Work in progress.
- Rosenberg, J., Schulzrinne, H. and Kyzivat, P. (2004). Indicating User Agent Capabilities in the Session Initiation Protocol (SIP), IETF RFC 3840, August 2004.
- Rosenberg, J., Schulzrinne, H. and Levin, O. (2005). A Session Initiation Protocol (SIP) Event Package for Conference State, IETF draft-ietf-sipping-conference-package-12, July 2005, Work in progress.
- Rosenberg, J., Schulzrinne, H. et al. (2002), SIP: Session Initiation Protocol, IETF RFC 3261, June 2002.
- Rosenberg, J. (2005). The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), IETF draft-ietf-simple-xcap-07, June 2005, Work in progress.