Adaptive Rich Media Presentations via Preference-Based Constrained Optimization

Abstract

Personalization and adaptation of multi-media messages are well known and well studied problems. Ideally, each message should reflect its recipient's interests, device capabilities, and network conditions. Such personalization is more difficult to carry out given a compound multi-media presentation containing multiple spatially and temporally related elements. This paper describes a novel formal, yet practical approach, and an implemented system prototype for authoring and adapting compound multi-media presentations. Our approach builds on recent advances in preference specification and preferences-based constrained optimization techniques.

1 Introduction

Multi-media presentations are messages containing multiple audio/visual elements that must be presented in some particular temporal and spatial relation. Such messages can now be sent to users over both the Internet and mobile networks. As an example, consider an ESPN promo containing two video segments of upcoming sports events, image and video advertisements, as well as sports results in the form of plain text. The author of this message would like the two video segments to be broadcasted one after the other, followed by a short commercial. Image-based ads will be displayed along side the video segments, and the scores will be displayed below.

Such presentations can be described using the standard SMIL format (for Synchronized Multimedia Integration Language `www.w3c.org/AudioVideo`), supported by popular browsers and media players. And they can be either streamed to the target device or downloaded. Our problem begins when we want to customize and personalize a presentation. Message recipients for the ESPN promo have diverse interests and may be using diverse devices that differ in their image quality, screen size, memory, processing power, media playback support, and more. We need to adapt each presentation to a format supported by each particular user's capabilities and to personalize it to suit her taste. Obviously, preparing a special presentation for each potential combination of user profile, target device, and network conditions is infeasible. This paper describes the principles behind a working system prototype implemented for a consortium of companies in the area of streaming multi-media. This system employs a novel customization and adaptation approach that is both flexible and extensible. Moreover, it provides an interesting application of the ideas of preference-based constrained optimization discussed in [Boutilier *et al.*, 2004].

In our approach the author need not consider explicit recipient scenarios. Rather she describes in a natural manner preferences and constraints on the content and form of the message. At presentation time, the author's specification is combined with user device and network properties, and a user profile. These define a preference-based constrained optimization problem whose outcome is a description of the optimal presentation for this particular user and user device. This presentation is generated in SMIL format on-the-fly by the system following the user's request, and can be downloaded or streamed immediately. The method has two parts: an authoring part that enables the presentation author to describe the basic elements of the presentation, as well as her requirements and preferences; and a presentation part that combines this information with information about the user and his device and executes an appropriate optimization algorithm that selects a concrete presentation for this particular case.

Our work contributes both novel ideas to the area of adaptive presentations as well as an interesting example of the use of qualitative preference-based reasoning techniques which have been gaining popularity recently. In this short paper, we concentrate on the general ideas and algorithms behind our system, and in particular, its approach to content personalization. For lack of space, we defer discussion of the spatial and temporal aspects to the full paper. We note that these aspects are mostly dealt with using existing techniques, although their combination with adaptive content selection does raise some interesting issues. Technical details of the implemented system as well as a working prototype are available online. A reference will be provided in the non-anonymous version.

2 Background and Overview

Content adaptation is a well known problem for multimedia presentations. Even for single-media messages, more advanced architectures take into account the need to adapt the particular video/audio/image format to one supported by the end-user's device and may utilize transcoders that can take into account the bandwidth of the user's connection (e.g., see `www.strimm.org`). But ideally, not only the message format should be adapted, but the actual content, too. This is often referred to as *content personalization*, a specific form of adaptation that has received wide attention in the literature (e.g., see [Riecken, 2000]). Personalization and adaptation of compound rich-media is more problematic. The choice of one element may affect that of other elements – e.g., if we have a large video file, then we may have a problem delivering it simultaneously with another media file, such as an audio file. Similarly, if we choose to display one image, then we have less screen space to display another image simultaneously. And if we select particular content for one media component, it may affect the desirable content of other components. Thus, the nature of the end-user's device and his network connection constrain the type of presentations that we can display. Moreover, these properties are known only at message presentation time, not at authoring time. If we combine these constraints with the desire to personalize the message based on a user profile, we are faced with a non-trivial problem.

