Generating Presentation Constraints from Rhetorical Structure

Lloyd Rutledge, Brian Bailey, Jacco van Ossenbruggen, Lynda Hardman and Joost Geurts*

CWI (Centrum voor Wiskunde en Informatica)
P.O. Box 94079
NL-1090 GB Amsterdam, The Netherlands
Tel: +31 20 592 41 27
E-mail: Firstname{.van}.Lastname@cwi.nl

*Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
Tel: +1 612 624-8372
E-mail: bailey@cs.umn.edu

ABSTRACT

Hypermedia structured in terms of the higher-level intent of its author can be adapted to a wider variety of final presentations. Many multimedia systems encode such high-level intent as constraints on either time, spatial layout or navigation. Once specified, these constraints are translated into specific presentations whose timelines, screen displays and navigational structure satisfy these constraints. This ensures that the desired spatial, temporal and navigation properties are maintained no matter how the presentation is adapted to varying circumstances.

Rhetorical structure defines author intent at a still higher level. Authoring at this level requires that rhetorics can be translated to final presentations that properly reflect them. This paper explores how rhetorical structure can be translated into constraints, which are then translated into final presentations. This enables authoring in terms of rhetorics and provides the assurance that the rhetorics will remain properly conveyed in all presentation adaptation.

KEYWORDS: Rhetorics, constraints, presentation generation, meta-structure, authoring

INTRODUCTION

Historically, the human authoring process was mostly expressed directly in terms of the final presentation. Hand-copied and printed books acted as both data storage and presentation. Hypertext and hypermedia systems are emerging that provide authoring at levels of abstraction higher than final presentation. Authors write specifications at these higher levels, and lower-level presentations are automatically generated that meet these specifications. Examples include the use of CSS and XSL style sheets for presenting HTML and XML documents.

The many types of presentation-independent authoring abstractions include presentation constraints and rhetorical structure. Constraints define the spatial, temporal and navigation limits and relationships that a presentation must have. Rhetorical structure represents the conceptual flow of a document — how the user is led to understand it. This paper argues that rhetorics are a higher level of abstraction than constraints, and can be used to derive constraints, which are then used to derive the presentation.

Presentation constraints are typically expressed in terms of the timeline, screen layout, or navigation structure. In most constraint systems, only certain aspects of the presentation are adapted to satisfy each constraint. Hypermedia presentation structure can be said to consist of multiple *dimensions*, including primarily *space, time* and *navigation*. Current constraint systems typically satisfy constraints of one dimension by generating presentation structure only of that dimension. That is, they will translate spatial constraints into spatial structure, temporal constraints into temporal structure, and navigation constraints into navigation structure. These systems do not provide inter-dimensional constraints — that is, constraints that apply simultaneously to multiple dimensions of presentation structure.

Rhetorics are another means of describing a presentation at a higher level of abstraction [14]. Rhetorics represent the conceptual flow along which a presentation progresses. Rhetorics have been used for analyzing existing documents, and as an instructive aid for writing new documents [13][15].

This paper describes how a presentation's rhetorical structure can be used to derive the presentation's constraints. Once the constraints are generated, the presentation itself can be automatically derived from these constraints. This means the presentation itself can be encoded in terms of rhetorical structure rather than in terms of less abstract representations such as the presentation's constraints or of the final presentation itself.

This paper first presents background information on automatic presentation generation. Next, it discusses constraints, how they are processed, and what their relationship is with a presentation's structure. Then rhetorical structure is described, as is the means of generating presentation constraints from it. Finally, an implementation of rhetorics and constraints processing for hypermedia presentation generation is presented.

PRESENTATION GENERATION

Abstraction of document presentation, such as that provided by constraints and by rhetorics, provides several benefits. One is that it removes much redundant encoding that the author needs to perform. Another is that it makes the authoring product adaptable to a wider variety of presentation circumstances. Such abstraction also has the effect of changing the authoring process, and the nature of what is being written. It enables some parts of the human authoring process to become automated. Other parts remain non-computable, but become performed more efficiently by humans in the context of what has become automatically processed. Finally, such abstraction introduces new aspects of the human authoring process that previously did not exist. Much of the code that was previously written by humans is now generated by machine. And what human authors encode themselves is of a different nature than before.

Of course, presentations generated by machines will often not have the aesthetics of presentations made by human authors. However, generated presentations are still useful. Some may simply be a quick means of providing selective access to massive amounts of data, such as the pages generated by Web search engines. These need primarily to be understandable, and not necessarily aesthetically appealing. Another use for such a generated presentation is as a first draft, or rough sketch, for a human author. The generated presentation can contain the basic information in a basic presentation structure, and then a human author can edit it into a more pleasing form. Here, the computer makes the job of the human author easier, not unnecessary.

Several prevalent examples of generating useful, but not necessarily pretty, presentations exist currently on the Web. The presentation of HTML is adapted differently by different browsers and different user preferences, and often by different user CSS style sheets. However, the variation in adapted presentation that CSS and browser distinctions provide is small. For example, the specific content, the order of its presentation and its navigational structure are fixed.

More variation is provided with HTML presentations that are generated dynamically. Many commercial sites generate presentations in response to filled-in HTML forms, but these often simply repeat the information provided in the form in a particular structure. News sites often generate presentations from fixed sources based on varying user profiles. These are typically generated from large document information stores and programs that pull components from them for inclusion in specific HTML presentations. But while the content of each generated page varies, it consists of different combinations of fixed components selected for the user and positioned in fixed templates.

The most prevalent examples of generated HTML pages come from Web search engines. Search engines routinely process presentation-independent data into widely varying final presentations. However, although the specific content and the order of presentation can vary greatly with search engines, the structure of presentation is simple and unvarying. Also, the information conveyed is quite simple in structure — typically a list of Web pages with short excerpts.

