Journal of Mobile Multimedia, Vol. 4, No. 3&4 (2008) 227–240

PROXY AGENTS FOR ADAPTIVE DELIVERY OF MULTIMEDIA[a]

ROCCO AVERSA BENIAMINO DI MARTINO

NICOLA MAZZOCCA SALVATORE VENTICINQUE

Department of Information Engineering, Second University of Naples
via Roma 29, 81031 Aversa, Italy

{*rocco.aversa, salvatore.venticinque*}*@unina2.it*
{*beniamino.dimartino, nicola.mazzocca*}*@unina.it*

Received April 20, 2008
Revised July 10, 2008

Adaptive delivery of multimedia means the capability of adapting content exploitation according to available resources, session parameters and user profile, in order to improve quality of service. It can be exploited to support people with disabilities in accessing available services, or to change the way of exploiting a resource according to working conditions. For example a text document can be delivered as audiostream when an user is driving, or when it cannot exploit a content as it is, because he is blind. Different kinds of multimedia fruition allow also to conceive new value added services and new models of interaction with contents. Here we present architecture design, and prototypal implementation, of a framework that can be integrated, transparently, in web applications, to add mobile agents based adaptive delivery. Mobile agents technology has been exploited to augment flexibility of classical solutions and to develop advanced facilities. We set up a case study on semantic and location based information retrieval to demonstrate effectiveness of our approach.

Keywords: Context awareness, Content Adaptation, Multimedia delivery

1 Introduction

Adaptive delivery of multimedia means the capability of adapting content exploitation according to available resources, session parameters and user profile, in order to improve quality of service that is perceived at client side. Intelligent delivery of multimedia is required not only to handle technological advances, but also to support new kinds of fruition of contents. Adaptive multimedia delivery can be relevant from a social point of view. It can be exploited to support people with disabilities in services access, or to change the way of exploiting a resource according to working conditions. A text could be delivered as an audio stream, by a text to speech adapter, when the device does not provide an easy interface for reading, when the user is blind, or he is simply driving. Furthermore context awareness can help users, in filtering large amount of data, which are available in Internet, but which are often not usable. This happens because it is difficult to retrieve, among all information distributed across the

[a]This work has been supported by LC3 - LABORATORIO PUBBLICO-PRIVATO DI RICERCA SUL TEMA DELLA COMUNICAZIONE DELLE CONOSCENZE CULTURALI - National Project of Ministry of Research MIUR DM1791

Network, the really relevant and exploitable ones. It is also difficult for providers to attract customers by identifying and offering those products, which optimize their satisfaction. It is a tedious and long work to select relevant information, because contents are not organized according to semantic criteria, which grant an effective and accurate retrieval. Actually information in Internet does not have a meaning. Meaning is not an available attribute that applications can take in account for their elaboration.

As a partner of LC3 project, the research unit of Second University of Naples is designing and developing an adaptive interface for multimedia delivery. The project puts together efforts by academic partners and industrial ones in order to define techniques and technological solutions to support digitalization, classification, retrieval and communication of cultural resources. Results from these activities fill storages of multimedia, which will be retrieved and exploited by different kinds of users by heterogeneous devices.

New techniques and technologies are going to be provided to address these issues. Research and innovation along this direction has provided a new vision of web that is called Web 2.0. Web 2.0 is a new approach to interact and work with available information in Internet. It is not a special middleware, neither a registered mark, but a new way of using the Network by innovative techniques and technologies.

Dion Hinchcliffe described in Web Services Journal [1] relevant features of Web 2.0:

- Web is built by little parts, which are independent. A single site is not flexible, it cannot be connected, it cannot be distributed and does not live for a long time.
- Self service and participation. Some examples are: tagging, ranking, trackback.
- Distribution. A single source represents a single point of failure. Furthermore single sources do not improve delivery and retrieval of relevant contributions.
- Facilities of Web 2.0 can be reused, changed, composed, discovered and their augmented value can be integrated again in any applications.

Statements mentioned above provide also a new vision of web users, whose role changes from consumer to participant. In fact, when information is delivered, users' interactions modify data related information, making users authors themselves. Data acquire a new identity, each time they are used or modified according one's goal. They do not depend anymore on their producer and on the context where they have been published, but they are self-contained.

This improves information sharing and allows collaboration. This approach allows to create new services,by reusing data and services themselves in a way that was not conceived by their original provider.

