An automatic formatting system for multichannel cross-media contents

Paolo Nesi
DISIT LAB
Department of Systems and Informatics
University of Florence
Via S. Marta, 3
Florence, Italy
nesi@dsi.unifi.it

Paolo Vaccari
DISIT LAB
Department of Systems and Informatics
University of Florence
Via S. Marta, 3
Florence, Italy
vaccari@dsi.unifi.it

ABSTRACT

The market of digital media is now a heterogeneous environment where multimedia contents are accessed through different channels and devices. Many problems are related to the production and distribution of cross-media content for multiple channels, in particular when production on demand is required: adaptation features are needed to change radically the object usage according to the different environments, optimizing the layout of the visual presentation and adapting consequently digital resources. This paper presents the architecture of a system for automating the process of formatting multimedia presentations and adapting digital resources which compose them. SMIL and XSLT are used to describe presentation models that can be reused and adapted according to user profile, device capabilities and delivery channel characteristics. The work presented here has been performed for AXMEDIS (Automating Production of Cross Media Content for Multi-Channel Distribution) IP FP6 Research and Development Integrated Project of the European Commission.

Categories and Subject Descriptors

H.5 [**INFORMATION INTERFACES AND PRESENTATION**]: Hypertext/Hypermedia—*Architectures*

; I.2 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Cross-media adaptation and formatting

Keywords

cross-media, content adaptation, formatting, layout optimization, content processing

1. INTRODUCTION

In the last few years, the growing of multimedia market over the Net has interested personal computers and new devices as palm computers and smart phones. At the same time, market requires, for all those devices, rich contents which are natively composed by many different media. For instance: MP3s can be more attractive if provided in some presentation with singer's photos and lyrics for karaoke, movies should be offered with extra contents and a DVD-like menu, and so on.

Unfortunately, creating such a multimedia/cross-media presentation is not a simple task [2]; its manual authoring is very expensive and time consuming. On the other hand, reducing costs for formatting may lead producers and distributors to also offer less demanded contents from the so called "long tail" in nicer presentation, making them more attractive and thus opening new business opportunities. To be economically interesting, those presentations should be dynamically generated on the basis of contents, distribution channel, device capabilities and user preferences. Moreover, the opening of new business models is accompanied by the interest in exploiting a large amount of content that is in the hands of several actors along the content value chain and in cultural institutions.

Cross-media content production and distribution can be very difficult to be realized on demand by using traditional technologies of signal-level adaptation solutions. For cross-media content, the technologies of signal-level adaptation and quality of service control have to be rethought, since the adaptation may involve the redefinition of the content meaning and semantics. This tasks include redefinition of layout and interaction modalities, in addition to the adaptation of digital resources, taking into account natural environment, device/terminal, network and user profiles [8]. An alternative to the production on demand may be the creation of content in multiple formats in advance providing access and selection at the delivering time. This modality of working is space consuming and may be difficulty to cope with all the requests of the several kinds of devices, network, and user preferences. In this sense, some cross-media models allow to refer at different resources for different cases (e.g., in SMIL) or to provide different modalities of work in the same content – as, for example, in MPEG-21 with choice/selection mechanism.

More sophisticated models have been created to provide support for adaptation of resources and/or layout: a “cross-media adaptation” could be obtained with a semi-automatic selection of alternative contents based on their semantic value [1]; on the other hand, constraint based authoring environment with temporal and spatial specifications have been proposed for layout adaptation [7].

Mathematical and operational research techniques have also been adopted for layout optimization. The problem of laying out the regions of a presentation into the player’s window has been considered as a two-dimensional bin-packing problem [6] and solved using a Genetic Algorithm. GAs have also been used for determining the layout of a digital photo album [5], for optimizing the advertisement layout on a newspaper, and for optimizing the layout of a web page.

This paper describes a solution to automate production of multimedia presentations, enhancing possibilities of reuse and adaptation of contents. The production of cross-media content takes into account adaptation aspects regarding the object usage paradigm, the layout of the visual presentation and the format of digital resources. The solution uses models based on SMIL and XSLT style-sheets, integrated with a meta-heuristic algorithm based on Genetic Algorithms for layout parameters optimization. The system may be used totally automatic even for content production on demand and/or may assist the content producers, integrators, distributors in selecting and reusing existing models (template and style-sheets) suitable for the given resources (on the basis of their descriptors), and possibly in modifying them to create new ones that better fit the specific needs. This should help in dramatically reducing costs for mass-production of multimedia content production of the presentation aspects.

The proposed system has been developed for AXMEDIS (Automating Production of Cross Media Content for Multi-Channel Distribution) FP6 Research and Development Integrated Project of the European Commission [3]. AXMEDIS is based on MPEG-21 format and supports the content production and processing by means of GRID technology, so that the functionalities presented in this paper are available in the AXMEDIS Grid and multimedia processing language.

