A Framework for Media Resources Manipulation in an Adaptation and Negotiation Architecture

Tayeb Lemlouma *Nabil layaïda*

OPERA Project
INRIA Rhône-Alpes Research Unit.
ZIRST-655 Avenue de l'Europe -38330 Monbonnot Saint Martin, France.
{Tayeb.Lemlouma, Nabil.Layaida}@inrialpes.fr

Abstract

We define in this document a framework of media resources manipulation in developing an adaptive content delivery systems and technologies. We discuss the importance of media resources manipulation in an adaptation and negotiation architecture. The manipulation is discussed in term of extraction, description, binary relations definition (related resources approach) and resources adaptation. We give a simple recursive algorithm to extract media resources form an input tree structure; the media description is ensured by an extensible CC/PP profile that we define its form which should be followed on the rest of our work. We show how different resources can be related by predefined relations to be defined in order to help the adaptation and the negotiation tasks. We give also a general algorithm of media deliverance, according to media profiles (at both the client and the server side) and using the related resources approach. We show using a mathematical formula the importance and the influence of media choice at the deliverance of multimedia services in the heterogeneous environment. To see better these concepts, we give two practical applications of media manipulation: an html adaptation to WAP devices and a text to speech basic solution based on the related resource approach .

Key-words
Media resources, Adaptation, Negotiation, CC/PP, Heterogeneous environment.

1 Introduction

With the growing number of sophistical networks, hardware and software and the increased demand for access to multimedia services through these networks using several new terminals characterized by special constraints; comes a tremendous need for providing adaptation and negotiation techniques that end to deliver an understandable content to the service demander.

Offer an adaptable content is simply similar to express the same thing with two different vocabularies. The two vocabularies are not the same and may be different in the richness and the expression power. In multimedia contexts; the vocabulary represents the set of supported functionalities by the environment; the problem of the adaptation become difficult when the target vocabulary is poor or limited.

Media resources manipulation represents a great deal of the work concerning the adaptation and the negotiation of multimedia content. Indeed, negotiate the service to be used is based, at the end level of the deliverance, to determine which media to deliver, to adapt or simply to remove. In this document we give the framework of media resources manipulation in developing an adaptive content delivery systems and technologies. We focus on how medias will be processed in our multimedia content adaptation and negotiation architecture and this in order to enable clients terminals that present special constraints (limited communication, storage, processing and displaying capabilities, etc.) to access to servers content in the heterogeneous multimedia environment.

In Section 2, we give a general view of the considered multimedia architecture under which we aim to provide adaptation and negotiation solutions. Section 3 introduces the related media resources approach and how can it applied in content servers. The rest of the document presents the different sides of media manipulation: Section 4 defines the document instance vision rather than the document class or profile and this to process directly the media to be used at the end level of the deliverance. Section 5, shows how medias are extracted from instances using an algorithm that we give, and gives the general form of the CC/PP instance profile that we adopt, from the media resources consideration. Media client side and CC/PP client resource profile are discussed in section 6. Media manipulation using the related media resource approach and resources profile declaration and extraction are discussed in Section 7. in this Section we give two significant examples: the first concerns the adaptation of text medias to voices using related media resources approach, the rest of the work to implement really this application is just to create a rich data base of voices. The second example discusses the adaptation of HTML documents to WAP devices using a simple XSLT style sheet. In Section 8, we give the position of media resources in the negotiation of multimedia services in heterogeneous system, we give a formula that shows clearly the influence of the media choice on the total content delivery time. Finally, we present our conclusion in Section 9.

2 Content adaptation and negotiation architecture

The global architecture under which we aim to provide solutions for efficient services deliverance, includes three main components: the client, the server of content and the protocol layer that ensure client/server interaction in a good way.

The architecture considered is composed of the following entities:

1- *The server of content*: it maintains the multimedia content. Services may contain many heterogeneous medias, such as text, video stream, audio, etc. and may be authored in different versions. A server can use the content of another one belonging to the same multimedia system.
2- *Clients*: which have several characteristics and which requests their service demands to the servers of content.
3- *A connection network*: the connection network ensures continues communication between servers and clients. No assumption is posed on the connection bandwidth,

latency and accuracy. This means that client and servers may interact in bad conditions, which must be taken into account when delivering multimedia content.

And eventually:

4- *Intermediate proxies*: which can exist between clients and servers. A proxy may play the role of a client when the considered interaction is oriented proxy-server, and the role of server when the considered interaction is oriented proxy-client.

3 The related media resources approach

Most of works concerned by the adaptation in heterogeneous multimedia systems discusses the adaptation in terms of transformations to apply. This means to adapt a content *C*, we should have methods to transform it from its original state to the adapted state. The approach of *related resources* that we define here, enriches the adapting system by substitution possibilities rather than transformations, since that these last can not applied in all the cases. The role of related resources is to complete the work of the existed transformation methods and this to achieve the adaptation task efficiently.

Related resources can be defined as a set of binary relations that can be defined between a pair of media resources. A media resource can be an image, a text file, an audio file, a video stream, etc. and can be used by more than one document or application. A resource can be authored locally or imported and may be used by the local server or a remote one. It can be also obtained after applying some transformations techniques. A relation gathers two resources and helps the adaptation process; note that resources may exist in two different servers.

