NAC: A Basic Core for the Adaptation and Negotiation of Multimedia Services

Tayeb Lemlouma and Nabil Layaïda
OPERA Project
INRIA Rhône Alpes
Zirst 655 Avenue de l'Europe – 38330, Montbonnot, Saint Martin, France
Tel: +33 4 7661 5281, +33 4 7661 5384
Tayeb.Lemlouma@inrialpes.fr Nabil.Layaida@inrialpes.fr

ABSTRACT
We present in this paper NAC: a basic core for the negotiation and adaptation of multimedia services in heterogeneous environments. The objective of the implemented core is to allow clients (PDA, WAP phones, laptops, etc.) to use a multimedia content which is adapted automatically to their preferences and capacities. Client descriptions (i.e. profiles) are declared in CC/PP structures stored in an XML format and can be modified at anytime. NAC includes two kinds of adaptations: structural adaptation such as adapting XHTML to WML, SMIL and HTML filtering, and media adaptation such as image compression, text to SMS, remote text to speech. The proposed core doesn't make any assumption on the existing platform and used browsers. Players must only point the network connection to the server or the proxy that uses the Adaptation and Negotiation Module (ANM). Services will be then adapted automatically. The architecture of NAC is flexible to be used at any particular system, and enriched by transformation programs or style sheets to meet particular needs.

Keywords
Multimedia adaptation, content negotiation, UPS, CC/PP, SMIL, heterogeneous environments, proxy architecture, HTTP.

1. INTRODUCTION
Providing adaptable services for different clients in heterogeneous environments is very important since the use of a wide diversity of digital storage and small devices is nowadays increasing. Making multimedia services understandable and usable by this range of clients is a hard task to achieve. This last requires the knowledge of users, servers and network contexts; but also demands efficient mechanisms that allow delivering the aimed service in the best way. Hence the implementation of a good architecture is necessary. The objective of such architecture is so adapting services to different contexts no matter what is the kind of the user device.

Existing servers use several multimedia models to store and handle multimedia services. Among the existing models, we find: HTML [10], MHEG [7], HyTime [5], SMIL [22] etc. Clients in heterogeneous environments are characterized by several limitations which make theme unable to use directly such multimedia models without achieving some adaptation tasks. Indeed, clients and networks in such systems are subject of many constraints such as low power, risks of data, small user interface, small storage and processing capacities, low and high variability of the network bandwidth [4] etc.

Nowadays, several adapting systems exist such as: the *VideoZoom* system which handles video streams in terms of spatial and temporal view and resolutions [6], application specific stream transformation [2], reducing the network bandwidth consumption [3], transforming HTML to some markup languages (e.g. WML, HDML) [16] etc. These systems are benefic but unfortunately remain limited to a particular kind of architecture. They lack of content negotiation flexibility and can't support the large devices heterogeneity increasing today.

The problem can be felt more if we take a look on the actual situation of the web. Web services tailor on them many rich media resources like images, scripts, audio, and video of high quality etc. Unfortunately, protocols used to transfer these services do not take into account the limitations and constraints of the target context. If we take the example of the HTTP protocol [8], we find content negotiation mechanisms based on the versioning principle and with limited expressive powers [8][9]. Following such approach, requires providing the content in many versions for each target context which is hard to do in a wide diversity of clients. Furthermore, the description of the client capabilities and preferences isn't well expressed using the HTTP accept headers.

In this paper we present the NAC (Negotiation and Adaptation Core): a new basic core for multimedia services adaptation and negotiation in heterogeneous environments. Our final objective is to ensure that the diversity of clients existing in the global system can accede on services provided by servers. To allow a maximum of heterogeneity tolerance, we have built our core on the base of no assumptions neither on the client context nor on the player (or the browser) used in the communication with the rest of the environment. Another fixed target is to make the core extensible for other models and kind of adaptation. Hence, the same architecture can be reused to meet the needs of other particular environments with their proper specifications.

