Web Services based Approach for Integrating Multimedia Content

Fatma Zohra Lebib[1], Hakima Mellah[1] and Youssef Amghar[2]

[1] Research Center in Scientific and Technical Information, CERIST, Algiers, Algeria

[2] National Institute of Applied Sciences, INSA, Lyon, France

Abstract. Multimedia content is derived from various autonomous, distributed and heterogeneous content sources. To address problems posed by content sources heterogeneity, a service-oriented architecture is proposed to assure a dynamic integration of multimedia content. In this work we propose a model that we call *MaaS* (for *Multimedia as a Service*), through which multimedia content providers expose their content. Once the MaaS is discovered, its classification into category of concepts, based on domain ontology, is made. However, content access is done through a concept hierarchy. The sport domain is used to validate the proposed model.

1 Introduction

With recent progress in computing technologies, multimedia content such as images, video clips, animations, graphics, and audio have proliferated over the past several years. Users have begun to expect that multimedia content should be just as easily accessed as alphanumeric data. They want to see video clips related to text article they read, listen to music contained in a video clip they see, and find relevant photo images that appear in a movie or news video clips. To support such user needs, it is important to provide integrated access to diverse types, of multimedia data, stored in disparate data sources. However, many multimedia applications today deal with multimedia data from disparate sources separately.

The need for an integrated and unified access to multiple information sources has stimulated research on mediators [1] and data warehouse [2], [3]. However, mediator and data warehousing based integration approaches fail to meet the requirements of a changing environment (evolving needs of users, business and technical environment change, etc...). It is difficult to dynamically unify data sources, since data sources must be known at design time. Another drawback of these approaches, successive maintenance operations on software systems tend to make the system software increasingly hard and costly to maintain [4], [5].

Service orientation such as Service-Oriented Architecture (SOA) has emerged as a better methodology for Enterprise Architecture [6]. The companies are more oriented toward SOA to share data, making their data sources behind service and providing an interoperable way to interact with their data. This category of service is known as

Zohra Lebib F., Mellah H. and Amghar Y..
Web Services based Approach for Integrating Multimedia Content.
DOI: 10.5220/0004087800510060
In *Proceedings of the 10th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems and 1st International Workshop on Web Intelligence* (WEBI-2012), pages 51-60
ISBN: 978-989-8565-14-3

DaaS (Data-as-a-Service). This concept was recently introduced as the first step to virtualize access to data sources in the Cloud and SOA architectures [7], [8], [9], [10].

Inspired by the concept of DaaS, *a service-oriented architecture for integrating multimedia content*, is proposed to provide a unified view of various autonomous, heterogeneous and distributed multimedia content sources (relational or object databases, multimedia applications, multimedia servers, etc…). This architecture allows combining diverse multimedia content type (image, text, audio and video), which are semantically related. The multimedia content are encapsulated into a web service model, named *Multimedia as a Service* (MaaS). MaaS is typically implemented using Web service technologies. Services providers publish their multimedia content sources as MaaSs, which may be discovered, bound at the time they are needed and disengaged after use (loosely coupled).

The paper is organized as follows: We start Section 2 by giving a motivating example; Section 3 discusses related work; and Section 4 describes our approach for integrating multimedia content. Section 5 presents experiments. Finally, we conclude the paper and outline our future work in Section 6.

2 Motivating Scenario

Let us consider, a user wants to receive rich content that meets his request that is: *the exploits of Zinedine Zidane at the beginning of his career*.

Different and multimodal (image, text, audio and video) contents can contribute to compose the answer to this query, for example:

- a video interview of the player at his first title event,
- an article about this interview,
- some player photos,
- another video of his first football match.

The most important steps to meet this request are:

- transparent access to heterogeneous and distributed content sources to retrieve content which semantically meet the request. The result may have different type (image, text, audio and video),
- combination of a set of retrieved content, to build a more complex content (composite content), which is delivered to the user.