The current status of presentation generation can be improved upon by increasing the varying complexity of the structure of the presentation and of the information conveyed. The benefits of this include increased adaptation to the user’s preferences, language, perceptive abilities and previous knowledge. Constraints and rhetorics are two types of structure that enable wide adaptation. Constraints allow for more complex presentation structure that still adapts to varying presentation circumstances. Rhetorics represent the effect of the information conveyed, and thus understanding their processing provides an important key for increasing its complexity.

A consortium of researchers has developed a model for the automatic generation of hypermedia presentations called the Standard Reference Model for Intelligent Multimedia Presentation Systems (SRM-IMMPSs) [4]. The SRM-IMMPSs describes a general framework for producing final presentations from presentation-independent specifications. It is intended as a model for comparative discussion rather than for implementation, making its application to multimedia presentation generation similar to the Dexter model’s application to hypertext [7].

PRESENTATION STRUCTURE AND CONSTRAINTS

This section provides a definition of presentation structure and discusses how constraints on it can be specified and processed. It introduces new types of constraints and constraint resolution that are needed to help the translation of rhetorics to constraints.

Presentation Structure

This paper treats presentation structure as consisting of multiple dimensions, including primarily *space*, *time* and *links*, as shown in Figure 1. Spatial structure defines the layout of visual media items on the screen. Temporal structure is the document timeline which represents when media items are played (temporal issues at runtime, such as those related to user interaction and network delays are beyond the scope of this paper, see [8]). The linking structure is how the user can navigate through the presentation.

Each dimension of hypermedia presentation structure has, in turn, its own sub-structure. Space has two dimensions representing the layout of the screen display. Time has one dimension, along which the playing of the media items is scheduled. The linking structure of a document can be represented as a directed graph, in which each node represents a state of the presentation, and each edge represents a user interaction that moves the presentation from one state to the next. These static states are represented by combinations of space and time structure. They represent how the presentation will progress in the absence of user interaction.

When the structure of the whole presentation is considered, it consists of combinations of these dimensions, also shown in Figure 1. Space and time can be combined into a three-dimensional structure representing how a scene changes during the progression of a passive presentation. Placing media items along these three dimensions thus represents how a presentation would progress without user interaction. In a simple, scene-based hypermedia presentation, space/time structures can be represented as nodes in a link graph. Certain areas in the space-time of each scene represent areas of the screen that can be clicked on during certain periods of time to activate the traversal to another scene.

Constraint Processing

The details of presentation structure are often rendered from *constraints*. That is, some constraints on what presentation structure is allowed are set, and then a presentation structure that matches these constraints is found. PREVISE is a system that resolves spatial constraints through modifications to spatial structure [26]. An example of the use of spatial constraints is shown in Figure 2. An example of spatial constraint code is shown in Figure 3. The Madeus system processes both temporal and spatial constraints, but each type is resolved only with modifications to the presentation structure of the corresponding type [12]. Desired linking structure has been represented with hypertext metrics [5]. The EUCLID project explores the use of spatial constraints in hypertext to maintain desired patterns of user interaction, a conceptual equivalent of linking constraints [23].

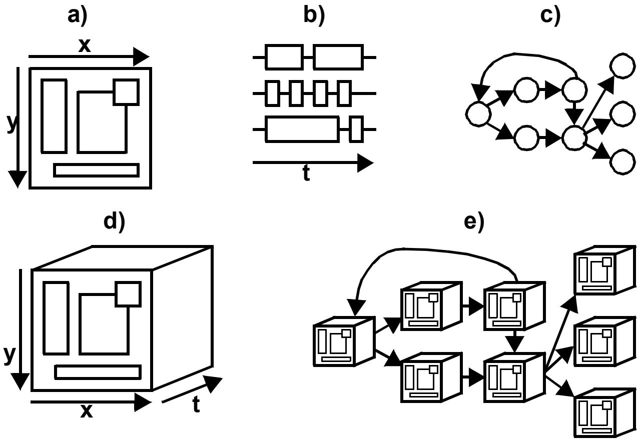


Figure 1: The Hypermedia Presentation Structure of a) Space, b) Time, c) Links, d) Space and Time Combined and e) Space, Time and Links Combined

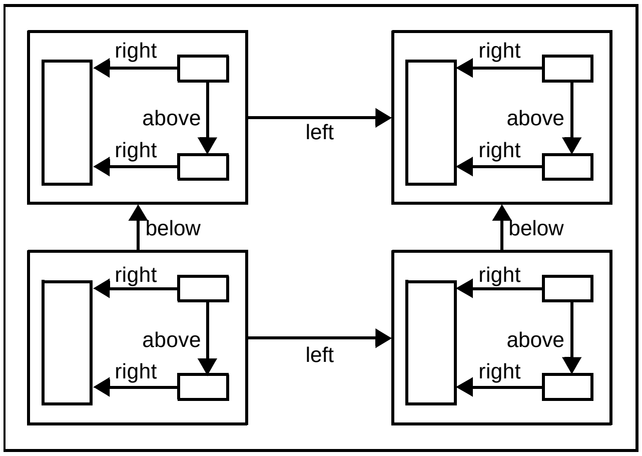


Figure 2: Example of Spatial Constraints

(Ensure that ImageA remains a rectangle)

```
ImageA.UL.X = ImageA.LL.X
ImageA.UR.X = ImageA.LR.X
ImageA.UL.Y = ImageA.UR.Y
ImageA.LL.Y = ImageA.LR.Y
```

(Ensure that ImageA maintains its width/height)

```
ImageA.UR.X - ImageA.UL.X = ImageA.Width
ImageA.LR.Y - ImageA.UR.Y = ImageA.Height
```

(Ensure that ImageA is left of ImageB)

```
ImageA.UR.X < ImageB.UL.X
ImageA.UR.Y = ImageB.UR.Y
```

(ImageA is being constrained to be left of ImageB, and A is constrained to remain rectangles and maintain their dimensions.)

