Abstract

This paper addresses the adaptation of dynamic and synchronized multimedia presentations built by querying XML compatible data sources. We provide WIS designers with facilities for describing presentations whose content is not known at design time in terms of quantity, but only after the execution of queries. Our approach relies on the definition of a template. A template consists of a model that aims at automatically adapting the multimedia content of a presentation to both the user’s profile and the characteristics of her/his access device. We show here how a template is built and how adaptations of the presentation are performed when the quantity of information and/or the material capabilities of the access devices (e.g. display size), do not match the template’s spatiotemporal specifications.

1. Introduction

Nowadays, documents available on the Internet include data of several media types (text, 2D/3D graphics, video, audio, etc.) which are temporally and spatially combined in so-called synchronized multimedia presentations. Exploited in Web-based Information Systems (WIS) [4] as an access interface to their functionalities (delivering, querying, modifying, etc. information), these presentations are moreover considered as dynamic since they are built at run time using multimedia contents extracted from databases. To face the increasing diversity of both users and access devices (workstation, laptop, PDA, etc.); WIS designers should be provided with tools for specifying dynamic and synchronized multimedia presentations adapted to the diffusion context.

In this context, the global objective of our work is to offer a platform (called KIWIS [7]) for the design and the generation of adaptable WIS. The adaptations are both user and device oriented and concern the content, the functionalities as well as the presentation of information.

In [7], we have addressed the issue of adapting the content and functionalities of a WIS to its users. One aim of KIWIS is to allow the design of dynamic and synchronized multimedia presentations which will be, at delivering time, automatically adapted to both the user’s profile (interest, knowledge, preferences, etc.) and the device characteristics (screen size, memory capacity, etc.).

In this paper, we focus on the adaptation of dynamic and synchronized multimedia presentations built by querying semi-structured and XML compatible data. We assume that the content retrieved by the query is adapted to the user’s interest and preferences.

We specially aim at providing WIS designers with facilities for describing presentations whose quantity of information is not known at design time (but only after the execution of queries). This unknown set of information encompasses multimedia data, which have to be combined in space and time, some requiring a minimum surface to be displayed (except for audio data) and/or a minimum time interval in order to be played (i.e. continuous multimedia data such as video and audio).

Designing presentations of multimedia data obtained from queries implies to anticipate problems related to their spatiotemporal organization. One issue we address here is the adaptation of the data spatiotemporal organization inside the available displaying surface which differs according to the device used (workstation, laptop, mobile devices, etc.).

Our approach relies on the construction of a template which gives the specifications of a presentation model and which is automatically adapted at instantiation time (i.e.
when the final presentation is generated). Templates are built in accordance with a model called STAMP (Synchronized Templates for Adaptable Multimedia Presentations).

Based on the model STAMP, a WIS delivers multimedia contents which are adapted in terms of presentation without any need, for the designer, to explicitly specify how the adaptation process performs. Our solution relies on the dynamic generation of multimedia presentations which is controlled by an adaptation engine. This engine is able to transform the presentation in order to compensate both a too high number of information elements and a lack of available surface for display, compared to the specifications of the chosen template.

This paper is organized as follows. Templates built using the STAMP model are presented in section 2. In section 3, we briefly describe some processes performed by the generation engine when adaptation is required. The instantiation scenario is presented in section 4 before we conclude.

2. The STAMP model

In order to build adapted multimedia presentations, our approach is based on the description of templates. Templates can be seen as predefined patterns, which are instantiated for the generation of adapted presentations of multimedia data. A STAMP template describes the dynamic content of a multimedia document according to six dimensions: three traditionally used (spatial, temporal and navigational) and three innovative ones (data, composition and decorative). The spatial and temporal dimensions respectively define how the media elements of the presentation are organized in space and time. The navigational dimension manages the hypermedia links which define the network of Web pages of the WIS. In the remainder, we focus on the data and the composition dimensions.

The Figure 1 shows the general structure of a STAMP template described in XML. Each dimension is described independently. This description is modular and flexible. If one dimension can not be taken into account by the access device, the other still can be.

The data dimension of a template is made of a set of elements called datasources. Each datasource corresponds to a query and describes the structure of the expected result. The data dimension is described independently from any structure so that the corresponding content can be presented in many (and personalized) ways.

To cope with the indeterminism (in terms of number and formats of the media objects to be displayed), we propose a set of composition structures which adapt the presentation to the number of items to be presented. In STAMP, we propose four composition structures in order to describe presentations (oneDataStruct, kDataStruct, indexStruct and mixedStruct).

A structure element is associated with one frame in the presentation. For instance, if the WIS designer wishes to build a presentation including one index zone and one zone for displaying the content of the selected item, she/he has to describe three structures: one for the index zone, one for the selected item presentation zone, and one main structure (<structure id="main" ...>). This includes the first two. In each structure, four optional sub-elements (spatial, temporal, navigational, and decorative) describe the corresponding dimension.

Within each structure, the designer can organize the elements of a result item according to their importance in the presentation. This organization will guide the adaptation process by pointing out which elements must be preserved and displayed in priority. We propose to organize elements by extending the nucleus-satellites and multi-nucleus metaphors introduced in the Cuypers system [6]. The orbits of a nucleus are ordered so that the closest the orbit is, the more relevant its satellites are.

