Context-Aware Web Services Orchestration for Multimedia Conference Process Management

Bo Cheng, Xiangtao Lin, Junliang Chen
State Key laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing 100876, P.R.China
chengbo@bupt.edu.cn
doi:10.4156/aiss.vol3.issue4.8

Abstract

Service oriented architecture is employed to cooperate with different software assets as long as they are wrapped up into web services. Under the architecture, the interoperation of telecom services and IT services is no longer an obstacle lying between the two domains. Besides, with the Web services orchestration technology, combining telecom services and IT services to rapidly develop and deploy new services can effectively improve software assets reusability. However, without context information, it is still hard to provide personalized services adapting to subscribers’ demands and market changes. In this paper, we bring forward context services to acquire context information and introduce context services into Web services orchestration. Taking multimedia conference business process management scenario as an example, we illustrate that context information acquired from context services can assist to provide personalized services to rapidly adapt subscribers’ demands and market changes.

Keywords: Web Services, Context, Context Ontology, Context Services, Multimedia Conference

1. Introduction

Developing and deploying new services rapidly and adapting to subscribers’ demands and market changing are the marked features of today’s services providers. Today telecommunication networks, internet and cable TV networks are merging into a common network [1][2]. Web service is based on the concept of using a structured XML document contained within a standard based SOAP message, to access, control various services remotely. Web Services orchestration is a technology to combine a series of loosely coupled services into an integrated business process in an open, platform independent manner to deliver rapidly new business services like value-added service to adapt to subscriber’s demands and market changes. Web Services Business Process Execution Language [3] (BPEL) has been a standard of OASIS in April, 2007 and is a de facto industrial specification for web services orchestration among vendors. With web services orchestration technology and BPEL supported tools, we can easily develop killer services and rapidly deploy the business process adapting to subscriber’s demands and market changes.

With the development of hardware, more and more portable devices are equipped with audio and video players, and some of smart mobile phones are even distributed with a variety of sensors, which take in charge of catching environment conditions like lighting, temperature, location etc. which together usually are called “context” [4]. In our approach, we have classified resources, from which context information comes, into three facets: users, physical environment and operational environment. According to the definition, in Multimedia Conferencing (MMC) scenario, context information contains participant profiles about name, age, gender, address, contacts etc., physical environment conditions like time, location, lighting, weather etc. and operational environment conditions e.g. the number of running processes on server, workloads and remain resources of networks and so on. The unique MMC in this paper involves telecom services, to be specific, OSA/Parlay [5] X Web services, and IT services, for example, weather web service, location web service and so on. Web services orchestration is the very mechanism to combine those loosely coupled hybrid services into one unique MMC business process compliant to WS-BPEL specification in structural XML. At present, WS-BPEL specification itself does not support dynamically binding web services in execution phase, that is, concrete web services must be specified in business process in design phase, which is typically named static binding mechanism. The lack of dynamic binding mechanism prevents services providers from
providing personalized services to adapt to subscribers’ demands and market changes. In our approach, context-aware services are introduced to try providing personalized services.

The paper is organized as follows: section 2 gives an overview of context aware Web service, a discussion of context ontology and a proposal about context services. Section 3 describes context-aware multimedia conference architecture and BPEL based multimedia conferencing process management. Section 4 is the conclusion.

2. Context aware Web services

Context information exhibited here are the final results returned by various context processing programs over a huge indirect results and raw data sensed by a number of various kinds of “hard sensors” and “soft sensors” residing in physical environment and operational environment respectively. “Hard sensors” referred here are the traditional electrical sensors to detect some aspects of information about physical environment. In contrast, “Soft sensors” are components, whose functionality are to monitor the status of applications and their containers in which applications are residing, of operational environment.

2.1. Context hierarchy

The term “context” referred so far is abstracted from indirect results and raw data detected by sensors. To illustrate the concept “abstract”; for example, the number of ID card is abstracted to a person, a pair of longitude and latitude is abstracted to a meeting room in MMC scenario while in traveling scenario it is abstracted to a hotel or an interesting place. And in addition, in our approach, we maintain an indirect context information knowledge base for later utility so that they will save a lot of time for applications and their containers in which applications are residing, of operational environment.

2.2. Context Ontology

The basic concept of context abstraction is based on ontology which provides a vocabulary to define concepts and relationships between them of a domain in a machine interpretable manner. Just like other knowledge bases, divide-and-conquer methodology is applied to design context ontology. In our
approach, as Figure 2 shows, the context ontology consists of two levels of formulations. One is *general context ontology* describing the basic and common knowledge about several domains having an is-a relationship instead. The other is *domain context ontology* presenting domain concrete concepts and relationships between concepts.