SMIL [SMIL, 2001] is the most popular format for synchronized presentations.[1] SMIL 1.0 specifies a set of *content modules* that let the author control the content of the presentation based on parameters such as bit-rate, CPU, and language. Control is achieved by allowing the presentation of an element to be conditional on the value of these parameters. SMIL 2.0 has added to these capabilities the ability of the author to specify additional customized attributes beyond the standard attributes. SMIL's conditional primitives provide important flexibility to authors, but they are still limited: the choice of whether to display one element is individual and independent of other choices. This is the core of the problem we try to address. For instance, a device's buffer size or screen size imposes a global constraint on the whole presentation, not on a single element of it.

A number of multimedia authoring systems attempt to address this problem (see [Brusilovsky, 1996] for a survey of adaptive hyper-media). [Boll *et al.*, 1999] describe a system supporting cross-media adaptation, i.e., media elements, or entire multimedia presentation fragments, can be replaced by other fragments of a different type. A rich semantic model is used to identify adequate substitutions, and a strong underlying multimedia database that can address these semantic issues is required. Adaptation consists of filtering semantically inadequate options. Madeus [Jourdan *et al.*, 1998] uses a temporal constraint-based approach to specify allowable media element combinations. The Cuypers system [van Ossenbruggen *et al.*, 2001] uses more sophisticated constraint-programming techniques, as well as higher level semantic specifications. All these systems require a rich semantic model and do not differentiate explicitly between different presentations that satisfy their constraints.

In this paper, we propose a flexible approach that views presentation adaptation as a preference-based constrained optimization problem. Our approach is modular, flexible, and pragmatic, and can be used as a basis for supporting even more complex settings, such as live feeds. It is much more akin to the process of specifying a SMIL presentation than the above systems – in fact, it can be viewed as specifying a flexible SMIL template. The basic idea is for the author to specify a set of possible media elements and a number of possible instantiations for each such element. This defines the space of potential presentations. Now, the author specifies a preference model over this space of possible presentations using a simple language, and can state some hard constraints as well. At presentation time, the author's preference model and constraints are combined with constraints on which the author has no control: the basic capabilities of the device, the network conditions at the time of delivery, and the user's profile (e.g., age, gender, income, past choices). Together, the preferences and the constraints pose a preference-based constrained optimization problem. Its solution is the best feasible presentation (from the author's point of view) for this particular user. Thus, our approach lets the author bias the adaptation process. Moreover, the authoring process is relatively simple, requires no special semantic data,[2] and is easy to master.

Our work presents a sophisticated extension of the work on static adaptive documents in [Brafman *et al.*, 2004; Gudes *et al.*, 2002] that addresses three new issues: (1) The need to handle complex constraints, requiring the introduction of constrained optimization techniques, as opposed to the simple unconstrained optimization used there; (2) A richer specification language; and (3) The ability to handle media elements with diverse durations and temporal constraints which do not arise in the case of static web-pages and documents discussed in the above applications.

Viewing content adaptation as constrained optimization is useful only if we can provide: a simple way for the author to specify her presentation, and an algorithm for performing constrained optimization given such a representation. To specify preferences, we the language of TCP-nets [Brafman and Domshlak, 2002]. This language supports an efficient constrained optimization algorithm.

Finally, we note that our work is on a completely different level of abstraction from work on synchronizing multi-media streams (e.g., a video stream and its matching audio element) such as that of Little and Ghafoor [Little and Ghafoor, 1990]. Our work determines, among other things, which media elements need to be synchronized. Their work provides techniques for actually carrying out such synchronization. In addition, much work has been carried out on the temporal model of presentations (some of it related to the synchronization issue). Our work does not attempt to contribute to that area either; our representation uses the well-known simple temporal constraints on start and end time of media elements.

3 Specifying Presentations

To prepare a presentation, an author first selects the basic presentation elements and their possible respective content options. This defines a set of possible presentations. Next, she defines a preference-order over this space of possible presentations using an appropriate set of preference statements – the preference language. Constraints can be introduced as well (e.g., "no two ads for the same company"), indicating which of the possible presentations are unacceptable. We explain this process in Section 3.1. In section 3.2, we examine more

[1] See [van Ossenbruggen *et al.*, 2003] for a comprehensive analysis of formats for time-based, media-centric presentations.

[2] A semantic model specified using constraints can be integrated into our approach naturally.

closely our preference language and the graphical structure it induces – called called a TCP-net. We illustrate these ideas with the ESPN promo example in Section 3.3.