Figure 3: Examples of Spatial Constraint Java Code

In addition to specifying direct relationships between pairs of components, compositional structure has also been used to specify constraints. The W3C multimedia standard SMIL has several types of temporal constraints, including sequential and parallel composites and synchronization attributes [6][9].

Overflow and Compensation

With most current hypermedia constraint solvers, a constraint on one dimension of presentation structure is typically matched by considering variations of the structure only along that same dimension. This section introduces techniques with the constraints on one dimension of hypermedia presentation structure can be resolved by affecting not just that single dimension of the final presentation, but affecting constructs from the other dimensions as well. This process starts by considering the presentation structure as a combination, or intersection, of all the dimensions, and then seeking to meet the constraints set by considering variations in the overall structure instead of in just a single dimension.

To illustrate overflow and compensation, a running example was developed that emulates an electronic program guide (EPG) for movies broadcast on television. Specifically, this work enables the user to query a simulated multimedia database containing program guide information, and in response to the query, automatically generate a hypermedia presentation. An EPG screen display is shown in Figure 4.

To define spatial relations between objects, a set of specific-to-general rules are defined. The rules are in the format:

```
spatial_relation(MediaA, ConceptA, MediaB, ConceptB, Rhetoric,
Position).
```

For example, a spatial relation stating that the image media object of the movie concept is to be placed underneath the title media object of the movie concept could be defined as follows (using Prolog syntax, with '_' as a wildcard symbol):

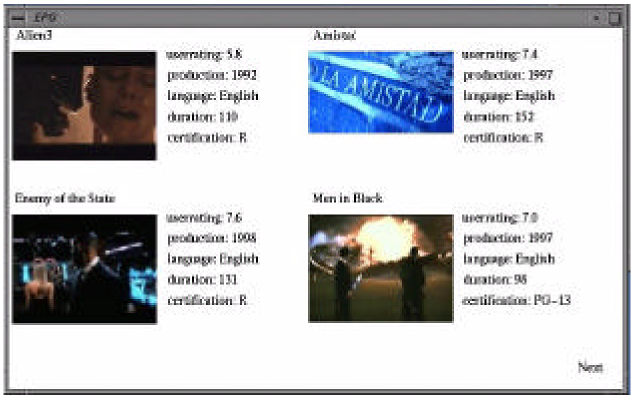
```
spatial_relation(image, movie, title, movie, _, below).
```

However, the same relationship could also be defined on a more generic level by relating the media objects and not the concepts:

```
spatial_relation(image, _, title, _, _, below).
```

This states that images are to be placed underneath titles regardless of which concept they belong to.

Figure 5 shows some code from this EPG that specifies overflow and compensation. A sequence of items is to be displayed in the hypermedia presentation. This means that each item is placed within the spatial, temporal and navigational structure of the presentation.



Alien3
userrating: 5.8
production: 1992
language: English
duration: 110
certification: R

Amistad
userrating: 7.4
production: 1997
language: English
duration: 152
certification: R

Enemy of the State
userrating: 7.6
production: 1998
language: English
duration: 131
certification: R

Men in Black
userrating: 7.0
production: 1997
language: English
duration: 98
certification: PG-13

Figure 4: An EPG Screen Display

```
% sequence(+Concepts, -Scene)
% Apply a sequential ordering to the given concepts and
% return the result in Scene

sequence([Concept|Concepts],
[SpatialConcept|RestofScene]) :-
        spatial_layout(Concept, SpatialConcept),
        sequence(Concepts, RestofScene).

% spatial failed, try temporal
sequence([Concept|Concepts],
[SpatialConcept|RestofScene]) :-
        temporal_layout(Concept, SpatialConcept),
        sequence(Concepts, RestofScene).

% temporal failed, try linking
sequence([Concept|Concepts],
[LinkConcept|RestofScene]) :-
        link_layout(Concept, LinkConcept),
        sequence(Concepts, RestofScene).
```

Figure 5: Prolog Code for Generating Spatial, Temporal and Navigational Structure from Sequences

The items are first to be positioned horizontally across the page. Whenever this fails due to the horizontal page boundary being crossed, a new row is allocated and the initial sequence goal is recursively called with the remaining items. Whenever the allocation of a new row fails due to the vertical page boundary being crossed, a new page is created and either temporally sequenced or linked from the previous page. The initial sequence goal is once again recursively called with the remaining items.

Overflow occurs when constraints cannot be meet using only one type of presentation structure, and thus another type of presentation structure must be used as well. In the above example, the items that remain after a single screen display is full are overflow resulting from using only spatial structure to present all the items.

Compensation is using other presentation structure types when the needs of the presentation cannot be met by using only one. Compensation is triggered by overflow. In this example, temporal presentation structure is used if the spatial structure alone is insufficient. That is, if all the items cannot be placed in a single screen display, then they are put in multiple screen displays that are shown one after the other.

A time limit can also be placed on the duration of the full presentation. Such a constraint is useful, for example, in broadcast presentations with which schedules must be adhered to. If the number of screen displays is so large that the resulting presentation would exceed this time limit, then the combination of spatial and temporal structure is not enough to satisfy the constraints put on the presentation. This is another case of overflow, and again compensation is used. This time, the navigational structure is extended as compensation. Links are added to guide the user to particular screen displays instead of having the user passively see all of them. These navigational links would exist as a next, and possibly a previous, button, a menubar, or both.