As we will see, NAC allows multimedia services delivery according to the client context. The same service, that exists on the server side, can be so delivered to different devices with respect to their capabilities and preferences. The way of declaring these descriptions is based on the XML [18] model which makes this task very easy to achieve and opened to different platforms. NAC is experimented into two different environments: a wired and a wireless network using different players. Adaptations can be enriched at any time (i.e. dynamically) by transformation methods in the form of XSLT [23] style sheets or other transformation programs which enlarge the adaptation possibilities of the proposed solution.

2. CONTEXT OF THE WORK

The core that we propose in this paper is built to support different contexts of multimedia services use such as through: desktop computers, cell phones, personal device assistants, workstations etc. The XML multimedia model is chosen as base technology since it ensures a good separation between data and presentation. This doesn't exclude any other kind of services adaptation like HTML adaptation for instance. The use of the XML model can be found in mainly two different parts of NAC: in the services model and the environment description using the UPS schema [13] based on CC/PP [17] and RDF [19].

Our aimed goal is providing a complete solution of multimedia services delivery independently to: 1) the original and the target service format (SMIL, WML, XHTML, Voice XML, etc.) 2) the user context (desktop model, mobile device or other), and 3) the communication network (wired or wireless). The proposed architecture is extensible and doesn't handicap users or servers with a fixed multimedia document or protocol.

One of the target contexts of this work that we are actually focusing on is adapting SMIL2.0 [22] services to SMIL 2.0 Basic [20]. The SMIL2.0 -which has been recently a W3C recommendation-, is an efficient multimedia model for services handling and authoring since it covers a wide range of multimedia presentation functionalities. The target format, (i.e. the SMIL basic) is chosen thanks to the minimal set of functionalities that it presents. Therefore, it can be used as a common base for heterogeneous devices. The use of the SMIL modules description knows already an important use, such by the third Generation Partnership Project (3GPP) [24].

3. ARCHITECTURE

NAC (negotiation and adaptation core) can build in a simple architecture which includes three main components: The server of the content which uses documents bases (XML, SMIL, HTML etc.), the ANM module which can run on the original server or on an intermediate proxy, and the UCM module on the client side (figure 1).

The main component of the architecture is represented by the adaptation and negotiation module (ANM). The ANM is an opened module which can receive client requests and responds by delivering the wanted service. The module can be used by a local server and so it can exploit the server transformation processes and other services. It can also be shared by many servers; no assumptions are made on the number of the used ANM modules in the global architecture.

In the client side, the UCM (user context module) is responsible of sending the client profile (file or URI). We can distinguish two cases: static and dynamic situations. In the first situation, the client profile doesn't change frequently therefore the UCM sends the profile once and then the client player interacts with the ANM in order to use multimedia services adapted according to the profile. The second situation concerns frequent changes of the client context, in this situation the client profile can be sent at any moment. The ANM takes into account the current version of the client profile.

The client player's requests are received by target hosts through the ANM module. The wanted service can tailor on it any kind of media (text, image, audio, video streams, scripts, applets, etc.) and can have any format. For example, the wanted service can be an HTML or a SMIL file. The ANM module delivers the wanted service after applying some adaptation methods to meet the client context; therefore unsupported functionalities and media must be removed or filtered.

ANM uses a negotiation protocol [12] to achieve the multimedia services delivery efficiently. The server negotiates with the UCM to the best service to be delivered; according to the local context, the server capabilities and to the network profile. The decision to apply adaptation techniques is the result of this negotiation step. In this last, different profiles, concerning the content, the client and the server are matched.

4. THE NAC CORE

NAC is designed to provide a basic solution that resolves the problem of multimedia services adaptation and negotiation in heterogeneous systems. Who says heterogeneous, says a large diversity of devices with different capacities and also a wide range of services format which are different in their potentialities of multimedia representation [1]. Our classical view of services delivery, which follows the general scenario: "service demand then service deliverance" must change. The actual situation is characterized by many constraints which makes direct deliverance difficult to ensure. NAC can be seen as a low layer for multimedia deliverance that supports personalization and adaptation enrichment to be used in the actual situation of modern multimedia environments.