3.1 Possible Presentations and Preferences

Consider the ESPN promo. It consists of three consecutive parts. Each part consists of a main video segment, two images, and running text. The running text element is constant in all stages. Thus, altogether, we have 10 different media elements. For each element, there are multiple choices. For instance, the first video segment could describe an upcoming broadcast of a football, baseball, or basketball game, and each such content choice may come in different quality levels (e.g., frame-per-second rate) and format. For each ad, we have multiple options too. Thus, the potential set of concrete presentations is large.

To model this, we associate a variable with each content element – let V denote the set of these variables. The set of different options for the content of element v constitutes the variable's domain, denoted $\mathcal{D}(v)$. These options can differ both in their content and their quality. A distinguished null value can denote the choice of not presenting the element at all. The Cartesian product of the variables' domains corresponds to the set of all possible presentation content choices. We use $\mathcal{O}$ to denote the set of all these options, i.e., $\mathcal{O} = \times_{v \in V} \mathcal{D}(v)$. Each element of $\mathcal{O}$ provides a concrete choice of components, but can give rise to *multiple* presentations that differ in the timing and layout of these components.

In addition to the presentation variables, it is desirable to include in V additional variables that denote properties such as: user profile aspects, network parameters, and user device parameters. While we cannot influence their value, they do affect our preference over presentation element choices and participate in related constraints. For instance, personalization can be achieved by conditioning the values of content variables on user properties.

Having specified the set of possible presentation content, the author's next step is to provide information that will bias the choice of which particular presentation the user is actually provided with. Formally, our goal is to specify a preference order over $\mathcal{O}$, the set of possible presentations, based on which we will select the best feasible presentation given each user request. This specification should be based on simple and intuitive statements, so that novice users could design presentations easily and quickly. There are two types of preference information people find natural to express: (1) statements of relative importance of different variables, e.g., "The sports video is more important than the commercial." We take such statements to mean that if we must compromise on the choice of the sports video or the commercial (e.g., because of bandwidth limitations) we prefer to compromise on the choice of commercial; (2) statements of (conditional) preference over values of a variable. For instance, a statement like "For young male users, we prefer the football video over the bowling video," can be used for personalization. The statement "If the video segment is a football game, I prefer the Budweiser ad, and if it is ice-skating, I prefer the Pepsi ad" expresses the fact that the author's preferred ad depends on the choice of video. Thus, when comparing between two *similar* presentations featuring a football game in their main video segment the author prefers the one with the Budweiser ad. Our system takes as input both relative-importance statements and conditional preferences statements.[3]

Finally, the presentation author also specifies a set of constraints. These could be content constraints, such as: "Ads for alcoholic beverages cannot be shown to users under 18." But they can also be temporal and spatial constraints. For example: "The commercial starts immediately following the end of the video"; "The two ads are displayed at the same time"; "Ad1 and Ad2 should have the same size"; "The width of Ad1 should be twice its height"; "Ad1 should be centered above Ad2", etc. Note that preference information is allowed only with respect to content choices, and not with respect to temporal and spatial properties. On the latter we allow only constraints. These constraints indirectly limit the set of content options because, e.g., we may not be able to find appropriate layouts for certain content combinations. As noted earlier, we mostly ignore layout and timing issues in this short version. They are reasoned about using standard techniques such as linear programming.

In general, constraints are specified separately from the preferences, using a standard syntax. To specify temporal and spatial constraints, the author refers to distinguished variables denoting the start and end time of each element, as well as bottom-left and top-right positions. This decoupled approach is convenient because we can add additional device and network constraints later on, at presentation time, without affecting the preference information.

3.2 TCP-Nets

The preference specification language we use consists of (conditional) relative importance statements and (conditional) value preferences. Such statements can be depicted in a graphical manner using a formalism called TCP-nets [Brafman and Domshlak, 2002]. TCP-nets can be used both as an input tool or simply as an internal representation of preference statements provided by the user directly or by means of an appropriate interface. Their graphical structure plays an important role in analyzing the information in such statements and its consistency, and in the constrained optimization process. We use the semantics of TCP-nets, explained below, to interpret the meaning of the author's preference statements.

TCP-nets are an annotated directed graph. The nodes of the graph correspond to the variables of interest (i.e., the elements of V). Each node is annotated with a table describing the author's preference over the different values of the variable associated with this node. Edges describe preferential dependencies and the relative importance of variables.