Intra-dimensional Constraints vs. Inter-dimensional Constraints

Considering compensation results in the distinction between two types of constraints: intra-dimensional constraints and inter-dimensional constraints. *Intra-dimensional constraints* involve only one type of presentation structure. Each intra-dimensional constraint defines a relationship among media objects within a single dimension of space, time, or links without crossing over or including other dimensions. For example, a spatial constraint between two media objects may not also include a temporal constraint within the same specification. However, a single media object itself may separately be involved in both a spatial and temporal constraint. Examples of intra-dimensional constraints include "left of", which involves only spatial structure, and "afterwards", which involves only temporal structure.

The set of intra-dimensional constraints for each type of presentation structure is solved independently of the others.

For example, the solution to the set of spatial constraints will have no impact on the solution to the set of temporal constraints, and vice versa. Intra-dimensional constraints do not result directly in compensation. The inability to meet them, however, can cause overflow, which may in turn trigger compensation.

Inter-dimensional constraints define how one type of presentation structure can be altered to allow the meeting of constraints involving another structure type. Inter-dimensional constraints do result directly in compensation. In the current example, an inter-dimensional constraint is used to specify that a group of items is to be played on multiple subsequent screen displays if they cannot all fit in one. The overflow caused by the inability to meet intra-dimensional constraints typically triggers the use of inter-dimensional constraints. Compensation strategies are composed of inter-dimensional constraints and their relationships with intra-dimensional constraints.

Single-pass vs. Order-changing Compensation

The compensation strategy in the current example is simple in that it is linear and one-directional. First only spatial structure is used. When that fails, the temporal structure is used. And when that fails, navigational structure is used. Furthermore, the ordering of the placement of the items in the spatial-temporal structure is fixed. That is, once the first screen display is filled, the remaining items are never candidates for placement in that screen. The list of items is only passed through once and its order is not changed.

One example of a more complex compensation strategy is illustrated in Figure 6. Here, overflow of the spatial layout still results in temporal structure being used. However, the placement of items in the different screen displays is changed to have items of the same media type displayed together. Images are displayed on one screen, and text objects on another. The ordering of the items can be changed to make the presentation look and work better. The input list of items would be passed through multiple times and its order effectively changed for its translation to the final presentation structure. This order-changing compensation strategy is not appropriate when the input order of the item list is significant to the user.

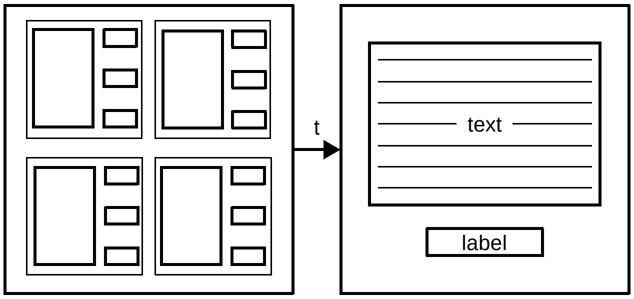


Figure 6: Grouping of Similar Items in Same Temporal

Compensation with Linear vs. Tree-shaped Navigation

Temporal progressions are by nature linear. The navigation structures presented above for handling spatial overflow, that of next and previous buttons and menubars, are also linear. Another alternative when using link structure to handle overflow is to provide tree-shaped instead of linear navigation. Tree-shaped navigation has been long established in hypertext as more efficient than linear. The primary example of tree-shaped navigation is the use of a table of contents, with which sections are selected, and then subsections within them can be selected.

Figure 4 provides a high-level navigational view of the movies selected. The user can click on one movie in this screen display to access another screen display showing more information on that movie. If few enough movies are to be shown, such a detailed screen display may be able to show all movies at once. If overflow happens on this screen display because too many movies are to be shown, a linear navigational compensation could be used to keep the same screen display with next buttons or a menubar. Alternatively, compensation with tree-shaped navigation could be used to generate the display in Figure 4 with links to more detailed screen displays.

RHETORICAL STRUCTURE AND CONSTRAINTS

Presentation constraints were just described as the specification of hypermedia at a level of abstraction higher than that of presentation. There are many other, higher types of hypermedia abstraction as well. They can be used to derive constraints, which in turn can be processed into final presentations. Rhetorical structure is one such higher-level abstraction. This section discusses the generation of constraints from rhetorical structure.

This mapping of rhetorics to constructs will be guided in part by compensation strategies. The compensation strategies introduced earlier in this paper make the mapping of rhetorics to constraints more flexible. They allow a presentation to adapt by altering structure in multiple presentation dimensions and still convey the same rhetorics.

The processing of rhetorical structure into constraints and its conceptual context within this paper are shown in Figure 7. As shown, rhetorical structure can be used in combination

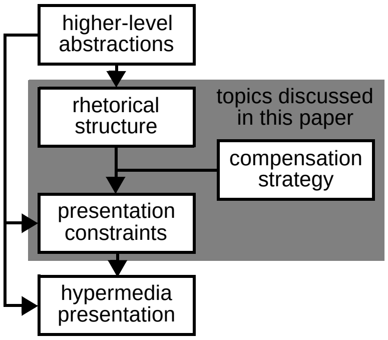


Figure 7: Rhetorical Structure and Constraint Processing

with other higher-level abstractions. This paper already discussed how presentation constraints are processed into final hypermedia presentations. It also discussed how compensation strategies guide this process. This section focuses on how rhetorical structure, as one abstraction above constraints, can be processed into constraints.

Background on Rhetorics and Other Presentation Abstractions

The levels of presentation abstraction that are higher than constraints represent the bases on which constraints are automatically determined. They are equivalent to the considerations human authors may make when defining constraints for the presentations they desire, except that they are modeled in a way that can be automatically processed, either independently of or in conjunction with human authoring. Defining a presentation in terms of higher levels of abstraction enables more varied adaptation. It also gives adaptation a better chance of staying consistent with the author's wishes — that is, it helps the inevitable degradation that comes with adaptation be more graceful.

