Device Independent Principles for Adapted Content Delivery

Tayeb Lemlouma[1] and Nabil Layaïda[2]
OPERA Project
Zirst 655 Avenue de l'Europe - 38330
Montbonnot, Saint Martin, France
Tel: +33 4 7661 5281 +33 4 7661 5384
[1]Tayeb.Lemlouma@inrialpes.fr [2]Nabil.Layaida@inrialpes.fr

Abstract

In this short paper, we describe briefly different mechanisms and techniques that can ensure an adapted content delivery and this based on the device independent principles. We identify some points that represent in our point of view some main keys toward an adaptable environment.

1 Device Independent Languages

Markup languages play a major role to provide content adaptation solutions for different contexts. Thanks to their advantages (platform independence, open standards, flexibility, extensibility, etc.), the adaptation processes can be applied easily than on a pure non structured content.

In the context of devices independent delivery, some actual languages offer many advantages and adaptability facilities. Among these languages we find XHTML that combines the advantages of both XML and HTML, SMIL with a whole module (content control module) dedicated to support device heterogeneity and the adaptation for different delivery contexts, XEvents for user interactions, XForms for processing forms, etc. Unfortunately, there is no languages -in present- that are completely device independent and integrates all the aspects of content adaptability. From this point, two directions arise: a) Extending existing languages and content models with new elements and attributes in order to approach the objective of device independent, and b) The definition of new languages that address directly that objective.

In [7] and [8] the first approach is adopted. In [7], the presented product family (called Mariner) tries to ensure a solution for developing Web sites that support access from a wide range of devices with adopting device independent techniques and avoiding the use of heavy traditional approaches. The approach is based on the extension of XHTML since this last is not completely compatible with device independent application such as its coupling to the implementation of its presentation. For example, the approach achieves the presentation according to a set of stylistic information associated to the content to be generated exactly as in the classical use of CSS and XSL Formatting Objects. The adopted authoring model starts from XHTML and adds new elements and attributes and removes some others. Here we find some similarities with SMIL, especially in the definition of the *pane* element which can be seen as a simplified definition of SMIL regions. The physical layout of the final presentation is done by arranging the *panes* within the layout definition used for the target device. So, *panes* can be arranged but also substituted (by an explicit mapping from the author) to links that point to their content if the display is limited. From this point, the approach requires preventing layout definitions for each device or group of devices which makes the support of dynamic context change (concerning profiles and especially those related to the user preferences) less supported.

In [8] single authoring techniques are presented using a user interface. Here, the used markup language is XHTML and CSS for the style and the transformation process uses XSLT and XSL for transformation rules. Here, the markup language is also extended with new elements defined in a DTD for adaptability reasons. The criteria that is considered in the adaptation process is represented mainly by the displaying capabilities of the device. In this approach, the extension is based on the definition of a new tag library (called AUI) that helps to perform the adaptation engine. Using these new elements, different stylesheets and medias are defined for different contexts and are included in the original content. Following an approach similar to the SMIL *switch*, the context in which a stylesheet or a media resource can be used is expressed using some attributes in the form of: terminal='pc', terminal ='pda', etc. The alternative that best fits the target device is selected. Including different parameters of the style and media context inside the original content has many disadvantages: length of the original content, incomplete description of the target context which can cause some problems in the adaptation, reuse of the same description inside different contents especially for media resources, etc.

As we have already seen, the second approach is based on the definition of new languages that should be completely device independent. The work in progress presented in [3] belongs to this approach. The objective is to build a new markup language that can be used to adapt application mobile interfaces for heterogeneous devices. To reduce the effort that can be made for the wide range of clients, devices are classified into different classes (four in present) depending to their capabilities. The definition of the new markup language (called Renderer Independent Markup Language or RIML), will be based one some existing markup languages thanks to their device independent advantages. The actual identified languages are: XFroms, XEvents and XHTML. The approach will help, certainly, the process of delivering adapted content but remains tailored to the user interface side of the complete delivery process.

2 Content Fragmentation

The classic concept of a "document page" depends generally to a particular context and more exactly to the physical characteristics of the target client. Indeed, a physical page depends widely to the displaying capabilities of the device. This is why this vision must be left to a more flexible one that depends more to the content and less to the delivery context.