TCP-nets have three edge types. The first type of (directed) edge captures preferential dependence, i.e., an edge from X to Y implies that the user has different preferences over values of X given different values of Y. The second (directed) edge type captures relative importance relations. Existence of such an edge from X to Y implies that X is more important than Y. The third (undirected) edge type captures conditional importance relations, i.e., importance relations that hold only when certain other variables have particular values. For example, a good choice of ad in the ESPN

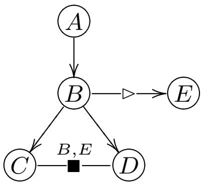
[3]We allow conditional relative importance statements, too.

a	$b \succ \bar{b}$
$\bar{a}$	$\bar{b} \succ b$

b	$c \succ \bar{c}$
$\bar{b}$	$\bar{c} \succ c$

$e \succ \bar{e}$

b	$d \succ \bar{d}$
$\bar{b}$	$\bar{d} \succ d$



be	$C \rhd D$
$\bar{b}e$	$D \rhd C$
$b\bar{e}$	$D \rhd C$

Figure 1: Illustrations for Example 1.

promo is more important than a good choice of video segment only when the user is an affluent male in his 40's.

Each node X in a TCP-net is annotated with a *conditional preference table*. This table contains the author's preference order over $\mathcal{D}(X)$ for every possible value assignment to the parents of X (denoted $Pa(X)$). In addition, each undirected edge is annotated with a *conditional importance table* (CIT). The CIT associated with the edge (X, Y) describes the relative importance of X and Y given the values of the conditioning variables.

Example 1 *In Figure 1 we see a TCP-net over five binary variables A, B, C, D, and E. Standard directed edges in this graph capture preferential dependence; double directed edges capture relative importance relations; undirected edges capture conditional relative importance; $\succ$ denotes preference over variable values; and $\rhd$ denotes variable importance. The graph shows that the preferences over the values of B depend on A's value, and those of C and D depend on B's value. These dependencies follow from the presence of conditional-preference edges from A to B and from B to C and to D. The actual preferences are provided in the associated table. For example, when B is* true*, we also prefer that D will be* true*. Additionally, there is an importance edge connecting B and E. This indicates that the value of B is more important to us than that of E. Finally, there is an undirected edge between C and D. This indicates a conditional importance relation between these variables. Thus, sometimes C is more important than D, and sometimes D is more important than C. The relative importance of C and D is conditioned on the assignment to B and E, and this information is annotated on the edge from C to D. The precise dependence is shown in the associated conditional-importance table. For instance, we see that when B and E are assigned $b\bar{e}$ or $\bar{b}e$, then D is more important than C. When B and E are assigned $\bar{b}\bar{e}$, C is more important than D. Note that although we used binary variables for simplicity, there is no such restriction in the theory.*

A TCP-net specifies a *partial* order over the set of possible variable assignments. This means that not all pairs of assignments are comparable. The statements embodied in a TCP-net are intuitive, but subtle issues in their interpretation require that we clearly define the preference relation induced by the conditional preference tables, the importance relations, and the conditional importance relations. The transitive closure of the union of these preference relations yields the partial order induced by the whole TCP-net.

Conditional preference tables tell us which values of a variable are preferred and under what conditions. This information is interpreted under the *ceteris paribus* semantics as follows: the conditional preference table of variable X specifies the relation between any two complete assignments, o and o', that differ *only* in the value of X. To compare o and o' we examine X's table and check which one of them assigns X a more preferred value. This depends on the value of $Pa(X)$, which must be identical in both o and o'.

For example, according to Figure 1, $\bar{a}\bar{b}\bar{c}de$ is preferred to $\bar{a}\bar{b}cde$ because $\bar{c}$ is preferred to c given $\bar{b}$, and the other attributes have identical values in both outcomes.

Importance relations provide similar information. When X is more important than Y, we can compare any two outcomes o and o' that differ in the value of X and Y *only*. o is better than o' if o assigns X, the more important variable, a better value than o' assigns to Y.

