Multimedia Conferencing Management Using Web Services Orchestration over Public Networks

Bo Cheng*, Xiaoyan Wu, Xiaoxiao Hu, Chen Junliang
State Key laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing 100876, P.R.China
chengbo@bupt.edu.cn

Abstract

The multimedia conference control and management interfaces are still in an early stage of development, and need to be extended in order to provide a high flexibility, scalability, and process management level that allows the deployment of multimedia conferencing solutions adapted to the user requirements. In this paper, we introduce the multimedia conferencing management using Web service and orchestration approach over public networks. For multimedia conferencing is a converged voice and data applications, which management involves both real-time communications and non-real-time information processing tasks, and mainly focus on the design and implementation for communication Web service and extensible BPEL (Business Process Execution language) based multimedia conferencing communication services process logics and also are encapsulated in a coarse grained Web service for the purposes of flexible multimedia conferencing process management. And also, we give the multimedia conferencing process management system and performance measurement and analysis. Finally, we give the conclusion and future work.

Keywords: Multimedia Conferencing, Process Management, Service Orchestration

1. Introduction

Developing multimedia conferencing over IP networks is a challenging task, since the number of end users is increasing that are able to attend multimedia conference that support diverse set of terminals including mobile phone, fixed phone, and personal computer etc. Furthermore, increased number of users require scalable and flexible multimedia conferencing management system that can deliver thousand or tens of thousands of simultaneous multimedia streams. Therefore, it is particularly difficult to develop a flexible and scalable multimedia conferencing management system that supports high number of users with diverse capabilities [1][2]. Web service is an emerging trend in the industry today for providing distributed Internet information services over IP networks and it relies on some structured XML based SOAP message and service WSDL to access, control and integrate various services remotely for complex transactions. Recent advances in Web services have made it practical to provide communication Web services. Service oriented communication is a new trend in the industry to enable communication through SOA and thereby package communications as services. It is receiving increasing attention, because it is well-suited for new, unified, converged communication initiatives and applications [3][4].

BPEL (Business Process Execution Language, BPEL) [5] is an XML-based language for Web service orchestration language. However, BPEL has not been developed for control of specific, in particular real-time, communication services [6][7][8]. To the multimedia conferencing management using BPEL based Web services orchestration approach over IP networks, where a composed service orchestrates a set of services that interface existing communication capabilities, is a crucial and a challenging problem. In particular, multimedia conferencing management strongly rely on asynchronous communications to handle heterogeneous events generated by network resources, and they usually handle large and dynamically evolving sets of concurrent processes [9][10]. Multimedia conferencing interfaces need to be extended to provide a fine-grained management level that allows the deployment of innovative multimedia conferencing solutions adapted to the user requirements.

Our work confront both with the issue of the multimedia conferencing process management for communication Web services and communication Web service orchestration using extensible standard
languages as BPEL. The remainder of the paper is organized as follows: Section 2, multimedia conferencing process management using Web service orchestration. Section 3, is the multimedia conferencing system and performance analysis. Section 4, conclusions and future work.

2. Multimedia conferencing management using Web service orchestration

The management of multimedia conferencing process involves both real-time communications and non-real-time information processing tasks. Especially, the real-time communication capabilities are encapsulated in a coarse grained Web service with above described method for the purposes of inter-working with other Web services orchestrated with BPEL via multimedia conferencing communication Web service bus. Figure 1 shows the multimedia conferencing management framework using Web service over public IP networks.

![Conference process management: Manages the multimedia conference service logic by Web Service orchestration approach via BPEL.

Web service bus: Acts as a message broker between services interaction, and also provides the capabilities for the messages transformation and routing.

Multimedia conference Web Service: The interface module between the multimedia conference process management and the resource adaptors. The multimedia conference Web service provides the same interface to the BPEL process independently from the resource adaptors used in the implemented solution. In multimedia conferencing process management scenarios, there exists real-time Web service, such as multiple party call control service and short message service, and also exists non real-time Web services, such as user authentication service and account service. Which is the same adopted and defined in the Parlay X approach.