3 Related Work

We classified the works related to the integration problem into the following categories:

Mediator Approach. In systems based mediator [1], the local data sources maintain their independence. Data integration is realized by defining a global view, or an integrated schema, which is shared by all the sources. A middleware component "*mediator*", translates request from the global view to local data. Typically, each local source needs an adapter component (wrapper), which exports a local data view into a common format for the mediation. This approach required mapping to capture the relationship between the source descriptions and the mediator and thus allow queries on the mediator to be translated to queries on the data sources. Specifying this correspondence is a crucial step in creating a mediator, as it will influence both how difficult the query reformulation is and how easily new sources can be added to or removed from the integration system.

Data Warehouse Approach. A data warehouse addresses data-intensive problems which are localized in multiple heterogeneous systems. To be exploited, all data from distributed systems must be organized, coordinated, integrated and finally stored to provide the user with a global view of information [11]. This approach accelerates query processing because it does not require accessing sources in order to answer to the queries, for against it needs a very important storage cost and a particular update cost. Any change in the local sources must be re-imported in warehouse.

Service Oriented Approach. In the service oriented approach two important categories of works are considered:

Service-based Software. System change is inevitable, expensive and very hard [5]. Both Bennett [4], [12] and Ghezzi [13] suggest that traditional static bound (early bound) supply-side systems cannot meet the needs of continually changing business environments. Two categories of works emerge:

Concept Software as a Service (SaaS), which was proposed by the Penning Research Group, in which services [4], [14] are composed out of smaller ones (and so on recursively), procured and paid for on demand. This solution offers the potential for users to create, compose and assemble a service by bringing together a number of suppliers to meet their needs at a specific point in time. The central technical issue for this solution is very late binding at system execution. The general scheme for service-based software [4], [12], [15] is: publish, find and bind[1].

Due to the growing enlightenment on SOA technologies [16], there is an opportunity for organizations to use these emerging technologies as a platform to modernize legacy systems. According to [17] « any software artifact that was built using pre-SOA techniques is legacy ». The challenge is to adopt such software in the new development service-oriented. Data are provided as a service, thereby increasing the longevity of the core functionalities of the legacy system. These services have been generated by modernization [18].

Service-based Data. Taking advantage of Web services technologies, the software as a service model [4] and cloud computing [19], recently, various research efforts have concentrated on the development of the concept of data/information as a service (DaaS) [8]. DaaSs offer functionalities to allow their consumers to acquire or provide data under the service model, regardless of whether the offerings are free or commercial [10]. The implementation of DaaS offer a set of service operations, each one takes a data request and returns a data resource that meets the demand.

Reference [20], the authors suggest to use SOA to integrate biological data from different data sources. Data is stored into a wide variety of formats in heterogeneous

---

[1]We need to be able to find and bind services at the moment of need, and then unbind (disengage) as soon as the service is finished.

systems. SOA provides a standard method to integrate both data sources and software applications by regarding them as interoperable services. SOA can provide dynamic service discovery and binding, which means that service integration can occur on demand. The proposed system has been built using web services on Microsoft .Net environment. The web service is used to provide a uniform regime for "plumbing together" data resources that present themselves as services with programmatic interfaces. The authors show that SOA will solve the problems that facing integration process and the biologist scientists can access the biological data in easier way.

4 Service Oriented Architecture for Integrating Multimedia Content

In order to provide a unified view of disparate and heterogeneous multimedia content from different sources (relational or object databases, multimedia applications, multimedia servers, etc...), we propose a *Service-oriented Architecture for Integrating Multimedia Content* (Fig. 1), which is based on the idea of *SaaS/DaaS* [5], [9], [13] and *SOA technologies.*

In the proposed architecture, multimedia content providers expose their multimedia content as a web service, according to proposed model named Multimedia as a Service (MaaS). MaaS is a kind of web service which can provide multimedia content by querying heterogeneous multimedia content sources. Service provider implements MaaS using web services technologies, describes the service using description languages such as WSDL (Web Service Description Language) and publishes the service description file in a service registry, such as UDDI (Universal English Description Discovery and Integration). The MaaS can be then discovered, bounded and disengaged after use.

Since multiple MaaSs may contribute to answer to a user request, we introduce a middleware component, named *Multimedia Broker Service* (MBS), between MaaSs and the user. MBS allows the user to acquire a unified integrated view of the multimedia content.

