Modeling Service-Based Multimedia Content Adaptation in Pervasive Computing

Girma Berhe
Lab. d'InfoRmatique en Images et Systèmes d'information (LIRIS)
INSA de LYON
20 Avenue Albert Einstein
69621 Villeurbanne, France
+33 4 72 43 63 48
girma.berhe@lisi.insa-lyon.fr

Lionel Brunie
Lab. d'InfoRmatique en Images et Systèmes d'information (LIRIS)
INSA de LYON
20 Avenue Albert Einstein
69621 Villeurbanne, France
+33 4 72438392
Lionel.Brunie@insa-lyon.fr

Jean-Marc Pierson
Lab. d'InfoRmatique en Images et Systèmes d'information (LIRIS)
INSA de LYON
20 Avenue Albert Einstein
69621 Villeurbanne, France
+33 4 72 43 88 97
Jean-Marc.Pierson@insa-lyon.fr

ABSTRACT

Pervasive computing applications allow users to access information from anywhere while traveling and using variety of devices. Heterogeneity and limitation of resources involved in this application demand adaptation of content according to the current context (device, user, network etc.). The dynamic nature of adaptation mechanisms together with emerging opportunities of Web Service technology provides new approach of adaptation which is service-based. While this approach would provide a valuable service for the end customer, the service provider, and the content provider, it is important to have an architectural framework which is simple, scalable, flexible and interoperable. Moreover, in order to provide a complete service-based content negotiation and adaptation solution, we must have a model, or a tool, that allows defining environmental constraints, mapping them to appropriate adaptation service requirements and finding an optimal service configuration.

In this paper, we present service-based content adaptation architecture, enabling the use of third-party adaptation services and a novel content negotiation and adaptation model. The proposed architectural framework is validated through a prototype.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]

General Terms

Algorithm, design, performance

Keywords

Multimedia content delivery, content adaptation services, media transformation, pervasive computing.

CF'04, April 14–16, 2004, Ischia, Italy.
Copyright 2004 ACM 1-58113-741-9/04/0004…$5.00.

1. INTRODUCTION

Pervasive computing is the ability to extend applications and services normally conducted on office computers to handheld and wireless devices enabling anywhere, any time, with any device access to information systems. Pervasive computing environment involves a variety of devices with different capabilities in terms of processing power, screen display, input facilities and network connectivity which can vary between wired and different types of wireless links. For pervasive computing to succeed in general, fundamental issues in different areas such as network, data management and delivery and security among others, have to be well addressed [2].

Heterogeneity and mobility of devices poses new challenges for information delivery applications in this environment. To meet the demands in this heterogeneous environment, it is necessary for the information to be customized or tailored according to the user's preferences, client capabilities and network characteristics [11]. This tailoring process is called content adaptation which may include format transcoding (e.g. XML to WML, JPEG to WBMP), scaling of images as well as video and audio streams, media conversion (e.g. text-to-speech), omission or substitution of document parts (e.g. substitution of an image by a textual representation), document fragmentation, language translation, etc. Providing complete list of content adaptation types is not possible and includes others in the future. This dynamicity is one of the reasons that makes difficult to develop a single adaptation system that can accommodate all types of adaptations hence third-party adaptation services are important.

On one hand, Web Services are becoming popular technologies for publishing various services on the Internet [22]. On the other hand, there is a trend of developing content adaptations as value-added services [6, 20]. However, the linkage between them has not been yet explored, that is using Web Services for the purposes of developing content adaptation services. Moreover, deciding what adaptations to perform and which services to select in order to maximize performance and minimize costs can be a complex constraint satisfaction problem.

The objective of our work is to bridge these two technologies, Web Service and content adaptation, in a common dynamic and extensible framework in order to have scalable, flexible and extensible architecture and defining a negotiation and adaptation model that enables the adaptation engine to select an optimal adaptation path.

The rest of this paper is organized as follows. In section 2, we present briefly some of the related works followed by motivation of our work (section 3). Section 4 describes the system architecture and its components. In section 5, we present a negotiation and adaptation model and formal definition of cost and user requirement constraint problem. Description of the prototype implementation of the proposed architecture with language translation as an example service is presented in section 6. Section 7 provides short discussion of our work. In section 8 we conclude the paper and list future works.

2. RELATED WORKS

One of the important issues with content adaptation is the location where the adaptation operation should be done. With respect to location of adaptation operation, adaptation frameworks can be categorized into three groups [13]: server-side approach, proxy-based approach and client-side approach.

In the case of server-side approach (e.g [18], [19]), the functionality of the traditional server is extended by adding content adaptation. In this approach both static (off-line) and dynamic (on-the-fly) content adaptation can be applied and better adaptation results could be achieved as it is close to the content; however clients experience performance degradation due to additional computational load and resource consumption on the server [10].

In proxy-based approach (e.g. [10], [16]), a proxy that is between the client and the server, acts as a transcoder for clients with similar network or device constraints. The proxy makes request to the server on behalf of the client, intercepts the reply from the server, decides on and performs the adaptation, and then sends the transformed content back to the client. In this approach there is no need of changing the existing clients and servers. The problem of proxy-based adaptation approaches is most of them focus on particular type of adaptation such as image transcoding, HTML to WML conversion, etc. and they are application specific. In addition, if all adaptations are done at the proxy it results in computational overload as some adaptations are computational intensive and this degrades the performance of information delivery like the server-side approach.

Client-side approach (e.g [13], [23]) can be done in two ways; performing transformation by the client device or selection of the best representation after receiving the response from the origin server. This approach provides a distributed solution for managing heterogeneity since all clients can locally decide and employ adaptations most appropriate to them. However, adaptations that can benefit a group of clients with similar request can be more efficiently implemented with server-side or proxy-based approaches. Furthermore, all of the clients may not be able to implement content adaptation techniques due to processor, memory resource constraint and limited network bandwidth [7].

The above three approaches do not deal the problem of content adaptation from service perspective that can be commercialized and utilized by user, content provider or other service providers (like Internet Service Providers). Introducing content adaptation as a service (service-based adaptation approach) distributes the activities and results in performance enhancement especially for computational intensive applications such as key-frame extraction as it is done by specialized adaptation services. For example, a server that handles only language translation is inherently more efficient than any standard web server performing many additional tasks [9]. It also opens new opportunities to service providers as additional revenue.