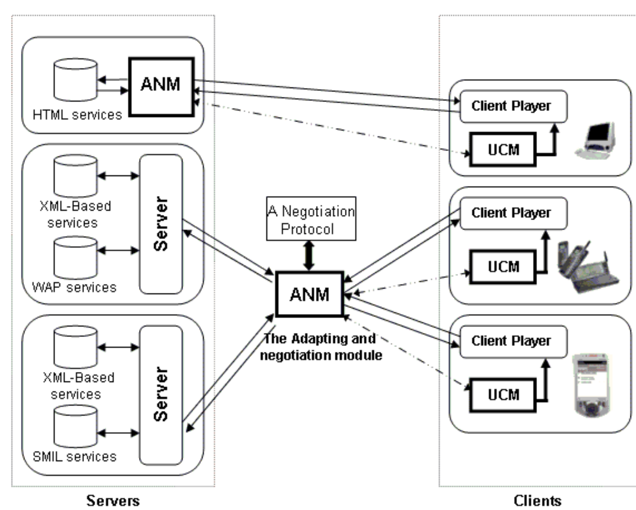


Figure 1. NAC: The architecture of multimedia services adaptation and negotiation

4.1 The UPS: Universal Profiling Schema

The Universal Profiling Schema (UPS) [14][13] is created in order to resolve the problem of the content negotiation and adaptation of multimedia services. UPS defines the framework that allows a well formed description of all the elements that can intervene in the final deliverance of services. These elements can be in the general case: the content of the server (the requested documents), the server capabilities in terms of available adaptation and transformation methods, the client preferences and capabilities, the network profile (available bandwidth, etc.). On the base of UPS, servers can define their proper negotiation strategies to match these profiles.

Since networks in heterogeneous environments are very heterogeneous and knows a high variability of the bandwidth; the UPS was defined in a way that minimizes the exchange of profiles between clients and servers or proxies. Only utile information is exchanged and this when that's needed. To ensure this, UPS uses some techniques such as: - separating the document structure (SMIL, XHTML, etc.) from the used media resources (images, videos, etc.); - using links inside profiles which makes the server able to retrieve the required profiles only when it's necessary. All the descriptions of the NAC components follow the UPS schema.

4.2 The UCM: User Context Module

The role of the UCM is to give the required view of the user context or which we call the client profile. This description concerns the material side of the client device and the software side in terms of the player and the system used to accede to multimedia services. To achieve this task, we have chosen to use the CC/PP format [17] - based on RDF [19] and XML [18] models - which is designed specially for describing devices capabilities and user preferences. The client profile is so stored in an XML file which is very easy to edit and to modify at any time.

The client profile contains the description of the device hardware and software and the services constraints. In the profile file, the user can declare service categories that it can handle (SMIL, HTML, etc.) and its abilities to play different media resources of the supported services. The example of the figure below concerns a device in the form of a PDA that accedes to SMIL services category using a local SMIL player which doesn't support videos.

```
...
<ccpp:component>
  <rdf:Description rdf:about="TerminalHardware">
  <rdf:type rdf:resource="HardwarePlatform" />
  <neg:DeviceName>iPAQ 3600</neg:DeviceName>
  <neg:display>101x52Pixels</neg:display>
  <neg:PixelPitch>0.24mm</neg:PixelPitch>
  </rdf:Description>
</ccpp:component>
  …
<ccpp:component>
  <rdf:Description rdf:about="OnlySupportedResources">
  <rdf:type rdf:resource="Resources" />
  <neg:OnlySupportedResources>
  <rdf:Bag>
  <rdf:li rdf:parseType="Resource">
  <neg:type>service</neg:type>
  <neg:serviceDetail>SMIL</neg:serviceDetail>
  <neg:format>smil</neg:format>
  <neg:profile>device-profiles/smil-profile.xml</neg:profile>
  </rdf:li>
  <rdf:li rdf:parseType="Resource">
  <neg:type>audio</neg:type>
  <neg:format>wav</neg:format>
  <neg:profile>device-profiles/wav-profile.xml</neg:profile>
  </rdf:li>
  </rdf:Bag>
  </neg:OnlySupportedResources>
  </rdf:Description>
</ccpp:component>
      ...
<ccpp:component>
  <rdf:Description rdf:about="NonSupportedResources">
  <rdf:type rdf:resource="Resources" />
  <neg:NonSupportedResources>
  <rdf:Bag>
  <rdf:li rdf:parseType="Resource">
  <neg:type>video</neg:type>
  <neg:format>mpeg</neg:format>
  <neg:profile>device-profiles/mpeg-profile.xml</neg:profile>
  </rdf:li>
  …
  </rdf:Bag>
  </neg:NonSupportedResources>
  </rdf:Description>
</ccpp:component>
    …
```

Figure 2. An example of client profile.