The rest of the paper is organized as follows. Section 2 provides a system overview and architecture, together with some details regarding the design; some examples and experimental results are presented in Section 3. Finally, Section 4 presents the conclusions of this work.

2. SYSTEM OVERVIEW

The major tools for automated rendering of web documents or presentations offer separated models and tools between the content and its appearance. For example: a template system may allow to automate creation of documents or presentations; while style-sheets allow the definition of the appearance and make easier to apply presentation level changes.

In the proposed solution, separation between cross-media content model and presentation is also used to simplify the production of presentations and to enhance their reuse. In fact, all presentations that offer the same type of content elements have almost a similar structure, therefore the structure (a *template*) can be reused in similar situations. The presentation layout, instead, may be very different and may be customized through *style-sheets* for creating different visual effects and fruition paradigms (slide show, karaoke, subtitle, advertising, news, menus, and their composition, etc.). Moreover, style-sheets allow adapting models for multi-channel delivering, exploiting just those elements of the template that are supported in the target devices/players (considering size, device capabilities, resource types, etc.).

The creation of templates and style-sheets is almost transparent to the content designer, that may author a SMIL presentation with a commercial or open source tools. From that, the proposed solution provides support to generate from authored SMIL the related template and style-sheet, and their descriptors. Subsequently, advanced content designer may adopt further customizations, editing them separately to improve them and/or to create from them new models (see Figure 1).

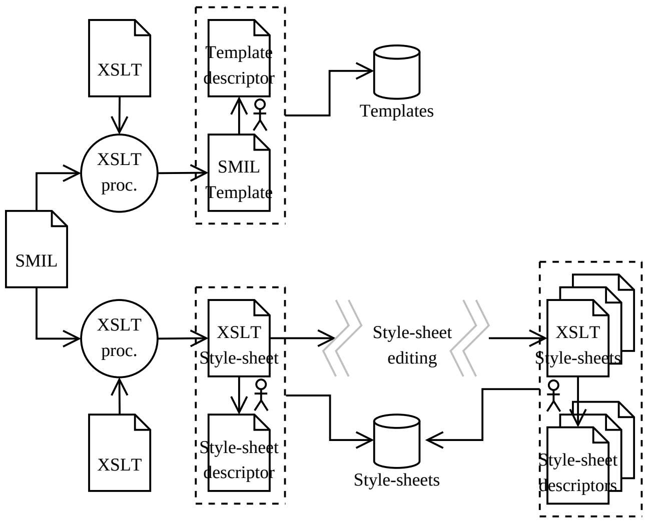

Figure 1: Creation of templates and style-sheets

Please note that the usage of SMIL makes simpler to separate content and layout: templates declare elements that compose the presentation and their dynamic behavior (which mainly is the content of the SMIL `<head>` section); all elements and attributes of the SMIL language involved in defining the appearance (which mainly are those contained in the SMIL `<body>` section) can be defined by an XSLT style-sheet.

Although the system uses models based on SMIL, the output language could be different from SMIL. For instance, MPEG-4 BIFS [11], Laser, SVG, and HTML could be used for delivering the final presentation according to their limitations. The final presentation in some cases can be a set of files or a unique file. In the presented system, they can be embedded in any case into a MPEG-21 compliant model.

2.1 System architecture

In this section, the cross-media content adaptation process is described considering its application for multichannel distribution, where cross-media content is organized as a set of SMIL. SMIL cross-media content allows defining the rendering of the content scene.

According to Figure 2, the adaptation of cross-media content has to take into account:

- semantic-level adaptation with layouting aspects by means of the Formatting Engine;
- signal-level adaptation as single digital resource adaptation by means of the Adaptation Engine supported by some extended Adaptation Decision Taking module [10].

In this solution, the Formatting Engine has the duty of producing the adapted cross-media content layout, reshaping the whole cross-media content (e.g., elimination of some digital resource) if some functional aspects cannot be converted to device supported formats. For example, by converting video in images, animations in videos, passing from a SMIL template to a different layout by using style sheet, or from a HTML to SMIL, or from SMIL to MPEG-4, etc. This process is performed on the basis of:

- Terminal and Network capabilities, User and Natural Environment characteristics [9].
- Distributor Characteristics coming from the distribution channel features and/or intentions such as: device type (e.g., PC, PDA, mobile), output format (e.g., SMIL, HTML, MPEG-4), set of preferred templates/styles to be used for the cross-media content layouting, etc. In a scenario of content sharing among devices this set of information may be recovered from some extended terminal capabilities.
- Formatting Parameters: content layout – e.g., formalized in SMIL and XSLT; criteria for the selection of the most suitable templates and style sheets, additional user preferences such as the minimum font size.
- Resource Descriptors which contain information largely automatically extracted from the resources and/or from the metadata provided for their semantic annotation.