Some well promising technologies, which aim at representing the pillars of Web 2.0 are: Web Services, XML, RSS, Ajax, ... Ajax is a technology for developing web contents by JavaScript and XML. Web pages begin to work as desktop applications, which support complex interaction models, rather than static information. RSS allows users to get automatic updates of web information. Web content is filtered and readers can easily retrieve what is new in papers, web sites and other kinds of data. It represents a capability to select, and process again. information by Web 2.0 technology. Many web providers, such as Amazon, Google, Yahoo, eBay, Skype, make available open APIs to create new applications. Web becomes a software development platform. It can be used for management, marketing, communication, multimedia information and pure development.

We present in the following an example of application that can be extended by addressing all relevant issues of LC3 Project and Web 2.0, semantic, personalization an so on ... We focus on integration of Mobile Agent based multimedia delivery.

General objectives of our activities and the methodology that has been conceived are presented in section 2. In section 3 we discuss related work. Section 4 describes basic components of Service Oriented Architecture model of LC3 project. In section 5 our test-bed implementation to test adaptive multimedia delivery is presented. Section 6 provides details about software architecture and implementation of the prototype that has been developed. Finally conclusion is due.

2 Objectives and methodology

General objective of our work is to provide a methodology:

- to extract and represent any kinds of information resulting from data processing;
- to organize and exploit available information by a high level approach that involves expertise in application domain, but not a technological skill;
- to develop advanced facilities such as profiled or semantic based information retrieval.

Italian Digital Library (BDI) is designing a standard schema of Metadata Administration and Management (MAG[2]). MAG is a schema that has been conceived in order to provide a formal specification for collecting and storing digital data and their metadata. It is based on METS[b], which is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed by a XML schema language of the World Wide Web Consortium. The standard is maintained in Network Development and MARC Standards Office of the Library of Congress, and is being developed as an initiative of the Digital Library Federation. MAG schema is composed of following sections:

- **gen**: general information about project and digitalization process;
- **bib**: description of digital object;
- **img**: description of fixed images;
- **ocr**: information about optical character recognition.

Actually MAG does not allow representation of other kinds of information. At the state of the art it is not possible to associate semantics to data, to add georeferences or requirements for their correct utilization and so on. For this reason our first goal is to extend MAG and to pursue following objectives.

- Definition of a schema to represent information, that describes data missing attributes, like semantic annotation, georeferences, ...
- Definition of techniques for data processing an indexing based on metadata elaboration.

[b] http://www.loc.gov/standards/mets/

- Definition of a model to build new digital objects by selecting components or by composition of available ones.
- Automatic ontology creation and ontology based annotation to associate meaning to data.
- Semantic retrieval of contents.
- Adaptive delivery of multimedia contents

In the following we present intermediate results. We provide also a prototype implementation of a platform that allows us to experiment original techniques, for adaptive multimedia delivery, and to test different technological choices. The test-bed described into Section 4 has been used to test the integration of an original service for client profiling and content adaptation. A design of this component has been previously described in [3]. We present here its extension and an experience of integration into a prototypal architecture.

3 Related Work

Adaptive delivery depends on the capability to react not only to users input, but also to input (i.e. context) from users environment. For example, in learning environments, it allows both to support the utilization of heterogeneous devices for distant learning, and to personalize the learning process, according to a community context[4]. An example of context aware electronic tourist guide is described in [5].

A profiling mechanism is required in order to describe both users and devices, so that users' equipments, preferences and needs are defined. Several standards and solutions have been proposed [6, 7]. Adaptation is often based on available resources of client devices: display resolution, badwidth, processor speed and energy [8]. On the other hand XML based technologies (discussed in the following), such as OWL, RDF, DARPA (Agent Markup Language+Ontology Inference Layer), provide a support for integrating information from different domains, by facilitating semantic matching of elements[9]. Last evolution of web technologies leads to design context-aware services like agents, which adapt the way they behave according to the current context [9]. From an architectural point of view some distributed solutions have been proposed in [10].

We describe here a proxy based solution that exploits Mobile Agents technology [11]. Proxy architecture allows us to grant transparent integration of adaptive interfaces in web applications. Mobile Agents are the last evolution of Mobile Code based systems. They add mobility to a set of features, such as reactivity, proactivity, communication and social ability, which characterize common software agents systems. A mobile agent is a program able to migrate across a network, together with its own code and execution state. At client side agents' mobility allows for a local access to client information, and to peripheral devices, like GPS (Global Position Systems) for localization. They can be used also to reconfigure software profile of users' devices, by providing libraries and applications which can extend client capabilities. Agents' proactivity can be used to let the user be guided by the application itself, in order to leverage the discovery and exploration of available information. In fact mobile agents support both PULL and PUSH mechanisms to implement different interaction models, which are relevant in content and services exploitation, as it has been discussed in

[12]. At server side Mobile Agents facilities, such as cloning and migration, can be used to distribute, dynamically, the management of different sessions, among different nodes. They can be delegate to perform from remote intensive computations on client behalf.