However its is very important to have an architectural framework to enable content delivery system to incorporate such functionalities and a mechanism able to configure various services required to support the constraint by selecting suitable services from a list of those available and decide on the most appropriate service configuration. There are few works done in this direction. For example, works in progress such as the Open Pluggable Edge Services architecture (OPES) [20] and the Internet Content Adaptation Protocol (iCAP) [9] are closely related to our proposed approach. The OPES work deals with standardizing mechanisms to extend intermediaries with application-specific value added services and provides a standard way for the content provider and/or third parties to offer services. This work focuses at general level and does not address in depth the issues of content adaptation.

ICAP distributes Internet-based content from the origin servers, via proxy caches (iCAP clients), to dedicated iCAP servers [9]. For example, simple transformations of content can be performed near the edge of the network instead of requiring an updated copy of an object from an origin server, such as a different advertisement by a content provider every time the page is viewed. Moreover, it avoids proxy caches or origin servers to perform expensive operations by shipping the work off to other (iCAP) servers. However, it only defines a method for forwarding HTTP messages, that is, it has not support for other protocols and for streaming media (e.g. audio/video) and only covers the transaction semantics (how do I ask for adaptation?) and not the control policy (when am I supposed to ask for which adaptation from where?)

3. MOTIVATION

Most of the solutions and architectures proposed to help in designing adaptive multimedia systems for heterogeneous devices are limited to specific applications [15] and can not be extended to large scale systems like MediGrid [1]; a project we are currently involved. In this project, content adaptation plays major role in providing accessibility of patient data from any location with any type of devices. For example a physician can have access to his/her patient's record with PDA or any other device. Therefore, content adaptation is required for at least the following reasons: a) the user's device may not have the necessary hardware or software capabilities to present different types of data possibly multimedia b) the bandwidth connection for the devices such as PDA may not support large volume of data c) accessibility of data based on the user's role may be important for the purpose of security.

One of the problems of content adaptation is that it is difficult to generalize the procedure of adaptation and provide it by single application or system that is why existing adaptation approaches deal with simple or one dimensional adaptation (e.g. dropping frames or layers of a multi-media stream) [6]. Dynamism and diversity of content adaptation mechanisms (see Figure 1.) combined with the opportunities came to exist like Web Service technologies, open new approach of content adaptation which is service-based. In this approach adaptations are done by third-party services on the fly. These services can be accessed through subscription by users or content providers. To realize this approach at least the following things must be addressed well:

- need of flexible and scalable architectural framework to incorporate easily third-party services
- depending on the complexity of the content and context of the environment, it may be necessary to apply multiple adaptation techniques on the data before it reaches the user (e.g. changing a text in French to English, extract summary of the text and then translate the text to audio). Furthermore, one or more services could provide similar adaptation service, for example, video adaptation with different size, cost and quality values. In order to determine adaptation services suitable for the client or the environment context in general with some quality of services, a formal model of content negotiation and adaptation is important.

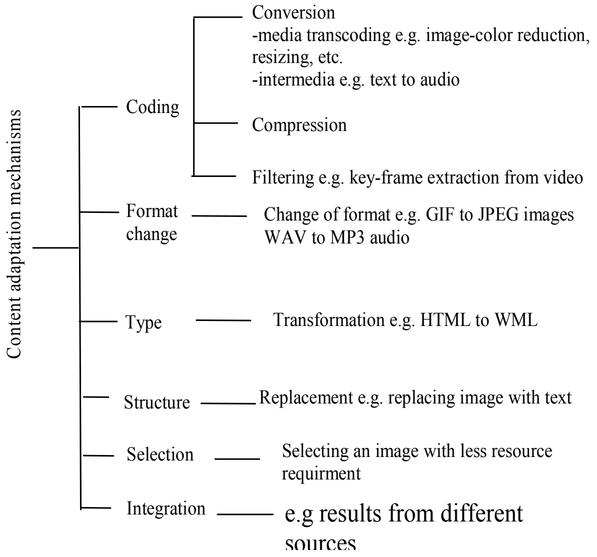

Content adaptation mechanisms
Coding
Conversion
-media transcoding e.g. image-color reduction, resizing, etc.
-intermedia e.g. text to audio
Compression
Filtering e.g. key-frame extraction from video
Format change
Change of format e.g. GIF to JPEG images WAV to MP3 audio
Type
Transformation e.g. HTML to WML
Structure
Replacement e.g. replacing image with text
Selection
Selecting an image with less resource requirment
Integration
e.g results from different sources

Figure 1. Content adaptation mechanisms.

4. SYSTEM ARCHITECTURE

The system architecture has to serve as a flexible platform for efficient exchange of message and data between different components. We propose an architecture which consists of the following components (see Figure 2.): local proxies, content proxies, adaptation service proxies, client profile repositories (user and device) and adaptation service registry.

- Local Proxies (LP): intercept user request and server response and initiate adaptation services
- Content Proxies (CP): retrieve content and its profile and forward to local proxies
- Adaptation Service Proxies (ASP): perform content adaptation on behalf of the user or content provider and are implemented as Web Services
- Adaptation Service Registry (ASR): stores service profile.
- Client profile repositories: store user and device profile

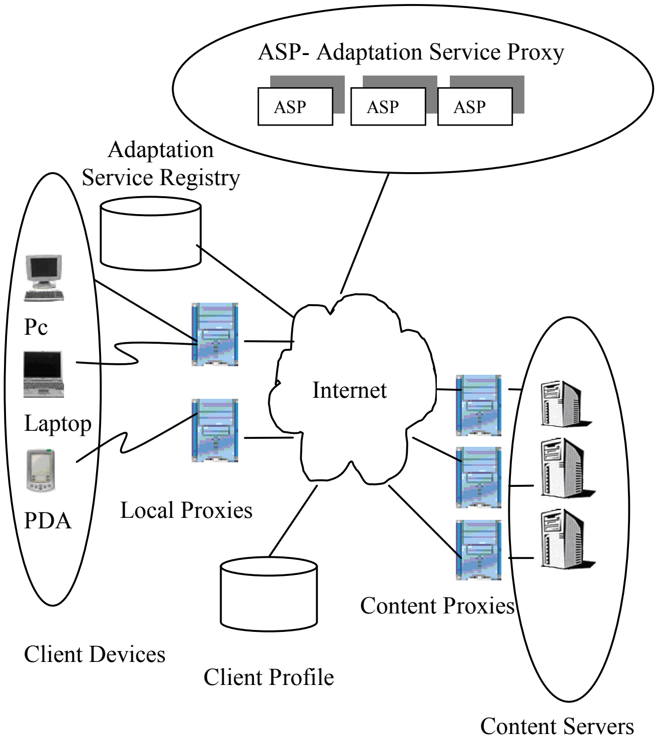


Figure 2. Distributed service-based content adaptation architecture.