In following Section we present a detailed description of the architecture.

4.1 Multimedia as a Service (MaaS)

A MaaS is a service that ensures a transparent access to multimedia content. It is a wrapper implemented as Web service which provides a WSDL interface to allow remote service invocation over SOAP (Simple Object Access Protocol). The standard interface ensures a transparent access to all available distributed heterogeneous and autonomous multimedia content sources. By the mean of service oriented approach, each MaaS can be developed in different implementation languages, can access sources managed by different systems and can run on different platforms. However, regardless of how a MaaS is programmed, it provides a unified and seamless way to access underlying multimedia content. Each MaaS provides standard interface method to query the appropriate multimedia content sources, thus the information about how actually access to multimedia content sources is hidden inside.

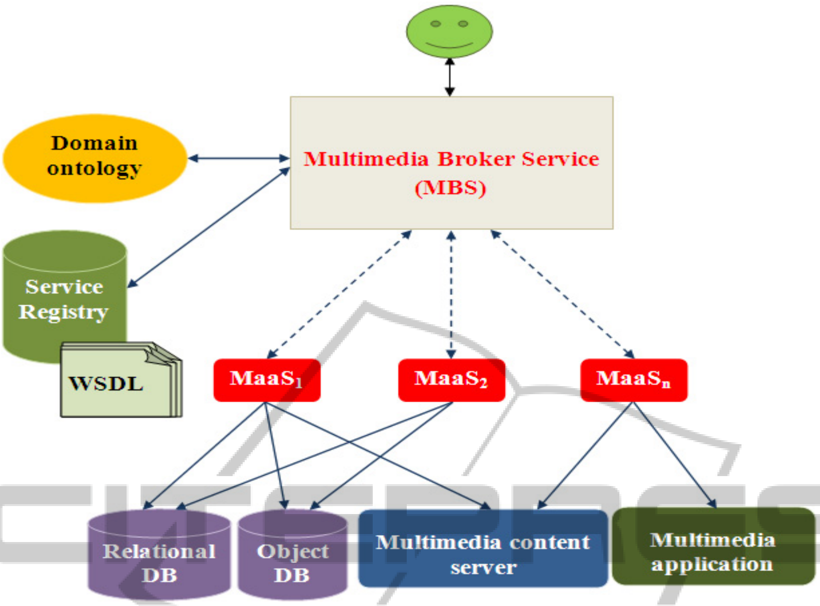

Fig. 1. Service-oriented architecture for integrating multimedia content.

4.2 Multimedia Broker Service (MBS)

MBS is a composed service, which integrates different MaaSs in order to provide the end-user an integrated uniform view of the multimedia content. When the MBS is invoked, it dynamically finds appropriate MaaSs and then binds to them. However the MBS acts as an interface between the user and multimedia content sources.

To offer this functionality, the MBS consists of a Graphical User Interface, a Classification Component and a Query Processing Component (Fig. 2), in order to complete the following tasks:

- classifying and managing the MaaSs description,
- analyzing and executing the user query,
- managing communication with the MaaSs.

The main components of MBS are: Classification, Graphical User Interface and Query Processing that we describe respectively.

Classification Component. To support the MaaSs discovery and invocation, the MBS builds a MaaS repository and query schemas that we explain in this sub section.

MaaS Repository. To prepare the MaaS discovery process, the MBS builds, during system configuration, a *MaaS repository* from MaaS interface specifications that are discovered on the fly. The repository assists searching and invocating MaaSs, which description is published in service registry.

In our approach, the discover function is to query the service registry for web service whose name or description contains the relevant keywords. For example a name/description keywords list might be (multimedia, article, image, video, etc...).

Relevant WSDLs and corresponding schemas of identified MaaSs are then downloaded.

