Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

# The role of high-level and low-level features in semi-automated retrieval and generation of multimedia presentations

F.-M. Nack, M.A. Windhouwer, E.J. Pauwels, M.W.J.H. Huijberts, H.L.Hardman

Information Systems (INS)

**INS-R0108 June 30, 2001**

Report INS-R0108
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB  Amsterdam
The Netherlands

# The Role of High-level and Low-level Features in Semi-automated Retrieval and Generation of Multimedia Presentations

Frank Nack[1], Menzo Windhouwer[2], Eric Pauwels[3], Michèle Huijberts[1], Lynda Hardman[1]
Centrum voor Wiskunde en Informatica (CWI) Amsterdam
*Firstname.Lastname@cwi.nl*

*CWI*
*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

ABSTRACT
In this article we argue that for the automatic generation of adaptive multimedia presentations we are in need of expandable, adaptable style descriptions which provide both high-level conceptual and low-level feature extraction information. Only the combination of both facilitates the retrieval of adequate material and its user-centred presentation. We discuss the requirements for an adaptable Web-based environment for museums presenting visual artefacts. We then present the framework of our prototype multimedia generation environment which transforms a high-level user query into a concrete multimedia final-form encoding that is playable on an end-users' platform. We describe the underlying architecture and provide a working example.

*1998 ACM Computing Classification System:* H.5.2,I.3.4,I.3.8
*Keywords and Phrases:* Multimedia semantics, feature grammars, multimedia presentation generation, multimedia retrieval
*Note:* The work is carried out under the "TOKEN2000" and "Dynamo" projects

## 1. Introduction

Over the last decade , museums have been contributing to the information revolution by digitising their collections and providing access to them for the general public. Currently, many sophisticated and visually pleasing environments, such as exhibitions in virtual galleries, art collections and presentations of different cultural artefacts, are available. The problem with nearly all of these environments is that they are hand-crafted. In nearly all cases these environments lack adaptive [6] or adaptable qualities [25], which could otherwise facilitate the adjustment of the multimedia presentation to the specific context of an individual user. For overcoming these problems, various attempts to explore and develop innovative presentation techniques are described by [1, 2, 5, 17, 25, 29]. These approaches facilitate the synthesis of multimedia documents and plan how this material is presented to various users.

The complex semantic systems in visual media which form the basis for a subjective interpretation by each viewer [3, 11, 13, 14, 23], and their combinatorial potential, make it difficult to follow established approaches. Thus, we claim that an approach is required that uses both high-level conceptual representations, such as provided in queries and presentational goals, and low-level features, such as automatic feature extraction, to enhance the generated presentation.

That is the methodology followed in the Cuypers project [29]. Cuypers provides a Web-based, database-driven hypermedia interface to information made available by the Rijksmuseum in Amsterdam. The system provides access to the collection of paintings and sculptures for diverse user groups. The goal is to facilitate user context-specific adaptation by guiding the automated presentation generation not only with a user model but also with intrinsic information about the

[1] INS2
[2] INS1
[3] PNA4

material to be presented. The system allows us to provide the user with the required information and also to improve the presentation on the basis of the conceptual structure of the material.

In this article we first present the requirements for an adaptable Web-based environment for museums presenting visual artefacts. We then present the framework of our prototype multimedia generation environment which transforms a high-level user query into a concrete multimedia final-form encoding that is playable on an end-users' platform. We describe the underlying architecture and provide a working example.

## 2. The Art environment

The aim of museums is to exhibit, preserve and support the study of historical, artistic or scientific artefacts. As far as the exhibition is concerned, usually the space is limited and thus mere fractions of the collection can be presented at a time. Hence, most museums use the Web for presenting large parts of their collections to the general public. On the whole the interfaces are visually pleasing but the common way of providing information is a static combination of text and different sorts of visual data, where the text tries to convey the main information. Most museums also perceive themselves as educational institutions - offering the visitor some initial information on their collections and thus enabling them to consult more specialized sources.

The Rijksmuseum (http://www.rijksmuseum.nl/) in Amsterdam provides information on 1,200 of its top exhibits, in the form of texts, photos, video and animation. Links enable objects to be connected from different departments of the museum's vast collection, for example joining a painting of a 19th-century landscape with its 17th-century forerunner. Access is via four categories: artists' names, themes, encyclopaedic terms and a systematic catalogue.

The current system depends on the designer's vision of how the different media items should be presented so that a typical user can understand the relationship between the different units and the ideas they represent, as well as the differing meanings that can be attributed to the objects within the media units. The challenge for the designer of such a media-based information environment is to foresee the circumstances and presuppositions of the user at the time of accessing the information. This is in particular hard since visual material is composed of various codes and sub-codes that are understand by every user in a choice of interpretational channels [3, 11, 13, 14, 23].

Given the unpredictable nature of such situations it is infeasible to create all relevant media documents in advance. However, in order to combine various materials to establish conceptual models that support a better understanding of art, a system must possess knowledge of what is contained in the different media used. Furthermore, strategies are needed for combining the materials on the basis of context so that the system can find satisfactory solutions for upcoming questions (e.g. based on the content of an image), misunderstandings (rearrangement of the material) or non-understanding (creation of a new sequence).

From the point of view of visual media this is in fact the problem. Even though an image might provide a limited amount of visual information, it still provides a continuum of meaning. This functionality is based on the two formal structures within visuals, the content (realised through the image) and its spatial and temporal relationship with other media items (as realised through montage in a presentation). Since the relationship between the two representational systems is complex, it is useful to quickly examine them separately to determine their relevance to the generation of meaningful presentations.

### *2.1 The image - an access problem*

If we look at an image as an abstract element, then we can experience it optically (objectively, realistically) and mentally (subjectively). On the optical level we try to identify as many objects as we can in the available time of perception. Each object is mentally transformed into a sign. A sign usually consists of two distinguishable components: the signifier (which carries the meaning) and the signified (which is the concept or idea signified) [23]. The relation between the signifier and the signified is arbitrary, which enables the creation of higher order sign systems and their diversity. Figure 1 summarises the compositional and the interpretational structures that enable a perceiver to understand an image.