In this step of work, we define such relation as a value of the following set {<equivalent-to>, <adapted-to>}. Two resources related by an *equivalent-to* relation, means that in a multimedia use, they will be treated with the same manner. Generally, this can be found when the two resources keep the same characteristics regardless the aimed use. For example, at the server side, the author can define an equivalent relation between a JPG and a GIF images that contain the same information and use the same resolution. A resource *a* can be related by an *adapted-to* relation to another resource *b*, if this last can replace b in the multimedia use. This happens when the two resources represent the same object and have the same meaning but with different level of detail. For example, an *adapted-to* relation can relate a 256-color video to a 2-color one or simply a text file to its resume.

Related resources relations must be defined at server sides between common resources susceptible to be used by different clients. This definition is used by the adapting task and accelerates its execution. When a content *C* is to be adapted, the adaptation task is guided by related resources to the ones used by *C*. Resources found, and which can respond to the aimed context, are directly used. Note that this definition of related resources is not necessary between all resources, indeed some resources can exist in a single state or version and haven't another related resource.

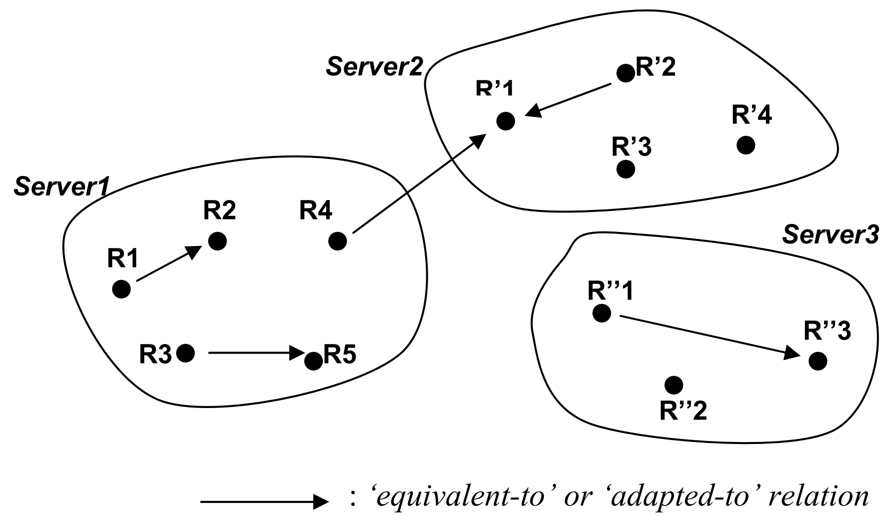

Figure 1: Related resources approach

4 Document instances

Document profiles describe a kind or a class of document, these documents may have some differences from the resources used or required point of view. The principle of *document instance* is different than the document class; an instance may use limited sub set of particular resources compared to the high document class. For example, an instance belonging to the same document class that supports videos can contain just a particular format of video or don't contain videos at all.

Considering document instances, at a low level of the negotiation task, guides more services delivery and makes it more efficient to respond exactly to the need of the client according to wanted resources in the multimedia use.

The following figure gives a global view of document organization -regardless existed resources- into a set of document profiles:

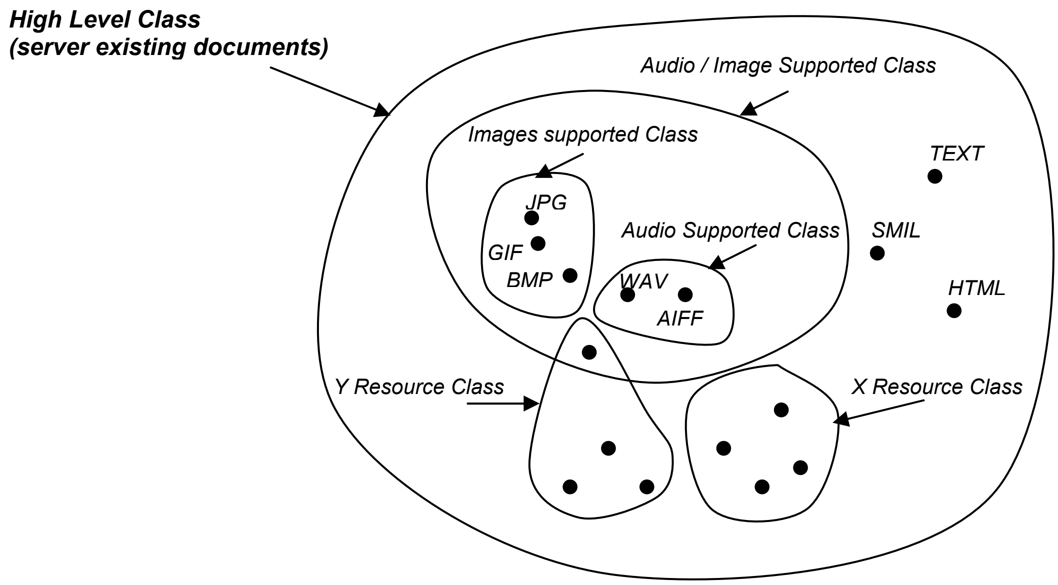

Figure 2: Classes of server resources organization

The instance principle let us thinking about the abstraction level of functionalities and resources used in defining and organizing documents. The lower bound of profiling definition is to consider each instance of document as an independent profile, the super bound is to make all existing instances belonging to the same profile class. A good choice is the one that applies an intermediate organization according to the existed functionalities and the heterogeneous environment constraints. Thus from the profile point of view, a SMIL Basic [11] class for example, contains the list of modules defined is the SMIL Basic profile. A SMIL Basic instance is defined by this list of modules and the detailed set of used resources by the document instance.