4.1 Profile Information

For efficient and adequate content adaptation, the availability of content and usage environment (or context) description is essential since content and context descriptions (also called metadata or profile) provide important information for the optimal control of a suitable adaptation process. In our architecture four types of profiles are identified: content or media, user, device, network and service.

- Content or media profile represents object description (meta-data) and characteristics. It includes:
  - o media features such as type, format, size, color depth etc. For example image (width =600, height=350)
  - o available modalities or content version. For example an X-ray image might have textual description.
  - o location where the media is stored
- Client profile represents user profile and device profile
  - o User profile includes preferences such as display preference, language preference, media type (image, text), response time etc.
  - o Device profile composed of hardware (display, memory, storage capacity, and CPU among others) and software

capabilities. For example, PDA (width=240, height=320). Device profile can be gathered from the user or retrieved from profile repository of Vendor Company using device identification.

- Network characteristics such as bandwidth and latency can be gathered from dedicated network monitors [13].
- Service profile stores information about adaptation services such as service type, location, supported media formats, network connection, price, status (on/off), performance etc.

Content profile can be calculated on-the-fly or stored with content at the content server. The later approach avoids characterization of the object during the request and thus optimizes the processing. Client profile can be stored anywhere in the network [14]: in the client side, in the server side, in a common profile repository, etc. Putting the profile on the client side is not recommended since pervasive devices have in general less computational and storage capacity. It is not as well a good choice to put it on the server side due to lack of interoperability and flexibility. Thus, in our architecture we decided to store the profile in a server (profile repository) and can be accessed through network. Centralizing profile information helps to address the problem of user and terminal mobility, however for the sake of efficiency; profile information can be cached at the local proxy. User/device profiles are represented according to the W3C composite Capabilities/Preferences Profile (CC/PP) specification [12]. Service profiles are expressed in the Web Services Description Language (WSDL).

4.2 Adaptation Service Proxy

Now a day, both content providers and content consumers are increasingly looking for additional services beyond basic content delivery. They are interested in services that operate on and provide additional value to the content to be delivered. Examples of services include dynamic content assembly, content personalization, virus scanning, and content adaptation for different device types [17]. In the case of content adaptation, the use of services is very important as it can handle diverse adaptation mechanisms (see Figure 1.) and provides better adaptation results. The service approach also gives possibility of parallel adaptation since each adaptation task could be executed by different adaptation services.

The adaptation service proxy is the part that provides adaptation service for content delivery. We suggest implementation of adaptation services using Web Services due to their capabilities to address the issues of flexibility, extensibility and interoperability and for the purpose of simplicity. It also provides possibility of including new adaptation services which might be developed in the future that provide more efficient adaptations. The Web Services will be implemented as call out services. Their openness and support of direct interactions with other software agents using XML based message exchange via internet-based protocols is making Web Services tools of the infrastructure choice for publishing various services like language translation [21, 3, 20].

4.3 Adaptation Service Registry

The adaptation service registry (ASR) is a server that allows for the look up of adaptation service providers interfaces and stores information (service profile) such as service type, location, supported media formats, network connection, price, status (on/off), performance etc. Performance of an adaptation service is measured in terms of throughput, latency, execution time, and transaction time. Higher throughput, lower latency, lower execution and faster transaction times represent good performing adaptation service.

In relation to Web Service, a customer sends a request to UDDI server to discover possible Web services that can provide the demanded services. UDDI is a mechanism by which service providers uses to publish their services and clients use to discover the services. In the future there will be a number of Web Services for different application areas or services. For the purpose of facilitating the task of discovery and usage of Web Services, UDDI servers can be organized based on particular domain or service category. For example, adaptation Web Services can use a specific UDDI server to publish their services and user of the service can consult this server. In our architecture, adaptation service registry is such type of UDDI server.

4.4 Local Proxy

The tasks performed by local proxy (LP) are retrieving client requests and profile, analyze client and content/media profile, plan adaptation strategy, integrate and forward the adapted content to the user. It also caches adapted content for further request. Discussion of cache is out of the scope however in the future techniques of collaborative caches developed in the MediGrid project [5] will be incorporated for the purpose of cache exchange protocol between local proxies.

One of the main components of the LP is Content Negotiation and Adaptation Module (CNAM). CNAM has two main parts: the adaptation manager and the adaptation path selector. The adaptation manager controls the processing of client profile, content profile and network characteristics and produces adaptation constraints. The approach proposed in this paper uses adaptation services hence the adaptation constraints will be mapped to suitable adaptation service (AS) based on some pre-defined rules. It is possible that more than one adaptation services would be activated and/or an adaptation service activates another adaptation service, hence it is important to develop path selection strategy so that an optimized adaptation path is activated (see Section 5). Computation and activation of an adaptation path is handled by the adaptation path selector.

4.5 Content Proxy

The content proxy (CP) provides access to content server, formulates the user request to source format, performs generic adaptation and caches generic documents. It also manages and provides content/media profile to local proxy and does general adaptation such as anonymity (security). Upon receiving user's request, it retrieves content and its meta-data and sends to local proxy.

4.6 Interaction Protocols

In architectures like the proposed one, flexibility and simplicity depends on the mechanisms and tools used for interaction between the components (see Figure 3.). For the sake of openness and scalability, standard protocols of Web Services and socket based communications are utilized. An adaptation service discovery protocol (ASDP) which is basically UDDI is used to get list of services from adaptation service registry. For passing content to adaptation service, SOAP or ICAP protocol can be used.