Multimedia conference services enabler: Which provides core network services like call control or presence to higher level services for re-use of the functionality in a more complex application scenario.](image-url)
Services enabler: Allows an easy integration of legacy network technologies and provides an access network independent view on service building blocks. Thus value added services can be well provisioned in a fixed and mobile converged telecommunication environment. Here, the multimedia conference services enabler contains the session management, media controller, and data management modules.

Public networks: Different public networks are involved, such as Telecom networks, including fixed and mobile telecom networks, Internet, 3G network and so on.

2.1. Multimedia conferencing communication Web service model

SIP [11] relies on a set of special SIP operations to establish the communication session and set up a call. Despite the success of SIP as a primary Voice over IP (VoIP) protocol, SIP is neither XML nor Web service based, and a communication Web service model for multimedia conferencing that combines Web service with SIP for VoIP communication is needed.

2.1.1. Media signal session management

Media signal session is perhaps one of the most important relations in communication. This is because the interactions in communication are typically stateful and the stateful transactions are usually based on a stateful context “Session”. However, Web service is intended to be transport protocol neutral and it cannot rely on the lower-level transport protocol to manage the session. In order to use Web services to enable communication, it is needed to incorporate certain application session based web services in service-oriented communication to manage the session [12]. State machines are implemented using the state pattern. The state pattern is described in terms of context and state objects. The context object delegates handling of commands and events to the current state object, which takes action appropriate for the state. State objects further delegate handling of commands and events back to the context object via primitive functions. Changing the state of a context object is accomplished by changing the state object to one of a different class. The state machine is used here to manage the SIP session, which makes the application server play a role as a back to back user agent (B2BUA). The handler of message and the next state depend on both the current state and the incoming message. As a B2BUA, the application server communicates with both the user agent (UA) and the media server (MS). Several kinds of SIP call capability are needed in multimedia conferencing. One is answering call from user agent and to connect it to the media server. The other call is to connect some UA and our MS by a third party. The third is to connect with the MS directly, and the disconnecting and canceling of these connections. Besides, there are also some abilities of SIP message not related to call, such as INFO and REGISTER, among which calling just one sip entity and call independent SIP messages are all limited within one dialog.

2.1.2. MSML/MOML based media control module

The Media Server Markup Language (MSML) [13] is an XML language which can be used to control the flow of media streams and services applied to media streams within a media server. Clients can use it to define how multimedia sessions interact on a Media Server and to apply services to individuals or groups of users. For example, to control Media Server conferencing features such as video layout and audio mixing, create sidebar conferences or personal mixes, and set the properties of media streams. As well, clients can use MSML to define media processing dialogs, which may be used as parts of application interactions with users or conferences. Transformation of media streams to and from users or conferences as well as Interactive Voice Response (IVR) dialogs are examples of such interactions, which are specified using MSML. MSML is itself transport independent, but SIP INFO messages are typically used. MSML commands are sent to the Media Server by the Application Server as XML within a SIP INFO message body. The media server using MSML also sends asynchronous events to a client using MSML scripts in SIP INFO. Events are sent based on
previous MSML requests and are sent within the SIP dialog on which the MSML request that caused the event to be generated was received. If this dialog no longer exists when the event is generated, the event is discarded. Here, MSML provides a number of activities for interacting and controlling the flow of media streams and services applied to multimedia conference within a media server. We illustrate six types of activities in MSML with regard to the multimedia conference scenarios. Figure 2 shows the request activities for the conference, and <msml> is the root element when received by a media server, which defines the set of operations that form a single MSML request. <dialogstart> instantiates an MSML media dialog on a connection. <dialogend> terminates a MSML dialog.