5 Determining media resources in document instances

In a heterogeneous system and at a low level of the negotiation task, determining resources in the multimedia service to be delivered is very important. This process helps to deliver exactly -when possible- the demanded service regardless the client requirements, and avoid any error (due to unsupported types or functionalities) during the service use by the client.

In our approach, the deliverance of multimedia services requires resources determination in two sides: the exited document instance side and the wanted service side. To achieve the determination in the first side, an instance is explored and a resources CC/PP profile is created. From the service side, the client send a resources CC/PP profile to show its capabilities and preferences about resources to be used. The CC/PP [3] choice is justified by the power of CC/PP and RDF [10] vocabulary in the environment constraints description.
Determining resources from a document instance can be done by exploring tags contained in the source file of the document. Only media resources tags are considered and all their attributes are examined. Note that in order to facilitate the exploring process, the instances to be examined must follow an XML[13] structure. So structure transformations must be applied in the opposite case to transform the initial structure to an XML one.
The process of exploring instances can be given by the following algorithm which represent a recursive procedure that treats a node *n*:

```
Procedure Treates_node(n)
{
        if (n represents a media resource)
        then
                create an entry in the output CC/PP profile;
                explore the n attributes;
                create media output attributes;
        else
                if (n contains other child  nodes)
                then
                        for each child s do Treats_node(s);
                fi;
        fi;
}
```

Figure 3: media resources extraction

The algorithm above, requires the knowledge of all media nodes or tags that can exist in a document instance i.e. in the source tree.

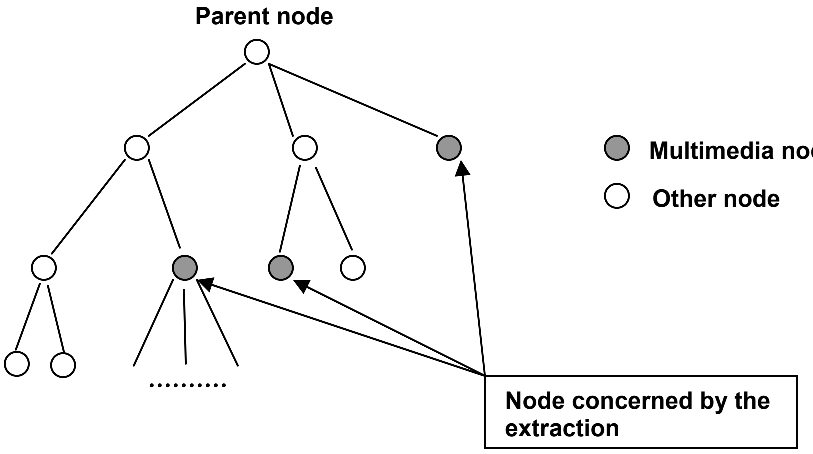

Figure 4: source tree structure

The set of tags depends widely to the syntax used in the input document for example in an HTML document the *img* tag denotes a media resource which is the image object. The algorithm of extraction is generally launched at the server side. The server must be able to determine resources attributes either with storing them initially when they start to be created, or using methods of resources evaluation. Attributes can give detailed information about the media object such as the format, the size, the resolution, etc.

5.1 Example of instance resources determination

We give here an example of media resources extracting from an input document. To simplify the comprehension we use a simple HTML input document. The extraction algorithm explores the input document and outputs the result in a resources CC/PP profile form.

The input instance is given as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
     "http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html">
  <title>WAP Technology</title>
</head>

<body bgcolor="#fafafa"
background="http://yap.inrialpes.fr/smil/images/background.jpg">
<center>
<h2>WAP Technology</h2>
<img src="images/phone.jpg" alt="phone" border="1" height="281"
width="99"><br/><br/>
<h3>WAP phones use Wireless Markup Language (WML) instead of HTML.</h3>
<br/>
WML is very simple by comparison of HTML, and easy to be automatically created
from monitoring scripts <br/> <br/>
```

```
<a href="audio/mobile/WapTechnology.ram">WAP Technology (audio stream)
<img border="0" src="images/realplayer.gif" width="20" height="20"/>
</a>
<p><img src="images/thank_you.jpg" alt="thank you for visiting" height="38"
width="167"/></p>
</center>