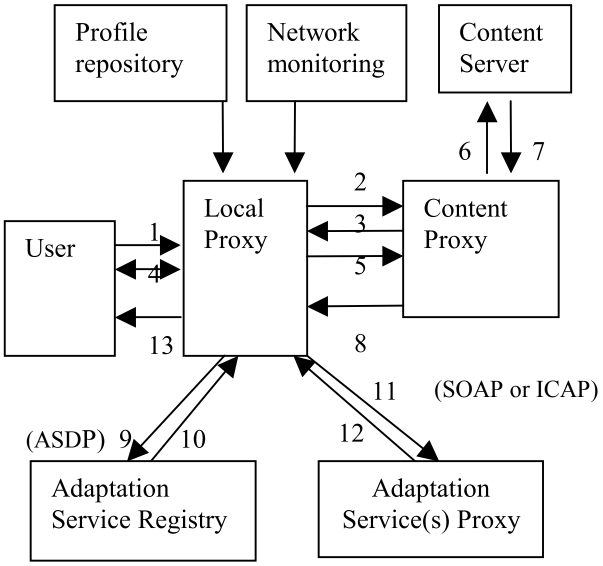


Figure 3. Interaction protocol.

1. A client sends a request (in the form of meta-data) for data to local proxy. At this point, the LP gathers profile information (user, device, network etc.).
2. If no cache hit the local proxy forwards the request to the content proxy.
3. The content proxy returns content profile (meta-data) response to local proxy.
4. The local proxy negotiates with the user to filter the request
5. The local proxy forwards the adapted request to content proxy.
6. If no cache hit, the content proxy forwards the request to content server.
7. Content server retrieves the data and delivers to content proxy.
8. Content proxy delivers the data to local proxy.
9. LP requests for list of adaptation services
10. The ASR provides list of adaptation services
11. LP forwards the data to Adaptation Service Proxy.
12. After the Adaptation service performs the adaptation, it sends back adapted data to the local proxy. The local proxy may perform some integration on the result(s).
13. The adapted data is delivered to the client and it may also be cached in the local proxy.

5. NEGOTIATION AND ADAPTATION MODEL

The problem of path selection strategy can be stated as follows: given a set of adaptation operators and their logical links and given a set of adaptation services; the best mapping of the operators, required to support the request's constraints, onto the adaptation services that supports them must be computed. We can say that selection strategy of content negotiation and adaptation is an optimization problem. Therefore, it is important to incorporate formal representations and mathematical model in solving the problem.

The model consists of media/content source and its profile, adaptation constraints (user, device and network), media adaptation operators, adaptation services, and adaptation service configuration and adaptation path. In our model a media source is the content delivered to the user and it can have one of the modalities (image, video, audio, text).

Definition 1: Adaptation constraints

An adaptation constraint defines conditions that must be satisfied to deliver the content to the user. It is a logical relation among profile values. We consider three categories of profile: media, environment and service profile. The media profile contains detailed characteristics of the media. Environment profile consists of information about the user preferences and acceptable level of QoS, device capabilities and network conditions. Service profile provides information about the adaptation service. The set of adaptation constraints will be noted C.

Media profile= (color depth, size, dimension, modality, format, language, etc.)

User profile= (color perception, scaling, language, maximum cost, response-time, etc.)

Device profile= (color depth, screen size, memory, processor speed, videoCapability, audioCapability, imageCapability, modality, encoding, etc.)

Network profile= (bandwidth, latency, etc.)

Service profile= (description, input, output, condition, type, reduction, charge, etc.)

Adaptation constraints can be described as lower or upper bound limit value, e.g. c_1 = (media size < =device size) or value checking e.g. c_2= (VideoCapability=Yes). Condition c_1 limits the media dimension to fit the device screen size and condition c_2 verifies video capability of the device.

Definition 2: Media adaptation operators

A media adaptation operator (or simply operator) is an abstraction of media adaptation mechanism or transformation. Effects of adaptation operators include media size reduction, media format change, media modality change, and media language translation, etc. The set of adaptation operators will be noted T.

Adaptation operators include (but not limited to):

-compression

-decompression

-scaling

-color depth reduction

-conversion (format)

-translation (modality)

-extraction, summary etc.

Definition 3: Adaptation services

An adaptation service (or simply service) is third-party service hosting one or more adaptation operators. The set of adaptation services will be noted S.

Formal representation

We define adaptation operator set T, adaptation constraint set C and adaptation service set S as follows:

T= $\{t_i \mid 1<=i<=n\}$, n represents the number of supported media adaptation operators

C= $\{c_i \mid 1<=i<=m\}$, m represents the number of constraint conditions

S= $\{s_i \mid 1<=i<=k\}$, k represents the number of adaptation services

Two mapping functions: operator and service are defined using the above representations.

Operator mapping function

It is a rule-based operation that associates adaptation constraints C to a set of media adaptation operators T.

Formal representation

f: C→ρ(T), where ρ(T) is power set of T.

$c_i \rightarrow T_i \subseteq T$

e.g. c_1 = (media size < =device size), c_2= (VideoCapability=Yes)

If the value of c_1 is not true, adaptation operator(s) should be applied on the media. Let us say that an operator t_1 = scaling. Therefore, f (c_1) = $\{t_1\}$ = T_1.

There are cases in which a constraint is mapped to more than one operator. For example, a network bandwidth constraint c_2 may be mapped to compression and scaling operators. Let us assume that, t_1 and t_2 are scaling and compression operators. In this case, f (c_2) = $\{t_1, t_2\}$ =T_2.

In general, operator set T_c for a user request with constraint R_c is given as

$T_c = \bigcup_{i=1}^{R} T_i$, where f (c_i) =T_i, $c_i \in R_c \subseteq C$ and R is the cardinality of R_c.

Service mapping function

The output of the operator mapping is the set of adaptation operators $T_c \subseteq T$ that should be applied on the media to deliver to the user. The next step is to associate T_c to set of adaptation service groups S_c that support each operator.

Formal representation

h: T→ρ(S), ρ(A) is power set of A.