Rhetorics specify the conceptual structure of how information is conveyed. Figure 8 contains one established list of rhetorical relations that has been developed for text analysis [14]. It remains to be shown how well these rhetorical structures apply to hypermedia as well as text. This paper does not address this issue, but focuses instead on how rhetorics in general can be used to generate presentations.

Rhetorics, like any presentation-independent abstraction, do not necessarily have a direct correlation with the spatial layouts, timelines, navigational links or media content of the presentations they describe. There can be many different mappings between rhetorics and hypermedia presentation, resulting in many different presentations for the same source rhetorical structure.

The Textnet system provides link types that are often rhetorical structures [22]. This conveys to the browsing user what the rhetorical structure of the information is. In contrast, this paper focuses not on what the structure is but on how varied presentations can be generated from it.

Some research has studied generating multimedia explanations from rhetorical structure [2][15]. This research focuses on explanation as both a particular type of rhetorical structure and a particular presentation structure. The final generated result of these projects' rhetoric process was a detailed presentation specification, not an adaptive abstraction such as constraints.

Nucleus-satellite Relations

Evidence	Circumstance	Restatement
Concession	Background	Antithesis
Elaboration	Volitional Cause	Solutionhood
Motivation	Non-volitional Cause	Enablement
Condition	Volitional Result	Purpose
Evaluation	Non-volitional Result	Interpretation
Justify	Otherwise	Summary

Multi-nuclear Relations

Sequence	Constrast	Joint

Figure 8: Some Established Rhetoric Relations [14]

The EUCLID project explores the generation of spatial constraints from the structure of reasoned discourse [23]. Reasoned discourse could be considered a type of rhetorical structure. Given this, EUCLID describes constraints from rhetorics with a focus on spatial constraints and on the rhetorics of reasoned discourse.

One type of presentation-independent representation used as input for rendering hypermedia presentations is relational grammars. Relational grammars define domain-specific conceptual relationships between document objects. They have been used as input for deriving spatial constraints of final presentations [25].

The Fiets hypermedia application also explores the mapping from abstract hypermedia representation to presentation structure [18]. Fiets' representation of the city of Amsterdam and its history is represented in terms of time, as years, space, as location, and with various types of relationships. This sense of Amsterdam's time, space and links is contrasted with the time, space and links in the structure of presentations about Amsterdam. For example, the temporal structure of Amsterdam, its history, will not necessarily map directly to the temporal structure of a presentation, its timeline. Fiets demonstrates how the structure of a document can be independent of the structure of its presentation. This property allows multiple presentations that vary widely to be generated from the same document. A screen display of Fiets is shown in Figure 9.

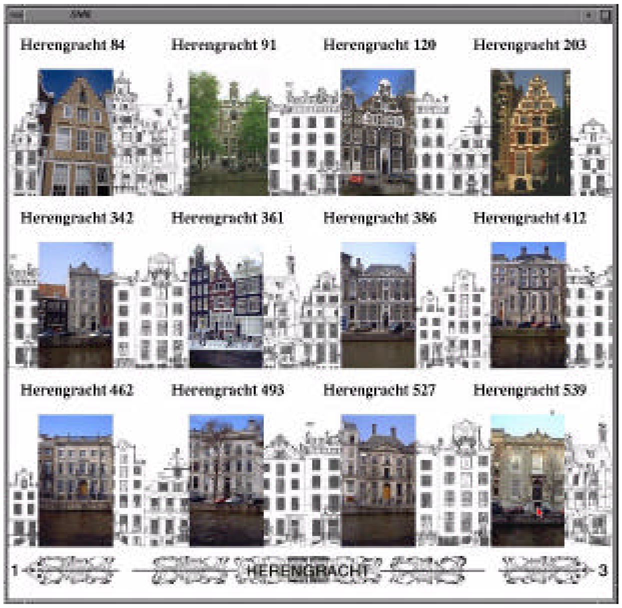


Figure 9: A Fiets Screen Display

Sequence

Sequence is a rhetorical structure consisting of an ordered list [14]. It is arguably the most important and most frequently used type of rhetorical structure. Typical hypertext examples include narrative structures as discussed in [24] and navigation structures such as guided tours and Walden's Paths [21]. Most documents consist of sequences of described concepts, each of which is decomposed into smaller sequences or other rhetorical structures.

The sequence rhetorical structure is processed in the EPG application. Its use is illustrated in Figure 5. The abstract concept of a sequence relation does not explicitly specify how it must be realized in terms of presentation format. The goals encoded in Figure 5 define how the rhetorical construct of a sequence of items can be presented in terms of spatial, temporal or navigational structure. One impact that the specification of a rhetoric sequence would have on the resulting constraints is that single-pass compensation would be used, not order-changing compensation. This is because the order of items in the sequence must be maintained in the final presentation.

The sequence rhetorical structure is used in the Fiets application. The higher-level abstractions such as year, address and relation translate directly into sequences. A rhetorical sequence can state that a collection of buildings is to be presented in order of time or, alternatively, in order of street address. Once such a sequence is established, it can be mapped to the presentation structure with a focus on either the spatial, temporal or navigational structure of the final presentation. The Fiets presentation in order of address shown with spatial structure is shown in Figure 9. This would result in, in order, the display of multiple buildings in left-right and top-down order on one screen, a temporal sequence of screen displays that each show one building, or the display of one building at a time with a navigational menu for selecting another building to display.

Strict and Loose Sequences

One variation of rhetorical structure processed by Fiets is that a sequence can be either strict or loose. A *strict* sequence indicated that the items must be presented to the user in the order given, with none skipped. A *loose* sequence, on the other hand, indicates that it is only necessary to indicate to the user that the order of the sequence is important. With a loose sequence, the user has the option to access the items in any order, but what the ordering is will be conveyed.