Figure 1. The General Structure of a template

By associating a composition structure with a data source, we establish a link between the presentation and its content extracted from a query. The next step consists in specifying, independently and for each structure and sub-structure, its spatial, temporal, navigational and decorative dimensions.

3. Adaptability in STAMP

We discuss here how a template is adapted in case of conflicts. Conflicts occur when i) too much data are contained by the result compared to the chosen composition structure, ii) when space is lacking compared to the size of the media elements, iii) when time is lacking compared to the fixed duration. We focus here on the first two cases of conflict (for the third case, see [4]).

All the adaptations presented hereafter are automatically handled by the adaptation engine and are transparent to the designer (and the end-user).

3.1 Adaptation to the number of data

The number of items of the result being known only at run time, the STAMP instantiation engine may have to face a lack or an excess of data. The first case is not
awkward because the presentation of the items can be achieved within the same structure. Places reserved for displaying missing items are left empty. However, if space is lacking we have to reorganize the template.

One solution is to reduce the number of items to be simultaneously displayed (e.g. passing from 8 items to 4 items per page). However, for a composition structure presenting one item (a oneDataStruct) complementary solutions are needed.

Three solutions can be adopted in this case:
- ignoring the additional items by presenting only the first one,
- producing an item by item presentation (for instance, a slide show based on the initial oneDataStruct),
- presenting k items simultaneously (inside a kDataStruct). The value of k is determined dynamically according to the item sizes. Changing a oneDataStruct into a kDataStruct may require some adaptations to the display surface, because it consists in reproducing k times the same structure in a surface initially designed for one.

### 3.2 Adaptation to the available surface

A lack of space is detected when a spatial constraint is not satisfied because the display surface of the access device is too small for presenting one of the media object or because one of them does not satisfy a spatial constraint.

In some cases, a spatial conflict can be solved by reducing the size of some media or by substituting some of them (multiple format adaptation). Several negotiation techniques are proposed in [5].

When no multiple formats are available, the solution leading to the reduction of the media object – unavoidable when the media objects are larger than the screen size – often decreases the quality of the presentation.

In STAMP, we prefer to act on the number of media objects to be presented using three possible solutions: i) excluding one of the media objects, ii) keeping all the elements and modifying the temporal scenario, iii) building a new structure containing some elements excluded from the initial presentation and linking the two separated structures by using hyperlinks.

### 4. Instantiating STAMP templates

The Figure 2 illustrates the transformations applied to a STAMP template throughout the instantiation process.

First, the template is filled up with the result items. If a data overflow occurs, it is solved by selecting subsequent subsets of result items whose sizes fit. According to the chosen compensation, navigational links or temporal relations are introduced in order to link the main presentation to the subsequent presentations concerning the other result items, if any.

The adaptation engine relies on two agents for acquiring contextual information and an optional third one concerned with user’s preferences in terms of adaptation strategies. The first agent provides some information related to the size and duration of media objects. The second one provides some information on the CC/PP profile associated to the user’s access device. This information is used for validating the set of temporal and spatial constraints for each composition structure.
For instance, the spatial relation $A$ is on the left of $B$ is rewritten using a linear inequality $A.left+A.size>B.left$. The same transcription is performed for each temporal relation: $A$ meets $B$ corresponds to $A.begin + A.duration = B.begin$, etc. The set of generated constraints is completed with implicit ones which ensure a consistent presentation. Each object must be fully displayed on the screen ($A.left+A.size<=DISPLAY.width$ and $A.left>0$). Each object must be completely played during the presentation ($A.begin>0$ and $A.begin + A.duration <= TOTAL\_TIME$).

This set is validated against the spatio-temporal properties (size and length) of media objects. The $left/begin$ attributes of an object are free variables while the attributes width/height/duration are linked to the values extracted using the first agent. The $DISPLAY$ and $TOTAL\_TIME$ properties are extracted from the CC/PP profile of the access device and user’s preferences.

Consistency is checked by a constraint engine. When a spatial or temporal conflict occurs, the ‘guilty’ element is the one with the lowest relevance (according to the composition dimension). However, other strategies can apply. For instance, the shortest (in time) element or the composition dimension). However, other strategies can apply. For instance, the shortest (in time) element or the composition dimension. Each object must by fully displayed on the screen ($A.left+A.size<=DISPLAY.width$ and $A.left>0$). Each object must be completely played during the presentation ($A.begin>0$ and $A.begin + A.duration <= TOTAL\_TIME$).

The adaptation strategies (performed by an adaptation engine at instantiation time) rely on the exclusion of the elements which have the lesser relevance and/or on spatial, temporal, and/or navigational compensations, respecting as much as possible the initial intention of the presentation described by the template.

Our future work concerns the formalization of adaptation strategies used in STAMP. Some conflicts can be solved using more than one strategy. Giving a weight measure to each adaptation strategy according to some criteria (relevance, computing resources needed, user’s preferences, loss of quality etc.) could allow the adaptation engine to calculate the optimum series of adaptations. For this purpose, the semantic adaptation model proposed in [3] could be used. It associates the transformation process of a document with a graph whose nodes represent the different versions of the initial document and where the links between nodes correspond to the applicable strategies along with their weight. The optimal path (from the initial template to a template respecting the diffusion context) in the graph matches the optimum series of adaptations to be applied on the initial template.

6. References