$t_i \rightarrow S_i \subseteq S$ since an operator might be provided by several adaptation services,

e.g. $h(t_1)=S_1=\{s_2, s_4\}$

$S_c= \{S_i \mid h (t_i) =S_i, 1<=i<= R_t\}$, R_t is cardinality of T_c

Definition 4: Logical Adaptation Path (P_L)

A logical adaptation path P_L is a sequence of operators $[t_1,t_2,\ldots t_n]$, such that for every t_i (1<i<n), the following holds:

- there exist c_i such that t_i ε $f(c_i)$
- for t_i and t_{i+1}, t_{i+1} executed after t_j

The P_L is constructed from the results of the operator mapping function and does not include operator implementation details such as the specific machines on which these operators executed (service location).

e.g. 1 Let us say T_c= {extraction, scaling, compression} = $\{t_1, t_2, t_3\}$, if we consider compression as a zipping operation there are two possible logical adaptation path configuration t_1-t_2-t_3 and t_2-t_1-t_3.

e.g. 2 Let us consider a scenario that a user requests data with his mobile phone. Assuming that there is one result of pdf data, at least the following two possible logical adaptation path configurations can be constructed (see Figure 4.1 and Figure 4.2):

P_{L1}=conversion-summarization-translation

-the data will be converted (pdf to text), summarized (long text to short text) and finally translated (text to audio).

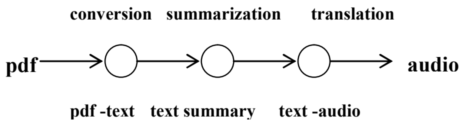


Figure 4.1 Logical adaptation path

P_{L2}=conversion- translation- summarization

-the data will be converted (pdf to text), translated (text to audio) and finally summarized (audio summary)

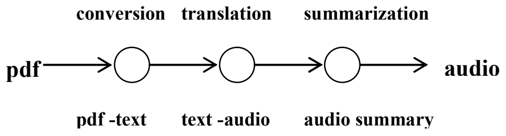


Figure 4.2 Logical adaptation path

Even though, both ordering are possible, P_{L1} seems more acceptable than P_{L2} since it is less expensive to translate summarized text than the whole text to audio.

Ordering of the operators can also be restricted semantically, like putting language translation or format/modality conversion (for compatibility with the device) as last operation. This avoids repeating the adaptation process in case of providing the same content to other request.

Definition 5: Physical Adaptation Path (P_P)

A set of pairs P_P= $\{(t_1, s_{i1}), (t_2, s_{i2}),\ldots(t_n, s_{in})\}$ is a physical adaptation of a logical adaptation path P_L iff:

- $\{t_1,t_2,\ldots.t_n\}$ is the set of operators in P_L
- for each 2-tuple (t_j, s_{ij}) in P_P, service s_{ij} executes operator t_j or in other word s_{ij} ε $h(t_j)$

A P_P is a mapping of a particular P_L onto services which execute the operators. It should be noted that if successive services are not type-compatibly additional service will be inserted to do type conversion. Hence, a P_P can have more services during execution. For example, Figure 5 depicts possible physical adaptation paths corresponding to the logical adaptation path in Figure 4.1 and 4.2.

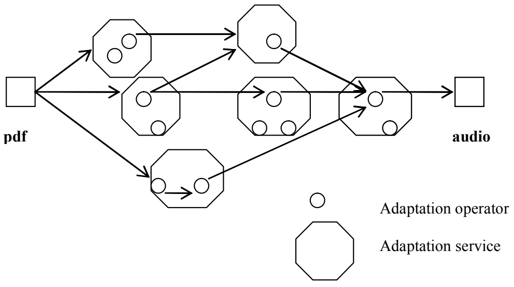


Figure 5. Physical adaptation path

Path selection strategy

It is possible to have different physical adaptation paths for a given logical adaptation path as different services can be allocated to the operators in the path (an operator can be mapped to more than one service). We define a physical adaptation path set P that is a collection of all possible paths.

For example, given service groups S_1, S_2 and S_3 with cardinality n_1, n_2 and n_3 respectively, a physical adaptation path set P_{123} is defined as follows:

P_{123}= $\bigcup p_{ijk}$ where 1<=i<= n_1, 1<=j<= n_2 and 1<=k<= n_3

In general, a physical adaptation path set $P_{123\ldots v}$ where v is the number of service groups is defined as

$P_{123\ldots v}$=$\bigcup p_{ijk\ldots v}$ where i, j, k, …, v run from 1 up to n_1, n_2, n_3, …, n_v

The number of possible physical adaptation paths ***pn*** that might support a user request with constraint R_c is equal to

$\prod_{T_i \in f(R_C)} card(h(T_i))$ assuming that we have fixed logical adaptation path. card(S) denotes the number of elements of the set S.

Therefore, it is very important to have some strategy we call it **path selection strategy** that that takes into consideration some service quality criteria during path selection process. We adopt a Web Service quality model based on a set of quality criteria (i.e. non-functional properties) that are transversal to all Web Services, for example, their charging and duration. We consider two quality criteria: service charge and duration but it is extensible as new criteria can be added without fundamentally changing the path selection strategy developed on the model.

Service cost

Given an operator t of a service s, the services charge or cost S_{cost} (s, t) is the amount of money that a service requester has to pay for executing the operation t. S_{cost} (s, t) is the sum of processing cost $S_{process}$ (s, t) and transmission cost $S_{transmission}$ (s, t).

Service time

Given an operator t of a service s, the services time S_{time}(s, t) measures the expected delay in seconds between a request is sent and the results are received and it is computed using the expression S_{time} (s, t) =$T_{process}$ (s, t) + $T_{transmission}$(s, t), where $T_{process}$ (s, t) is the processing time and $T_{transmission}$(s, t) is the transmission time.

$T_{process}$ (s, t) and $T_{transmission}$(s, t) are retrieved from the service profile. While services usually publish their processing time, $T_{transmission}$(s, t) can be estimated as an average of past observations of the transmission time.