</body>
</html>
```

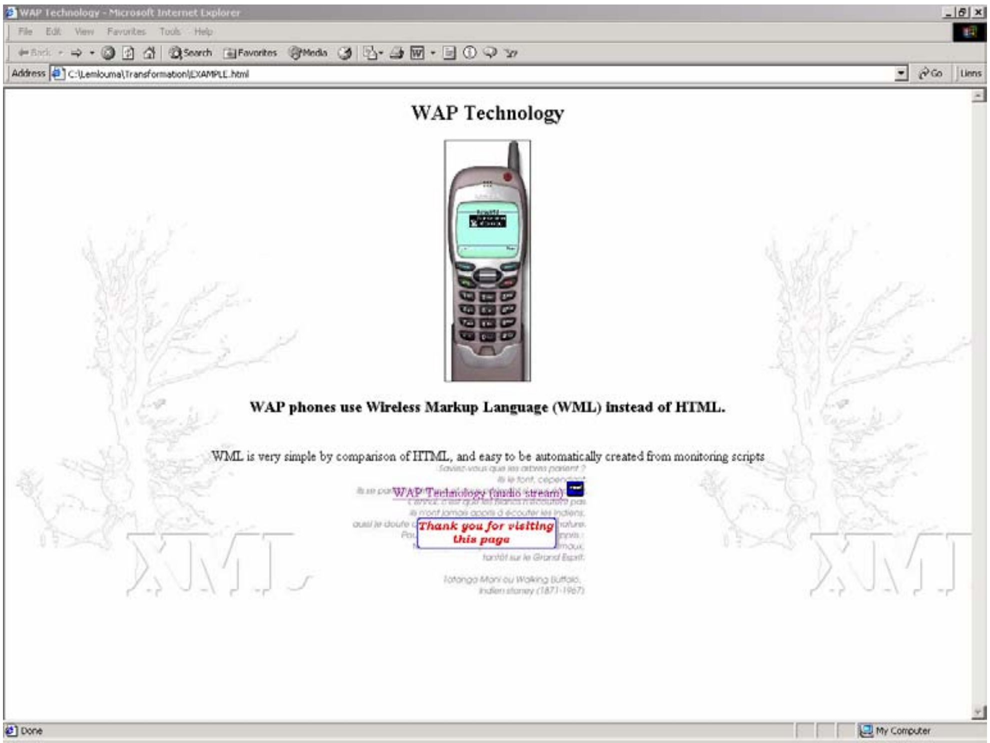

Figure 5: example of document instance

The following CC/PP profile corresponds to the correspondent profile after applying the extraction algorithm on the input instance:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
xmlns:neg="http://www.tayeb.negotiation.org/2002/01/01-neg#">

<rdf:Description rdf:about="InstanceResourcesProfile">

<ccpp:component>
<rdf:Description rdf:about="Document">
<rdf:type rdf:resource="DocumentClass"/>
      <neg:serviceType>HTML</neg:serviceType>
      <neg:version>4.01</neg:version>
      <neg:doctype>html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
      "http://www.w3.org/TR/html4/loose.dtd"</neg:doctype>
      </rdf:Description>
</ccpp:component>

<ccpp:component>
<rdf:Description rdf:about="DocumentInstance">
<rdf:type rdf:resource="DocumentInfo"/>
      <neg:name>EXAMPLE.html</neg:name>
      <neg:size>910Bytes</neg:size>
      <neg:location>WAP/</neg:location>
```

```
      <neg:server>LocalServer</neg:server>
</rdf:Description>
</ccpp:component>

<ccpp:component>
<rdf:Description rdf:about="MultimediaContent">
<rdf:type rdf:resource="MediaResources" />
      <neg:content>
      <rdf:Bag>
      <rdf:li rdf:parseType="Resource">
      <neg:type>image</neg:type>
      <neg:format>Joined Picture Expert Group</neg:format>
      <neg:name>background.jpg</neg:name>
      <neg:size>25824Bytes</neg:size>
      <neg:location>smil/images/</neg:location>
      <neg:server>http://yap.inrialpes.fr/
      </neg:server>
      <neg:dimension>800X600</neg:dimension>
      <neg:resolution>
      <rdf:Bag>
      <rdf:li rdf:parseType="Resource">
      <neg:horizontal>300</neg:horizontal>
      <neg:vertical>300</neg:vertical>
      </rdf:li>
      </rdf:Bag>
      </neg:resolution>
      <neg:color>
      <rdf:Bag>
      <rdf:li rdf:parseType="Resource">
      <neg:ColorBit>24</neg:ColorBit>
      <neg:ColorRepresentation>true RGB</neg:ColorRepresentation>
      </rdf:li>
      </rdf:Bag>
      </neg:color>
      </rdf:li>
      .
      .
      <rdf:li rdf:parseType="Resource">

      <neg:type>image</neg:type>
      .
      .
      </rdf:li>
      .
      .
      <rdf:li rdf:parseType="Resource">

      <neg:type>audio</neg:type>
      .
      .
      </rdf:li>
      </rdf:Bag>
      </neg:content>
</rdf:Description>
</ccpp:component>

</rdf:Description>
</rdf:RDF>
```


Figure 6: document instance profile

The form of the profile that we have adopted is simple but extensible to any kind of multimedia document. The profile is subdivided into to main parts: the first part concerns the document instance (format, characteristics, etc.). The second part concerns the multimedia content used by the instance. As we can see in this part, every media object contained in the input instance is represented by a ‘Resource’ element. The element contains a description of the media object (size, location, etc.). In the case where media resources are used by different document instances; media descriptions can be stored separately from the document instance profile and this to avoid storing the same information in different places. In this case, the media description will be presented as a link which points to the media description profile. Note that more the description is detailed, more the negotiation and the adaptation become efficient and thus the deliverance of services will be well performed.