Conditional importance provides similar information but in a more restricted context, i.e., when the selected set has the appropriate value. For example, according to Figure 1, $abcd\bar{e}$ is better than $a\bar{b}cde$ because B is more important than E. Thus, it is better to get a less preferred value of E, as in $abcd\bar{e}$ than a less preferred value of B, as in $a\bar{b}cde$, all else being equal. Similarly, $abc\bar{d}\bar{e}$ is better than $ab\bar{c}d\bar{e}$ because C is more important than D given $b\bar{e}$. Thus, it is more important to get the preferred value for C than for D, all else being equal. On the other hand, we cannot compare $a\bar{b}c\bar{d}\bar{e}$ with $a\bar{b}\bar{c}d\bar{e}$ directly, since we don't have an explicit importance relation between C and D when B and E are assigned $\bar{b}\bar{e}$.

A formal definition of TCP-nets appears in [Brafman and Domshlak, 2002]. Here we note that not all sets of preference statements are representable as TCP-nets, nor are all TCP-nets consistent. We restrict ourselves to the class of *conditionally acyclic* TCP-nets, which are always consistent. This property, which can be verified by the authoring tool.

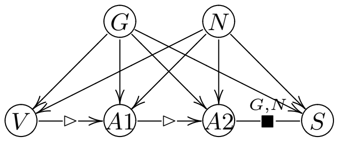
3.3 Defining an ESPN Promo

We now look at how we could model the ESPN promo using a TCP-net. We simplify it by assuming 4 basic elements only: *video, scores, ad1, ad2*. One variable will correspond to each element. In addition to the presentation elements, we have variables denoting: user's gender and nationality. The variable domains are as follows: *Video* has two possible values *football* and *soccer*, and each can be displayed at two quality levels: high and low. The high level requires bandwidth of 56Kbs, and the low level requires 30Kbs. *Ad1* and *ad2* both have the same domain, containing ads for Nike, Adidas, Pepsi, Tuborg, and Budweiser. Each image has two possible formats: JPEG and GIF. GIF files are 4KB each, JPEG files are 40KB. Finally, the *scores* are 20KB each and in SMIL format. Content options include: *sports news, general news, basketball scores, baseball scores, and none*.

Next, we need to specify preferences over the choice of content. We start with preferences over the values of variables. For the video, if the user is an American male, football is preferred, otherwise, soccer. For ad1, the preference is for Nike and Adidas over the drinks, and for ad2 the other way around. The actual ranking depends on whether the user is European or American and on the user's gender (e.g., Tuborg for Europeans, Pepsi for females, etc.). In addition, there

Video (V)	
am	football ≻ soccer
otherwise	soccer ≻ football

Scores (S)	
am	baseball ≻ basketball ≻ sports ≻ news ≻ none
af	news ≻ sports ≻ basketball ≻ baseball ≻ none
em	basketball ≻ sports ≻ news ≻ none ≻ baseball
ef	news ≻ sports ≻ basketball ≻ none ≻ baseball



Ad1 (A1)	
am	Adidas ≻ Nike ≻ Bud ≻ Tuborg ≻ Pepsi
af	Adidas ≻ Nike ≻ Pepsi ≻ Tuborg ≻ Bud
em	Nike ≻ Adidas ≻ Tuborg ≻ Bud ≻ Pepsi
ef	Nike ≻ Adidas ≻ Pepsi ≻ Tuborg ≻ Bud

Scores vs. Ad2	
ef	Scores ⊳ Ad2
otherwise	Ad2 ⊳ Scores

Ad2 (A2)	
am	Bud ≻ Tuborg ≻ Pepsi ≻ Adidas ≻ Nike
af	Pepsi ≻ Tuborg ≻ Bud ≻ Adidas ≻ Nike
em	Tuborg ≻ Bud ≻ Pepsi ≻ Nike ≻ Adidas
ef	Pepsi ≻ Tuborg ≻ Bud ≻ Nike ≻ Adidas

Figure 2: A TCP-Net for ESPN promo. a, e, m, f stand for American European, male, and female, respectively.

is a constraint that states that the two images should not be the same. Regarding the scores, for European males we prefer basketball scores, sports news, general news, no scores, and baseball scores. The preferences are shown in Figure 2. Throughout, we prefer higher quality options to lower quality options and JPEG to GIF, but content is more important than quality. Thus, for American males, a low quality football segment is preferred to a high quality soccer segment. Finally, importance relations must be specified. The video is most important, next is ad1, then ad2, and finally the scores. However, for European females, the scores are more important than ad2. This information is expressed in Figure 2 (with the quality alternatives omitted).