Given the above quality criteria, the quality vector of a service s is defined as follows:

Q(s) = (S_{cost} (s, t), S_{time} (s, t))

The quality vector of a P_P is calculated as sum of quality vectors of its component services and defined as follows:

Q (P_{Pi}) = (Q_{cost}(P_{Pi}), Q_{time}(P_{Pi})), where

$$Q_{cost}(P_{Pi})=\sum_{i=1}^{n} S_{\cos t}(s_i,t_i)+\sum_{i=1}^{n-1} S_{\cos t}(s_{ii+1},t_{ii+1})$$

$S_{cost}(S_{ii+1},t_{ii+1})$-is type conversion cost.

$$Q_{time}(P_{Pi})=\sum_{i=1}^{n} S_{time}(s_i,t_i)+\sum_{i=1}^{n-1} S_{time}(s_{ii+1},t_{ii+1})$$

$S_{time}(S_{ii+1},t_{ii+1})$-delay introduced by type conversion service

Once the physical adaptation path set is generated and quality vectors of each path is computed, the path selection strategy can select the optimal adaptation path using Simple Additive Weighting technique [8]. However, it is important that the raw quality values must be adjusted to a common scale so that they can be added to form an aggregate.

$$Qs_{ij}=\begin{cases}\dfrac{Q_j^{\max}-Q_{ij}}{Q_j^{\max}-Q_j^{\min}} if \;\; Q_j^{\max}-Q_j^{\min}\neq 0 j=1,2\\ 1 \quad if \;\; Q_j^{\max}-Q_j^{\min}=0\end{cases}$$

Where $Q_j^{\max}$ and $Q_j^{\min}$ are the maximal and minimal quality values of criteria j.

Users' preference or requirement of quality criteria such as fastest or cheapest can be incorporated using weighting values in calculating overall quality score value of each path as follows

$$Score(P_{Pi}) = \sum_{j=1}^{2} Qs_{ij} * W_j \quad \text{Where } W_j \, \varepsilon \; [0,1]$$

$$\text{and} \sum_{l=1}^{2} W_j = 1 \,. \quad W_j \text{ represents the weight of each criteria}$$

If the user specifies maximum waiting time, the path selection strategy would be an optimization problem that that finds a path with minimum service cost while not exceeding the maximum waiting time and can be reformulated as follows:

$$\min_{1 \le j \le pn} \left(Q_{\cos t}(P_{Pj}) \right) \text{ and}$$

$$Q_{time}(P_{Pj}) \le T_{\max .waiting} \quad \text{Where } pn \text{ is the number of paths}$$

6. PROTOTYPE IMPLEMENTATION

For the purpose of realizing the architecture proposed, we have developed a prototype. There are some components of the system that have not been implemented so full experimentation could not be contacted, however good results have been discovered in terms of interoperability of the system. The system has been developed in Java using MySQL. An experiment has been conducted in the context of football news delivery which consists of texts and images. Our prototype implementation contains a negotiation rule and supports invocation of language translation Web Service.

6.1 Local and Content Proxy Communication

In our prototype, there are two phases of communication: meta-data and content. During meta-data communication, the content proxy retrieves meta-data for the user request and sends to local proxy. When negotiation of meta-data against profile information is done, content communication is initiated. The introduction of first phase communication enables to avoid deliverance of huge and unnecessary volume of data. Moreover, the meta-data is used in deciding the process of adaptation. For purpose of simplicity and flexibility, both communication protocols are implemented with java sockets.

6.2 Path Selection Strategy

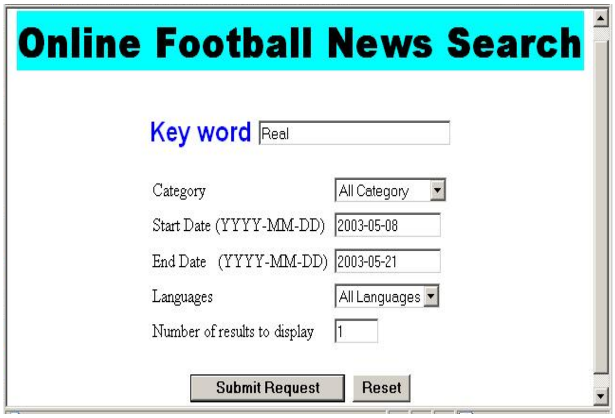
Since this prototype is to verify feasibility of our architecture in terms of flexibility and inter-operability, the focus mainly on the interaction protocols of the different components. This prototype involves only one adaptation service and simple selection strategy is employed. The path selection strategy described in our model has not been implemented so far and envisioned in the near future.

6.3 Example Language Translator Service

We have considered language translation as an example service in our prototype since it is one of content adaptation mechanisms [10, 20]. For example, user requests latest news on football and the system produces results which all are in French text but the user understands only English. It is therefore important that the results are translated to English text in order to be useful for the user. There are two solutions: either to enable the system to do the translation (French-to-English) or sending the content to translation service to do the translation on behalf of the user or content provider. In our approach we consider the later. Currently, there are Web Services that offer translation services free of charge (e.g Babelfish AltaVista [4]). In our system, we used Babelfish for the purpose of testing. Moreover, SOAP protocol is used for communication between local proxy and adaptation service (language translator).

6.4 Status of the Prototype

The communication protocol developed enables the exchange of metadata and content of a request. Instead of delivering the content for the request, the system responds with metadata or content profile, and then the local proxy do some filtration according to context of the environment. Once the content to be retrieved is decided, the content proxy sends the content to local proxy. Then, the local proxy sends the content to adaptation service. We have considered a language translation Web Service for the purpose of testing the prototype. There are remaining components of the architecture like the path selection strategy to make it a full-fledged system, however initial results have shown interoperability and flexibility of the architecture. Figure 6 and 7 shows the login interface and metadata result of a request. Figure 8 and 9 show the text result in French and its corresponding English version which is the result of translation respectively. We can observe that the translation is not hundred percent correct but that is the problem associated with quality of the translation service which might be considered as one criteria in selecting services.