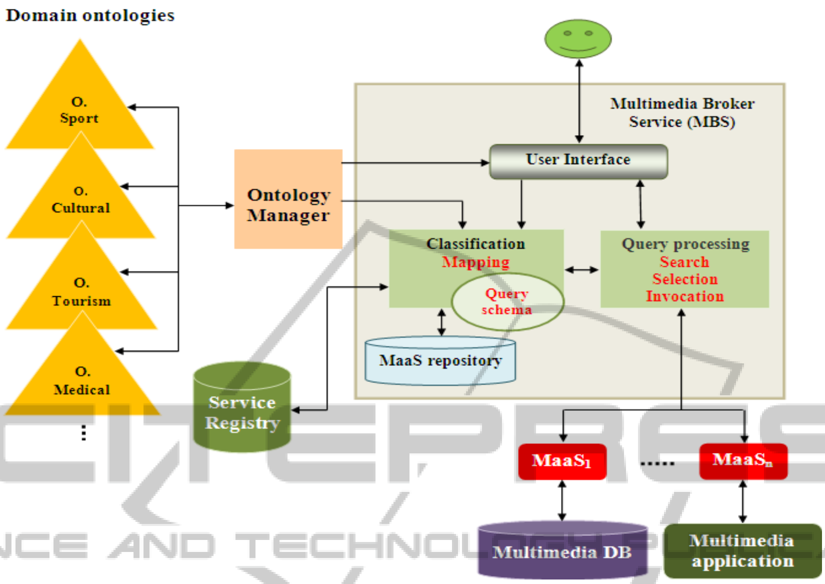

Fig. 2. The architecture of multimedia broker service.

The next step is the processing of the discovered WSDLs. This step involves the examination of each newly discovered WSDL and recording particular information (methods and their inputs) about the MaaS to enable integration. The MBS adds the newly found MaaS to its repository. The repository provides a quick means of identifying MaaSs that retrieve specific multimedia content. It is updated every time the MBS identifies a new MaaS or detects an update to a previously discovered MaaS.

The information added in the repository includes:

– method name,
– input parameters /data types,
– WSDL location.

To satisfy user query processing, the MaaSs are classified and organized in activity domains concepts. The MaaSs classification is based on domain ontologies (sport, medical, cultural, etc…) to which the MaaSs pertain. A domain concept relevant to the content, provided by the MaaS is associated with the corresponding MaaS method as shown in Fig. 3. For example, a method, which contains "name-player" as input parameter and provides a photo of "Zinedine Zidane" as output message, would be related with the concept "*football player*" in sport domain. This step is done by a domain expert.

Additional information necessary to meet user requests includes:

– the content type (image, text, audio, video), provided by MaaS,
– the MaaS properties (resolution, duration and size of provided content).

Query Schema. Multiple MaaSs can contribute to answer user's query, they may have different WSDLs/Schemas. In this step, a global query schema is created from discovered MaaSs definition. This global query schema is named “*Query schema*”.

Domain	Concept	WSDL Localisation	Method Name	Input Parameter / Data Type	Content Type
Sport	players	http : //...	getphoto	$Param_1/Type_1$ $Param_2/Type_2$	image
Sport	Football match	$MaaS_{11}$ loc WSDL.	getvideo	$Param_1/Type_1$ $Param_2/Type_2$ $Param_3/Type_3$	video
Cultural	movies	$MaaS_{13}$ loc WSDL	getmovie	$Param_1/Type_1$	video
Medical	drugs	$MaaS_2$ loc WSDL	getarticle	$Param_1/Type_1$ $Param_2/Type_2$	text
Geography	cities	$MaaS_4$ loc WSDL	getmap	$Param_1/Type_1$	image

Fig. 3. MaaS repository.

To allow MBS to present the query schema to the user, a repository maintains a query schemas catalog, which contains a set of XML files. Each file contains query schema description, composed of:

– The associated concept,
– The parameters required for MaaSs invoking.

Fig.4 shows an example of XML description of a query with one parameter.
Identity (1): gives the concept associated to MaaSs and a provided multimedia content description.

Parameters (2): contains the description of the required parameters invoking MaaSs. For each parameter, XML file gives a name, type, and a text description.