**Figure 2.** The request activities for conference

It is observed from Figure 3 shows the response activities for the conference, the <event> element is used to notify an event to a media server client. The <result> element is used to report the results of an MSML transaction. Figure 4 shows the activities for the conference, and <createconference> is used to allocate the media mixing resources for conferences. <destroyconference> is used to delete mixers or to delete the entire conference and all state and shared resources. <modifyconference> means all of the properties of an audio mix or the presentation of a video mix may be changed during the life of a conference.

**Figure 3.** The response activities for conference

**Figure 4.** The activities for the conference

It is observed from Figure 5 shows the activities for the streams, <join> is used to create one or more streams between two independent objects. <unjoin> removes one or more media streams between two objects. Individual streams are specified using the <stream> element. <modifystream> is used to change the properties of a stream by including one or more <stream> elements that are to have their properties changed. Figure 6 shows the activities for the DTMF, and <dtmf> is used to collect DTMF digits from a media stream. <dtmfg> means DTMF generator originates one or more DTMF digits in sequence.

**Figure 5.** The activities for the streams

**Figure 6.** The activities for the DTMF
Multimedia Conferencing Management Using Web Services Orchestration over Public Networks
Bo Cheng, Xiaoyan Wu, Xiaoxiao Hu, Chen Junliang
Journal of Convergence Information Technology, Volume 6, Number 6, June 2011

Figure 6. The activities for DTMF

Figure 7. The activities for Play/Record

It is observed from Figure 7 shows the activities for the play/record, and <Play> is used to generate an audio or video stream. It MUST play in sequence the media created by the child media elements <audio>, <video>, <media>, <tts>, and <var>. Received media becomes part of the recording when <record> is in the create state and is discarded in the suspend state.

2.1.3. Data management module

The data management module has the functionality of authentication in the process of SIP register and provides interfaces to persist the information in the process of SIP register, such as called ID, SIPURI, and so on. The SIP register authentication and register information provides data access interface for the registration and enables the latter to accomplish the authentication in SIP register and persist the information needed by SIP protocol. Especially, plenty of information will be generated in the process of conference beginning. Hibernate as the tool for the interaction between the application program and database, and Hibernate acts as an object-relational mapping framework which is a lightweight encapsulation of JDBC and can directly carry out to create, read, update and delete operation to database via an object. Here, the frequent data access exists in the operation of the whole system. Massive disk I/O and even the system bottleneck will be caused if every access is operated on the database directly. Especially, the Least Recently Used (LRU) cache algorithm was adopted, and the data management stores the recently used data in cache, so the data access is speeded up greatly and disk I/O is reduced.

2.2. Multimedia conferencing service orchestration and management

Multimedia conferencing process management allows the creation of a multimedia conference and the dynamic management of the participants involved. Here, some main processes are illustrated.

2.2.1. Create conference process management

An application setting up a multimedia conference must initially invoke the createConference Web service to ready for the media resources. The result of such invocation is the creation of a context that represents a virtual room where users can meet. The BPEL flow for process management to create conference is shown as Figure 8.
Here, if the conference created successfully, and a unique identifier, a conferenceIdentifier result is assigned to the just created conference, and then to start record and update the conference information item with the request parameters in memory database.

2.2.2. Invite participant process management

The process of invite participant requests to add a new participant specified by participant to the multimedia conference call identified by conferenceIdentifier. The media used for the initial connection of the new participant depends on the conference type and the participant's supported media. The process will fail if the conference has already reached the maximum number of participants as specified in the creation multimedia conference service. The BPEL flow for process management to invite participant is shown as Figure 9.
There is a need to compare the numbers values when the inviteUser request comes, and the process will return the failure if the conference has already reached the maximum number of participants. Otherwise, begins to invoke the createConnection service to create a SIP connection for participants. Especially, the participant can also use fixed/mobile phone terminal to join the multimedia conference, it is necessary to invoke the DTMF service, there will have a voice to notify the user that invited by the chairman if the DTMF service opened successfully, and also inform the foreground that the user has joined successfully and to update the user information in memory database. If the continue DTMF service startup failure, and then to notify the foreground that fails to join the conference for a user, and then disconnect the corresponding SIP connection.