4 Architecture design

Let us to introduce the basic blocks of our architecture. *Digital object* represents the main concept. A digital object is composed by a metadata and by related contents[13]. It is realized by a XML document that collects information about semantic, data organization, authoring, ..., of an abstract or real concept. Each time data is processed by annotation, encoding, part extraction, semantic extraction,... , a new metadata is produced and a new digital object is created. Digital objects can be results of many activities such as data digitalization and organization. Classification of data according to a new ontology can also produce a new digital object. Such a model is relevant to coordinate national and international activities, on data digitalization and on design and implementation of archives.

The methodology we are adopting consists in defining new XML elements, which will be added to MAG. Information consumers and producers are skilled on a specific application field. Their roles consists in defining concepts of a specific domain, that will be used to annotate digital objects and to query the CMS for retrieval purposes. This will be done using ontologies. An ontology formalizes a knowledge domain. It represents a scheme that is composed of entities, which are hierarchically organized, and of properties, which link them. Rules, preconditions and axioms can be defined too. Some relevant technologies, upon which is being to be built ontologies for the Semantic Web [14], and which will have been used in this work, are RDF and OWL. RDF (Resource Description Framework) [15] is a standard for producing metadata, which describe web resources, and makes it much easier their automatic management. Resources can be semantically annotated by defining statements in the form of subject-predicate-object expressions. It provides a flexible approach for representing data by a set of basic semantics. It exploits XML ability to define customized tagging schemes. OWL (Web Ontology Language) [16] extends RDF. It is a language for describing ontologies, to be used when information, which is contained in documenots, needs to be processed by applications. OWL can be used to explicitly represent meaning of terms in vocabularies and relationships between those terms. It is conceived to allow machines to perform useful reasoning tasks on web documents. In our design OWL is exploited to describe both ontology and annotations. For each object, a semantic annotation can be represented by: a link to the application ontology, concepts of the same ontology and properties. Properties belong to the current digital objects and link other concepts of the same ontology.

What has been said for semantic can be extended to any kinds of information. We mean that MAG can be extended to represent any kind of information.

Requirements, to be satisfied for data exploitation, can be defined by a link to a standard taxonomy that describes client profile [17](e.g.: CCPP,UAPROF ,...) and a set of capabilities which are needed to be provided. In order to allow storing, management and retrieval of digital objects we store them in a Content Management System (CMS), with advanced functionalities, for handling both data and metadata. In order to support interoperability with services provided by third parties, and to allow the utilization of CMS facilities by any applications, we support loosely coupled integration by adopting Service Oriented Architecture (SOA) model and Web Services technology. In our activities this choice makes easy the integration of prototypes developed by other partners, which implement their service autonomously, but agree on a common model of digital object.