Defining a logical content fragmentation represents an important side of the device independent authoring model. Pieces of fragments serve to make easier the application of the adaptation engine and helps to keep a semantic and presentation coherence inside the adapted content. Once a model defines this, authors can create easily their content and fragment it according to the content characteristics. The fragmentation will be used to perform the adaptation in order to deliver the content for many contexts such as for different display sizes or for different stylistic presentations.

The problem that arises here is the definition of atomic units that can build fragments and consequently build the whole content. In [7] this atomic unit is represented by the *pane* element. This one regroups different elements of the initial language (XHTML) and is used during the adaptation process. Thanks to that definition, fragments can serve, for example, to deliver a long page in the form of small pages for a small device or to rearrange the layout with changing the order of the fragments. However, this approach presents some difficulties when the fragment nature is incompatible with the target context (ex. between the fragment size and the atomic unit accepted by the device) which requires further sophistications in the adaptation engine. A similar approach in defining fragments is presented in [2]. In this framework, fragments definition is enriched with added attributes and elements that can help more to perform advanced adaptations. We find for example, the definition of areas (again similar to SMIL regions) associated to a priority attribute where affecting the value is the content author responsibility. Propriety attributes in conjunction with the model expression of alternatives enable selecting the most suitable variants during the adaptation engine. The proposed framework is based on the concept of device classes which does not always ensure satisfactory results since the individual characteristics of the client within a class need to be considered separately. Furthermore, associating a global priority value for a mixed fragment or area can result, in some situations, on removing the fragment even if it contains elements that can be supported by the target device.

The approach followed in [8] can be seen as a more fine-grain definition of the fragmentation. Here, any element of the original content and that belongs to the adopted authoring model, can be seen as a fragment. Paginating the content (or breaking large bodies according to the device capabilities) is achieved by patterns using delimiters and by fixing the target 'page' size. This approach allows the application of more advanced adaptation. However, it lacks for semantic regrouping of different objects inside a fragment unit which increases the risk of generating incoherent presentations after applying the adaptation.

3 Managing Content Versions

One of the important sides of device independent principles is to manage content variants or versions. This aspect must not be ignored even thought that advanced techniques have the tendency to replace using variants by creating them dynamically. Many languages have taken this into account starting from the simple management principle of the HTTP protocol [6] [5], to the simple alternative expressing of HTML, to the SMIL *switch* [4], to new efforts that uses the same principle or simplify it.

Managing the content variants can require a complicated process that includes: retrieving the variant characteristics from the content and from the resource itself, checking the coherence after the variant substitution and, determining the context in which the variant can be used i.e. the context related to the client capabilities, such as the colour displaying capabilities, but also to the environment such as the required bandwidth. Since variants exist generally in the content provider side, the author has generally an idea about their context. Consequently, the device independent model should offer to the author the possibility of describing and expressing that context using a minimal set of attributes. In this context, SMIL represents a good model to be followed with its *switch* element and the different delivery context attributes such as: systemScreenSize, systemBitrate, systemLanguage, etc. [6]. This is why most of the new efforts reuse the same SMIL approach.

Another point that meets the need of the definition of a good variant managing strategy is alternative and media referencing. Resources referencing problem is not restricted to variants managing but concerns in general all the process of content adaptation. Actual languages show clearly that direct referencing is pure device dependent and thus makes the content adaptation hard to achieve. Solving properly this problem will be the progression of the work that we have presented in [6].

4 Gathering Device Profiles, Classes Definition and Dynamic Support

Since content adaptation solutions target the end user and the delivery context; it is logical that proposed strategies will be based on the client characteristics: capabilities and preferences or which is simply called *client profile*. With the wide range of devices that exist today, the problem is how to deal with this large amount of characteristics. Should we classify devices according to large criteria such as sound or input capabilities, or should we create and gather profiles for each existing device

Existing solutions based on predefined classes of devices and static profile repositories show their limitations to guarantee a fine-grain adapted delivery. Based on predefined classes, two problems arise: classifying the device and adapting the content for a given class. The first problem is presented when the considered device shares many criteria from more than one class which is due to the way adopted in the initial definition of classes. The second problem is presented generally when the device belongs to a class but needs some further adaptations related to its specific characteristics. Based on static profile repositories, the major problem of dynamic context change such as the change of the user preferences is presented. Furthermore, static repositories approach does not guarantee an adapted delivery for future and unknown devices.