Note that the same client profile can contain the description of many service categories and resources. For example, the user can define a new 'service' by adding the corresponding tag and giving the appropriate description. Since a constraint is a logic formula (atomic or combined) [11], constraints expressing in profiles requires strong logic expressing tools and syntax and this to be large and precise in the same time. In this step of work, constraints expressing is very simple and covers logic formulas of the kind: C_1="The media is not supported", C_2="The media is supported", C_3="The service must be adapted using a precise adaptation method" or C_4="The service must be adapted to some existing versions".

NAC includes two implemented variants of UCM: one that can run using classical platforms (such under a desktop model) and one under a PDA (an iPAQ 3600) as it shown on the figure below.

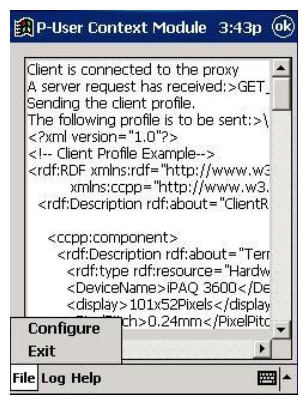


Figure 3. The UCM implementation in an a PDA

4.3 The ANM: Adaptation and Negotiation Module

ANM can be run on the content server or on an intermediate proxy. It allows services deliverance after receiving client requests. The ANM treats the client request and obtains the wanted service from the original server. Services are then matched to the client preferences and capabilities and eventually to other parameters. Finally the result is delivered to the demander. In order to install the ANM module in the global architecture (composed by the set of clients and servers). Client players must point their network connections to the ANM host by declaring it the intermediate proxy and with indicating the corresponding port number.

The ANM adaptation process adapts dynamically services stored temporarily by the ANM host using the appropriate transformation process (style sheet, program, etc.). The adaptation follows the client profile which is transformed - using an intern XSLT style sheet- in a memory form (variables, arrays, lists, etc.) which facilitates its direct exploitation by the ANM adaptation program.

Actually, the protocol supported by NAC in the exchange of the requests: ANM/Client and ANM/server is the HTTP protocol [8]. In next steps of our work, when completely integrating the negotiation protocol introduced in [12], interactions between the ANM and the UCM will follow an XML based protocol like the SOAP [21] one, and this to provide more flexibility and simplicity offered by the XML technology. The approach of using an XML-based protocol will make easier to achieve which we can call by *protocol adaptation*, especially in the case where the protocols used by the client and the server are not the same (for example WAP [15] for the client and HTTP for the server).

In present, the implemented ANM module includes a set of adaptation methods which can be classified into two kinds: document structural adaptations and media adaptations. In the first kind, we use mainly the XSLT [23] model since it offers easy structure adaptations. This includes: XHTML to WML, XML to SVG, XML to LaTeX and documents filtering. The second kind of the ANM adaptation concerns media transformation and direct methods. Since this kind depends widely to the encoding of the media, the adaptation is done by independent programs. In this kind we find: image compression, text to speech using remote hosts and text to SMS messages using an SMS gateway.

The use of an advanced negotiation protocol in the global architecture is necessary. The way in which content negotiation is done using existing protocols (like the HTTP negotiation mechanisms) is very limited. However, interactions with client players and servers, concerning the content delivery, must remain in the same original protocol. Thus, we must enlarge the NAC supporting of other protocols, for example with the WAP [15] protocol to cover players integrated on mobile phones.