A XML Resource Descriptor may include:

- technical properties extracted from the digital resource: MIME type/subtype, format, size, resolution, codec, duration, etc., (some of these are specific properties for video, audio, text, images, etc.). A part of these data are used only by the adaptation engine (see Figure 3);
- other information related to the semantic aspects of the elements used in the cross-media content such as: the identification of resources into the cross-media content, resource category of usage defined as in the template descriptor (title, header, subtitle, body, footer, menu item, background, logo, header, etc.) (some of these are specific properties for video, audio, text, images, etc.), metadata and other annotations.

The proposed architecture (see Figure 3) is logically constituted of three parts that provide different functionalities: the first stage manages and select templates according to the cross-media content; the second allows the selection of suitable style-sheets; the third provides functionalities for optimizing the layout and adapt consequently the digital resources. All blocks reported in the figure take into account user, device and network profiles; such information are defined according to standards and can be collected statically or dynamically.

2.2 Template selection

Firstly, the Template Selection Engine analyzes all templates into the database and rates them according to their matching with the given set of Resource Descriptors and with the Profile Information. The solution allows mapping resources and resource placeholders present in the template, based on type, category and number of repetitions of each placeholder. The score of the template allows producing an ordered list of Templates (identified by their Template ID) on the basis of their score obtained considering their matching with the Profile Information and the Resource Descriptors. The Template with the highest score is selected for the next phase, while in some case more templates can be selected to produce a set of possible solutions/versions.

Each template in the database of templates has a XML formal Template Descriptor that includes: Template ID (TID), target format (e.g., SMIL, MPEG4, SVG), target devices (e.g., PC, mobile, PDA, etc.), category (e.g., slide show, video clip, interactive music, interactive video, hypertext, audio/video karaoke, electronic book, kiosk, training tutorial, etc.), a list of the resource placeholders used into the template including categories (e.g., title, body, footer, media, graphics, menu item, etc.) and other information (similarly to what is described in the resource descriptor), metadata, a textual description and a mockup example in as a snapshot. Such information can be partially automatically extracted by using a Template Descriptor Extractor, while the rest can be completed by the template author. In the lack of the full set of information the Template Selection Engine can apply only a limited set of criteria to the template.

The Template Selection Engine works on the basis of a set of filtering and some evaluation criteria. The filtering actions allow to select only templates that are suitable for the Distributor Characteristics provided - e.g., device category (e.g., PC, PDA, mobile, STB/TV, HDTV), output category (e.g., SMIL, HTML, MPEG-4, MHP) – for some Terminal and Network capabilities, and/or for some User and Natural Environment characteristics. This allows reducing the number of processed templates that are not suitable for producing acceptable results for the expected device and format. Once performed the filtering, among the criteria that can be used to identify the bet matching we have defined the Draft Structural Matching (DSM), and the Category Matching (CM) which is a more precise structural matching model. The Draft Structural Matching, DSM, allows estimating how a template is similar to a cross-media object

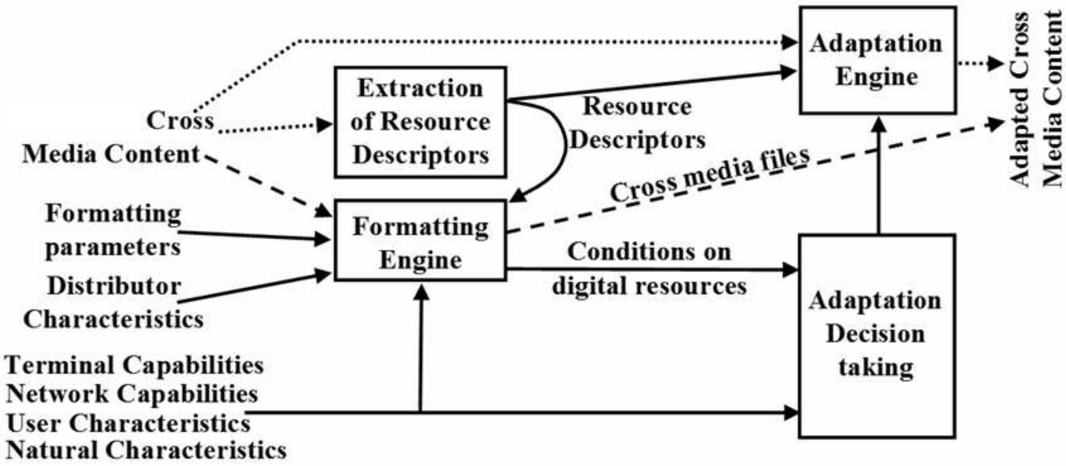

Figure 2: Cross-media content adaptation for multichannel, both semantic-level and signal-level are depicted (dotted lines represent the flow of digital resources; dashed lines are the flows of cross-media files such as SMIL; continuous lines represents the flow of metadata and information used as parameters and/or to control the adaptation process)

scene in terms of content types:

$$DSM(CMoS, TID) = \\ = 1 - \frac{\sum_{t=T_{res}}^{N_{rt}(CMoS)} \frac{|N_{res}(CMoS,t) - N_{res}(TID,t)|}{N_{res}(CMoS,t)}}{N_{rt}(CMoS)}$$