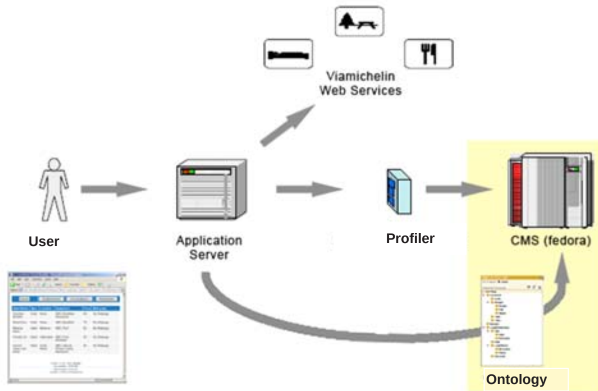


Fig. 1. System architecture

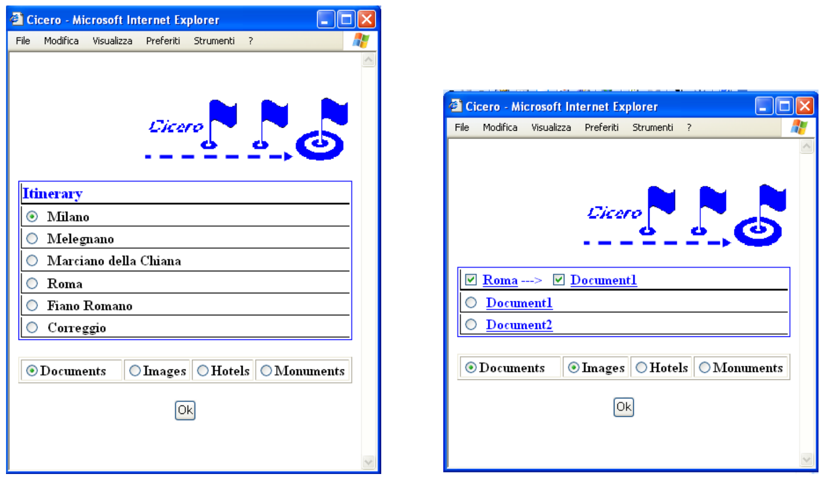
A significant case study is relevant to experiment techniques and technological solutions, which address the issues described before. We called our application Cicero, because it aims at representing a virtual guide, who is aware about the meaning of retrieved objects and about their relationships. Cicero is a web application that allows users to retrieve digital objects belonging to different categories in order to compose a multimedia tour between two cities. An user can choose a departure location and an arrival one. The application extracts a list of cities, which can be enjoyed along a possible itinerary, and supports an intelligent discovery of documents, monuments, images and hotels related to locations and interests. Digital objects can represent different kinds of multimedia contents. Discovery is performed composing a semantic query, which specify relationship between objects and locations. Users must be aware of the particular application field and not of software technologies. Functionalities provided by Cicero are:

1. Computation of an itinerary between two locations of interest.
2. Semantic retrieval of digital objects which are related to locations, to other objects and to any concepts defined by a specific ontology.
3. Easy solution to allow composition of semantic query.
4. Multimedia exploitation by heterogeneous devices.

5 Test-bed implementation

The test-bed described in this section has been conceived according to the SOA model. Web Service technology grants interoperability among components. The prototype integrates all components of a complete system: external services, original ones, a web application, ontologies and a repository with semantic based facilities for information retrieval.

In Figure 1 it is shown its software architecture. ViaMichelin Web Services have been used to compute an itinerary from departure location to destination. Users composes its query step by step. After he has chosen departure and destination he can select a city along the provided itinerary (Figure 2 (a)). Supposing to select Rome as location of interest, the application allows to search for objects belonging to a particular category: documents, images ... . As it shown in Figure 2 (b), search results are shown as a list of hyperlinks

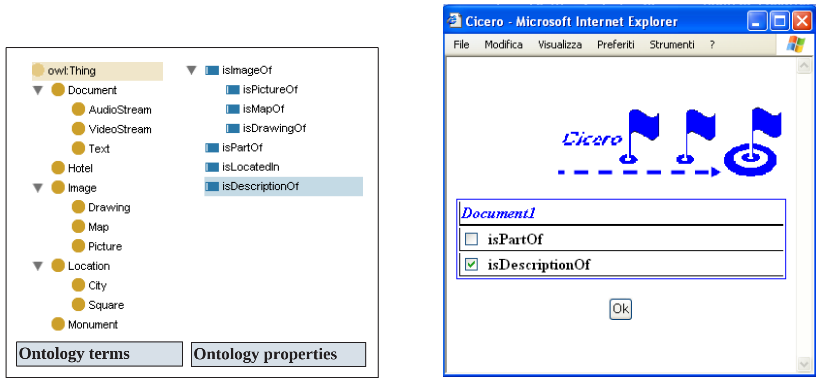


(a) Itinerary (b) Documents

Fig. 2. (a) Itinerary is automatically computed by ViaMichelin Web Services. (b) We can look for all documents which are semantically related to a location and with results of a previous query

and radio buttons. Hyperlinks can be used to retrieve available objects. Radio buttons are used to selected the correspondent object as a new term of the query. Next object, which the application is looking for, can be linked to *Rome* and to *Document1* according to the properties defined into the application ontology. The kind of object is chosen by checking a radio button at the end of the page. First line of Figure 2 (b) represents a query that is composed of two terms: *Rome* and *Document1*. Cicero ontology is shown in Figure 3 (a). Documents and locations are main classes. Multimedia documents can be audio and video streams, text documents or images. Locations can be cities, hotels, monuments, restaurants, historical places. Furthermore ontology defines properties of these entities, which correspond to real or to abstract concepts.

All checked elements will compose the next query. By clicking on a query element it is possible to select which properties are being considered. In Figure 3 (b) we can ask that *Document1* is part of a next document, or it could be a description of an image we are looking for. A Content Management System (CMS) allows to store digital objects and their metadata. Metadata will describe different kinds of information, such as author, version, location, encoding, ... Fedora CMS [18, 19] has been installed as content repository. It allows to create digital objects similar to the model described above and offers many useful facilities, even if some limitation to our theoretical model must be considered. In particular



(a) Application ontology (b) Property checks

Fig. 3. (a) Application ontology. It define resources and properties which can be used to compose the query. (b) A friendly web interface can be used to check which properties must be satisfied by query results.