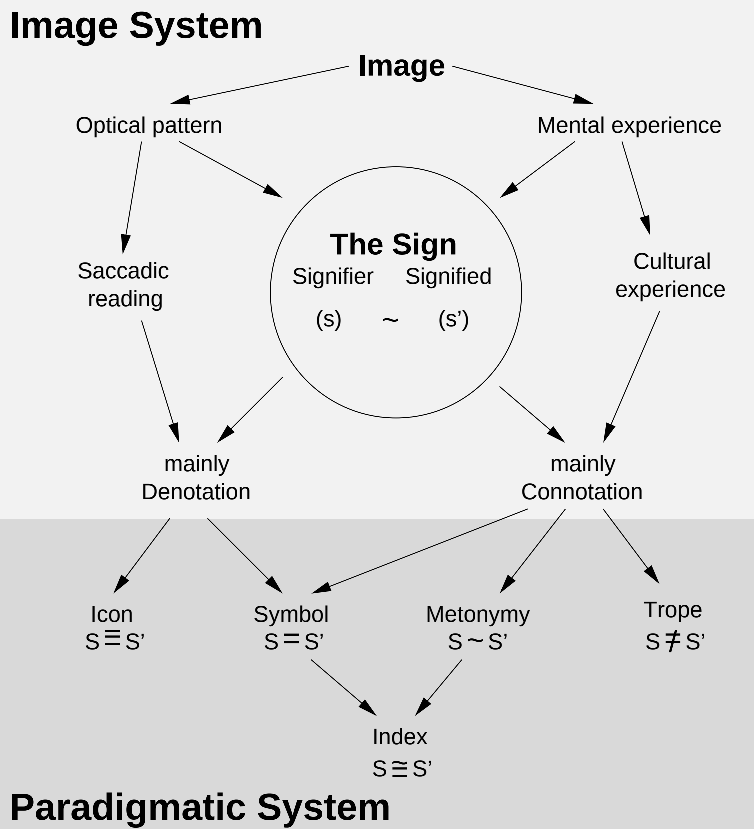


Figure 1: The compositional and interpretational structures that make up the image (based on [20])

It should be stressed here that the diversity of the semantic system described above provides, with its combinatorial possibilities, the foundation for a subjective interpretation by each viewer. Consider Figure 2, which shows Rembrandt's "Philemon and Baucis". In this image Rembrandt made use of the technique *clair-obscur* or *chiaroscuro*. Both terms mean 'light-dark' and are used to describe strong contrast of light and dark shading in paintings, drawings and prints. Rembrandt uses light typically in two ways, either as religious symbol or as means to portray it as the center and the range of a narrow world. Light used as a religious symbol has typically its source outside the portrayed scene and the objects within are merely reflections of it. The observer is thus led to understand that the world is dependent on the existence of an unknown and invisible power [3].

While a single image is in the position to trigger multiple interpretational levels, the use of numerous visuals simultaneously in one presentation is even more problematic. It is in particular the fact that humans tend to search for cues as support for the creation of meaning. Important are for example associative cues which facilitate the combination of indicators. Imagine, for example, that we would like to present a number of paintings in *clair-obscur* style. Cues to guide the user into the understanding of the style are, for example, a title stating the main purpose of the presented images. However, we could also present the images in the style they are drawn in. For this, we have to judge the order of the paintings depending on the way the dark and light parts are distributed in each painting for coming up with a reasonable arrangement of images. In other words, we use space as an associative cue. In fact, the conceptualisation of space is, therefore, an elementary principle of the analysis and organisation of material in the presentation of multimedia presentations for dynamic digital environments. In addition, aspects of the style in the underlying content can also be carried over into the temporal dimension, e.g. in the *clair-obscur* example by fading each image completely to black before displaying the following image.

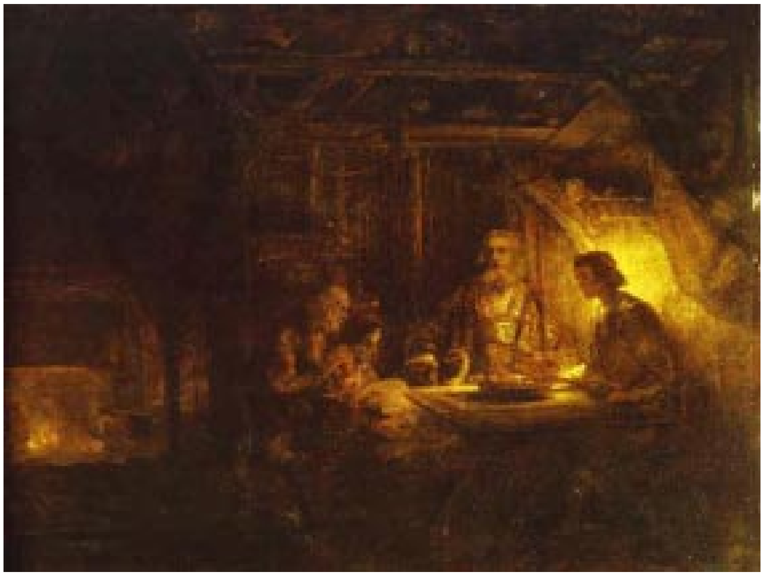


Figure 2: Rembrandt's Philemon and Baucis. (1658, The National Gallery of Art, Washington, DC, USA)

### *2.2 Requirements for adaptive multimedia systems*

For the representation and use of visual media in dynamic digital environments we have to represent both its intrinsic and conceptual aspects. This means that we can use the intrinsic representation to analyse, interpret or generate presentations of visual material. Relating back to the argument of applying this to digital museum environments, we now see that for the presentation of a collection on the basis of free interaction with the user on a rhetorical, e.g. educational basis, we have to be able to transform a fairly vague information request, i.e. about a particular style, in such a way that we can display not only the appropriate material but enforce the core element of the query.

We are in need of an expandable, adaptable description of the style based on a low level feature extraction and high level conceptual descriptions [8, 26]. Within the domain of arts metadata typically means text. Even though textual metadata provides conceptual structures, it usually covers aspects such as painter, date of painting and title of painting. The reason for this limited amount of useful data is that more semantic information is usually provided through further manual annotation - an expensive endeavour normally not covered by the archival budget. Hence, we require search mechanisms which are able to analyse and use the required low level information while retrieving.

In the following section we describe an approach to address these problems. The aim of the work is to improve existing search techniques on category and image features, and to combine these with presentation independent knowledge in order to improve the quality of the generated presentations.

## 3. A Framework for the automated generation of user-centred presentations

We incorporate automatically extracted image features in our existing framework for generating hypermedia presentations. The architecture, as described in Figure 3 on the following page, consists of three major units:

- the style repository, which embodies style schemata, style grammars and rule-bases for different presentation styles
- the data repository, containing the images and related metadata, and the retrieval engine
- the presentation environment, including a presentation generator and a hypermedia browser.

We provide now an overview on the different units, modules and the way in which they interact with each other. In section 4 we discuss an example of how the described units behave with real material.