6 Client side: resources declaration

From the client side, the determination of wanted resources consists on the creation of a profile to be declared to services providers. The profile is sent in the context of the actual use and depends widely to the demanded service. This means that profiles can change from an application to another (for example two different browsers, etc.), and according to services intended to be used by theses applications (wml, voice XML, etc.). Furthermore, profiles sending can also follow a predefined protocol. For instance, in the first step the client can send only its hardware description (the device description, etc.) which is -relatively static- and then software description in terms of media resources.

In the following, we give a CC/PP profile that corresponds to the supported resources declared by a WAP phone according to technical information found in [4]:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
xmlns:neg="http://www.tayeb.negotiation.org/2002/01/01-neg#">

<rdf:Description rdf:about="ClientResourcesProfile">

<ccpp:component>
<rdf:Description rdf:about="TerminalHardware">
<rdf:type rdf:resource="HardwarePlatform" />
                <neg:DeviceName>Ericsson-R320</neg:DeviceName>
                <neg:screen>30X23mm</neg:screen>
                <neg:display>101X52Pixels</neg:display>
                <neg:PixelStretch>1.24</neg:PixelStretch>
                <neg:row>
                  <rdf:Bag>
                      <rdf:li rdf:parseType="Resource">
                         <neg:type>Latin</neg:type>
                         <neg:value>5</neg:value>
                      </rdf:li>
                      <rdf:li rdf:parseType="Resource">
                         <neg:type>Chinese</neg:type>
                         <neg:value>3</neg:value>
                      </rdf:li>
                  </rdf:Bag>
                </neg:row>
                <neg:col>14</neg:col>
           </rdf:Description>
</ccpp:component>

<ccpp:component>
<rdf:Description rdf:about="Services">
<rdf:type rdf:resource="SupportedServices" />
                 <neg:formats>
                    <rdf:Bag>
```

```
                        <rdf:li>WML</rdf:li>
                     </rdf:Bag>
                  </neg:formats>
</rdf:Description>
</ccpp:component>

<ccpp:component>
<rdf:Description rdf:about="MultimediaContentRequierement">
<rdf:type rdf:resource="MultimediaContent" />
                 <neg:content>
                    <rdf:Bag>
                         <rdf:li rdf:parseType="Resource">
                            <neg:serviceType>WML</neg:serviceType>
                         </rdf:li>
                         <rdf:li rdf:parseType="Resource">
                            <neg:type>WmlCard</neg:type>
                            <neg:maxSize>3000Bytes</neg:maxSize>
                         </rdf:li>
                         <rdf:li rdf:parseType="Resource">
                            <neg:type>image</neg:type>
                            <neg:SupportedFormat>
                                 <rdf:Bag>
                                 <rdf:li>WAP bitmap</rdf:li>
                                 <rdf:li>Graphics interchanges format
                                 </rdf:li>
                                 </rdf:Bag>
                             </neg:SupportedFormat>
                             <neg:Size>
                                 <rdf:Bag>
                                 <rdf:li rdf:parseType="Resource">
                                 <neg:type>wbmp</neg:type>
                                 <neg:maxValue>1500Bytes</neg:maxValue>
                                 </rdf:li>
                                 <rdf:li rdf:parseType="Resource">
                                 <neg:type>gif</neg:type>
                                 <neg:constraint>heightXwidthLE12000Bytes
                                  <!-- LE: less or equal -->
                                  </neg:constraint>
                                 </rdf:li>
                                 </rdf:Bag>
                              </neg:Size>
                              <neg:alignment>no</neg:alignment>
                              <neg:link>yes</neg:link>
                         </rdf:li>
                         .
                         .
                         .

                         <rdf:li rdf:parseType="Resource">
                            <neg:type>text</neg:type>
                            <neg:style>
                                 <rdf:Bag>
                                  <rdf:li>small</rdf:li>
                                  <rdf:li>bold</rdf:li>
                                  <rdf:li>emphasis</rdf:li>
                                  <rdf:li>strong</rdf:li>
                                 </rdf:Bag>
                            </neg:style>
                            <neg:alignement>
                                 <rdf:Bag>
                                  <rdf:li>left</rdf:li>
                                  <rdf:li>center</rdf:li>
                                  <rdf:li>right</rdf:li>
                                 </rdf:Bag>
                            </neg:alignement>
                         </rdf:li>
                         .
                         .
```

```
                          .