Fedora allows to define a digital objects as a set of heterogeneous information. We can define a dublin-core[c] record for each object, different versions and different data streams. It is possible to store a semantic description of objects, but according to a proprietary built-in ontology that cannot be changed. Furthermore a reasoner that processes semantic query is available for objects retrieval. A web service interface is provided for most of all available facilities. Fedora RDF Ontology is described in Figure 4. Properties shown in Figure 5 can link two objects stored into the repository. Semantic description of a digital object consists in a list of RDF statements. A RDF statement is composed of three terms: a subject, a verb and an object. *Document1* and *Milano* are two digital objects of our repository and *isDescriptionOf* is a property that links them. Of course that property is part of metadata of *Document1*.

Fedora provides the user with a search engine that accepts ITQL queries. Our application translates a visual query composed by a web form into an ITQL query. Interactive Tucana Query Language (ITQL) is used to search for digital objects, whose properties, which are described in a XML file, satisfy the criteria expressed by the query. Queries are composed by three clauses: *select*, *from* and *where*. They must be used by specifying subject, verb and object, which must be found in desired data description. In order to exploit the reasoner provided by Fedora we rewrote the original OWL application ontology in terms of Fedora ontology.

Following query selects from the repository all documents describing *Milano* :

```
select $obj from <#ri>
where  $obj <fedora-rels-ext:isMemberOf>  <info:fedora/demo:document>
and    $obj <fedora-rels-ext:isDescriptionOf>  <info:fedora/demo:milano>
```

$obj represents all the objects we are looking for.

[c] The Dublin Core metadata element set is a standard for cross-domain information resource description (see http://dublincore.org/)

```
<rdf: RDF base="info:fedora/fedora-system:def/relations-external#">
   <rdf:Property rdf:ID="fedoraRelationship"></rdf:Property>
   <rdf:Property rdf:ID="isPartOf"></rdf:Property>
   <rdf:Property rdf:ID="hasPart"></rdf:Property>
   <rdf:Property rdf:ID="isConstituentOf"></rdf:Property>
   <rdf:Property rdf:ID="hasConstituent"></rdf:Property>
   <rdf:Property rdf:ID="isMemberOf"></rdf:Property>
   <rdf:Property rdf:ID="isSubsetOf"></rdf:Property>
   <rdf:Property rdf:ID="isMemberOfCollection"></rdf:Property>
   <rdf:Property rdf:ID="hasCollectionMember"></rdf:Property>
   <rdf:Property rdf:ID="isDerivationOf"></rdf:Property>
   <rdf:Property rdf:ID="hasDerivation"></rdf:Property>
   <rdf:Property rdf:ID="isDependentOf"></rdf:Property>
   <rdf:Property rdf:ID="hasDependent"></rdf:Property>
   <rdf:Property rdf:ID="isDescriptionOf"></rdf:Property>
   <rdf:Property rdf:ID="hasDescription"></rdf:Property>
   <rdf:Property rdf:ID="isMEtadataFor"></rdf:Property>
   <rdf:Property rdf:ID="hasMetadata"></rdf:Property>
   <rdf:Property rdf:ID="isAnnotationOf"></rdf:Property>
   <rdf:Property rdf:ID="hasAnnotation"></rdf:Property>
   <rdf:Property rdf:ID="hasEquivalent"></rdf:Property>
</rdf:RDF>
```

Fig. 4. Fedora RDF ontology

```
<info:fedora/demo:document1>
      <fedora-rels-ext:isDescriptionOf>
<info:fedora/demo:milano>
```

Fig. 5. RDF statement

6 Transparent client profiling and content adaptation

The original service integrated into the presented architecture is a proxy based platform for client profiling and content adaptation. During service exploitation it is completely transparent both to client applications and to web applications. It configures a content adapter according to a client profile. An adapter intercepts client requests and redirects them to a CMS. Responses are processed along the backward path to transform original data.

Using a PDA, or any other devices, users are able to access web applications and to download multimedia contents. Client profile is described by an XML document that extends UAPROF specifications. It can be acquired as follows:

1. Not automatic profiling. User provides device information filling a form or specifying an URL that links a specific document.

2. Automatic and static profiling. A document, or its URL, is automatically provided by the client when user signs in.

3. Automatic and dynamic profiling. A client application or mobile code, which is uploaded on the client device, evaluate dynamic parameters such as battery status, network bandwidth, available memory, cpu workload,... , which complete the static profile. Dynamic profiling can be performed also during service exploitation in order to provide a real time adaptation meanwhile session parameters are changing.