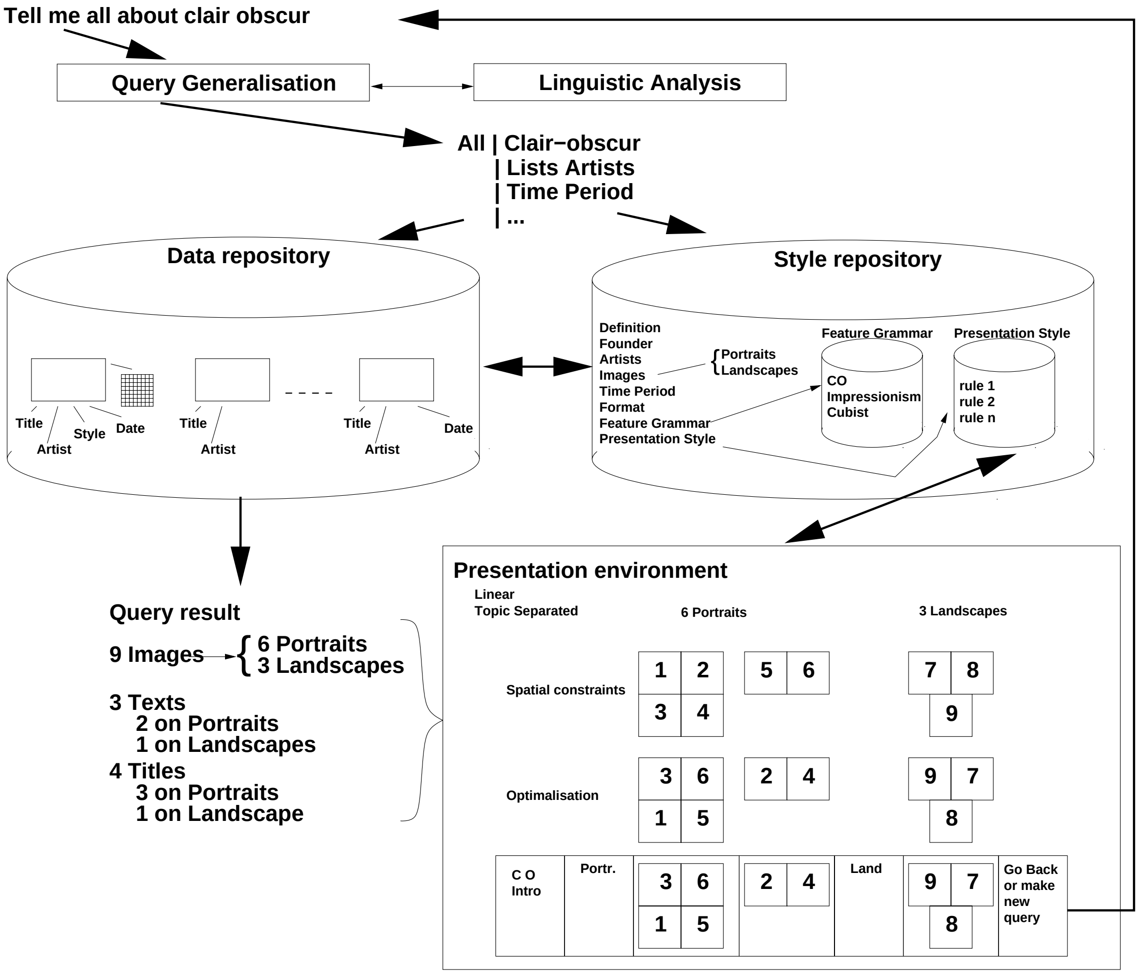


Figure 3: General scenario framework

### *3.1 The style repository*

The aim of the style repository is twofold. On the one hand it provides a collection of representations describing styles in fine art, such as *clair-obscur*, impressionism or cubism, in a structured way. These collections are designed to improve the retrieval of images or other metadata in the data repository (see the data repository section below). On the other hand it provides a presentation rule-base in which rhetorical structures describe how retrieved material can be presented.

The collection of representations of fine art styles provides for each style mainly text-based schemata. A schema holds information about the definition, the main period, the inventor of this style, other artists using or improving it, etc.. The advantage of these style ontologies is that they allow an enlargement of the search-space, if the style plays a prominent part in the query. This is state-of-the-art technology within the retrieval community [26]. The development of text-based ontologies is still a mainly manual task, which is today quite well supported [15].

This information, while important, is insufficient, low level feature descriptions are also required. Rather than a random choice of features as a style description, features that represent the intrinsic characteristics of a particular style need to be collected. In *clair-obscur* images, for example, we find a clear distinction of light and dark areas. Usually there is one dominant light source, predominantly filled with high luminance colours, alongside dark areas with a high proportion of brown colours which can be blended with other objects [3]. Thus, a collection of features such as colour, shapes, brightness, either in the form of their extraction algorithms or as threshold values for a particular style, facilitate the automatic identification of relevant material. Such a collection we call a feature grammar, and its functionality will be explained in more detail in section 4.

Note that the development of feature-based representations also requires human effort, in particular by specialised experts who have an understanding of the compositional structures of an image, as outlined in Figure 1. The collection of these features is, on the other hand, not too difficult, since tools for this particular task do exist [12].

The third representation form in the style repository is of a different sort. Here we collect rules describing rhetorical presentation structures, as addressed in the Rhetorical Structure Theory [19] or Cognitive Coherence Relations [18], which might vary between general and specialised levels. If, for example, the presentation environment is educationally oriented, it can build presentations on a larger level of the form: Introduction Topic; Introduction Subtopic 1; Details Subtopic 1; Introduction Subtopic 2; Details Subtopic 2; Introduction Next Topic. A more detailed level specifies what an introduction means, e.g. show a definition of the topic in combination with a visual example of the topic. Another detailed description might be concerned with the sort of interaction, such as a linear presentation in the form of a slide show, or a more interactive way in the form of additional buttons for individual traversal. The combination of these rules form themselves schemata, which can then be connected to relevant styles. The design of these schemata and the connection to particular style representations again requires human effort, such as that of a graphical designer of a museum Web environment. Development environments which support such tasks are described in [4, 21]. Once these presentation rules are in place, a system can react to the particular needs of a user.

### *3.2 The data repository*

The repository, as described in Figure 3, stores annotation schemata in the form of XML-based documents and media-based data, such as images in various formats (pic, gif, tiff, etc.). The repository itself can be realized using federated database technology.