Where: T_{res}, resource type, can be video, audio, images, text, etc.; $CMoS$ is the Cross-Media object Scene; $N_{rt}(c)$ estimates the number of resource types of the Cross-Media Object scene c; $N_{res}(c,t)$ estimates the number of resource of type t of the Cross-Media object Scene c; $N_{res}(tid,t)$ estimate the number of resource of type t of the template tid. In the case of perfect matching DSM is equal to 1.

The Category Matching, CM, allows estimating how a template is similar to a cross-media object scene in terms of content types and its related categories:

$$CM(CMoS, TID) = \\ = 1 - \frac{\sum_{t=T_{res}}^{N_{cat}} \frac{|N_{res}(CMoS,t,cat) - N_{res}(TID,t,cat)|}{N_{res}(CMoS,t,cat)}}{N_{rt}(CMoS)}$$

Where: $N_{res}(c,t,cat)$ estimates the number of resource of type t and of category cat of the Cross-Media Object scene c; $N_{res}(tid,t,cat)$ estimate the number of resource of type t and of category cat of the template tid. In the case of perfect matching CM is equal to 1.

The first criterion is simpler to be estimated while is draft in verifying the structural matching of a template with respect to the cross-media object scene under analysis. It should be noted that, both metrics estimate a value that is maximum when the template match the cross-media object placing all the resources of the cross-media object. This means that the template may have space for some additional resources with respect to those of the cross-media object scenes. This problem can be solved by pruning the template and/or replicating the same resources in some manner. In the later case, for example, producing the buttons of the menu with an image taken from the video, or by reducing an image; producing the back ground image of scene by processing some image with some funny filtering, etc.

2.3 Style-sheet selection

In the second phase, the style-sheets related to the Template ID selected are rated according to their matching with Profile Information. The style with the highest rate can be automatically used for the selection of the best style sheet. In some cases, more styles can be selected to produce different solutions. Almost all data into a style sheet regard dimensions and positions of the elements in the scene, color of the background, features of the text, etc. A style sheet contains defaults values about the size and position of regions in the template. Each style-sheet has a corresponding descriptor, formalized in XML, which includes: Style ID (SID), TIDs of the related templates, target devices (which are a subset of those specified by the related template), metadata, parameters that are intentionally used for the optimization (e.g., size of regions, size of images/video, font size, font style, colors, etc.), regions and their names and relationships. Different style sheets may leave to the optimization phase a different number of parameters. The Style Selection Engine works on the basis of some simple criteria, that are based on the matching with:

- TIDs, list of the possible template IDs related to the style sheet,
- device type (PC, PDA, STB, etc.),
- distributor preferences (number and selection of parameters left open)

Therefore, when the template has a corresponding style matching in terms of TID, the best fitting is based a simple scoring model that estimates the number of matches for the above parameters. On the other hand, when the adaptation of a