In the latter case, when mobile agents execution is supported, we provide advanced reconfiguration and profiling capabilities [20]. Mobile agents based facilities are also exploited to allow for web services composition and delivery [21]. Client devices, are reconfigured by hosting incoming software agents, which are able to read system properties. A coarse grain profile is build by exploiting java APIs, but native library can be download to exploit specific resources. For example battery level can be read by invoking OS APIs, and user position can be provided by accessing GPS functionalities, when they are available. Of course the same approach can be exploited to extend a client profile by dispatching agents, which implement new capabilities for service exploitation.

At server side Proxies are implemented by Mobile Agents themselves. In Figure 6 the class diagram of an Agent Proxy is shown. Each agent execute in parallel two behaviors.

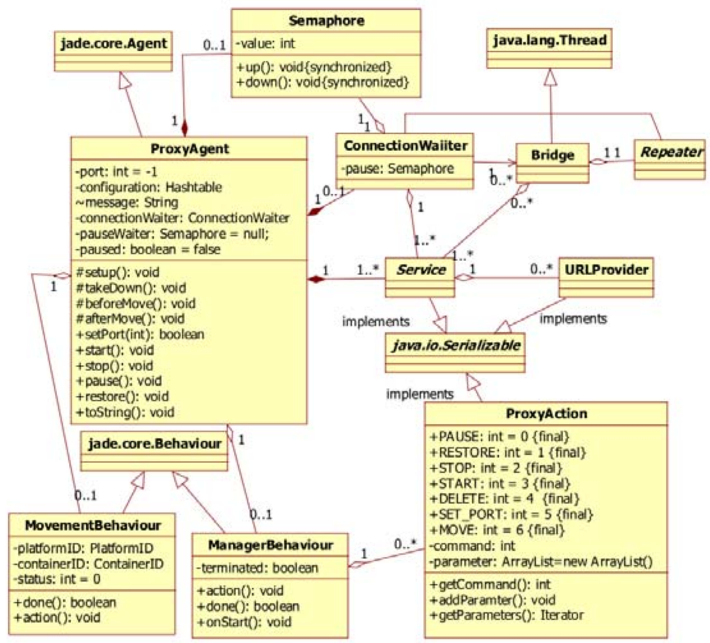


Fig. 6. Class diagram of an Agent Proxy

Each Proxy Agent is delegated to grant the best Quality of Service to clients by content brokering and content adaptation. It uses profiles to analyze compliance between client capability and requirements for data exploitation. It is able to configure special adapters which elaborate data before they have been delivered. At the same time agents wait for management directives. They can be asked to suspend their activities, to change their behavior or to move to other nodes.

In Figure 7 it is shown how profilers and adapters work in our prototype.

A proxy server intercepts http requests from a browser. It looks for the x-wap-profile header. If it has not been transmitted, user is asked for a manual profiling, by a web interface, or can skip this phase and will access without any adaptive interface. We installed a proxy on client devices to add a profile descriptor in each http request when it is not done automatically.

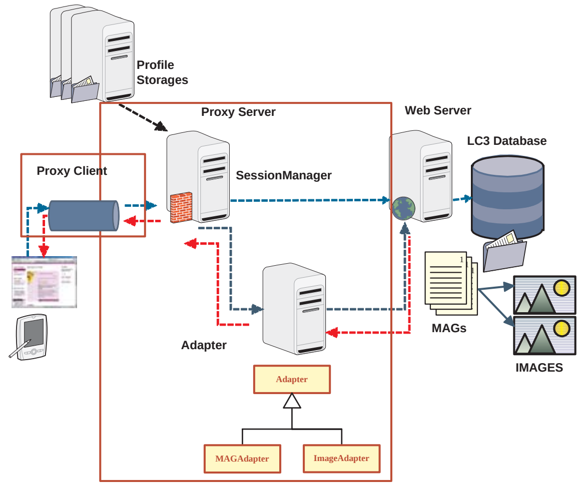


Fig. 7. Profiler architecture

This field can refer to a static UAPROF file or to an application that is able to collect dynamic parameters by a direct interaction with the device.

It is also responsible for binding profiles and sessions. It manages a list of cookies, (the application ones, it does not create new cookies), and, for each of them, it downloads and store the specific profile. A profile contains static parameters and dynamic ones, whose values can be valid for limited time. Incoming requests are forwarded according to the kind of resource that is going to be retrieved.