For example, one Fiets compensation strategy starts with spatial structure and then overflows to navigational structure, showing one page of items at a time, and providing links to other pages. A strict sequence rhetorical structure would be translated into pages that have only a "next" button for accessing other pages. With this, the user can only access the pages in one order. A loose sequence would be translated into pages with a menu bar along the bottom. The menubar has buttons that each indicate and provide access to another page. These buttons are displayed in order, conveying the order of pages to the user, but the user can select any button.

Nucleus-satellite Relations

Multi-nuclear rhetorical relations, such as sequence, are lists of equally significant components. Thus, they translate well to list-like presentation structures, as described above. A *nucleus-satellite* relation, on the other hand, associates a piece of knowledge — the *nucleus* — with a related, though less important, piece of knowledge — the *satellite*. These relations are conveyed best with non-list presentation structures, including adjacency spatial relations and directed binary navigational links. A list of multi-nuclear and nucleus-satellite relations is shown in Figure 8.

Adjacency spatial relations convey nucleus-satellite relations well by keeping the satellite concepts visually associated with its nucleus. For example, a piece of text that has a summary relation with an object may best be presented below that object in the final presentation. Translating summary relations as "below" spatial relations will help ensure that this association is visually conveyed to the user in a consistent manner.

Binary navigational links that start with a nucleus object can provide the user with quick access to one or more of its satellites. The information in such a satellite would be considered not essential enough to be shown with the nucleus object in the initial screen display, but significant enough to be readily accessible. This decision may be made in the context of a spatial layout overflow. When all the information cannot be displayed on one screen display, a compensation strategy using tree-shaped navigation would be used, providing links to the less important information. The satellites of certain relations would be assigned in the rhetoric-to-constraint mapping as conveying such less important information.

In the EPG application, each movie may be associated with text making up its summary description by using a summary rhetorical relation. Clicking on a movie image on an EPG screen display like that shown on Figure 4 would bring up a display showing that movie's summary.

IMPLEMENTATION IN BERLAGE

The Berlage environment is a system for authoring and generating hypermedia presentations. It is built in terms of the SRM-IMMPSs [19]. Berlage was developed to demonstrate the generation of presentations from hypermedia meta-structure. With Mix'n'Match, the encoding of how presentations are generated by Berlage was broken up into exchangeable modules [17]. More information about these earlier implementations of the SRM-IMMPSs in Berlage can be obtained from the earlier publications about them[17][20]. This section describes the components of Berlage and of the SRM-IMMPSs that relate to the translation of rhetorics into constraints.

The primary SRM-IMMPSs components related to constraints and rhetorics are the *design expert*, the *design* *layer* and the *realization layer*. The design expert stores and provides access to specifications for how various aspects of design, including how rhetorics are translated into constraints, are mapped. The design layer processes design specifications, including those for rhetorics, into constraints. The realization layer processes these constraints into specific mono-medium formats and multimedia formats.

Application Expert

The application expert in the SRM-IMMPSs supplies data relevant to a particular media format or a particular conceptual domain. The EPG relies on a class-instance hierarchy to define a small ontology relevant to the EPG domain, as shown in Figure 10. This ontology is extensible, making the addition of new concepts, attributes, and relations trivial, and the mechanism for searching and retrieving their values already defined. For each class in the ontology, a set of class attributes (in the form of name-value pairs) can be attached. Example class attributes are *width*, *height*, *border*, and *rhetorical relation* to another type. By default, each type inherits the attributes defined for its parent, but the value of an inherited attribute can be overridden by re-defining the same attribute for itself.

Each relational attribute of the movie database is an instance of a class listed in Figure 10. Each instance may also have a set of instance attributes attached that override the class attributes. When an instance attribute value is requested, the value of the first matching attribute starting from the current instance and moving up its class hierarchy is retrieved.

Control Layer

In the SRM-IMMPSs, a goal for the presentation is presented to the control layer. The control layer then processes these goals into sub-goals and communicates them to the content layer. In the Fiets application, there is no focussed goal entered into the control layer. The stored document and the initial settings of the experts, which define the mapping from storage to presentation, are used to provide a default presentation to the user. The EPG application, on the other hand, has as its input a goal. This goal consists of a query entered by the user specifying what programs should be listed in the presentation.

Content Layer

The content layer is responsible for selecting the appropriate media content that meets the information needs of the specified presentation goal. The selection process is non-trivial and must rely on extensive domain knowledge in order for the correct content to be selected. This domain knowledge is provided in part by the application expert. When the content is originally stored in the application expert, semantic information must be attached describing the content items and the relationships between them. Ultimately, the goal is to define semantic information germane to the current domain, but general enough so that the knowledge base does not become a collection of pre-defined presentations.

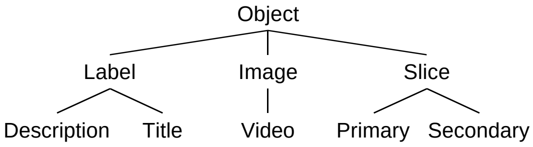


Figure 10: An Ontology for the EPG Domain

In the present system, the content layer relies on the simple concept of a slice, which is adopted from the RMM methodology [11], for semantic information. A slice represents a subset of the attributes of a single relation. Two slices are defined for each table: the primary and secondary slice. When the content layer receives the result of the goal query from the control layer, the set of slices represented by the attributes is retrieved. The query result is then augmented with the additional attribute members of each slice. This result is then passed on to the design layer.