![Diagram of context ontology](image)

**Figure 2.** Division iterations in ontology designing process

It is observed from Figure 2, the root ontology is firstly divided into various domains e.g. data mining ontology, context ontology and the other ontology. Secondly, context ontology is divided into general context ontology and domain context ontology i.e. wires domain ontology and wireless domain ontology in MMC scenario, marked in blue lines. Thirdly, wireless domain context ontology is continuously divided into GSM sub-domain, WiMAX sub-domain and other sub-domains. In previous sections, we have emphasized that reasoned context will be derived from other already available contexts. JENA 2 [6], RACER [7], FaCT++ [8] and Pellet [9] are well-known ontology reasoning tools. However, reasoning on large ontology is time-consuming and requires powerful processing ability of work stations. One feasible way to step over such a stumbling block is to divide a large ontology into smaller pieces, in which context only contains less concepts hence less complex relationships as less concepts will dramatically, at a rate of exponent, reduce the complexity of relationships referring to two concepts. In this condition, it is possible to reason on smaller ontology on soft sensors and even hard sensors, from which meaningful and machine interpretable context information is acquired. Thanks to the well-defined and good extensibility for the standardization domain ontology, we can add it to context ontology even if we are reasoning on that context ontology. With this plug and play property, our context ontology can migrate from one domain to another, and business process, in MMC scenario, will pick up different services with the migration of context domain ontology.
2.3. Context Services

Context information can be acquired in three methods: forma-filling, context detection and context extraction. In the past, most researchers considered catching context information as the behaviors of sensors and their corresponding programs instead of services [10][11][12]. In this paper, we introduce context-aware services to catch raw data from operational environment and physical environment, to generate indirect results, to return the abstracted final context information. Corresponding to these three context acquisition methods, we recommend three types of context-aware services: form-filling context services, detecting context services and extracting context services.

3. Context-aware Web services orchestration for MMC process

In our previous works [13, 14], we present the multimedia conference model with IT web services and telecom web services i.e. Parlay X Web services. However, without context information, we meet difficulties in providing personalized services in web services orchestration process as we can not acquire personalized information or preferences of the participants. In our novel MMC service, a participant is invited into a conference with his/her static context i.e. user profiles, acquired from form-filling context services, like ID, calendar, preferred terminals, and preferred situations; and, our MMC service will adjust its actions, almost transparent customers, to adapt to the changes of these context.

3.1. Context-Aware Multimedia Conference Architecture

In this section, we describe our context-aware multimedia conference architecture. Our architecture aims to orchestrate telecom web services and IT web services to provide customers with personalized MMC service. The architecture consists of the following components depicted in Figure 3.
**BPEL Based MMC WS Orchestration.** Ordinarily, the orchestrator works in a WYIWYG (What You See Is What You Get) manner, which can help designers to rapidly make out a process in visualization with less key hits. There are several visual integrated orchestration toolkit e.g. ActiveBPEL Designer, Oracle BPEL Designer.

**BPEL Process Executor.** Which is a BPEL execution engine compliant to BPEL specification, currently BPEL 2.0. The MMC business logic is running in the executor. There are many open source BPEL engine e.g. ActiveBPEL, Apache ODE (Orchestration Director Engine), and also enterprise level engine like Oracle BPEL process Manager.

**Results Evaluator.** After the execution of MMC process, the corresponding form-filling context service is called to ask participant for feedback of MMC experience.

**Parlay X Web Services.** The Parlay X Web Services mentioned here are multimedia conference related services in part 12 of OSA/Parlay X Web Services. They are createConference WS, getConferenceInfo WS, endConference WS, inviteParticipant WS, disconnectParticipant WS, getParticipantInfo WS, getParticipants WS, addMediaForParticipant WS and deleteMediaForParticipant WS. They together consist of the primitives of MMC control signaling.

**Context Web Services.** They are the three types of context services illustrated in to previous sections.

**Context Ontology.** Context ontology consists of general context ontology and MMC domain ontology. Context ontology is referred by the three types of context services. In form-filling context services and context detection services, concepts in context ontology are referred to abstract context information from raw data. And in context extraction services, concepts and relationships in ontology are the foundation of context aggregation and reasoning.

**Context Knowledge Base.** Context KB contains context information linked to participants. And context KB is referred in MMC creation and running process and altered in the service evaluation phase by the three types of context services.

### 3.2. Context-aware Web services orchestration for MMC BPEL process

To make it clear, Figure 4 depicts the MMC process control flow without faults and exceptions. Faults and exceptions can be processed in fault handlers in the same way.