                       </rdf:Bag>
                    </neg:content>


</rdf:Description>
</ccpp:component>

</rdf:Description>
</rdf:RDF>
```

Figure 7: client resources profile

The client resources declaration is written in the form of a CC/PP profile. The resources profile contains two main components: a component concerned by the hardware platform, and a component concerned by the service intended to be used. The service component contains a detailed description of media resources that can exist inside the service. The description follows the context of the client and the set of constraints presented in its environment.

7 The adaptation and related resources approach

Ensuring content adapting at the end level, is equivalent to adapting the general structure of the content and resources that it contains.

Related media resources approach doesn't represent a versioning choice to respond to the adaptation problem. The approach completes the work of the adaptation, and can also serves to build some transformation processes. To illustrate this, we give in the following the example of a transformation Text to Voice, which can be very useful in heterogeneous environment, or more precisely in the environments which provide voices services such as e-mail reading over the phone, talking web and screen reader.. etc. This application shows how this process of transformation can be entirely based on the related media approach.

Media resources in this case represent the data set of basic voices (all the single letters and some single words in different forms). Binary relations will be defined between text entities and voice entities. In this situation, relations can be expressed in a formal and general manner rather to explicit them between each pair of (text-entity, voice-entity). The related medias relations can be given as:

Text-entity → FormTextMedia (Voice.Media (Text-entity))

This relation associates a text entity to a voice one according to the state of the text. The function Voice.Media return the name X of the voice stored and which correspond to the vocal form of the Text-media. The function FormTextMedia determines the name of the voice file corresponding to the X voice at the same state of the text entity. The state of a text denotes simply the manner of how the text entity must be pronounced. The manner of the pronunciation is determined when parsing the input text document and according to the position of the text entity in the global sentence and/or the global tex. For example a letter which comes before a exclamation mark is related to the exclamation pronunciation state.
Relations between text and voice entities can be easily achieved following the same storage principle of audio files. For example by storing letter voices in files with the same name as the letter that they denote and concatenated to the state of the letter.

In this application, text medias are extracted from the input document intended to be adapted. The parsing of the text entities will be nicely done using a transformation mechanism. A simple XSLT[14] style sheet can be easily performed in order to achieve this task. The adaptation process follows so two main steps: The first step is to transform the input text document to an XML document containing the list of voices media to play. More exactly, it contains the list of links that point to the voices to play, and which are already stored in the server side. The heart of this transformation is to apply the media relation defined above, when parsing the input document. The second step consists to create the final voice document to be delivered. The output adapted document will have an audio format and contains the concatenation of all the voices included in the output XML document obtained after applying the XSLT style sheet.

We give in the following, an other example of multimedia adaptation based on transformation and substitution of resources. The example gives a simple style sheet which transforms HTML documents to WML. The style sheet is simple, a possible amelioration will be to parameterize it with the CC/PP client resource.

```
<?xml version='1.0' encoding="iso-8859-1"?>
<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform' version='1.0'>
<xsl:output doctype-public="-//WAPFORUM//DTD WML 1.3//EN" doctype-
system="http://www.wapforum.org/DTD/wml13.dtd" />

<xsl:template match="html">
<wml>
<card id="main" title="{head/title/text()}" newcontext="true">
             <xsl:apply-templates select="body/node()" />
</card>
</wml>
</xsl:template>

<xsl:template match="h1">
<big><b><xsl:value-of select="."/></b></big><br/>
</xsl:template>
<xsl:template match="h2">
<big><xsl:value-of select="."/></big><br/>
</xsl:template>
<xsl:template match="h3">
<b><xsl:value-of select="."/></b><br/>
</xsl:template>
<xsl:template match="h4">
<xsl:value-of select="."/><br/>
</xsl:template>
<xsl:template match="h5">
<small><xsl:value-of select="."/></small><br/>
</xsl:template>
<xsl:template match="h6">
<small><xsl:value-of select="."/></small><br/>
</xsl:template>

<xsl:template match="img">
<!-- This is for WAP devices that can't display images
[<i>IMAGE: <xsl:value-of select="@alt"/> </i>]
-->
<xsl:variable name="image_resource_name">
<xsl:value-of select="concat(substring-before( @src,'.'),'-adapted_to.wbmp')" />
</xsl:variable>
<!--
<xsl:if test="Exist_Resource(concat(substring-before( @src,'.'),'-
equivalent_to.wbmp'))">
$image_resource_name=concat(substring-before( @src,'.'),'-equivalent_to.wbmp');
</xsl:if>
The function Exist_Resource must be defined
-->
```

```
<img src='{$image_resource_name}' alt="[IMAGE: {@alt}]" />
</xsl:template>

<xsl:template match="br">
<br />
</xsl:template>

<xsl:template match="center">
<p align="center">
<xsl:apply-templates/>
</p>
</xsl:template>

</xsl:stylesheet>
```

Figure 8: A simple HTML to WML Style Sheet

The application of this style sheet on the document instance of the figure 5, gives the follow WML output document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>

<card newcontext="true" title="WAP Technology" id="main">
<p align="center">
<big>WAP Technology</big><br/>
<img alt="[IMAGE: phone]" src="images/phone-adapted_to.wbmp"/><br/><br/>
<b>WAP phones use Wireless Markup Language (WML) instead of HTML.</b><br/><br/>
WML is very simple by comparison of HTML, and easy to be automatically created
from monitoring scripts<br/>
<br/>
<img alt="[IMAGE: thank you for visiting]" src="images/thank_you-
adapted_to.wbmp"/>
</p>

</card>

</wml>
```

Figure 9: The output WML document

The Style sheet uses the related media approach in the adaptation of HTML documents to WAP devices. In addition to the transformation of the document structure, the sheet uses the binary predefined relation between media resources in the substitution of images which exist in the input document. The target of this substitution is the related wbmp images available in the server content. As it shown as comments, images can be only substituted by the alternative text and this for devices who doesn't support images at all. Concerning related media adaptation, a simple finding function can be developed in order to look for equivalent-to medias before the substitution by adapted-to ones. Don't forgot that the main objective of content adaptation is to provide understandable services, and to do in the same time the best effort to keep most of the content information. This is why equivalent-to medias have more priority than adapted-to ones.