Figure 9. BPEL flow for process management to invite participant
2.2.3. Invite chairman process management

When the chairman leaves the conference, all users are disconnected (such a user could be for instance the reference for the conference billing). Invite chairman process is charge of the process management for chairman to join the multimedia conference, which involves to set up a SIP connection, set user media status, maintenance user information, and so on. The BPEL flow for process management to invite chairman is shown as Figure 10.

![BPEL flow for process management to invite chairman](image)

**Figure 10.** BPEL flow for process management to invite chairman

When the inviteUser request comes, this process will return the “request has been sent” messages to the client immediately. Then the process begins to invoke the createConnection service to create a SIP connection for chairman, if the SIP connection created successfully, there is a need to judge whether the chairman use Web browse as client via the return value of the SIP connection because of the chairman role is only acted for online user which uses Web browse, and then begins to set the media status that guarantee the audio stream and video streams are available as the chairman use Web browse, including the bidirectional stream direction. Otherwise, which means the current user is not suitable to act the chairman role and should be kicked off from the multimedia conference. If the SIP connection failed, then reports the message to the client that chairman failed to join the multimedia conference, and end the
conference. Once the media status are set successfully, then this process continues to invoke the 
setUpVedio service, and put the chairman into the first window in the screen, then invoke the 
updateUser service to update the chairman information in memory database, and notify the 
foreground that the chairman has joined the conference successfully. Otherwise, the user should 
be kicked off from the conference when media status is failure.

2.2.4. Terminal handoff process management

Participants can use SIP soft-termination on personal computer, or mobile phone, or fixed 
telephone to join the conference. Terminal handoff process is charge of the transformation 
among mobile phone, fixed phone and Web client on PC. The terminal handoff directions 
include: from the PC to fixed phone, from the PC to mobile phone, from the fixed phone to 
mobile phone, from the fixed phone to PC, from the mobile phone to fixed phone and from the 
mobile phone to PC. The BPEL flow for process management to handoff terminal is shown as 
Figure 11.

![BPEL flow for process management to handoff terminal](image.png)

In particular, as defined press key functionality in advance, when user presses 3# or 4# key 
on the fixed/mobile phone, then the DTMF service collects the press key information and play a 
voice to notify the user with the message “request is being processed, please wait for a moment”, 
and then inform the foreground that is to ready for the user terminal handoff. First, the 
foreground dealt with some simple service logics based on the press key information, such as 
the key value whether is reasonable, if unreasonable, return illegal, and play a voice to notify 
the user that the terminal handoff is failed, and return the failure value. Otherwise, if it is 
reasonable, and return the terminal number and handoff directions, and then invoke the 
changeTerminal service to handoff the terminal.
2.2.5. Conference notice process management

There is a need to notify all the participants the conference start time in advance when created successfully. The conference notification process receives the noticeParticipant request, and then invokes the context Web service to send the conference notification message to the participants in the request list repeatedly. Here, the context Web service is charge of which user should apply what kind of conference notification Web service, such as short message or email Web service. The BPEL flow for process management of conference notice is shown as Figure 12.

![Figure 12. BPEL flow for process management of conference notice](image)

Here, we define two variables, one is Nparti, which stands for the number of participants that the conference notice is failure, and the other is Yparti, which stands for the number of participants that the conference notice that is success. Especially, if the user register time to the SIP server compared to the conference start time is less than two hours, and the short message notice service applied. Otherwise, if the time is more than two hours, then the Email notice service applied. Finally, the user lists are returned back that has been noticed successfully or unsuccessfully.