Through query schema a *correspondence table of parameters* is established. The latter contains for each parameter, the set of the corresponding methods.

```
<Query>
    <Identity>
        <Concept> city </Concept>
        <Description> Information about a city </Description>   1
    </Identity>
    <Parameters>
        <Parameter>
            <Name> city </Name>
            <Type> String </Type>                                 2
            <Description> Name of the city </Description>
        </Parameter>
    </Parameters>
</Query>
```

Fig. 4. Sample XML file for describing a query schema.

Graphical User Interface. The interface is based on domain ontology, which allows accessing to multimedia content through concepts hierarchy. The interface aims to assist user to specify his need on multimedia content.

The user specifies a concept into a chosen domain (sport, medical, cultural, etc…), then he introduces the following information:

– the search filter consisting in constraints set related to multimedia content (for example resolution, size and duration),

– the parameters values through a Q*uery schema* presented to the user.

Query Processing Component. The user query is processed by the MBS through the following steps:

– querying the *MaaS repository* to obtain all MaaSs that met user requirements,

– selecting MaaSs, which are more suitable to meet the user requirements, based on MaaSs historic: experience of using the MaaSs by MBS (for example: time necessary to execute service, quality of content provided by the service),

– invoking the selected MaaSs,

– receiving the result from each MaaS and forward it to requestor.

To invoke MaaS, the MBS:

– uses the previously established *correspondence table of parameters,*

– translates user query into candidates MaaSs query format,

– submits the queries to the MaaS providers.

The query result is returned to MBS that forward result to requestor.

5 Experimentation

To validate our proposal, for integrating multimedia content, we implement the Multimedia Broker Service using Java. It allows users to access to multimedia content from heterogeneous and distributed sources. The MBS is a web service that is published in UDDI in order to be discovered and invoked via the Web.

We used Axis (Apache eXtensible Interaction System) for implementing web services. For its use, Axis requires the JDK (Java Development Kit) and a Tomcat servlet container for web services deployment. Jena API is used to manipulate the OWL (Web Ontology Language) ontology.

First we create MaaSs that expose multimedia content stored in relational Databases. This content consists of sports information (photos, videos, articles...).

MaaSs are generated through a programming interface that we have implemented. This interface allows easily MaaSs generating. The MaaSs are then published in service registry to be discovered and invoked by the MBS. The attachment mechanism is used to associate a SOAP message with one or more attachments.

The sport domain is chosen where a user wants to receive information about the football player "Zinedine Zidane". Fig. 5 shows a sample of returned result. The result consists of a video containing a scene of player in a football match, a player photo and a text article about this player. The provided result is composed of a set of different type contents which constitute a semantic whole. However, the content returned by Search Engines meets simple queries. In particular, there's a separation between different types content. On Google for example, the result returned by the keyword "*zidane*" is shown in Fig. 6. Even the whole option of content type is selected results returned are categorized by type. Those that contain only textual documents, only images, only videos, etc (Fig. 6).

Fig. 5. Example of a returned result.

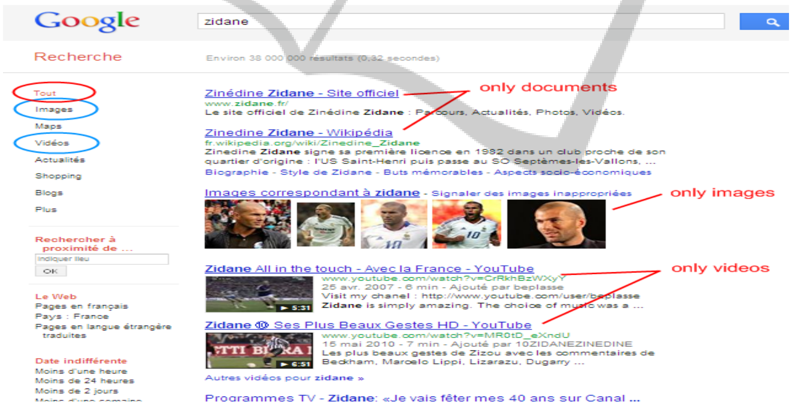

Fig. 6. Returned result on Google.