Let's consider a few simple illustrative scenarios demonstrating how these preferences affect the chosen presentation. Consider a European female viewing the presentation on her PC with an ADSL connection. Her PC supports both image formats. This viewer is practically unconstrained, and thus we can supply her with the optimal presentation for a European female: a soccer video together with Nike and Pepsi ads, and news. Suppose that our user now works with a mobile phone. Her bandwidth and buffer size limit the amount of information that can be stored on and transmitted to her device. Suppose that we cannot display high-quality video together with scores. Since the video is more important, the scores will be dropped (i.e., assigned a "don't present" value). Next, suppose that the user is working in off-line mode, and the whole presentation must be downloaded into her buffer. Thus, the sum of the sizes of the components must not exceed the size of the buffer. For instance, if there is no room for two JPEG images, *ad2* will be a GIF image. We explain how these solutions are actually computed in the next section.

As you can see, it is easy to add more complex dependencies. Constraints can be added as well, e.g., disallow ads for competing companies. Moreover, although the preference and importance tables in our example are completely specified, partially specified tables are acceptable, and the author may choose to ignore certain contexts. Our optimization algorithm works with such partially specified tables, although, naturally, with less preference information, fewer pairs of presentations are comparable.

4 Adapting Presentations

Having described the author's presentation preferences, we move to the actual generation of the presentation. This process is initiated by the presentation service provider following a viewing request from a customer. We assume this request contains information about the user and/or the user's device. Such capability-exchange protocols are standard now. At this point, we need to quickly compute an optimal presentation for this user, i.e., the best presentation (according to the author's preference order) among those that meet the constraints imposed by the user's device, network conditions, etc. Because we have a partial order over presentations, we may have a number of such (Pareto) optimal presentations, and any one of them will do. The rest of this section explains how we compute a Pareto optimal presentation.

A naive approach for solving various problems, including constraint satisfaction problems, is Generate & Test. We generate solutions in some systematic manner, and test each solution to see whether it satisfies the constraints. If it does, we can return it as a solution. Generate & Test is inadequate for optimization problems such as ours because we have no reason to believe that the first solution generated is optimal. A conceptually simple, but computationally taxing extension would be to generate all possible solutions to the constraint satisfaction problem, and then compare them. A much better approach, though one that is not always feasible, would be Ordered Generate & Test (OG&T). Here, solutions are generated in a non-increasing manner, i.e., no solution can be better than a solution generated earlier. (Solutions could be incomparable, though). Given such an ordering, the first solution obtained is, indeed, an optimal one; that is, no solution generated in the future will be better.

Fortunately, it is relatively easy to generate a non-increasing sequence of solutions (i.e., elements $o \in \mathcal{O}$) for conditionally acyclic TCP-nets. To generate such a sequence of presentations, we must build a tree whose nodes correspond to partial assignments. In particular, the root node corresponds to an empty assignment, and each leaf node corresponds to a complete assignment (i.e., a complete specification of presentation content, in our case). This is the standard search tree one constructs when solving constraint satisfaction problems (CSPs) in a systematic fashion: all children of a node extend its assignment by assigning one additional, identical variable. Each child corresponds to a distinct value for this additional variable.

Our construction must adhere to the following guidelines: a variable v can be assigned only *after* the following nodes have been assigned (1) all of v's parents in the TCP-net; (2) Any node that conditions a relative importance relation in which v is involved; (3) any variable that is more important than v given the current assignment. The fact that we are dealing with conditionally acyclic TCP-nets ensures that we can