The annotation documents are created by experts, using ontology-based environments for task-specific controlled vocabulary/subject indexing schemata for in-depth semantic-based indexing of various media [28]. Note that annotation schemata are different from the style representations. Annotation schemata provide information about one particular image or artist. For example, they capture information about the title of an image, its painter, production date, a list of exhibitions where it was presented, reviews, and so forth. The annotation process follows a strata-oriented approach, which allows a fine-granulated space-oriented description of media content, where particular areas within an image can be especially annotated. The connection can be based on linking mechanisms as described in XML path and pointer [7, 9] or MPEG-4 [16].

As visualized in Figure 3, the annotations will hardly ever be completed. Most of the time we will have merely the most basic data available. Thus, even if we are able to enlarge the potential search space, as described earlier, we might find a very limited information space to apply the query to.

Imagine that a user would like to know everything about Rembrandt and the different styles he painted his images in. With the textual representation we might be able to find images by Rembrandt in the database. However, if these images have no further annotation attached than 'Artist = Rembrandt' we would not be able to classify the retrieval results according to the query. Having access to the style specific representation of intrinsic features, we can now analyse the image during the retrieval process and decide based on the results which of the relevant styles is the most appropriate for this particular image. As a side effect, we can also use the gained feature information as additional annotation for the image, not only in classifying it as a particular style, but also providing several different representations of it, which can be, for example, useful for presentational purposes. An image in the style of *clair-obscur* can be additionally represented in a grid that contains the light and dark areas. This grid can form the basis for a presentation of images, where the style of the images and the style of the presentation correspond.

As the main goal of the suggested framework is to facilitate the automatic generation of user-centred multimedia presentations, we suggest that the result space will not only contain the retrieved data, associated metadata, and the relations between these different units but also information required for their presentation. Moreover, it also returns physical information about the retrieved data, i.e. image size and image file type.

### *3.3 The presentation environment*

The presentation environment, as displayed in Figure 3, is basically a constraint-based planning system, using the definitions provided in the style representation schemata and the presentation styles [29]. Since the system can access descriptions based on spatial, gradient and colour features, the presentation generator is in the position to analyse the retrieved material based on the relevant presentation design, according to design issues such as graphic direction, scale, volume, depth, shapes (i.e. physical manipulation of the material for better integration into the presentation), temporal synchronisation (interactive or linear presentation), etc., and provides a format that a hypermedia browser can interpret, e.g. SMIL or MPEG-4 [16, 30, 31].

Having introduced the main units within our framework, we are now in the position to explain the actual processes by means of a scenario. The scenario illustrates the current state of our prototype environment at CWI.

## 4. Scenario

Imagine a visitor of the web environment of the Rijksmuseum is reading the associated text of a painting. The visitor is intrigued by the technique *clair-obscur* and would like to know more about it. In the search field she types the following query: Tell me all about *clair-obscur*.

### *4.1 Query generation*

The query of the user is transformed based on simple linguistic syntax decomposition into two main units: a measure unit to describe the amount of data to be retrieved, and a topic unit, to explain what content needs to be searched for. The result for the example query is 'all' as a measure and *clair-obscur* as a topic. The former is interpreted by the system as: full database search on Topic, and results in the relevant qualifiers for the final query. The topic is analysed if it falls into one of the categories: Artist, Style or Art-Type, for which conceptual representations are available. Since *clair-obscur* is a style, the style representation is addressed, which covers the following information:

- Definition, a text describing what the style is about;
- Founder, which provides the name of the artist who came up with the first image of this style;
- Artists, a list of artists using this style;
- Images, a list of the most important examples of this style;
- Format, presents a list of typical formats, such as landscape, portrait, still life, etc.;
- Grammar, a link to a collection of feature detectors;
- Presentation style, link to a top level rule in the presentation rule base.

The query generator extracts the inventor, artists, images and format elements and adds them as additional attributes to the topic. This collection forms the basis for the query, which is represented in W3C's 'XML query' database format [10]. Definition, grammar and presentation style are marked as existing. This is important because related activities will only be performed if these mechanisms are available.

### *4.2 Retrieval*

In the context of this paper, we are primarily interested in the retrieval of simple concepts and will thus ignore the retrieval built on text-based content descriptions.

If we interpret the database, or a collection of databases, as the complete search space, then we can divide this space in three areas:

- material with annotations that provide evidence that the main topic is identifiable. For our example this means that the style is provided. If the image fulfills the query further processing on this image is not necessary. However, if additional information, such as brightness histogram, segmentation representation or grid abstraction cannot be retrieved the feature grammar will be activated for producing these items;
- material with annotations that allow a relation to the required topic through one of the additional attributes, but don't provide evidence that the topic is part of the image. An example would be an image that fits the time period and an artist but has no identification about the style. This is when the feature grammar is applied. A positive identification of the style results in an annotation of style and the other representations (histogram, segment, abstraction);
- material with no identification. Here the grammar can also be applied. Since the evidence is merely based on the execution of the grammar, extra validation needs to be collected, e.g. the judgment of a human expert.

Since the functionality of the feature grammar is of importance we will now explain it in more detail.

*A clair-obscur feature grammar* When there are no annotations of a multimedia item available, there is still the raw data of the item itself. In research areas like computer vision, speech recognition, etc. many algorithms have been devised to extract meaningful features from this raw data [8]. These low-level features can then be aggregated into high-level concepts. Features or concepts determined by one algorithm can, of course, be input to another algorithm. Using these chains of algorithms an annotation of the multimedia item could be created semi-automatically.

However, to create this annotation the algorithms should be executed in the right order, so a high-level description of their dependencies is needed. The Acoi system [32] provides the framework to specify this description and to create the annotations. The description is specified in the feature grammar language, which is at the core of the system.

A feature grammar describes the relationships between annotations (so called features), algorithms (so called feature detectors) and mutual detectors in a set of grammar rules. The grammar itself is basically context-free [27] (using the extended notation), but, additionally, some of the symbols are associated with feature detectors.

To build the annotation, the grammar has to be interpreted using the Feature Detector Engine (FDE). The FDE is a simple recursive-descent parser, which calls the detectors and manages a stream of tokens and a parse tree. The main goal of the FDE is to determine the validity of the start rule of the grammar. While doing this it will execute the feature detectors. These detectors will produce new tokens for the token stream. The tokens, when they match the grammar, will move from the stream into the parse tree. When the validity of the start rule has been proven the complete annotation is available in the form of a DOM structure.