6 Conclusion and Future Work

In this work, an architecture for integrating multimedia content, based on web services technology is proposed. The main objective is to provide an integrated view of multimedia content from autonomous, heterogeneous and distributed content sources. The proposed architecture is based on a model labeled MaaS, a kind of web service that allows easy and interoperable access to multimedia content from different sources (relational/object database, multimedia server, multimedia application). To address the semantic problem during MaaS discovery, domain ontologies are used to classify the MaaSs into categories and to assist the user to specify his request. Domain ontologies can be also exploited to deal with the semantic heterogeneity of MaaS.

An integration system, in sport domain, is implemented to validate our approach. We can consider the realized work as a first and important stage for an aggregated search that refers to service oriented computing.

As future work and taking into account the heterogeneous nature of user environment (user profile, terminal, network) we propose to adapt the returned results with user context. For this, it is suggested to compose MaaSs with adapting multimedia services.

References

1. G. Wiederhold, Mediators in the architecture of future information systems, IEEE Computer Magazine (1992).
2. Yig V. Raman, I. Narang, C. Crone, L. Haas, S. Malaika, T. Mukai, D. Wolfson and C. Baru "Data Access and Management Services on Grid", (2002).
3. J. L. Wiener, H. Gupta, W. J. Labio, Yue Zhuge, H. Garcia-Molina, J. Widom, The WHIPS Prototype for Data Warehouse Creation and Maintenance, In Proceedings SIGMOD (1997).
4. P. J. Layzell, K. H. Bennett, D. Budgen, P. Brereton, L. A. Macaulay, M. Munro, "Service-Based Software: The Future for Flexible Software" APSEC Conference (2000).
5. M. M. Lehman, "Laws of Software Evolution Revisited", Proceedings 5th European Workshop on Software Process Technology, Nancy, France (Oct 1996).
6. Ashrafi, R & Baghdadi, Y, 'E-business: issues, challenges and architecture ', IJBIS (2008).
7. Michael J. Carey, "Data delivery in a service-oriented world: the BEA aquaLogic data services platform." in SIGMOD Conference (2006).
8. A.Dan, R.Johnson, and A.Arsanjani, "Information as a Service: Modeling and Realization," in International Conference on Software Engineering (WSD in SOA Environments) (2007).
9. M.Gilpin, N.Yuhanna with K.Smillie, G.Leganza, R.effner, J.Hoppermann "Information-As-A-Service: Waht's Behind This Hot New Trend?," Forrester Research, RReport (2007).
10. H-L. Truong and S. Dustdar, "On Analyzing and Specifying Concerns for Data as a Service," in the APSC Conference (IEEE APSCC), Singapore (2009).
11. M-S. Hacid and Chantal Reynaud "Data sources integration", "L'intégration de sources de données" In Revue Information-Interaction-Intellegence (2004).
12. Bennett K. H., Gold N. E., Munro M., Xu J., Layzell P. J., Budgen D., Brereton O. P. and Mehandjiev N. "Prototype Implementations of an Architectural Model for Service-based Flexible Software", Proc. Thirty-Fifth Hawaii International Conference on SS (2002).
13. C. Ghezzi, "Ubiquitous, Decentralized, and Evolving Software: Challenges for Software Engineering". ICGT (2002).
14. C.Lovelock, S.Vandermerwe, B.Lewis, "Services Marketing", Prentice Hall Europe (1996).
15. H. Kreger, "Web Services Conceptual Architecture (WSCA1.0)", IBM (May 2001).
16. Gartner Group. (2009) Hype Cycle for Emerging Technologies (21 July 2009).
17. S. Cetin, N. Ilker Altintas, H. Oguztuzun, A. H. Dogru, O. Tufekci and S. Suloglu, «Legacy Migration to Service-oriented Computing with Mashups» (ICSEA 2007).
18. Barnes, A., M., and Cilliers, C., A Holistic Evaluation Strategy for the Modernization of Legacy Systems. In Southern Africa TNAConference, Stellenbosch (2010).
19. R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," 10th IEEE Conference (2008).
20. N. Al-Otaibi and A. Y. Noaman, Biological Data Integration using SOA, WASET(2011).