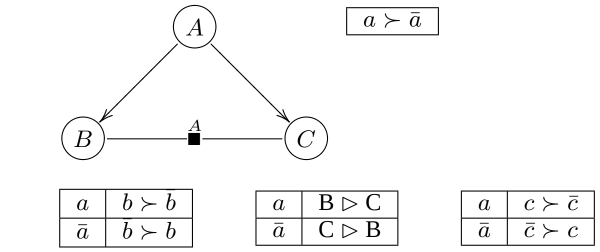


Figure 3: A TCP-Net

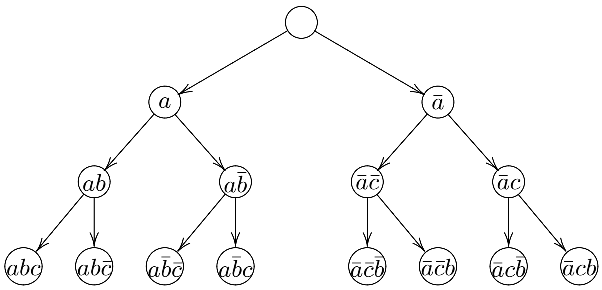


Figure 4: Search Tree for TCP-Net in Figure 3

satisfy these conditions. In addition, the children of each node must be ordered from left to right according to the preference ordering over the values of the newly assigned variable, as specified by its preference table. Because all the variables conditioning the preference for the newly assigned variable have been assigned earlier, this is well defined. The resulting tree has the property that the leaf nodes, ordered from left to right, constitute a non-increasing sequence of assignments. It is important to note that variable ordering can differ from one branch to another, as long as the above constraints are obeyed.

As an example, consider the TCP-net in Figure 3. Its corresponding search tree is shown in Figure 4. Because B is more important than C when A is *true*, we see that B is ordered before C. When A is *false*, C is ordered before B because now C is more important. We can also see that variable values are ordered based on the preference tables.

To use outcome ordering to implement OG&T, we perform depth-first search in order to incrementally generate the tree above. We test each leaf node to see whether it satisfies the presentation constraints. The first presentation generated that satisfies these constraints is optimal, because none of the following presentations is better.

OG&T is satisfactory for handling modest problems with a few thousand possible presentations. Our current system, uses a more advanced method that is semantically equivalent to OG&T but uses more clever pruning techniques that are described in the full paper.

5 Summary

We presented an approach for specifying adaptive synchronized rich-media documents and an algorithm for adapting and personalizing these presentations given each concrete user download request. Our algorithm combines the initial flexible presentation specified off-line with the information available online about the user and her device, and handles this as a preference-based constrained optimization problem. This problem is solved by utilizing the special properties of TCP-nets and their relation to CSP algorithms.

Our work provides a novel method for adapting the content and (some aspects of) the form of multimedia presentation, as well as a novel way of utilizing recent advances in preference specification and handling in AI. A system based on these techniques was implemented for a consortium of companies in the area of streaming technology, and users can interact with our presentation engine online.

References

[Boll *et al.*, 1999] S. Boll, W. Klas, and J. Wandel. A cross-media adaptation strategy for multimedia presentations. In *ACM Multimedia (1)*, pages 37–46, 1999.

[Boutilier *et al.*, 2004] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Preference-based constrained optimization with cp-nets. *Comp. Intelligence*, 20(2):137–157, 2004.

[Brafman and Domshlak, 2002] R. Brafman and C. Domshlak. Introducing Variable Importance Tradeoffs into CP-Nets. In *Proc. UAI'02*, 2002.

[Brafman *et al.*, 2004] R. Brafman, C. Domshlak, and E. S. Shimony. Qualitative decision making in adaptive presentation of structured information. *ACM Transaction on Information Systems*, 22(4):503–539, 2004.

[Brusilovsky, 1996] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2–3):87–129, 1996.

[Gudes *et al.*, 2002] E. Gudes, C. Domshlak, and N. Orlov. Remote conferencing with multimedia objects. In *2nd Int. Workshop on Multimedia Data Document Engineering*, volume 2490 of *LNCS*, pages 526–536, 2002.

[Jourdan *et al.*, 1998] M. Jourdan, N. Layaida, C. Roisin, L. Sabry-Ismail, and L. Tardif. Madeus – an authoring environment for interactive multimedia documents. In *ACM Multimedia Conference*, pages 267–272, 1998.

[Little and Ghafoor, 1990] T. D. C. Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communications*, 8(3):413–427, 1990.

[Riecken, 2000] D. Riecken, editor. *Special Issue on Personalization*, volume 43 of *Communication of the ACM*, 2000.

[SMIL, 2001] Synchronized multimedia integration language. Technical report, W3C, 2001. TR/2001/REC-smil20-20010807. URL: www.w3c.org/TR/smil20/.

[van Ossenbruggen *et al.*, 2001] J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, and L. Hardman. Towards second and third generation web-based multimedia. In *10th Int. WWW Conf.*, pages 479–488, 2001.

[van Ossenbruggen *et al.*, 2003] J. van Ossenbruggen, J. Geurts, L. Hardman, and L. Rutledge. Towards a formatting vocabulary for time-based hypermedia. In *12th Int. WWWW Conf.*, pages 384–393, 2003.