Lets have a look at the feature grammar for, among others, the clair-obscur annotation, see Figure 4. A small walkthrough of this grammar will show how the FDE processes it. The start rule of the grammar is *painting*. This declaration also states that the initial token stream needs to contain the *location* token. To prove that the start rule is valid, the FDE tries to prove the validity of all the symbols in the right-hand side of the *painting* rule. The rules are always interpreted in a top-down and left-right manner, so *general* is the first rule to be evaluated. The first symbol in this rule is *location*, which is an atom. If the name of the first token in the stream matches, the token is moved to the parse tree. The next symbol is a detector, *light_segment*. In its declaration

```
%start          painting(location);

%atom           uri;
%atom           uri location;
%atom           flt threshold, coverage;
%atom           int column, row, x, y, width, height;

%detector       light_segment(location);
%detector       histo_segment(location);
%detector       segment(location,flt);
%detector       global(location);
%detector       normhisto(uri,shape);
%detector       grid(location,int,int);
%detector       clair_obscur();

%detector       light coverage[0] > coverage[1];
%detector       dark coverage[0] <= coverage[1];

painting        → general global local technique;
general         → location light_segment;
light_segment   → location histo_segment segment(threshold[0]);
histo_segment   → threshold*;
segment         → location;
global          → shape features;
shape           → x y width height;
features        → light;
light           → normhisto(light_segment.segment.location)
                  (light | dark);
normhisto       → coverage*;
local           → grid(10,10);
grid            → (column row shape features)*;
technique       → clair_obscur;
```

Figure 4: Clair-obscur feature grammar

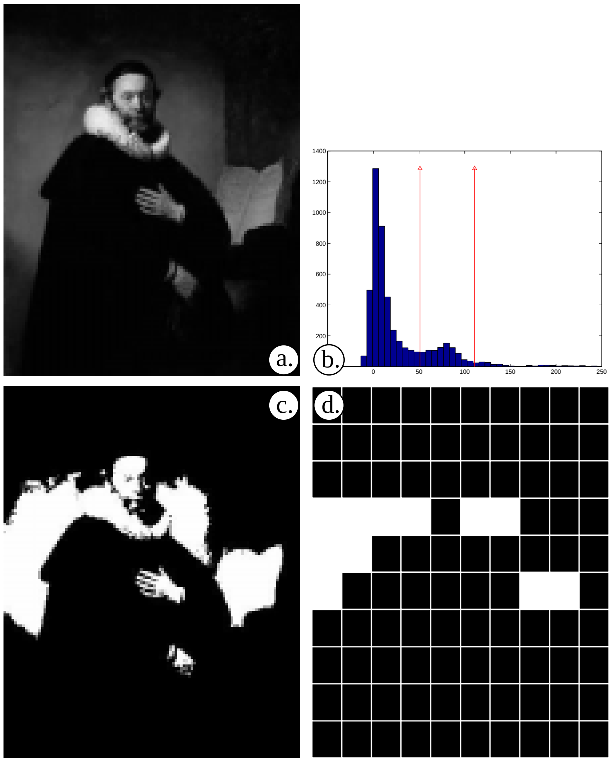


Figure 5: Feature detection steps

one parameter is specified, the *location*. These paths always refer to symbols in the parse tree, and a cursor to the proper position in the tree is passed on to the detector. The *light_segment* detector takes the original image of the painting and calculates a feature value for each pixel, *i.e.*, the gray value, and stores this in a new image. The location of this image is stored as a new token in the token stream, and, because it fits the grammar, will be moved by the FDE to the parse tree. In this way the FDE will continue to encounter atoms and detectors, and move tokens from the stream to the parse tree. When finally the whole annotation is available it can be dumped as a XML document. This document is then used as input for the next step in the presentation generation.

The feature grammar has a generic setup. For example it is easy to add new features, another localizing schema or another technique to the grammar. In these cases not the whole annotation has to be regenerated, but the FDE can do an incremental parse. An incremental parse means that the FDE will take the old parse tree and add new branches, therefore only executing the detectors for the new or updated rules in the grammar.

In the next section the global flow of execution embedded in this feature grammar is explained.

*Clair-obscur feature detection* Since we aim to design an approach that is data-driven and can therefore operate unsupervised, it is important to incorporate adaptive decision-making algorithms. For instance, in the case of the clair-obscur a vague high-level description of the style could be "a *brightly lit* object or person surrounded by a *dark* background". To translate this vague conceptual description into an operational low-level feature-extractor, we need to assign precise values to fuzzy concepts such as *bright* and *dark*. However, we cannot fix them in advance because these values depend on the context, *dark* and *bright* being defined relative to the rest of the painting.

The approach we propose is data-driven in that it inspects the data in search for natural thresholds, i.e. thresholds that are dictated by the structure apparent in the data [22]. To be more precise, assume that we have a numerical image-feature $x$ that can be computed at each of the $n$ pixel in the image (e.g. hue, or brightness, see Figure 5.a). This gives rise to a numerical dataset $x_1, x_2, \ldots, x_n$. To get an idea of how these values are distributed over the image, we can look at the histogram. If, in terms of this feature, the image has a clear structure then we expect to see a multi-modal histogram, with peaks over the most-frequently occurring feature values.

For instance, in the case of clair-obscur, computing the brightness histogram we expect to observe (at least) two peaks: one peak at low values created by the pixels in the dark regions, and one at high values corresponding to bright pixels, see Figure 5.b.

Locating the grey-value at the minimum in between these peaks determines a threshold that can be used to separate the bright from the dark regions in the image. This seemingly simple task is complicated by the fact that a data-histogram almost never has a clear-cut unimodal or multi modal structure, but exhibits many local maxima and minima due to statistical fluctuations. The challenge therefore is to devise a mathematically sound methodology that allows us to construct a smoothed version of the histogram, suppressing the spurious local extrema that unduly complicate the histogram structure.

To this end we introduce the empirical distribution function $F_n(x)$ which for each feature-value $x$ determines the fraction of observations $x_i$ that are smaller than $x$. The reason for switching to the empirical distribution is that it allows us to compute the precise probability that the given sample is drawn from a theoretically proposed distribution $F(x)$. The idea is simple: we search for the *smoothest* distribution $F$ that is compatible with the data, i.e. such that there is a high probability that the sample $x_1, \ldots, x_n$ has been obtained by sampling from $F$.

In mathematical parlance this amounts to solving the following constrained optimisation problem: given $F_n(x)$ find $F(x)$ that minimizes the functional

$$\Psi(F) = \int (F''(x))^2 \, dx \text{ subject to } \sup_x |F_n(x) - F(x)| \leq \epsilon.$$

(The value for $\epsilon$ is fixed in advance by specifying an acceptable level of statistical risk). This optimisation problem can be solved using standard spline-fitting routines. Once the shape of the smoothest compatible distribution $F$ is determined, its inflection points can be used to determine the genuine local minima in the histogram, thus yielding natural thresholds for the image-segmentation extractor.