Another possibility of ANM is to implement modules that control and retrieve the network characteristics such as the available network bandwidth. The retrieved network profile has a primary importance in some particular contexts such as in wireless networks and in networks where the client can be subscribed into different bandwidth speeds.

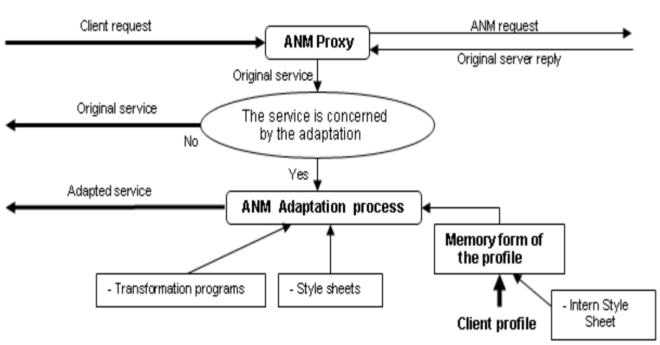


Figure 4. Overall view of ANM

4.4 Services Delivery

As it is shown in figure 2, the client profile can describe two kinds of resources: services and media resources (images, videos, etc.). Services are characterized by their type that denotes the format of the most handled documents such as SMIL or WML documents. A service type can tailor on it different media used in the final presentation. A media resource is described by its type, format and profile. A media resource *X* can be delivered to the user if it satisfies deliverance conditions included in the media profile. Formally, this can be written as follow:

Media-profile.xml_Conditions (Service=*Y*, Resource=*X*) = true

Where *Y* is the type of the service on which *X* belongs. The matching of the client profile with the multimedia service is equivalent to evaluate the deliverance condition for each resource that exists in the original service. As we have already said, client profile constraints are belonging to the set $\{C_1, C_2, C_3, C_4\}$, where C_i are defined as in section 4.2. The set of constraints can be enriched according to the available implemented methods. For example if an implemented module *M* can evaluate the available network bandwidth, we can tolerate the constraints of the following logic form: "available bandwidth ≥ numeric value", which can be traduced in XML as: <constraint>available bandwidth GE numeric value Kbps<constraint>.

5. EXPERIMENTAL RSULTS

In this section we give an experimental overall view about the implementation of the NAC core. This last stills extensible and can be enriched by other adaptation and transformation methods and profiles descriptions. The implementation of NAC was done using the JAVA Runtime Environment at the server side and the Microsoft embedded visual C++ for embedded devices at the client side. Experimentations were done in an architecture composed by three kinds of devices: a personal device assistant running under Windows CE and using a wireless network through a Compaq Wireless LAN card, a laptop computer using the same wireless network and personal computer using a wired network.

NAC becomes operational after executing the ANM (on the original server or a proxy) and pointing all the client players to it. In our case, we have used only one ANM module running in a proxy host, but as we have already said, the architecture can include more (see figure 1).

Different client profiles following the UPS schema were prepared to represent different users. To see the behavior of the NAC, several tests were done with modifying the clients' constraints using the different UPS profiles. The player used to play the received services is also changed in each test (SMIL player, IE browser, etc.).

Services to be played are represented by HTML, XHTML, SMIL documents and different media that can be used inside or independently to these documents. To see the work of NAC, the same document identified by its URL is kept for different profiles and for the same client player.

To understand how NAC can behave, we give here an example of a client that uses multimedia services using a SMIL player on a PDA device. The client accedes to the content in a wireless network through the NAC architecture. In this example and after the installation of NAC, the acceded content is represented by a SMIL document located at the address:

http://opera.inrialpes.fr/Smil/T/t1.smi. We give in the following the source of the SMIL document:

```
<smil>
<head><layout>
<regPoint id="middle" top="50%" left="50%" regAlign="center" />
<region id="r1" top="0" left="0" right="240" bottom="240"/>
<region id="r2" top="69" left="36" right="203" bottom="208"/>
<region id="r3" top="96" left="30" right="209" bottom="181"/>
<region id="r4" top="40" left="0" right="240" bottom="220"/>
<region id="r5" top="234" left="86" right="240" bottom="277"/>
</layout></head>
<body>
<seq>
<par>
<audio id="audio" src="tayeb.mp3" begin="0s" end="12s"/>
<img id="img1" region="r1" src="image01.jpg" begin="0s" dur="12s"/>
<img id="img2" region="r2" src="image02.jpg" begin="00:00:6" dur="6s"/>
</par>
<video region="r4" src="iceage.mpeg" begin="0s" end="30s"/>
<par>
<img id="img3" region="r3" src="image03.jpg" begin="0s" dur="6s"/>
<img id="img4" region="r5" src="image04.gif" begin="3s" dur="3s"/>
</par>
</seq>
</body>
</smil>
```