Design Expert

The design expert in the SRM-IMMPSs stores and provides access to different design specifications. Here, a design specification corresponds with the term “stylesheet” used in XML processing. A design could be made by a human designer who is not the document author. Ideally, one design can apply to multiple types of documents, and each document can be rendered with multiple designs.

For example, a multimedia presentation on a particular topic could be generated with a “BBC look” design. Alternatively, the same topic and information could be presented with an “MTV look.” Both would be presentations of the same document, with the same topic and information, and perhaps the same media content. The “BBC look” and “MTV look” would correspond with different design specifications provided by the design expert.

In Berlage, specifying the translation of rhetorics to constraints is an aspect of design. This mapping is stored in and provided by the design expert. A rhetoric design expert has been added to Berlage as a component of the design expert. This allows rhetorical structure processing to be specified separately from other aspects of design, such as font type and visual object color. It is left for future work to explore the ramifications of this distinction.

Design Layer

The design layer in the SRM-IMMPSs is responsible for media generation and overall layout of the media content. It makes the conceptual content fit a given design for how its presentation should look and feel. A general description of how the presentation should progress, specified at least in part by a rhetorical structure, is provided by the content layer. The design is provided by the design expert.

One aspect of design is how rhetorics are translated to constraints. This translation process, described earlier in this paper, occurs in the design layer. The design layer’s output is a set of constraints passed to the realization layer for processing into the final presentation.

The constraints passed from the design to the realization layer currently include both intra-dimensional and inter-dimensional constraints. It has not been fully determined whether inter-dimensional constraints are final presentation specification and not design or higher-level presentation abstraction. Future versions of Berlage may thus process inter-dimensional constraints in the design layer and pass only intra-dimensional constraints to the realization layer.

Realization Layer

The realization layer in the SRM-IMMPSs is responsible for creating concrete media objects as well as resolving their spatial, temporal, and linking constraints. The realization layer takes format-independent presentation details from the design layer and generates the presentation in a particular format. The design layer communicates the media objects along with their constraints to the realization layer through a design plan. The goal of the design plan is to be comprehensive enough to capture all of the necessary constraints within the EPG domain, yet be flexible enough to accommodate future requirements. The design plan is represented as an XML document.

To realize the design plan, the realization layer first parses the XML document instance and creates an in-memory representation of the document tree. The layer then separates the different types of relationships such as spatial, temporal, and link relationships. Once separated, both the temporal and spatial constraints are transformed into numerical constraints and subsequently solved using the constraint solver. Links are further processed for quick retrieval during output generation. The solutions from the solver are then passed to the SMIL generator, which maps the elements, attributes, and relationships from the design plan to corresponding SMIL constructs. This entire process is depicted in Figure 11.

IBM's Java *alphaWorks* XML package [10] is used to parse the XML document and manipulate the document tree through its DOM implementation. The *Cassowary* hierarchical constraint solver was developed as an improvement in performance to the DeltaBlue system [3]. The set of specified constraints are then solved using a modified Simplex algorithm. The realization layer transforms each spatial and temporal constraint into one or more linear equalities and/or inequalities and then passes them to Cassowary. The SMIL presentations generated were played using the *GRiNS* SMIL player [16].

SUMMARY

This paper introduces inter-dimensional constraint compensation, which extends this processing with constraints that apply to spatial, temporal and navigational layout simultaneously. It also discusses mappings for translating rhetorical structure into presentation constraints that help ensure that the author's wishes in terms of rhetorics will be meet. The Fiets and EPG applications demonstrated the effectiveness of compensation and rhetoric-to-constraint processing in a small scale for focussed domains. They also provided the first steps for increasing the complexity of compensation and rhetoric-to-constraint translation for hypermedia. Extensions were made to the SRM-IMMPSs-based Berlage hypermedia presentation generation environment to exercise the ideas discussed in this paper, both in terms of process modeling and final implementability.

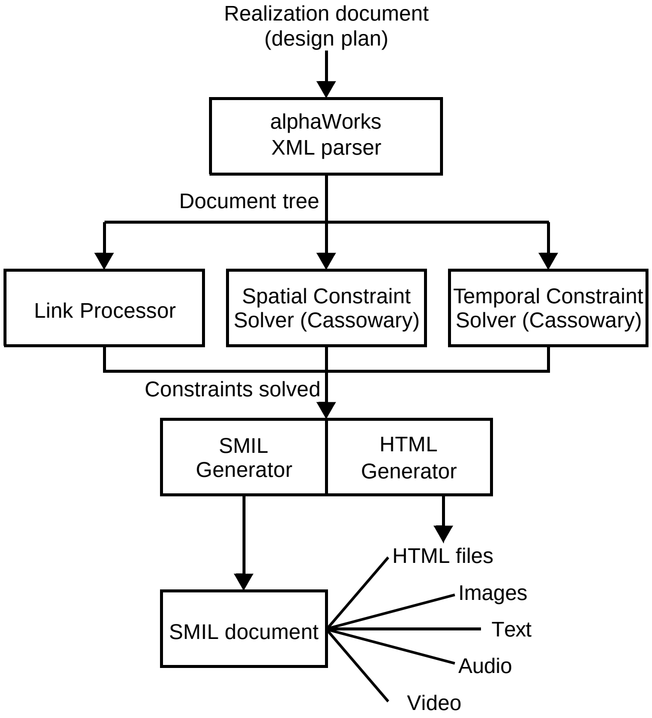


Figure 11: Processing Model of the Realization Layer

ACKNOWLEDGEMENTS

Brian Bailey developed the EPG application during a three-month visit with the CWI. This research was funded in part by the Multimedia Information Analysis (MIA) project. Many images in Fiets come from the Amsterdam Heritage Website [1].

REFERENCES

1. City of Amsterdam Municipal Department for Preservation and Restoration of Historic Buildings and Sites. Amsterdam Heritage, http://www.amsterdam.nl/bmz/adam/adam_e.html.