The lowest of these thresholds is then used to segment the image into dark and light areas, see Figure 5.c. And as a next step information about the areas is localized by overlaying the image with a grid, see Figure 5.d. The grid functions as the abstraction of the original image and forms the input to the final step: determinating if the painting uses the clair-obscur technique.

In the feature grammar of Figure 4 these steps are distributed over several detectors and their dependencies are described. For example, the *light_segment* detector calculates the brightness value of each pixel, this set of values is then taken as input by the *histo_segment* detector to determine the segmentation thresholds.

Based on the above discussion we are now in the position to describe the result space of a query.

*Result space* The result space provides a structured description of the retrieved material, represented in the form of an XML document. The structure reflects the types of retrieved data (text, images), the relation between them in the form of triplets (e.g. name-of, author, image) and additional information regarding valuable presentation information, such as sizes of data units, attached grid abstractions, and so forth. Note, the actual data, i.e. the file of an image, is referred to by using links.

Suppose that for the original query 'show me all about Rembrandt' the retrieval space consists of 6 images of the type portrait and 3 landscapes and for some of them author names are available. The final step is the transformation of the retrieval results into a presentation according to the layout functions, described in the rule set of presentation styles.

### *4.3 Generating the presentation*

Part of our prototypical implementation is a generation engine that is able to transform a high-level description of a presentation [24] in the concrete final-form encoding that is readily playable on the end-user's system. In our system the final encoding form is SMIL [30].

The presentation generation engine of our system is a constraint-based planning system. The constraint system is used for solving the design-based constraints, such as:

- the overall presentation dynamics (e.g. linear or interactive) and the resulting subdivision of information blocks;
- organising material for each information block, e.g. number of elements on a page and their spatial outline based on the actual size of each information unit;
- optimisation of ordered material based on additional style criteria, such as colour or brightness distribution, in particular to emphasize a particular style.

In this paper we are merely interested in the last bullet. The inner details of the other parts of the system itself, especially the transformation of the presentation structures generated by the constraint engine into a SMIL presentation, have been discussed in previous work [29].

We assume that the system constructs a linear presentation for educational reasons and creates topic blocks to present the material. Finally, based on spatial constraints, it calculates how many images for each topic block can be presented on a page.

At this stage the generator tries to arrange the images in each topic block in such a way that the style criteria for *clair-obscur* are fulfilled. A decision rule could look as follows:

```
style_order(Image_Style, List_Of_Images, Images_Per_Page, Presentation_List):
    gradient-match (Image_Style,List_Of_Images, Result__List),
    border-match (Result_List,Images_Per_Page, Presentation__List).
```

With this rule the system analyses not the image itself but rather its grid abstraction, as described in Figure 5d. The system first tries for a particular style (Image_Style) to order the images of one topic (List_of_Images) based on the pattern provided by those cells of the grid that represent light values. The analysis of these patterns is based on graphical shapes, such as triangle or rectangles. The direction of the light is derived from a number of criteria, such as solidness of a pattern (main light center), position in the grid (at the border indicates that the light source is outside the image), and the direction of the dissolve of this shape (direction of light beam). For Figure 5d the result is that the light source is outside the image, that light is coming from the left side and dissolves towards the right side in a rectangular way. The Result__List groups images in lists, where light follows similar directions, such as left, up-left, up, up-right, right, down-right, down, down-left, circular.

Once that is done, the system tries to align the images based on similar border pattern. If we take once again Figure 5d as the example, the system would try to find an image which shares a similar distribution of light and dark cells (up or down by one grid cell) but only on its right side. The combination of images is performed on the previous calculated maximum size of images per page.

Figure 6 and Figure 7 demonstrate how the unordered presentation looks before and after the transformation.

Using the rhetorics of an educational-oriented presentation, which requires introductions of topics and subtopics, a final presentation might look as presented in Figure 8.

The temporal duration for every single page is calculated by the number of presented objects and their graphical complexity, the number of words for text elements, or temporal presentation qualifiers such as fade-in or out times for media units. The last screen offers choices for the next step.