Figure 5. Example of a requested content

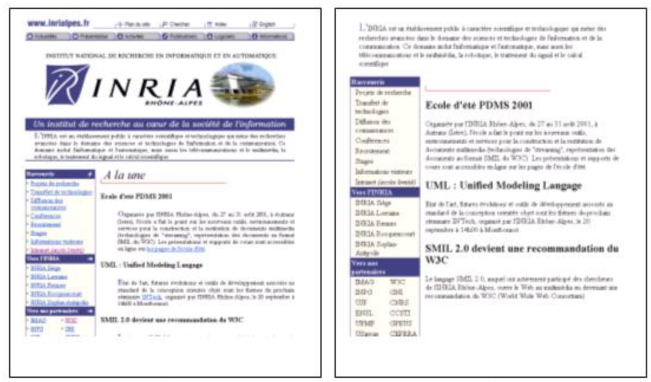
As we can note, the above document includes different types of media: image, audio and video presented in different ways. The document was tested first, with a client profile which doesn't contain any constraint concerning SMIL documents[1]. The multimedia presentation was played correctly with the respect of the source document semantic. The client profile (stored in *ClientProfile.xml*) was modified after, with indicating that video and audio media resources are not supported. As it shown in the following:

```
...
<ccpp:component>
  <rdf:Description rdf:about="NonSupportedResources">
  <rdf:type rdf:resource="Resources" />
  <neg:NonSupportedResources>
  <rdf:Bag>
  <rdf:li rdf:parseType="Resource">
  <neg:type>video</neg:type>
  <neg:format>mpeg</neg:format>
  <neg:profile>device-profiles/mpeg-profile.xml</neg:profile>
  </rdf:li>
  <rdf:li rdf:parseType="Resource">
  <neg:type>audio</neg:type>
  <neg:format>wav</neg:format>
  <neg:profile>device-profiles/wav-profile.xml</neg:profile>
  </rdf:li>
  …
  </rdf:Bag>
  </neg:NonSupportedResources>
  </rdf:Description>
</ccpp:component>
…
```

Figure 6. Example of client profile constraints

The PDA requests then the same SMIL file with the new profile. The received multimedia presentation is played without neither audio nor video and kept the same semantic of the original presentation: for instance the semantic of the parallel and the sequential media were not violated. Another profile modification which was tested also, is to render SMIL2.0 [22] document conformed to SMIL 2.0 Basic [20] which is more adapted to small devices. For example in the *t1.smi* document, we find the *regPoint* element, which is added to the SMIL Basic profile by the *HierarchicalLayout* module [22]. The client profile was modified and the adapted file received is a SMIL Basic file (the SMIL2.0 element was removed).

The next example concerns HTML services, the requested document is identified by the URL: http://www.inrialpes.fr and acceded by both a PDA device and a PC. Figure 7.a gives the original HTML document without any constraint. The document is then acceded through NAC with a profile that includes the constraint of the non-supporting of images and links. The document is adapted dynamically according to the client profile; the received document is shown in figure 7.b.



A B

Figure 7. Example of the NAC document adaptation

The two precedent examples can be classified into the document adaptation and negotiation class. NAC includes also media resources handling, an example of that is the compression of images according to the display capabilities of the user device. Both in the case where media or document resources must be adapted to meet the client context, a negative reply can be returned by the ANM if the original server or the proxy haven't the required adaptation method or an already existed version of the adapted content. The negative answer is used to avoid the blocking of the client and has the same format of the client supported services.