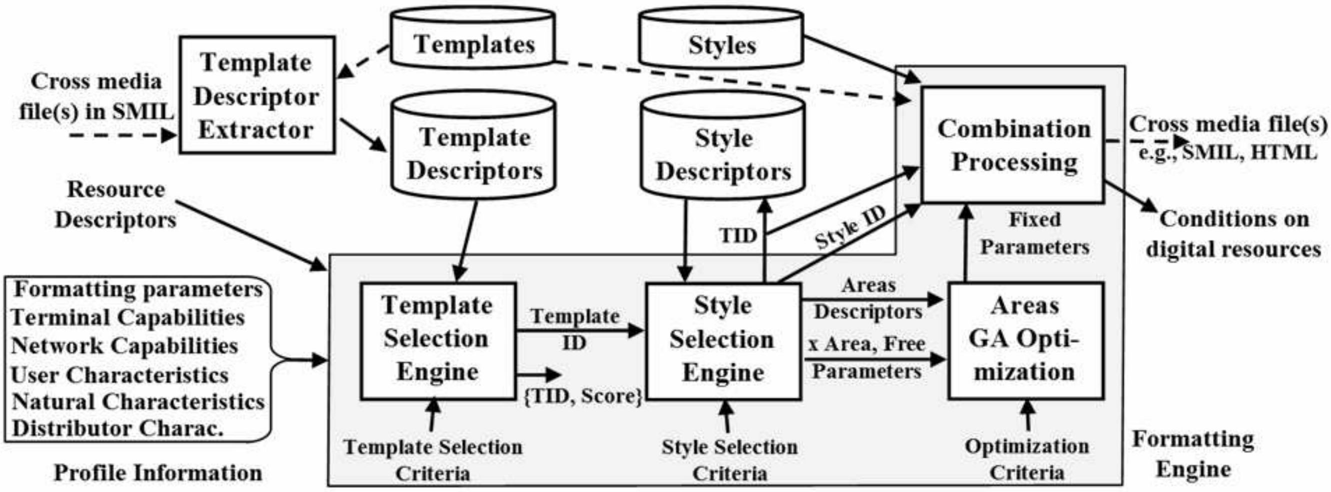

Figure 3: Formatting Engine process and architecture (dashed lines represent the cross-media content files, as SMIL and templates)

layout is requested for an unknown template (i.e., the template has no direct match into the style descriptor database), for example the adaptation of a cross-media from PC to a PDA in which the cross-media object does not contains several versions of the same content for different platforms. In this case, the criteria of selection are based on:

1. number of regions, and/or number of regions with the same name (if the objects are created with conventional names for regions), or

2. a measure of the match between the hierarchy of the regions used in the template and those managed by the style-sheet.

2.4 Layout optimization

The optimization process is used for the automatic definition of a number of style-sheet parameters optimizing the visual rendering layout of specific areas. Since the SMIL structure allows nesting regions (in each scene), a multilevel optimization is performed: after a structure analysis, the system optimizes regions/areas which belong to the same level (e.g., arrangement of the header, body and footer), then recursively can be applied at each sub-level. In fact, some properties of the Style can be left parameterized (e.g., shape, size and position of the resources that have to be integrated into the presentation, selection of the background color, dimensions of the fonts). Their values are set up just when the final XSLT transformation (combining template, style and parameters) is performed. Thus the visual layout for each area of the cross-media content presentation is finalized by using a Genetic Algorithm.

For each area of each cross-media content scene, the Genetic Algorithm produces a huge number of combinations of layout parameters (individuals) following the principles of the evolutionary theory and each of them is evaluated through a functional OF. Thus, the individuals undergo natural selection until a termination condition (computational time limit or convergence) is reached. The functional is expressed by:

$$OF(AreaParams) = \frac{\sum_{i}^{N_{to}} K_i LayoutTerm_i(AreaParams)}{\sum_{i}^{N_{to}} K_i}$$

Where: each $LayoutTerm$ of the functional estimates in a normalized manner a different aspect of the visual layout, weight K_i (ranging from 0 to 1) defines the relevance of the term in the model with respect to the other terms, N_{to} represents the number of non zero weights. When weight K_i is zero its corresponding term is disabled. A set of weight value is an Optimization Criterion. The weights forming a criterion can be manually chosen or can be learnt during a training phase using a multilinear regression analysis, in which some users are asked to give a vote to the presented results. Please note that the Genetic Algorithm optimization can be applied more than one time to identify different independent aspects. For example, size and spacing have to be optimized together while the identification of the best background image could be performed in second phase or together (if the background color is not uniform). Presently, the available $LayoutTerm$ terms are:

- Overlap: 1 minus the percentage of the overlapped surface among the visual regions in the scene. For this parameter, the weight is typically very low or zero, since the overlap is not typically acceptable and thus a strong penalty is assigned to configurations in which an overlap exists;

- Alignment: 1 minus a measure of the alignment among the visual regions in the scene. In some cases, the alignment is mandatory, thus the weight can be very low;

- SpareSpace: 1 minus the percentage of unused space in the scene. In the basic case, the whole area is used, while borders can be set up during the processing to avoid having only tiled images without any borders;

- DistColour: 1 minus the distance from background color/image/pattern and those of the resources overlapped. The distance is defined as an RGB distance or other color spaces between the back-ground and the dominant colors of the images/videos;
- TextFit: measure of the fitness of the text with respect to the space provided (for example defined as the percentage of text that fit in the region assigned changing the font and size, respecting some limits about the font size reduction).

As a result, the Area GA Optimization automatically fixes some parameters defined in the style-sheet, analyzing occurrences of that parameter within the SMIL structure. Parameters are mainly related to position and dimensions of some regions composing the layout and thus the search of good solutions is modeled similarly to a 2D bin-packing problem, in multiple areas [12]. Thus finally the selected template and style and the obtained parameters values are used to produce the final version of the cross-media gluing files (i.e., SMIL) and the conditions to adapt the digital resources to the new cross-media content and presentation.