![Figure 4. MMC process modeling](image)
To the MMC process control flow can be divided into three stages as the MMC reservation stage, MMC meeting stage and MMC ending stage. The items above line ① are about the steps and operations in MMC reservation stage. In this stage, context services, in general form-filling context services, are called to acquire context information about chairman and the other participants. The items between ① and ② demonstrate actions carried out in MMC meeting stage. In this stage, context services usually context detecting services and context extraction services are called to detect and extract dynamic context during the meeting. Just as the scenario described in the “Context Web Services” in the last section, context detecting services and context extraction services are called for many times in turn and corresponding actions are taken to adapting to context changes. The items below line ② are actions to end the MMC in MMC ending stage. In this stage, context services, in general form-filling context services, are called to ask participant for evaluation about the MMC and next, context about that participant can be extracted and abstracted from the evaluation information. Once we make out MMC process control flow, it is not hard to translate it into BPEL based process with web services orchestration technology. The BPEL based process is deployed into BPEL process executor described in the last section. Next, we will illustrate, step by step, how to orchestrate a BPEL based context-aware MMC process with web services orchestration technology. Lack of space forbids us to discuss every details of the orchestration and we only range over context services in the MMC BPEL process.

Firstly, we defined the partner link types about context services in the MMC BPEL process. A BPEL process describes a flow of interactions between the process and extern partners i.e. external Web services. BPEL process also defines the service partner links and service partner link types to identify the roles and their relationships in the process management interactions. Each process management interaction describes the roles, which the process and services play at that step, and the data, manipulated by the parties in these roles. A partner link type describes the kind of message exchange to be carried out by two web services. A partner link type characterizes the exchange by defining the roles played by each partner and by specifying the port type provided by that partner to receive messages appropriate to the exchange. A partner link type can include two or one role according to whether or not the calling service requires a callback from the target services in this step. Take form-filling context services as an example:

```
<plnk:partnerLinkType
 name="formFillingContextServiceLT">
  <plnk:role name="formFillingContextService">
   <plnk:portType name="mmc:FormFillingContextPT"/>
  </plnk:role>
  <plnk:role name="formFillingContextServiceCallback">
   <plnk:portType name="mmc:
   FormFillingContextServiceCallbackPT"/>
  </plnk:role>
</plnk:partnerLinkType>
```

To achieve well encapsulation and reusability, partner link type is recommended to be defined in WSDL files. Secondly, we define partner links about context services in the BPEL process. A partner link is a communication exchanges between the process and another partner i.e. another service. The role the process plays is defined in “myRole” tag of the partner link definition; and, the role the partner plays is defined in “partnerRole” tag.
Normally, partner links are defined in BPEL process. Thirdly, we define the BPEL process carrying the MMC business logic i.e. MMC control flow. The main body of MMC BPEL process is set up with tag <bpel:flow>, which describes a parallel behavior in a BPEL process, to achieve more flexibility and extensibility in the following process as there is no difference between flow and sequence if the process includes only one workflow.

In many situations, context services work in event handlers of the MMC BPEL process since the main part of context information is usually dynamic so that it is more efficient to acquire context with context interruption mechanism, that is, event mechanism than context rolling mechanism.

The definition we make so far is platform and product independent. Finally, we define a process deployment descriptor, in general platform and product specific, interpreted by BPEL process execution engine and its container to deploy the MMC BPEL process and make it run. Take the process deployment descriptor for ActiveBPEL for example.
With above detailed steps, we can create a BPEL based context-aware Web services orchestration for multimedia conferencing process management, and also the deployed BPEL process in turn is then wrapped into a Web service and exposes a WSDL interface for other external partners or third party to communicate with it.

4. Conclusion

Multimedia conference usually are developed by hard coding so that it will take a long time to make a MMC service developed and deployed, not to mention adapting to subscribers’ demands and market changes. Context services recommended in this paper can catch subscribers’ demands and market changes in different aspects in time. With context services, we can detect raw data from hard and/or soft sensors and then abstract these raw data into direct context referring to context ontology and even aggregate or reason on this direct context into indirect context. We introduce context services into multimedia conference business process to acquire context information from participants and environment. Cooperating with Parlay X web services, context services can bring us a personalized and unique MMC process; hence, we can rapidly adapt to subscribers’ demands and market changes. However, lack of real dynamic binding mechanism in BPEL leads to awkward services or subsequent process activities selection and provision, to provide personalized services, in context-aware web services orchestration progress. And take the characteristics of context and context services into consideration, it is better to put context services into event handlers of the BPEL process which may bring about lower efficiency for subsequent process activities may be first undo and then pick up new branch of the workflow.

5. Acknowledgments

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