The following table gives binary relations used in the adaptation HTML to WML:

Initial Resource	***Relation***	***Target Resource***
document.html	adapted-to	document.wml
background.jpg	nulle	nulle (removed)
phone.jpg	adapted-to	phone-adapted-to.wbmp
link to .ram	nulle	Nulle (removed)
realplayer.gif	nulle	Nulle (removed)
thank_you.jpg	adapted-to	*ALT* (text)
*h*1, *h*2	adapted-to	*Big*
*h*3	adapted-to	None*(normal)*
*h*5, *h*6	adapted-to	*Small*
Text	Equivalent-to	Text

Table1: A list of binary relations used in the adaptation HTML to WML

The presented principle of the style sheet can be used to cover other adaptation types, for example to substitute text to its resume or a kind of an audio format to another such as WAV to phone resolution voice, etc. Media substitution can also be obtained with launching real time transformations like transforming a text portion to voice. Remember that related media concept can be more detailed and complete in order to take into account many consideration related to the heterogeneous environment. The two values : *adapted-to* and *equivalent-to*, can be seen as a minimal set that can denote binary relations. More complete and rich relations based on detailed media attributes can be defined between medias in the context of the adaptation and the negotiation deliverance.

The general form of the server algorithm concerning *the deliverance of media resources* can be given as in the figure 10. The algorithm analyzes all the media resources which exist in the document instance profile that concerns the wanted service by the client. A resource is delivered if it can reply to the client user agent constraints. In the other case, the server tries to adapt it even thought that some characteristics of the initial media resource will be lost. If the resource can't be adapted using available local and remote adaptation methods of the system, the resource is removed.

```
For each
media resource category X contained in the profile of the document to be
delivered
Do
          If
     (1)  All attributes and elements responds to the corresponding resource
          Category X' (in  the client profile)
          Then
            X is to be delivered;
          Else
            Look for X-related resources (equivalent-to or adapted-to);
            Evaluate the resources;
                If (an appropriate resource exist) Then deliver it
                Else
            (2)  Look for available methods to make X adaptable regardless X'
                 Constraints;
                 If (such methods)
                 Then Apply the method on X;
                           Deliver the result
                 Else   Remove X;
                 Fi;
                Fi;
           Fi;
End for each
```

Figure 10: delivery algorithm

(1): The attributes of the initial resource can be accepted if its set of constraints is included or equivalent to the corresponding attribute in the required client resource. For example, a constraint like: 'monochrome color' for a video resource is included in the constraint '256 color' of the same resource category. To facilitate the achievement of this step, a high detailed description of resources must be done previously.

(2): Detailed description of the existed methods must already be stored. We propose that the description must be done in terms of input and output profiles description. After the creation of an adaptation method, the author must define the method capabilities in terms of what it accepts as input and what it outputs as a result. This means to give the profile that covers all the possible input documents and the profile of all which can be obtained after applying the adaptation method. The adaptation method can denote a simple style sheet, a program, or other combined techniques… etc.

Several adaptation methods can be applied in media resources. The MPEG-7 specifications document [8] gives a lot of descriptive for multimedia data, used in reducing the capacity requirement of large MPEG videos. Existing techniques of media adaptation [2][12][5] can be resumed in:

1- *Translation*: for instance video to image, text to speech, etc.
2- *Summarization*: text summarization, etc.
3- *Extraction*: key-frame extraction from video, etc.
4- *Substitution*: substitute a resource by a related-to resource: image substituted by the alternative text, etc.

8 The negotiation and media resources manipulation

The manipulation of media resources that exist in multimedia documents has a primary importance in the deliverance of multimedia services especially in heterogeneous systems. Media resources extracted from available services should not be sent directly to clients if they not respond to their needs and respect their constraints. They can be substituted, removed or transformed into an acceptable format. A detailed view of the existed resources intended to be used is so required to achieve properly the negotiation task. Media resources delivery is guided following the client description; since that the class of the service is not sufficient. For example, the demanded service can be a wml-1.3 document but the device doesn't support entirely the defined document type with absolute objects. This can be seen in the figure 7 (image resource, alignment element): the device –which correspond to an Ericsson R320 phone [4]- doesn't support the alignment of the image object. The server must avoid providing images with such attribute. Another solution, can be to include the image element in a <p align="center"> one.

The description form of medias supported by different user agent must be clear and extensible to all the heterogeneous components and specify the format of the wanted media resources that can be played locally.

In the negotiation of content based on the HTTP protocol [6], media resources represents an important dimension. The HTTP negotiation is based on the manipulation of multiple versions or variant of the same media resource which can be located under a single resource URI. The negotiation process uses media selections by matching the properties of the available variants to the capabilities and the preferences of the client expressed using Accept headers. These headers contain preferences for media resources types and eventually associated quality factors which indicate the level of the preference.

Using the HTTP/1.0 [7] negotiation, the client description is sent through Accept headers requests and the variants selection is achieved by the server of content. In the transparent content negotiation (TCN) [7], the list of the available variants and their properties are sent by the server and the selection of the best variant is done at the client side.