The Genetic Algorithm optimization process is initialized by using random values for the parameters to be optimized, while the models for the production of the new generations are the traditional crossover and mutation. Crossover is performed by exchanging parameters from a configuration to another (e.g., exchanging the position, dimension or background color of a region with another one, etc.). Mutations are obtained randomly generating the values of the style-sheet parameters.

2.5 Adaptation and exploitation

In the last step, the adaptation block may involve the automatic usage of a large range of content processing tools to perform transformations (e.g., transcoding, resampling, scaling, rotating, mirroring of images, video, documents, audio) necessary to adapt digital resources to generated layout, distribution channel and device capabilities. The resulting SMIL presentation may also be converted in MPEG-4 BIFS [11] or other similar languages.

The presented system has been implemented using C++ language and has been integrated within the AXMEDIS framework [4]. Most relevant features have been wrapped through the Mozilla SpiderMonkey library and are available in JS scripts, thus allowing automation and scheduling of main tasks through the functionalities of the framework. For that reason, the proposed architecture allows setting up either

- an interactive process that assists the content producer, or
- a completely automated production using models suggested by the system.

A simple visual editor, integrated into the framework MPEG-21 editor, allows creating SMIL presentations that can be used to produce templates and style-sheets and to store them with their XML descriptors to be processed by the system.

3. EXAMPLES AND EXPERIMENTAL RESULTS

The solution proposed for cross-media formatting and adaptation has been tested over a large set of resources embedded in AXMEDIS/MPEG-21 objects, exploiting content processing and profiling tools of the framework to create multimedia presentations suitable for multichannel distribution.

(a)

(b)

Figure 4: Example: style-sheets for different devices, PC (a) and PDA (b)

As an example, the process of creating a simple SMIL presentation about the Italian national football team is presented. The presentation, similarly to a web-site, is composed of different "pages" linked by a navigation menu. Each page presents several multimedia elements, with a dynamical behavior that may depend on the user interaction. The structure of each page is composed of "header", "footer" and "center" sections; the latter one is divided into a "nav" part (which includes the menu buttons that allow browsing the whole presentation) and a "body" part. The code used in a template to define this structure is the following:

```
<layout type="text/smil-basic-layout">
    <root-layout/>
```

```
    <region id="header"/>
    <region id="center">
        <region id="nav">
            <region id="menu1"/>
            <region id="menu2"/>
            <region id="menu3"/>
            <region id="menu4"/>
        </region>
        <region id="body"/>
    </region>
    <region id="footer"/>
    <region id="audio"/>
</layout>
```

In opposite to a normal SMIL presentation, the snippet above says nothing about position and dimension of the regions. Such attributes are specified through XSL Transformations defined in a style-sheet:

```
<xsl:param name="screenW" select="800"/>
<xsl:param name="screenH" select="600"/>
<xsl:param name="backgroundC" select="'#FFFFFF'"/>
<xsl:param name="navW" select="150"/>
...
<!--attributes for root-layout-->
<xsl:template match="sm:root-layout">
    <xsl:copy>
        <xsl:attribute name="width">
             <xsl:value-of select="$screenW"/>
         </xsl:attribute>
        <xsl:attribute name="height">
             <xsl:value-of select="$screenH"/>
         </xsl:attribute>
        <xsl:attribute name="backgroundColor">
             <xsl:value-of select="$backgroundC"/>
         </xsl:attribute>
    </xsl:copy>
</xsl:template>
<!--attributes for regions-->
<xsl:template match="sm:region">
    <xsl:choose>
        <xsl:when test="@id='header'">
            <xsl:copy>
                <xsl:attribute name="id">
                  header
                </xsl:attribute>
                <xsl:attribute name="width">
                  $screenW
                </xsl:attribute>
                <xsl:attribute name="height">
                  150
                </xsl:attribute>
                <xsl:attribute name="top">
                  0
                </xsl:attribute>
                <xsl:attribute name="left">
                  0
                </xsl:attribute>
                <xsl:apply-templates/>
            </xsl:copy>
      </xsl:when>
...
```

In the style-sheet above, some attributes are defined through parameters (for which a default value is given in the xsl:param element): this way the layout of the presentation may be customized or optimized on the basis of information that are available only when the contents are demanded by the final user.

Figure 4 illustrates two versions of the "index page" of the presentation: the PC version presents a large image of the Italian football stars while the Italian national hymn is reproduced; the PDA version just offers essentials elements as menu buttons, logo and copyright information. In this example, the same template provides behavioral relationships that allow browsing multimedia contents, while different style-sheets allow customization for different device categories. This requires a new usage paradigm, with a different layout to focus on basic elements of the presentation. According to the new layout, digital resources have been adapted to cope with the reshaped cross-media presentation.