Table1. User context examples

Client Profile	Described device	Main requirements	Player used
Profile 1 (P1)	PDA, Laptop, PC	HTML and SMIL content is entirely supported	SMIL and HTML player
Profile 2 (P2)	PDA, Laptop, PC	Only short HTML content is supported	Any HTML browser
Profile 3 (P3)	Mobile phone	Only SMS messages are supported	No player
Profile 4 (P4)	PDA	Only SMIL content without video	A pocket SMIL player
Profile 5 (P5)	WAP phone	Only WML 1.0 content is supported	Embedded browser

[1] As we have noted, the profile can contain constrains concerning other kind of documents like HTML, etc. In this case, services' description will be ignored by the ANM process since it's not concerned by the adaptation. (see figure 4)

Table 1 resumes some client contexts used in our experimentations. ANM negotiates with UCM module that exists in the client side. According to client preferences and capabilities expressed in its profile, the ANM try to adapt the original content to the user context using the capabilities of the server. The ANM can also use a service provided by a remote host. For example, the implemented Text to SMS adaptation method uses a remote SMS gateway. The following table shows the NAC behaviors according to different situations described in Table 1.

Table2. The NAC behavior

Client Profile	The Requested content	Returned content	Applied techniques
P1	HTML document	The original content	Direct reply
P2	A long HTML document	A short version of the original content	Versions selection
P3	Abstract HTML document	An SMS message to the described phone	Text to SMS adaptation
P4	SMIL content with videos	Original content without videos	Content filtering
P5	WML 2.0 document	Negative reply	No adaptation techniques

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented NAC: a negotiation and adaptation core for heterogeneous multimedia environments. NAC was designed to support heterogeneity and constraints presented in such environments. The ideal goal to reach is: the use of multimedia services whatever the format service, the client kind and the network nature. Clearly, if the wanted content doesn't meet the end user capabilities and preferences, several steps of adaptation and negotiation must be achieved. The architecture and the core that we have proposed ensure a framework to provide a basic solution that resolves that problem of services adaptation in multimedia systems. The core can be enriched with needed adaptation methods for a particular context. For example, if the target context uses the VoiceXML format, we need only to design style sheets or other transformation programs to achieve that task. Integration of this process is then done very easily to the ANM module.

As we have seen, we were based on the use of the XML technology, either to express the client profile using the UPS schema, or transforming profiles into memory form using an intern XSLT style sheet, or to use transformation processors. Our choice is justified by the flexibility and the simplicity offered by the opened XML model. The same direction will be followed when integrating completely an advanced negotiation protocol to the ANM module. In this case the XML model will be followed in ANM/UCM interactions and this time in the form of an XML-based protocol.

Experimentations done using NAC show that the defined architecture allows the deliverance of multimedia services which are adapted to the end user context. A client can easily obtain the adapted content only by declaring its profile which has a simple XML structure. It's important to note that there is a clear independence between the multimedia platform and the NAC core. This is very benefic because the same core can be used by different systems and provides personalized and understandable services for users.

In order to meet the needs of a particular environment, adaptation and transformation processes must be provided to the ANM module in order to ensure a universal access of the server content. For example: to provide processors which transform high audio quality to phone quality audio format in a heterogeneous environment that includes mobile phones. NAC ensures a very flexible integration of transformation processors by calling them when invoking the ANM adaptation process and after the step of the content negotiation between the ANM (at the server side) and the UCM (at the client side).

Ongoing work will focus on: 1) Developing the support of expressing content negotiation constraints and resolving them, 2) The enrichment of ANM with XSLT style sheets and transformation strategies, such as by adapting rich multimedia presentations (SMIL2.0) for limited devices (SMIL Basic), 3) The complete integration of an advances negotiation protocol (inspired from [12]) and 4) Developing a network context module (NCM), responsible to determine the actual context of the network and this following the same way as the UCM module.

7. REFERENCES

[1] Boll S., Klas W., and Westermann U. A Comparaison of Multimedia Document Models Concerning Advanced Requirements. Technical report, University of Ulm, Germany, http://www.informatik.uni-ulm.de/dbis/Cardio-OP/publication/TR99-01.ps.gz, February 1999.