3. Multimedia conferencing management system and performance analysis

The multimedia conferencing management system requirement includes the ability to handle asynchronous communications, suspend and resume execution upon events; the ability to handle long-running transactions, that is, to associate and dispatch the messages with the corresponding logical flows of interactions; the ability to handle event subscription, to create and synchronize parallel threads of execution. The BPEL language provides mechanisms to address these requirements and therefore to represent the patterns. While some of them are directly supported by the language (e.g., asynchronous interactions), the other require special ways to implement these patterns. Here, we have developed the BPEL based advanced communication Web service
orchestration for multimedia conferencing process management. Figure 13 shows the BPEL-based multimedia conferencing Web service orchestration and process management system.

In BPEL-based multimedia conferencing Web service orchestration and process management system, there are some important entities involved. Chairman identifies the user who manages the multimedia conference, and the client chairman can create and terminate a conference, and he can also invite or delete new participant users. It is assumed that only the chairman can invite a new participant to the conference. Clients, identifies the new participants invited by chairman. A new participant is invited to a conference and is notified of this invite. In order to receive the invite notifications, a client must previously register to a SIP register server providing its notification reference address (e.g., a URL of a notification Web Service). The client must accept the invite to connect to the multimedia conference. Media Server, which contains dynamic pools of media resources and the MSML, can be used to control the flow of media streams and services applied to media streams within a media server. Terminal, participants can use SIP soft-termination on personal computer, or mobile phone, or fixed telephone to join the conference.

When the chairman sends a request to create conference to the multimedia conference server, which deployed the BPEL-based multimedia conferencing process, and then begins to receive the request messages, and transfers the SOAP messages into normalized messages, and then routed the normalized message to the BPEL engine. At this time, BPEL engine receives the request message and invokes the BPEL flow to create conference which has been compiled. After some necessary process, BPEL engine invokes Web service provided by external communication server, which interacts with media server by SIP resource adaptor to the SIP server. Finally, RTP or RTCP media channel is established between the participants and the media server.

We also have illustrated the multimedia conferencing management system using different conferencing process management scenarios. The performance measurements have been tested on an Intel Xeon E5420 2.5GHz application server with the 10G RAM, under the RedHat Linux 9.0, in which BPEL engine are deployed. Especially, the LoadRunner tool is used to measure the load and performance with multimedia conferencing process scenarios that is running on the BPEL-based service orchestration bus, and the multimedia conferencing process sequence is to create conference, to create chairman connection, to get user status, to get media, to set
Multimedia Conferencing Management Using Web Services Orchestration over Public Networks
Bo Cheng, Xiaoan Wu, Xiaoxiao Hu, Chen Junliang
Journal of Convergence Information Technology, Volume 6, Number 6, June 2011

chairman media, to play voice, to create new chairman connection, to set new chairman media, to play the new chairman, and to change the chairman. Here, we configure 25 Vusers to use the multimedia conferencing services, and the measurement time last for 10 minutes. Running Vuser is used to determine the Vuser numbers that load on the integrated service bus at any given moment. The following Figure 14 shows the corresponding Hit per Second with 25 Vusers for the different multimedia conferencing processes.

![Figure 14. Hit per second for conference services](image)

Here, the hits per second, which means the number of hits made on the integrated service bus by Vusers during each second of the load test. Hits per Second diagram can evaluate the amount of load Vusers generate, in terms of the number of hits. From Figure 14 it is observed that the number of hits per second is stable and keeps 112 hits per second. The following Figure 15 shows the multimedia conferencing process transaction summary.

![Figure 15. Diagram of multimedia conferencing process transaction summary](image)

It is observed from Figure 15, that there are few of transactions that failed with errors. Specifically, there are only 2 number of total 7032 to create conference process transaction that failed with errors, only 4 number of total 7028 to create chairman connection process
transaction that failed with errors, only 5 number of total 7023 create connection for new chairman process transaction that failed with errors. Generally speaking, the errors are stochastic for unstable networks, and the result of transaction summary for those multimedia conferencing processes is reasonable and satisfactory. Next, the average transaction response time distributions diagram of conferencing process is shown as Figure 16.