Online Football News Search

Key word: Real

Category: All Category

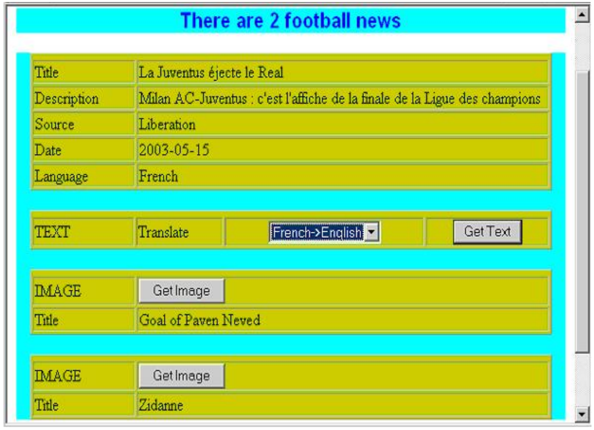
Start Date (YYYY-MM-DD): 2003-05-08

End Date (YYYY-MM-DD): 2003-05-21

Languages: All Languages

Number of results to display: 1

Figure 6. Search interface.



There are 2 football news

Title	La Juventus éjecte le Real
Description	Milan AC-Juventus : c'est l'affiche de la finale de la Ligue des champions
Source	Liberation
Date	2003-05-15
Language	French

TEXT	Translate	French->English	Get Text

IMAGE	Get Image
Title	Goal of Paven Neved

IMAGE	Get Image
Title	Zidanne

Figure 7. Meta-data result of a request.

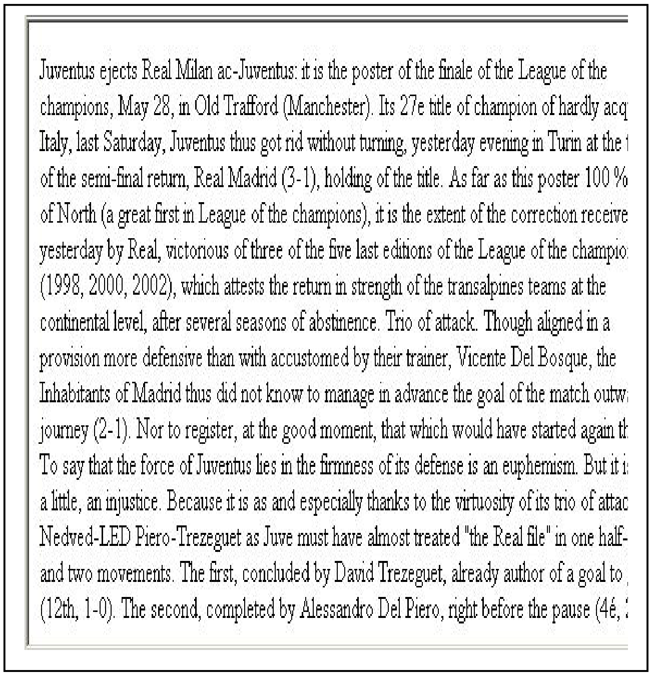
La Juventus éjecte le Real

Milan AC-Juventus : c'est l'affiche de la finale de la Ligue des champions, le 28 mai, à Old Trafford (Manchester). Son 27e titre de champion d'Italie à peine acquis, samedi dernier, la Juventus s'est donc débarrassée sans détour, hier soir à Turin lors de la demi-finale retour, du Real Madrid (3-1), tenant du titre.

Autant que cette affiche 100 % Italie du Nord (une grande première en Ligue des champions), c'est l'ampleur de la correction reçue hier par le Real, vainqueur de trois des cinq dernières éditions de la Ligue des champions (1998, 2000, 2002), qui atteste le retour en force des équipes transalpines au niveau continental, après plusieurs saisons d'abstinence.

Trio d'attaque. Quoique alignés dans une disposition plus défensive qu'à l'accoutumée par leur entraîneur, Vicente Del Bosque, les Madrilènes n'ont donc pas su gérer le but d'avance du match aller (2-1). Ni inscrire, au bon moment, celui qui les aurait relancés. Dire que la force de la Juventus réside dans la fermeté de sa défense est un euphémisme. Mais c'est aussi, un peu, une injustice. Car c'est aussi et surtout grâce à la virtuosité de son trio d'attaque Nedved-Del Piero-Trezeguet que la Juve doit d'avoir quasiment traité «le dossier Real» en une mi-temps et deux mouvements. Le premier, conclu par David Trezeguet, déjà auteur d'un but à l'aller (12e, 1-0). Le second, achevé par Alessandro Del Piero, juste avant la pause (43e, 2-0).

Figure 8. Original Text result in French.



Juventus ejects Real Milan ac-Juventus: it is the poster of the finale of the League of the
champions, May 28, in Old Trafford (Manchester). Its 27e title of champion of hardly acq
Italy, last Saturday, Juventus thus got rid without turning, yesterday evening in Turin at the
of the semi-final return, Real Madrid (3-1), holding of the title. As far as this poster 100 %
of North (a great first in League of the champions), it is the extent of the correction receive
yesterday by Real, victorious of three of the five last editions of the League of the champio
(1998, 2000, 2002), which attests the return in strength of the transalpines teams at the
continental level, after several seasons of abstinence. Trio of attack. Though aligned in a
provision more defensive than with accustomed by their trainer, Vicente Del Bosque, the
Inhabitants of Madrid thus did not know to manage in advance the goal of the match outw
journey (2-1). Nor to register, at the good moment, that which would have started again th
To say that the force of Juventus lies in the firmness of its defense is an euphemism. But it i
a little, an injustice. Because it is as and especially thanks to the virtuosity of its trio of attac
Nedved-LED Piero-Trezeguet as Juve must have almost treated "the Real file" in one half-
and two movements. The first, concluded by David Trezeguet, already author of a goal to
(12th, 1-0). The second, completed by Alessandro Del Piero, right before the pause (4é,

Figure 9. Text translated from French to English.

7. DISCUSSION

We have presented architecture of service-approach content adaptation. This architecture enables clients and content providers to utilize third-party adaptation services. The service-approach provides better adaptation performance since each adaptation task is handled by specialized adaptation techniques. Moreover, it is flexible (new service can be integrated easily) and scalable (can be easily extended to other systems). We have also defined mathematical model of content negotiation and adaptation. This model enables us to formulate content adaptation as an optimization problem so optimal adaptation path can be computed.