[2] Charles B., Murray S. Mazer, Scott M., and Jim M. Application-Specific Proxy Servers as HTTP Streams Transducer. In Proceedings of the Fourth International World Wide Web conference, December 1995.

[3] Chi Chi Hung and Lim Yan Hong. Adaptive Proxy-based Content Transformation Framework for the World Wide Web. IEEE 2000.

[4] George H. Forman. The challenges of Mobile Computing. IEEE Computer, 27:38-47 April 1994.

[5] ISO/IEC. Information Technology – Hypermedia / Time-based Structuring Language (HyTime), ISO/IEC IS 10744, 1992.

[6] John R. Smith. VideoZoom Spatio-temporal Video Browser. IEEE Trans. Multimedia, Vol. 1, No. 2, June '99.

[7] Joseph R., and Rosengren J. MHEG-5: An Overview. Technical Report, GMD-FOKUS, Berlin, http ://www.fokus.gmd.de/ovma/mug/archives/doc/mheg-reader/rd1206.html, December 1995.

[8] Network Working Group. Hypertext Transfer Protocol – HTTP/1.0. http://www.ietf.org/rfc/rfc1945.txt?number=1945, May 1996.

[9] Network Working Group. Transparent Content Negotiation in HTTP. http://www.ietf.org/rfc/rfc2295.txt?number=2295, March 1998.

[10] Raggett D., Le Hors A., and Jacobs I. HTML 4.0 Specification, W3C Recommendation.

http://www.w3.org/TR/1998/REC-html40-19980424, April 1998.

[11] Tayeb L., and Abdelmadjid B. Constraint Logic Programming (CLP), Study and Application to the Puzzle: “Send plus More equals to Money”. RIST, Vol. 10, N. 1-2, CERIST, Algiers, Algeria, 2000.

[12] Tayeb L., and Nabil L. The Negotiation of Multimedia Content Services in Heterogeneous Environments. The 8th International Conference on Multimedia Modeling, CWI, Amsterdam, The Netherlands, to appear, November 5-7, 2001.

[13] Tayeb L., and Nabil L. Universal Profiling for Content Negotiation and Adaptation in Heterogeneous Environments. W3C Workshop on Delivery Context, W3C/INRIA Sophia-Antipolis, France, 4-5 March 2002.

[14] Tayeb L., and Nabil L. Universal Profiling Schema (UPS). http://opera.inrialpes.fr/people/Tayeb.Lemlouma/NegotiationSchema/index.htm, specification January 2002.

[15] WAP forum. Wireless Markup Language Ver. 2.0. http://www.wapforum.org/ proposed version 26 June 2001.

[16] WebSphere. Transcoding Publisher. IBM Corp, http://www-4.ibm.com/software/webservers/transcoding/.

[17] W3C. Composite Capability/Preference Profile. http://www.w3.org/TR/CCPP-struct-vocab/, W3C Working Draft 15 March 2001.

[18] W3C. Extensible Markup Language (XML) 1.0. W3C Recommendation: http://www.w3.org/TR/1998/REC-xml-19980210, 10 February 1998.

[19] W3C. Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/1999/REC-rdf-syntax, W3C Recommendation 22 February 1999.

[20] W3C. SMIL 2.0 Basic Profile and Scalability Framework. http://www.w3.org/TR/smil20/smil-basic.html, 07 August 2001.

[21] W3C. SOAP Version 1.2. http://www.w3.org/TR/2001/WD-soap12-20010709/. W3C Working Draft 9 July 2001.

[22] W3C. Synchronized Multimedia Integration Language (SMIL 2.0) W3C Recommendation. http://www.w3.org/TR/smil20/, 07 August 2001.

[23] W3C. XSL Transformations (XSLT) Version 1.0. W3C Recommendation. http://www.w3.org/TR/1999/REC-xslt-19991116.

[24] 3GPP. Technical Specification Group Services and System Aspects, Transparent end-to-end PSS, protocols and codecs (Release 4). 3GPP TS 26.234 v1.5.1, March 2001.