![Figure 16. Diagram of process transaction response time](image)

It is observed from Figure 16, the different color lines represent different average transaction response time diagrams for multimedia conferencing process. Here, the response time of those multimedia conferencing processes transactions are measured, and also the precise response time value is listed in table 1.

**Table 1. Transaction response time for multimedia conferencing process**

<table>
<thead>
<tr>
<th>Color</th>
<th>Measurement</th>
<th>Min.</th>
<th>Ave</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purple</td>
<td>Action_Transaction</td>
<td>1.598</td>
<td>2.309</td>
<td>8.839</td>
</tr>
<tr>
<td>Green</td>
<td>ChangeChairman</td>
<td>0.045</td>
<td>0.06</td>
<td>0.15</td>
</tr>
<tr>
<td>Pink</td>
<td>CreateConference</td>
<td>0.205</td>
<td>0.295</td>
<td>0.628</td>
</tr>
<tr>
<td>Yellow</td>
<td>CreateConnChair</td>
<td>0.301</td>
<td>0.566</td>
<td>4.679</td>
</tr>
<tr>
<td>Blue</td>
<td>CreateConnNewChair</td>
<td>0.292</td>
<td>0.492</td>
<td>2.449</td>
</tr>
<tr>
<td>Cyan</td>
<td>GetMedia</td>
<td>0.036</td>
<td>0.055</td>
<td>0.096</td>
</tr>
<tr>
<td>Brown</td>
<td>GetUserStatus</td>
<td>0.046</td>
<td>0.072</td>
<td>0.542</td>
</tr>
<tr>
<td>Orange</td>
<td>NewChairPlay</td>
<td>0.154</td>
<td>0.244</td>
<td>1.64</td>
</tr>
<tr>
<td>Green</td>
<td>Play Voice</td>
<td>0.164</td>
<td>0.244</td>
<td>1.491</td>
</tr>
<tr>
<td>Pink</td>
<td>SetChairMedia</td>
<td>0.136</td>
<td>0.172</td>
<td>0.424</td>
</tr>
<tr>
<td>Brown</td>
<td>SetNewChairMedia</td>
<td>0.131</td>
<td>0.176</td>
<td>0.48</td>
</tr>
</tbody>
</table>

The preliminary measurement results show that the minimum response time for invoking of createConference process management is 0.205s and maximum response time is 0.628s, and the average response time is 0.295s. Other response time of multimedia conferencing process management transactions are listed above table 1. The result shows that our method is effective as a whole. Taking a closer look at the implementation of our multimedia conferencing process management system using Web service and orchestration approach, we found that the major part of time needed for the multimedia conferencing to complete is spent on Web services invocations. The relatively long delay for the service invocation can be explained to parse and transform the SOAP message from the Web service bus. Also, there some delays to allocate media resources from the media server at the beginning.
4. Conclusion and future work

The multimedia conference control and management interfaces are still in an early stage of development, and need to be extended in order to provide a high flexibility, scalability, and process management level that allows the deployment of multimedia conferencing solutions adapted to the user requirements. In this paper, we introduce the multimedia conferencing management using Web service and orchestration approach over public networks. For multimedia conferencing is a converged voice and data applications, which management involves both real-time communications and non-real-time information processing tasks, and mainly focus on the design and implementation for communication Web service and extensible BPEL (Business Process Execution language) based multimedia conferencing communication services process logics and also are encapsulated in a coarse grained Web service for the purposes of flexible multimedia conferencing process management. And also, we give the multimedia conferencing process management system and performance measurement and analysis. In the future work, we will put a focus on good performance and focus on how to integrate recent Web service standards, such as Web service notification, which standardize the way Web services can interact using the notification pattern, which specify a way for consumers to subscribe to a producer for notifications whenever a particular event occurs. More performance and scalability should be conducted with higher numbers in larger scales.

5. Acknowledgment

This study is supported by National Natural Science Foundation of China (Grant No. 61001118); “973” program of National Basic Research Program of China (Grant No2011CB302704)

6. References