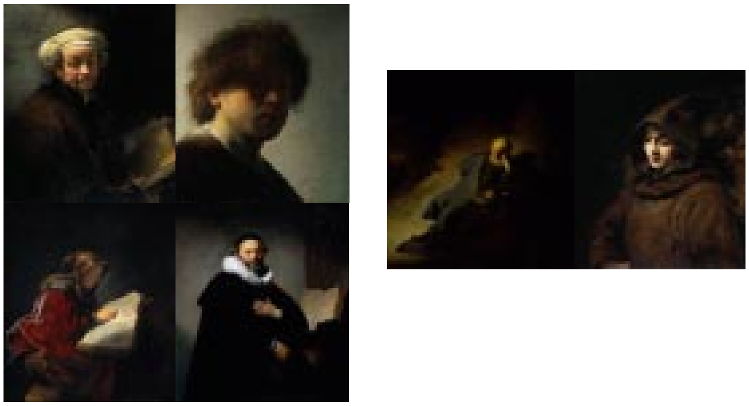


Figure 6: Ordered retrieval result before optimisation

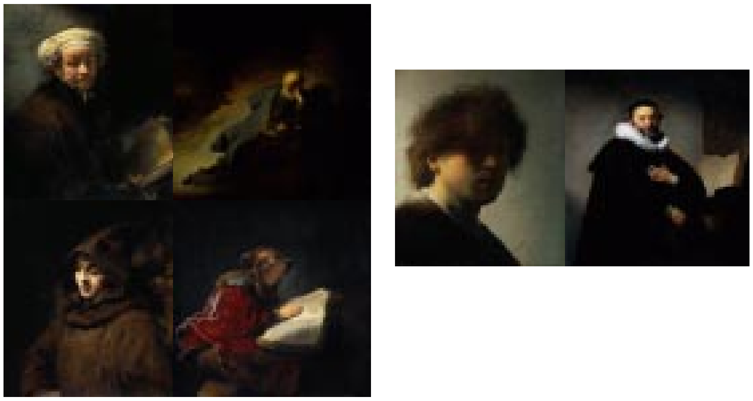


Figure 7: Ordered retrieval result after optimisation

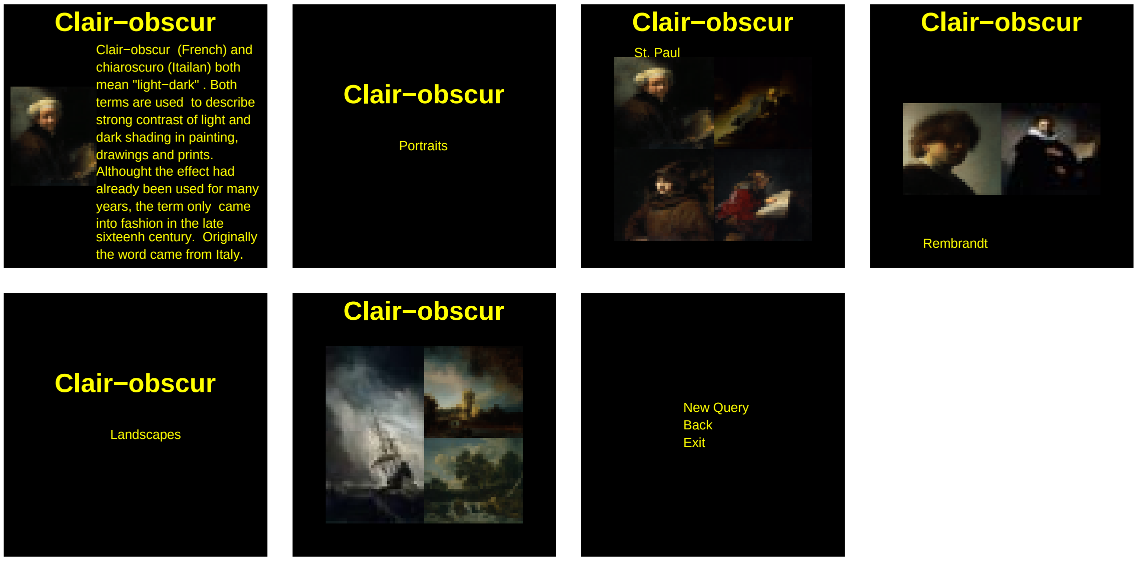
### *4.4 Evaluation of the example and the approach*

The current database for our prototype environment contains about 30 images, of which 10 can be identified as *clair-obscur*. Furthermore there is textual information for every image available, covering names of painters, image titles and smaller definitions of styles.

The retrieval, for cases where no further annotations are available and all the image processing needs to be done for style detection and the generation of the abstract image representation, is rather slow (about 3s per image). Under these circumstances the generation of a presentation as described in Figure 8 can take up to 1 minute. Once material is annotated the generation time amounts to seconds.

Regarding the collection of features in the grammar we do understand that *clair-obscur* is a misleadingly simple style. The grammar is simple because it only covers brightness elements. We are currently improving it with additional information on a colour level, to adapt it to the particular style feature of brown colours which nicely mingle with black. This will allow us to further improve the alignment of images.

We have not tested the grammar on material which is similar to *clair-obscur* images, such as keyframes of films of the film-noire genre, or extreme black and white images, as done by the artist Sanders. We assume that the system would describe them as *clair-obscur* images, which would not be false per se. In general it would only show the importance of light in art. However, in a presentation about Rembrandt those images would be ill suited because of the context. That is why we see the need for the extra information within the style schema. However, we are intrigued by the idea of using the low level features as one basis for the comparison of styles.



Clair−obscur

Clair−obscur (French) and chiaroscuro (Itailan) both mean "light−dark" . Both terms are used to describe strong contrast of light and dark shading in painting, drawings and prints. Althought the effect had already been used for many years, the term only came into fashion in the late sixteenh century. Originally the word came from Italy.

Clair−obscur

Portraits

Clair−obscur

St. Paul

Clair−obscur

Rembrandt

Clair−obscur

Landscapes

Clair−obscur

New Query
Back
Exit

Figure 8: The presentation

For obtaining a better understanding on how far the grammar will help us to handle different styles we are currently working on two different styles. One is impressionism, which is devoted to the representation of colour, and the other one is cubism, which centres on the representation of shapes. The question we would like to answer is not only if we can detect these styles, but what it would mean to use them for presentation. However, we have no illusion that even if we can deal with different colours, textures and shapes, there are still a large variety of style which we properly cannot handle. The image 'Lavender Mist' by Jackson Pollock, as portrait in Figure 9, is a good example.

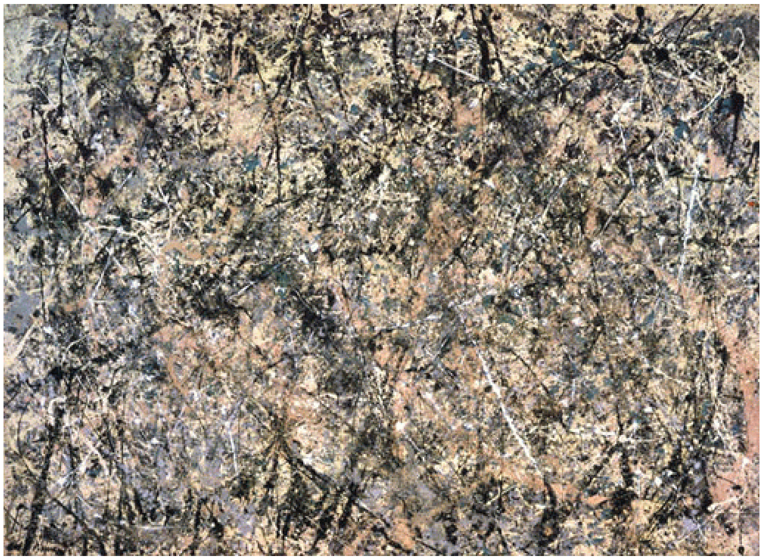


Figure 9: Jackson Pollock, Number 1, 1950 (Lavender Mist),1950, National Gallery of Art, Washington D.C., USA

## 5. Conclusions

In this article we argued that for the automatic generation of adaptive multimedia presentations we are in need of expandable, adaptable descriptions of style which provide both high-level conceptual and low-level feature extraction information. Only the combination of both facilitates the retrieval of adequate material and its user-centred presentation. We discussed the problems of visual signification for images and in dynamic systems and showed how the combinational approach can help overcome such problems. We proposed an architecture for such a system and presented its applicability for a museum-oriented multimedia system.

The described architecture and system behaviour and its theoretical foundation are best regarded as a platform that demonstrates the feasibility of automated stylistic-oriented multimedia generation in restricted, yet complex, domains.

## 6. Acknowledgments

The authors wish to thank the Rijksmuseum in Amsterdam for insightful discussion and helpful comments on art styles and for the provided access to their database. Furthermore we are thankful for their permission to use the images from their collection for this paper.

Part of this research was funded under the Dutch research projects Dynamo and ToKeN2000.