Main contributions of our proposed architecture are:

- Capability to apply different type of content adaptations The architecture enables content delivery systems to use external adaptation services which are basically Web Services. Utilization of Web Services for purpose of content adaptation is introduced for the first time here.
- Integrate easily in existing infrastructures Introduction of local and content proxies enable to extend existing content delivery systems easily without affecting their basic structure (client-server).
- Mathematical formulation of content negotiation and adaptation The negotiation and adaptation model defined in this paper enables to formulate content adaptation as constraint satisfaction problem so that optimal path can be selected satisfying the different constraints such as user, device and network.

8. CONCLUSION AND FUTURE WORKS

Due to the rapid pace of technological change, emerging standards and the constant evolution of mobile devices, the use of flexible and adaptable technology and methodologies is essential for pervasive applications. Hence, the first goal of the proposed architecture is to provide an all round solution to problems of content adaptation in wide area system which were not well addressed by existing adaptation approaches The second goal is to exploit the potential of emerging technologies such as Web Services for the purpose of adaptation. Web Services have demonstrated their flexibility, extensibility and interoperability; however their application for purpose of content adaptation in distributed environment has not been investigated. Therefore, this project is initiated to develop distributed content adaptation as a solution that combines several of these technologies and extends them to alleviate the drawbacks of existing adaptation approaches. By doing this, content adaptation can be developed as any value-added services provided by third party and used through subscription by users or content providers. Future work focuses on developing the heuristics required to solve the equation system that we have proposed for optimal path selection.

9. REFERENCES

[1] "ACI project MEDIGRID: medical data storage and processing on the GRID" (http://www.creatis.insa-lyon.fr/MEDIGRID 2002).

[2] Acharya, S., "Application and Infrastructure Challenges in Pervasive Computing", NSF Workshop on Context-Aware Mobile Database Management (CAMM), Providence, Rhode Island, January 24-25, 2002.

[3] Austin, D., Barbir, A., Ferris, C., Garg, S., "Web Services Architecture Requirements" W3C Working Draft 19 August 2002 (http://www.w3.org/TR/2002/WD-wsa-regs-20020819).

[4] BabelFish, (http://babelfish.altavista.com/).

[5] Brunie, L., Coquil, D. and Pierson, J-M., "Semantic collaborative web caching", WISE 2002, IEEE CS Press, Singapore, Dec 12-14, 2002, pp 30-39.

[6] Buchholz, S., Schill, A., "Web Caching in a Pervasive Computing World", Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SC12002), Orlando, FL, USA, Jul 14-18, 2002.

[7] Cardellini, V., Yu, P. S., Huang, Y.W., "Collaborative proxy system for distributed Web content transcoding" , Proc. of 9th Int'l ACM Conf. on Information and Knowledge Management, Washington, DC, pp. 520-527, Nov. 2000.

[8] C.L.,H. and Yoon, K., "Multiple Criteria Decision Making". Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1981.

[9] Elson, J., Cerpa, A., "ICAP - the Internet Content Adaptation Protocol", Internet Draft draft, the ICAP Protocol Group, June 2001. (URL: http://www.i-cap.org/spec/icap specification.txt).

[10] Han, R., Bhagwat, P., LaMaire, R., Mummert, T., Perret, V., and Rubas, J. "Dynamic Adaptation in an Image Transcoding Proxy for Mobile WWW Browsing". IEEE Personal Communication, 5(6):8--17, 1998.

[11] Held, A., Buchholz, S., Schill, A., "Modeling of Context Information for Pervasive Computing Applications", Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SC12002), Orlando, FL, USA, Jul 14-18, 2002

[12] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies" W3C Working Draft 15 March 2001. (http://www.w3.org/TR/2001/WD-CCPP-struct-vocab-20010315/).

[13] Lei, Z., Georganas, N.D., "Context-based media adaptation in pervasive computing", Proceedings of IEEE Canadian Conference on Electrical and Computer (CCECE'01), Toronto, May 2001

[14] Lemlouma, T., Layaida, N., "Content Adaptation and Generation Principles for Heterogonous Clients", OPERA Project, INRIA Rhone Alpes, Position Paper for the W3C Workshop on Device Independent Authoring Techniques, July 2002 (http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/DIAT-PositionPaper.pdf).

[15] Lemlouma, T., Layaida, N., "Adapted Content Delivery for Different Contexts", SAINT 2003 Conference, Orlando, Florida, USA, January 27-31, 2003.

[16] Lum, W.Y. and Lau, F.C.M., "A Context-Aware Decision Engine for Content Adaptation", IEEE Pervasive Computing, Vol. 1, No. 3, July-September 2002, 41-49.

[17] McHenry, S., Condry, M. and Tomlinson, G., "Open Pluggable Edge Services Use Cases and Deployment Scenarios", draft-mchenry-opes-deployment-scenarios.txt (work in progress), November 2000, (http://www.ietf.org/internet-drafts/draft-mchenry-opes-deployment-scenarios.txt).

[18] Mohan, R., Smith, J., Li, C.S., "Adapting Multimedia Internet Content For Universal Access", IEEE Transactions on Multimedia, March 1999, pp. 104-114.

[19] Noble, B.D., Price, M., and Satyanarayanan, M., "A Programming Interface for Application-aware Adaptation in Mobile Computing", Proc. 2nd USENIX Symposium on Mobile and Location-Independent Computing, Ann Arbor, MI, USA, Apr. 1995.

[20] Rahman, R., Menon, R.R. and Rafalow, L., "Enabling OPES to Use Web Service for Callout Service", draft-rrahman-web-service-00.txt (work in progress), May 2002 (http://standards.nortelnetworks.com/opes/non-wg-doc/draft-rrahman-web-service-00.txt).

[21] Using XML in Oracle Database Applications: Customizing Data Presentation Author: Brad Wait, Oracle Corporation Date: November 1999. Updated June 2001 by Robert Hall, Oracle Corporation. (http://otn.oracle.com/tech/xml/htdocs/xml custom_presentation.htm).

[22] William, L. Oellermanna, Jr. "Architecting Web Services", Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY, 2001.

[23] Yoshikawa, C., Chun, B., Eastham, P., Vahdat, A., Anderson, T., and Culler, D., "Using smart clients to build scalable services", Proc. Winter 1997 USENIX Technical Conf., Jan. 1997.