2. André, E and Rist, T. "The Design of Illustrated Documents as a Planning Task", in *Intelligent Multimedia Interfaces* (ed. Maybury, M.T.), AAAI Press / The MIT Press, 1993, pp. 94-116.

3. Badros, G.J. and Borning, A. *Cassowary: A Constraint Solving Toolkit*, http://www.cs.washington.edu/research/constraints/cassowary/.

4. Bordegoni, M., Faconti, G., Feiner S., Maybury, M.T., Rist, T., Ruggieri, S., Trahanias, P. and Wilson, M. A. "Standard Reference Model for Intelligent Multimedia

Presentation Systems." *Computer Standards and Interfaces*, 18(6,7) (December 1997), pp. 477-496.

5. Botafogo, R.A., Rivlin, E. and Shneiderman, B, "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics", *ACM Transactions on Information Systems*, Vol. 10, No. 2, April 1992, pp. 142-180.

6. Bulterman, D.C.A, Hardman, L., Jansen, J. Mullender, K.S. and Rutledge, L. "GRiNS: A GRaphical INterface for Creating and Playing SMIL Documents", *Computer Networks and ISDN systems*, 10, (1998), pp. 519-529.

7. Halasz, F., and Schwartz, M. "The Dexter Hypertext Reference Model". *Communications of the ACM*. Vol. 37, No. 2, February 1994, pp. 30-39.

8. Hardman, L., Van Ossenbruggen, J., Mullender K.S., Rutledge, L., and Bulterman, D.C.A. "Do You Have the Time? Composition and Linking in Time-based Hypermedia", *Proceedings of ACM Hypertext 99*, Darmstadt, Germany, February 1999, pp. 189-196.

9. Hoschka, P. (ed.). *Synchronized Multimedia Integration Language*, World Wide Web Consortium Recommendation. June 1998.

10. International Business Machines Corporation, *IBM alphaWorks*, http://www.alphaworks.ibm.com.

11. Isakowitz, T., Stohr, E.A., and Balasubramanian, P. "RMM: A Methodology for Structured Hypermedia Design". *Communications of the ACM*, 38(8), August 1995, pp. 34-44.

12. Jourdan, M., Layaïda, N., Roisin, C., Sabry-Ismail, L. and Tardif, L. "Madeus, an Authoring Environment for Interactive Multimedia Documents", *Proceedings of ACM Multimedia 98*, Bristol, England, September, 1998, pp. 267-272.

13. Landow, G.P. "Relationally Encoded Links and the Rhetoric of Hypertext", *Proceedings of Hypertext 87*, Chapel Hill, NC, Nov 1987, pp. 331-343.

14. Mann, W.C., Mattheissen, C.M.I.M and Thompson, S.A. "Rhetorical Structure Theory and Text Analysis", *Information Sciences Institute Research Report*, ISI/RR-89-242, November 1989.

15. Maybury, M.T. "Planning Multimedia Explanations Using Communicative Acts", in *Intelligent Multimedia Interfaces* (ed. Maybury, M.T.), AAAI Press / The MIT Press, 1993, pp. 60-74.

16. Oratrix Development BV., *GRiNS*, http://www.oratrix.com/GRiNS/.

17. Rutledge, L. Hardman, L., Van Ossenbruggen, J. and Bulterman, D.C.A. "Mix'n'Match: Exchangeable Modules of Hypermedia Style", *Proceedings of ACM Hypertext 99*, Darmstadt, Germany, February 1999, pp. 179-188.

18. Rutledge, L. Hardman, L., Van Ossenbruggen, J. and Bulterman, D.C.A. "Structural Distinctions Between Hypermedia Storage and Presentation", *Proceedings of ACM Multimedia 98*, Bristol, England, September, 1998, pp. 145-150.

19. Rutledge, L., Van Ossenbruggen, J., Hardman, L. and Bulterman, D.C.A. "Practical Application of Existing Hypermedia Standards and Tools", *Proceedings of Digital Libraries 98*, Pittsburgh, USA, June 1998, pp.

20. Rutledge, L., Hardman, L., Van Ossenbruggen, J. and Bulterman, D.C.A. "Implementing Adaptability in the Standard Reference Model for Intelligent Multimedia Presentation Systems", *Proceedings of Multimedia Modeling 98,* October 1998, pp. 191-199.

21. Shipman, F.M., Furuta, R., Brenner, D., Chung Chi Chung and Hao-wei Hsieh. "Using paths in the classroom: experiences and adaptations", *Proceedings of the ninth ACM conference on Hypertext and hypermedia,* June 20 - 24, 1998, Pittsburgh, PA USA, pp. 267 - 270

22. Trigg, R. A *Network-Based Approach to Text Handling for the Online Scientific Community*, Ph.D. Thesis, University of Maryland, November 1983, University of Maryland Technical Report, TR-1346.

23. Smolensky, P., Bell, B., Fox, B., King, R. and Lewis, C. "Constraint-Based Hypertext for Argumentation", *Proceedings of Hypertext 87*, Chapel Hill, North Carolina, USA, November 1987, pp. 215-245.

24. Walker J., "Piecing together and tearing apart: finding the story in afternoon", *Proceedings of ACM Hypertext 99*, Darmstadt, Germany, February 1999, pp. 111-117.

25. Weitzman, L., and Wittenberg, K. "Automatic Presentation of Multimedia Documents Using Relational Grammars", *Proceedings of ACM Multimedia 94*, San Francisco, USA, September 1994, pp. 443--451.

26. Zhou, M.X. "Visual Planning: A Practical Approach to Automated Presentation Design" *Proceedings of the International Joint Conference on Artificial Intelligence 99 (IJCAI 99)*, pp. 634-641.