References

1. E. André, J. Müller, and T. Rist. *WIP/PPP: Knowledge-Based Methods for Fully Automated Multimedia Authoring.* London, UK, 1996.
2. E. André, J. Müller, and T. Rist. Presenting Through Performing: On the Use of Multiple Lifelike Characters in Knowledge-Based Presentation Systems. In *Proc. of the Second International Conference on Intelligent User Interfaces (IUI 2000)*, pages 1–8, 2000.
3. R. Arnheim. *Art and Visual Perception: A Psychology of the creative eye, The new version.* London: Faber and Faber., 1974.
4. G. Auffret, J. Carrive, O. Chevet, T. Dechilly, R. Ronfard, and B. Bachimont. Audiovisual-based Hypermedia Authoring: using structured representations for efficient access to AV documents. In *Proceedings of the 10th ACM conference on Hypertext and Hypermedia*, pages 169–178, Darmstadt, Germany, February 21-25, 1999. ACM. Edited by Klaus Tochterman, Jorg Westbomke, Uffe K. Will and John J. Leggett.
5. S. Boll, W. Klas, and J. Wandel. A Cross-Media Adaptation Strategy for Multimedia Presentations. In *ACM Multimedia '99 Proceedings*, pages 37–46, Orlando, Florida, October 30 - November 5, 1999. ACM, Addison Wesley Longman.
6. P. Brusilovsky, A. Kobsa, and J. Vassileva, editors. *Adaptive Hypertext and Hypermedia.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
7. J. Clark and S. DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendations are available at http://www.w3.org/TR/, 16 November 1999.
8. A. Del Bimbo. *Visual Information Retrieval.*
9. S. DeRose, E. Maler, and J. Ron Daniel. XML Pointer Language (XPointer) Version 1.0. W3C Candidate Recommendations are available at http://www.w3.org/TR, 8 January 2001.
10. A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. XML-QL: A Query Language for XML. W3C Notes are available at http://www.w3.org/TR, August, 19, 1998.
11. U. Eco. *A Theory of Semiotic.* The Macmillan Press, 1977.
12. G. Frederix, G. Caenen, and E. J. Pauwels. PARISS: Panoramic, Adaptive and Reconfigurable Interface for Similarity Search. In *Proc. of ICIP 2000 International Conference on Image Processing. WA 07.04*, pages 222–225, September 2000.
13. E. H. Gombrich. *The story of Art.* Phaiadon Press Limited, London, 1999.
14. J. Greimas. *Structural Semantics: An Attempt at a Method.* Lincoln: University of Nebraska Press, 1983.
15. W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). Technical Report SMI Report Number: SMI-1999-0801, Stanford Medical Informatics (SMI), 1999.
16. International Organization for Standardization/International Electrotechnical Commission. Information technology – Coding of moving pictures and audio, 1999. International Standard ISO/IEC 14496:1999 (MPEG-4).
17. T. Kamps. *Diagram Design : A Constructive Theory.* Springer Verlag, 1999.
18. A. Knott and R. Dale. Choosing a Set of Rhetorical Relations for Text Generation: A Data-Driven Approach. *Trends in Natural Language Generation: an Artificial Intelligence Perspective*, 1996.
19. W. C. Mann, C. M. I. M. Matthiesen, and S. A. Thompson. Rhetorical Structure Theory and Text Analysis. Technical Report ISI/RR-89-242, Information Sciences Institute, University of Southern California, November 1989.
20. J. Monaco. *How To Read A Film.* New York: Oxford University Press, 1981.
21. F. Nack and C. Lindley. Production and maintenance environments for interactive audio-visual

stories. In *Proceedings ACM MM 2000 Workshops - Bridging the Gap: Bringing Together New Media Artists and Multimedia Technologists*, pages 21–24, Los Angeles, CA., October 31, 2000.

22. E. Pauwels and G. Frederix. Image Segmentation by Nonparametric Clustering Based on the Kolmogorov-Smirnov Distance. In *Proc. of ECCV 2000, 6th European Conference on Computer Vision, Dublin*, pages 85–99, June 2000.
23. C. Peirce. *The Collected Papers of Charles Sanders Peirce 1 Principles of Philosophy and 2 Elements of Logic, Edited by Charles Hartshone and Paul Weiss.* Cambridge, MA: The Belknap Press of Harvard University Press, 1960.
24. L. Rutledge, B. Bailey, J. van Ossenbruggen, L. Hardman, and J. Geurts. Generating Presentation Constraints from Rhetorical Structure. In *Proceedings of the 11th ACM conference on Hypertext and Hypermedia*, pages 19–28, San Antonio, Texas, USA, May 30 – June 3, 2000. ACM.
25. L. Rutledge, J. van Ossenbruggen, L. Hardman, and D. C. A. Bulterman. Mix'n'Match: Exchangeable Modules of Hypermedia Style. In *Proceedings of the 10th ACM conference on Hypertext and Hypermedia*, pages 179–188, Darmstadt, Germany, February 21-25, 1999. ACM. Edited by Klaus Tochterman, Jorg Westbomke, Uffe K. Will and John J. Leggett.
26. S. Santini and J. Ramesh. Integrated Browsing and Querying for Image Databases. *IEEE MultiMedia*, pages 26 – 39, July -September 2000.
27. A. Schmidt, M. Windhouwer, and M. L. Kersten. Feature Grammars. In *ISAS'99 The 5th. Int'l Conference on Information Systems Analysis and Synthesis*, Orlando, Florida, 1999.
28. F. van Harmelen and I. Horrocks. Reference description of the DAML+OIL ontology markup language. http://www.daml.org/2000/12/reference.html, December 2000. Contributors: Tim Berners-Lee, Dan Brickley, Dan Connolly, Mike Dean, Stefan Decker, Pat Hayes, Jeff Heflin, Jim Hendler, Deb McGuinness, Lynn Andrea Stein.
29. J. van Ossenbruggen, J. Geurts, F. Cornelissen, L. Rutledge, and L. Hardman. Towards Second and Third Generation Web-Based Multimedia. In *The Tenth International World Wide Web Conference*, pages 479–488, Hong Kong, May 1-5, 2001. IW3C2. This is a revised version of CWI technical report INS-R0025.
30. W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. W3C Recommendations are available at http://www.w3.org/TR/, June 15, 1998. Edited by Philipp Hoschka.
31. W3C. Synchronized Multimedia Integration Language (SMIL) 2.0 Specification. Work in progress. W3C Working Drafts are available at http://www.w3.org/TR, 1 March 2001. Edited by Aaron Cohen.
32. M. Windhouwer, A. Schmidt, and M. L. Kersten. Acoi: A system for Indexing Multimedia Objects. In *International Workshop on Information Integration and Web-based Applications & Services*, Yogyakarta, Indonesia, November 1999.