According to a specific resource we can have different actions:

1. *nothing*: resource/service exploitation does not need to be adapted;

2. *redirection*: there are more versions of that resource, so it needs a redirection to the correct path;

3. *dynamic adaptation*: resource needs to be adapted, so requests will be tunneled through specific adapters;

Of course a hybrid solution, which is cache based, can be exploited to improve performances, if the third case is expensive.

Let us consider a real case for LC3 project to show how dynamic adaptation works.We suppose we are retreving, by the application described in the previous section, a digital object resulting from digitalization and described by its MAG.

An example of MAG is shown in Figure 8. It is an XML document with Dublin Core and structural information [2]. It links several images, and different version for each of them.

An industrial partner of the project, provides a web access to MAGS (metadata) and to referenced resources (images resulting from digitalization).

In this case we developed two adapters. First one adapts visualization of MAG. Second one carries out an image resize and encoding, when it is required. An adapter is a proxy that intercepts client requests and transforms contents, which are returned by a Content

```
<metadigit>
<bib level="a">
   <dc:identifier>IT_ASEAO_Cagliari_B_01_F01</dc:identifier>
   <dc:title>Societ{\'a} Vinalcool (gi{\'a} Birraria Ichnusa)</dc:title>
   <dc:creator>S.E.S - Societ{\'a} Elettrica Sarda</dc:creator>
   <dc:date>1920</dc:date>
   ...
</bib>
<img>
  <sequence_number>1</sequence_number>
  <nomenclature>B_01_F01_0001</nomenclature>
  <usage>1</usage>
  <file Location="URL" xlink:href="./B_01_F01/tif/B_01_F01_0001.tif" />
  <md5>d872b68b12a367c741ab25e068fbedfd</md5>
  <filesize>29657844</filesize>
  <image_dimensions>
    <niso:imagelength>3814</niso:imagelength>
    <niso:imagewidth>2592</niso:imagewidth>
  </image_dimensions>
  <format>
    <niso:name>TIF</niso:name>
    <niso:mime>image/tiff</niso:mime>
    <niso:compression>Uncompressed</niso:compression>
  </format>
  <datetimecreated>2008-01-09T17:52:16</datetimecreated>
  <altimg imggroupID="jpeg300">
    <usage>2</usage>
    <file Location="URL" xlink:href="./B_01_F01/jpeg300/B_01_F01_0001.jpg" />
     ...
  </altimg>
  ...
  </metadigit>
```

Fig. 8. MAG example

Management System. Each adapter retrieves client information and requirements for data exploitation. It computes a matching between them and notifies the differences.

As only a subset of relevant parameters, which characterize the client profile, are relevant to adapt a particular kind of multimedia, an adapter can ask for a particular element of the XML profile to Session Manager of Figure 7.

A matching between profile description and service (or data) requirements could return one of following results:

1. All XML elements of data requirements can be found in the client profile, with compliant values. No adaptation is required, content can be delivered as it is.

2. XML elements are not present in the client profile, or their values are different, but a solution for content adaptation is not supported.

3. XML elements are not present in client profile, or their values are different, but a solution for content adaptation is supported.

In the second case we can chose to return the content as it is, or to forbid the access. In the third case adaptation is performed. Composite adaptation can be built by a tunneling of request across different proxies. An example is described in [3], where the first proxy along the path distributes the workload, and the second one adapts contents. Proxies can be replicated on different machines and requests can be forwarded according to load balancing strategies (Readers can refer to [3] also for quantitative results).

In a typical interaction among components of our architecture, at each http request the profiler looks for its cookie in a list. If that cookie is not found a new item is created and the value of x-wap-profile header is used to download the profile. Profiler returns an XML description of client capabilities, together with dynamic session parameters if they are available.

Profile Manager checks each request and, according to configuration parameters (resource path, resource type, ...), forwards it to a next proxy.

Each proxy adapter, providing the cookie value, which identifies a particular session, asks for relevant parameters of client profile needed for adaptation. At the same time it forwards the request to a next proxy or to the content provider. When content has been received adaption is performed and results are returned on the backward path. Each adapter asks in parallel for contents, to a next Proxy or to a content Provider, and for those parameter, which are relevant for adaptation, to the Profile Manager.

MAG adapter is a proxy that transforms the XML file by an XSLT transformation. It returns an html page that presents a selection of available images for that object. In particular, according to the screen size of the device, it shows only one link of an alternative image with a maximum resolution among supported ones. If all versions are bigger that the screensize, the littlest one is chosen. When a client asks for an image in that list, its request is forwarded to an ImageAdapter that compares image resolution with screensize, and performs a resize if it is necessary. Of course many other examples can be presented, and a lot of them will be designed according to LC3 project requirements.