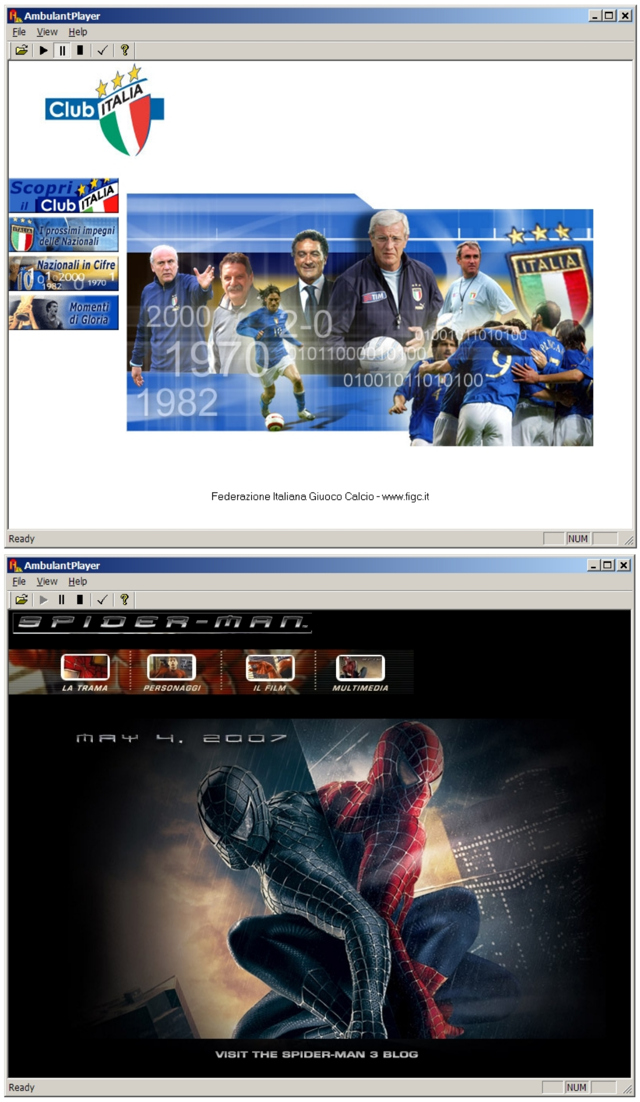

Figure 5: Example: style-sheet optimization for repurposing of a presentation

Figure 5 illustrates two presentations where the same style-sheet has been declined differently to optimize the layout according to the different images/resources provided in input for the presentation. The different proportions of graphics used for buttons and image used for the page body have required a complete reshaping of the layout.

In Figure 6 the structure of the two presentations is made evident: although their dimensions have changed, the three main regions are still arranged in the same order; regions contained in the “center”, instead, have been rearranged, to fit the different proportions of the involved resources.

Figure 6: Example: structure of the two presentations

In the above figure, the dashed lines indicate regions that contain other regions: in these cases, a “multi-level optimization” is needed. The SMIL code of the template is scanned to create a map of regions that belong to different levels; after that, the optimization is performed for the outer levels and then for inner levels. I.e., firstly the three main regions have been arranged in the screen region; then “nav” and “body” regions have been arranged in the “center” region; lastly, menu buttons have been arranged in the “nav” region. Therefore, three optimization processes have been requested to get an optimized layout for the new presentation.

Different GA algorithms and models have been tested changing several of their parameters, and the results have demonstrated the steady state as the best of them. This algorithm is characterized by the fact that a part of the population is maintained in the successive generation. In general, the best results have been obtained with a probability of crossover of the 90%, a probability of mutation of the 80%, maintaining the 40% of the best individuals for the successive generation.

Figure 7 reports the trend of optimization algorithm related to the arrangement of above mentioned regions. Please note that in this case, the convergence has been obtained for the functional OF including as terms: Overlap, Alignments and SpareSpace (white space).

4. CONCLUSIONS AND FUTURE WORKS

This paper has presented an automatic formatting system that aims to reduce costs of authoring multimedia presentations for multichannel cross-media contents, by means of presentation models written in SMIL language and using XSLT style-sheets. The production of cross-media content takes into account adaptation aspects regarding the object usage paradigm, the layout of the visual presentation and the format of digital resources. All those activities are based on the information collected by user, device and network profiles that follow the MPEG-21 DIA standard.

Usage of standard tools as SMIL and XSLT may make easier integrating the proposed architecture in other systems for management and authoring of cross-media contents. The proposed architecture has been described, as well as an implementation using C++ and JavaScript, and has been integrated in the AXMEDIS framework.