In the context of heterogeneous environments, HTTP negotiation principles present many limits. The sending of Accept headers in every request would be hugely inefficient, moreover the syntax of user agents preferences and capabilities is not large and extensible to cover the diversity of the clients and the existing media resources of servers content. In [9], we have present a protocol and basic tools to give a more extensible negotiation solution which can be applied independently to the used protocol (for example, it can be applied using an XML based protocol rather than the HTTP). In our approach, media resources can be easily and nicely described using the CC/PP and the RDF model; the expressing of media acceptance priorities is more opened which makes the negotiation task more efficient and easy to achieve.

In an other side, the deliverance of media services can depend to network constraints, for example a media resource can be substituted by the adapted-to related resource, if the later consumes less the bandwidth. The extraction of media resources helps to evaluate locally the effort to apply before services deliverance and consequently to choose the sweet resource and to use the best method to maintain a good QoS. To see better this, we give in the following a formula that indicates the relation between media resources and time delivery. Our formula is based on the evaluation done in [1]. However in our case we don't disregard the time necessary to execute transformation methods if no equivalent-to or adapted-to media exists.

We suppose that the original content *C* to be adapted contains *n* media resources *MR* and assume that fetching media time is negligible. A media resource can exist in a single or many versions. The document that represents the global content and which declares different resources is considered as a media resource too.

Let MR_i^v be the v^{th} version of the i^{th} media resource, $Size(MR_i^v)$ be the size of MR_i^v, $Transformed_k(MR_i^v)$ be the media obtained after applying the $Transform_k$ method on MR_i^v, $T\_Transformation_k(MR_i^v)$ be the time necessary to transform MR_i^v with the $Transformation_k$ method.

The size of the adapted content is:

$$Size = \sum_{i=1}^{l} Size(MR_i^v) + \sum_{i=l+1}^{n} Size(Transformed_k(MR_i^v))$$

Where media resources from $l+1$ to n are adapted using transformation methods. If B_A is the average network bandwidth and D_{RTT} is the network roundtrip time, the delivery time of the adapted content of *C* is:

$$Delivery\_Time = 2.D_{RTT} + \frac{Size}{B_A} + \sum_{i=l+1}^{n} T\_Transfomation_k(MR_i^v)$$

Our formula shows clearly the relation between media resources and the delivery time of the adapted content. This evaluation is very useful by the server of content while delivering services and this to respect the heterogeneous environment (clients, servers and network) constraints.

9 Conclusion

Media adaptation have a primary importance in a global architecture of content adaptation and negotiation. We have presented in this document an extensible framework for media resources manipulation in heterogeneous environments. Media manipulation was discussed in terms of extraction, description, binary relations definition and adaptation techniques. We have given also media extraction and delivery algorithms necessary to respond to clients needs, and shown the influence of the media choice in services deliverance with a formula that calculates the delivery time of a multimedia content.

Two useful applications were also given, in order to show how media resources can be processed in practical cases. The framework that we have defined in this document should facilitate the achievement of the two main tasks of the global architecture, which are the adaptation and the negotiation of multimedia content.

Bibliography

[1] J. Chen, Y. Yang and H. Zhang, *An Adaptive Web Content Delivery System*, In the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2000), 28-30 August 2000, Italy.
[2] Hongjiang Zhang, *Adaptive Content Delivery: A New Application Area For Media Computing Research*, Microsoft Research, China.
[3] W3C Working Draft, *Composite Capability/Preference Profiles(CC/PP): Structure and Vocabularies*, 15 March 2001.
[4] Ericsson mobile communications AB, *Mobile Phone R320 Design Guidelines for WAP Services*, January 2000.
[5] W. Y. Ma and Al, *A Framework for Adaptive Content Delivery in Heterogeneous network environments*, Multimedia Computing and Networking 2000, pp 86-100, San Jose, January 2000.
[6] Network Working Group, Hypertext Transfer Protocol - HTTP/1.0, http://www.ics.uci.edu/pub/ietf/http/rfc1945.html, May 1996.
[7] K. Holtman TUE, A. Mutz, Transparent Content Negotiation in HTTP, http://gewis.win.tue.nl/~koen/conneg/rfc2295.html. Network Working Group, March 1998.
[8] *MPEG-7 Specification document*.
[9] Tayeb Lemlouma, Nabil Layaïda, *The 8th International Conference on Multimedia Modeling*, CWI, Amsterdam, The Netherlands, to appear, November 5-7, 2001.
[10] Ora Lassila, Ralph Swick, *Ressource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation: http://www.w3.org/TR/1999/REC-rdf-syntax.
[11] Kenichi Kubota, Aaron Cohen, *SMIL Basic Language Profile*, 22 June 2000, http://www.w3.org/TR/2000/WD-smil-boston-20000622/smil-basic-profile.html.
[12] L. Sampath, A. Helal, J. R. Smith, *UMA-Based Wireless and Mobile Video Delivery Architecture*.
[13] W3C, *Extensible Markup Language (XML) 1.0*, W3C Recommendation: http://www.w3.org/TR/1998/REC-xml-19980210, 10 February 1998.
[14] World Wide Web Consortium. *XSL Transformations (XSLT) Version 1.0* W3C Recommendation. See http://www.w3.org/TR/1999/REC-xslt-19991116.