7 Conclusion

We provided a methodology for information management based on an original meta-model of digital objects. We described a platform that allows to test our methodology in a Service Oriented Architecture and we presented its prototypal implementation. Mobile Agents technology has been exploited, both at client side and at server side, to improve quality of service of multimedia delivery. This work is part if LC3 research activities. We aim at providing knowledge and innovative technologies to support semantic management of multimedia archives and delivery of information by exploitation of digital objects. Objectives are definition of new approaches and development of new functionalities to enhance availability, knowledge and usability of national and international cultural properties. The presented architecture addresses project requirements and can be exploited as a platform that allows us to integrate and test original components, which will be results of different research activities. In particular multimedia adaptation is going to be exploited in different way. In fact a personalized delivery of the same content, or service, can provide added value to applications and can enhance usability in different conditions.

References

1. O'Reilly, Tim (2007): What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Published in: International Journal of Digital Economics 65 (2007): pp. 17-37.

2. A. Scolari, M. Pepe, M. Messina, C. Leombroni, G. Cirocchi, G. Bergamin, MAG reference and MAG Schema v.2.0, 2005, *http://www.bncf.firenze.sbn.it/progetti/mag/*
3. 5. R. Aversa, B. Di Martino, N. Mazzocca, Venticinque S. (2008). An Agents Based Middleware for Personalization of Multimedia Service Delivery in SOAs. JOURNAL OF MOBILE MULTIMEDIA. vol. 4, pp. 2-18 ISSN: 1550-4646.
4. Klamma R., Spaniol M. and Cao Y., Community Aware Content Adaptation for Mobile Technology Enhanced Learning, Innovative Approaches for Learning and Knowledge Sharing (2006), pp. 227-24.
5. Cheverst K., Davies N., Mitchell K., Friday A., and Efstratiou C. Developing a context-aware electronic tourist guide: some issues and experiences. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2000.
6. M. Nilsson. UAProf An overview. Available from: http://www.w3.org/2002/02/DIWS/presentations/nilsson/nilsson.pdf, 2002.
7. World Wide Web Consortium. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. Available from: http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115, 2004.
8. Yung-Hsiang Lu, David S. Ebert, and Edward J Delp, Resource-Driven Content Adaptation,
9. Tao Gua, Hung Keng Pung, Da Qing Zhang, A service-oriented middleware for building context-aware services, Journal of Network and Computer Applications, Vol. 28, No.1. (2005) pp. 1-18.
10. M. Colajanni and R. Lancellotti, System Architectures for Web Content Adaptation Services. In IEEE Distributed Systems online, Vol.5, No.5, IEEE Communications Society, May 2004.
11. Pham, V.A., Karmouch, A.: Mobile software agents: An overview. IEEE Commun. Mag. 36(7), 2637 (July 1998)
12. Cheverst, K., Mitchell, K., et al.: Investigating Context-aware Informaiton Push vs. Information Pull to Tourists. In: MobileHCI 2001 workshop on HCI with Mobile Devices, Lille, France (2001)
13. California Digital Library, CDL Guidelines for Digital Objects,September 2006 http://www.cdlib.org/about/publications/CDLObjectStd-2001.pdf
14. Berners-Lee T., Hendler J. and Lassila O., The Semantic Web, Scientific American, (2001).
15. RDF (Resource Description Framework): www.w3.org/RDF.
16. Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, et al. OWL Web Ontology. Language reference. W3C Recommendation (2004). www.w3.org/TR/owl-ref/
17. W3C consortium, Composite Capability/Preference Pro les (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation 15 January 2004 http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/
18. Staples, Thornton, Ross Wayland and Sandra Payette, The Fedora Project: An Open- source Digital Object Repository System, D-LIb Magazine, April 2003.
19. Carl Lagoze, Sandy Payette, Edwin Shin, Chris Wilper, Fedora: An Architecture for Complex Objects and their Relationships, in Journal of Digital Libraries, Special Issue on Complex Objects, Springer 2005.
20. 5. R. Aversa, B. Di Martino, N. Mazzocca, Venticinque S. (2008). An Agents Based Middleware for Personalization of Multimedia Service Delivery in SOAs. JOURNAL OF MOBILE MULTIMEDIA. vol. 4, pp. 2-18 ISSN: 1550-4646.
21. Rocco Aversa, Beniamino Di Martino, Salvatore Venticinque and NicolaMazzocca, "A Mobile Agents based Infrastructure to deliver value added services SOAs", in Scalable Computing: Practice and Experience, Special Issue: Practical Aspects of Large-Scale Distributed Computing, Edited by Dana Pectu, Vol. 8, N. 3, September 2007, ISSN 1895-1767, pages 313-324