5. ACKNOWLEDGEMENTS

The authors would like to thank all AXMEDIS project partners: ANSC, AFI, EUTELSAT, Giunti ILABS, HP Italy, TISCALI, XIM, Univ. of Leeds, Univ. of Reading, etc., the Expert-User-Group and all affiliated members for their contributions, supports and collaborations. The authors would also like to thanks the EC DG INFSO for the partial funding of the AXMEDIS project.

6. REFERENCES

[1] S. Boll and W. Klas. ZYX - a multimedia document model for reuse and adaptation of multimedia content. *IEEE Transactions on Knowledge and Data Engineering*, 13(3):361–382, 2001.

[2] D. C. A. Bulterman and L. Hardman. Structured multimedia authoring. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):89–109, 2005.

[3] AXMEDIS Consortium. Automating production of cross media content for multi-channel distribution. http://www.axmedis.org.

[4] Ivan Bruno et al. Content composition and formatting. Technical Report DE4.3.1, AXMEDIS

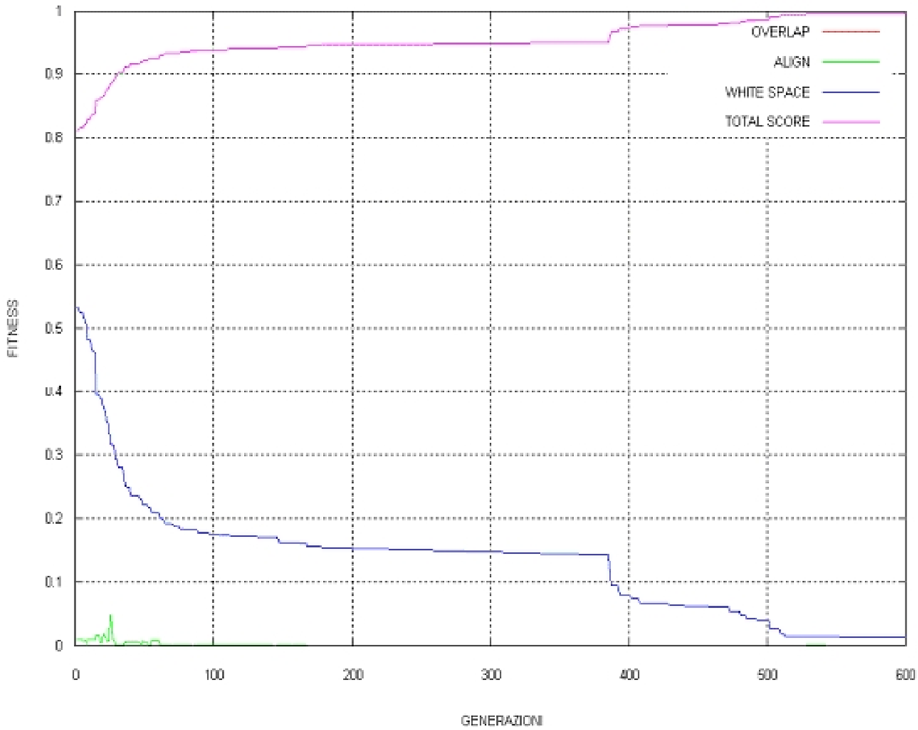

Figure 7: Example of the GA optimization convergence (OVERLAP is zero for all the generations, WHITE SPACE is reporting the term (1 - SpareSpace), ALIGN is depicting the term (1 - Alignment)

Consortium, May 2005.

[5] J. Geigel and A. Loui. Using genetic algorithms for album page layouts. *IEEE MultiMedia*, 10(4):16–27, 2003.

[6] E. D. Goodman, A. Y. Tetelbaum, and V. M. Kureichik. A genetic algorithm approach to compaction, bin packing, and nesting problems. Technical report, Michigan State University, 1994.

[7] M. Jourdan, N. Layaida, and L. Sabry-Ismail. MADEUS: An authoring environment for interactive multimedia documents. In *International Conference on Multimedia Computing and Systems*, pages 644–645, 1997.

[8] R. Mohan, J. R. Smith, and C. S. Li. Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, 1(1):104–114, 1999.

[9] MPEG. Introducing MPEG-21 DIA, digital item adaptation. www.chiariglione.org/mpeg/technologies/mp21-dia/.

[10] D. Mukherjee, E. Delfosse, J. G. Kim, and Y. Wang. Optimal adaptation decision-taking for terminal and network quality of service. *IEEE Transactions on Multimedia*, 7(3):454–462, 2005.

[11] B. Shao, L. Moro Velazquez, N. Scaringella, N. Singh, and M. Mattavelli. SMIL to MPEG-4 BIFS conversion. In *AXMEDIS '06: Proceedings of Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, pages 77–84, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[12] D. Smith. Bin packing with adaptive search. In *Proc 1st Int. conf. on Genetic Algorithms and their application*, pages 202–207, July 1985.