For a dynamic support of the device context, adaptation engines should be based on the actual characteristics of the client. Thought that -initially- the client is only entity that knows its current context, the adaptation should be based on the information conveyed by the device included in its request (request headers, etc.) and eventually other information that can be obtained using other particular protocols (negotiation protocols for instance). In all the situations, any static information that concerns the device (considered for example after classes identification) must be replaced by the received one. Following this way, a good approach would be the one that distinguish between static and dynamic client characteristics. Static characteristics concern generally the device capabilities while the dynamic ones concern generally the user preferences. Thus static profiles can be predefined and dynamically enriched by current information. The resulting profile will be the one considered by the adaptation process.

5 Client Side Adaptation

The general framework of an adaptable multimedia system should not ignore the role that can be played by the end client in order to help in the achievement of a complete content adaptation. CSS represents an ideal example of such approaches [1] that was exploited in several efforts.

In spite of some opinions, we think that even thought that devices become more and more sophisticated; making the target device responsible of whole adaptation process is a bad choice. Even if the client has an advanced capability to process the received content, the content provider (with eventually intermediaries) remains the entity that can manage and adapt well the content. The server can have a global idea of the delivery context including the content characteristics and related metadata, the media to be delivered, the existing variants, the client (using its request and other received information), the network, etc.

The adaptable environment should delegate a part of the adaptation process (such as some stylistic tasks) to the end device. This delegation could nicely take a profit from the client profile. Thus, if the content provider finds that the client is able to perform a part of the adaptation, a semi adapted content is delivered and finishing the adaptation task will be done in the client side.

6 Content Semantic Need

From the experience of current implemented solutions that address the problem of content adaptation in heterogeneous environments, the problem of adaptation flexibility and content coherence remains unsolved. In many situations, the adapted content: does not reply to the need of the client, example: a principle section of the original content is removed, or is semantically incoherent, example: the client receives a page in which a figure title still existing and the figure does not.

These two examples do not cover all the problems that can occur in the resulting adapted content. Both of them are related to the same reason that is widely related to the lack of semantic information inside (or used by) the authoring model. Such semantic metadata can include the definition of relationships between objects or fragments of the same content, relative level of importance concerning different parts of the content, metadata concerning media resources and documents, semantic relationships between resources and variants (media or structured resources), etc. Semantic information could help the adaptation process and avoid to deliver a well structured content that does not mean anything from the client side. For instance, if a semantic relationship expresses the dependence between two objects (ex. X is the title of Y); the adaptation process should not remove one of them and let the other.

Device independent languages need to develop advanced semantic vocabularies for adaptability considerations. Some existing technologies can be nicely exploited. Among them, RDF [9] takes an important part and can be adopted to satisfy this need. Adopting RDF will develop the adaptation efficiency and extend its applicability from stylistic considerations to semantic considerations.

7 Conclusions

In this short paper, we have tried to identify some points that we consider important in the framework of developing an adaptable multimedia system. As we have seen, the design of such systems could not be possible if we do not adopt the main device independent principles because we target the delivery of an adapted content for a wide rang of clients.

An efficient solution requires certainly a sound authoring model but also advanced vocabularies that should guarantee to understand the functionalities and structure of the content and the used media resources. In parallel, there is the adaptation process to be applied and that must not ignore which is offered in the environment: client capabilities related to achieving the adaptation, existing variants, available metadata, etc. This is why, fixing -initially- where the adaptation should take part is not preferred.

References

[1] B. Bos. The Place of CSS in Device-Independent Publishing. W3C. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[2] F. Giannetti. Device Independence Web Application Framework(DIWAF). HP Labs. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[3] G. Grassel, M. Lauff and A. Spriestersbach. Definition and Prototyping of Renderer-Independent ML. Nokia, SAP and IBM Germany. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[4] T. Lemlouma and N. Layaïda. Content Adaptation and Generation Principles for Heterogeneous Clients. OPERA project, INRIA Rhne Alpes. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[5] T. Lemlouma and N. Layaïda. A Framework for Media Resources Manipulation in an Adaptation and Negotiation Architecture. http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/RSRS.pdf

[6] T. Lemlouma and N. Layaïda. SMIL Content Adaptation for Embedded Devices. Submitted to SMILEurope 2002 Conference.

[7] R. Lewis. A Device Independent Method for Web Site Authoring. Volantis Systems Ltd. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[8] S. Mandyam, K. Vedati, C. Kuo and W. Wang. User Interface Adaptations: Indispensable for Single Authoring. Covigo Inc. *W3C Workshop on Device Independent Authoring Techniques*, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

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