Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

INS

Information Systems

INformation Systems

INS

Colour picking - the pecking order of form and function

F.-M. Nack, A.S.K. Manniesing, H.L. Hardman

REPORT INS-R0303 APRIL 30, 2003

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Colour Picking - the Pecking Order of Form and Function

Frank Nack, Amit Manniesing and Lynda Hardman
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT
Multimedia presentation generation has to be able to balance the functional aspects of a presentation that address the information needs of the user, and its aesthetic form. We demonstrate our approach using automatic colour design for which we integrate relevant aspects of colour theory. Colour selection takes the relative importance of form and function into account through the use of weights in the generation process. We do not provide a definition of the relative importance of form versus function, but seek to explore the roles of subjective elements in the generation process.
1998 ACM Computing Classification System: H.5.1, H.5.2, H.5.4, I.7.0
Keywords and Phrases: Multimedia semantics, automatic colour design, colour harmonisation, style-driven multimedia presentation generation
Note: This work is carried out under the Dutch national ToKeN2000/CHIME project

1. Introduction

The automatic generation of multimedia presentations has been a focus of multimedia research for over a decade. The aim is to establish generation mechanisms with adaptive [5] or adaptable qualities [19] that adjust the multimedia presentation to the specific context of an individual user. Various attempts to explore and develop innovative presentation techniques have been described [1, 2, 4, 9, 12, 26]. These approaches facilitate the synthesis of multimedia documents and plan how this material is presented to various users. The underlying assumption of these systems, however, is that all material and user requests are known. In dynamic environments, where neither the individual user requirements nor the requested material can be predicted in advance, the established planning approaches are insufficient.

Instead, we claim that a system that automatically generates functional and aesthetically pleasing presentations in dynamic environments needs the knowledge to provide a balance between media content, meaning, usability and aesthetics. As a result, it facilitates the communication of information by addressing the circumstances and presuppositions of the user at the time of accessing the information. To enable this the system requires knowledge of low-level codes, collections of objective measurements [7, 21] representing prototypical style elements, in combination with high-level conceptual descriptions [20] that support contextual and presentational requirements.

We focus on achieving parity between form and function within the process of designing visualisations of information as multimedia presentations. The goal is to establish an equilibrium between the inter-dependent sources that drive the creation process of the presentation, namely models of discourse, user, media and domain characteristics, graphic design and device capabilities. In the following discussion we concentrate on the aspects of graphic design, as this micro-world illustrates essentially all the design trade-offs in the complete production process.

Automating the graphic design aspects of the presentation generation process has to replace the human designer in two ways. First, a dynamic-response environment needs to determine how the different information items should be presented through the use of the paradigmatic elements and structures of design: namely colour, framing and composition [18, 15, 23, 27]. Second, it has to communicate to different people, at different times, by facilitating the transformation of functional qualities into aesthetic values thus ensuring appropriate cognition.

In this paper, we illustrate how our experimental system utilizes colour, framing and composition to automatically pick appropriate colours for the communication goal. In addition, it assigns them to functional components to address the needs of the user with respect to the media content and navigation, retaining an overall balanced, coherent appearance.

We choose colour as the basis of our exploration of this new direction of multimedia generation because colour both functions as part of the visual rhetoric [13] and refers to the sphere of decoration [22]. It thus forms part of dynamic aesthetic codes that are particular to social class and culture [8]. Moreover, automatic colour design has the advantage above other design steps, such as the spatial or temporal layout of information units, that it allows maximal control of the system over the process itself, once the essential design constraints are formulated. We refer to the design of spatial-temporal layout and typography only where these are required for colour design.

The aim of this work is not to provide a universal definition of the balance between form and function, but to recognize subjective elements, such as the emotional and symbolic meaning of a particular colour for a viewer, and to include these explicitly in the system architecture.

The paper is structured as follows. We present the framework of our prototype multimedia generation environment to provide a brief overview of the overall representations and generation methods the colour design process can rely on. We briefly outline the semantics of colour. We then describe in detail our approach towards automatic colour design. For the sake of clarity we explain the colour design process in the context of generating the introduction page of a particular presentation. Within this exploration we also integrate the aspects of colour theory upon which our approach is based. The article concludes with an evaluation of the presented approach and an outline of further research.

2. Underlying framework for the automatic presentation generation and scenario

The process of designing visualisations of information, be it for cultural or technical purposes, is complex and resource demanding, where the end result might establish various presentational forms, such as textbook illustrations, advertising images, or a fine art multimedia presentation.

Consider the introduction screen for a presentation on the art movement “De Stijl”, as portrayed in Figure 1. This page emerged from the wish of a visitor of the web environment of a museum to learn more about the “De Stijl” movement after she was intrigued by a Mondrian image she found during browsing the collection.

Despite decisions for the overall logic of the presentation, for example that further investigation of the subject or related topics should be possible, a considerable number of decisions need to be made with respect to layout, typography and colour design of this particular page to ensure its functionality. The prerequisite of legibility of text, for example, might result in a provision of black text on a white background. Such a decision then requires that the different functionality of textual links can be easily distinguished, for example by assigning a particular colour scheme to them. On an aesthetic level the page establishes a “De Stijl” feel by advocating pure abstraction and simplicity: form is reduced to the rectangle, typography for headers uses the “De Stijl” font set, and colour uses the primary colours red, blue and yellow, along with black and white.

The Cuypers system, our experimental workbench, provides a framework for automatically generating such multimedia presentations [24]. The architecture includes 5 modules, namely the user module, the platform module, the discourse module, the domain ontology, and the design module, covering the various knowledge representations in the form of facts and task-solving routines (mainly constraint solving) required during the presentation generation process. The presentation engine controls the organisation of this process. Note, all modules have their own conceptual representations in RDF and each representation is accessible by every other module. In this article we focus on the design module.

Having introduced the main components within our framework, we are now in a position to explain our approach that addresses the initial requirements for the balance between functional and stylistic aspects in dynamic multimedia presentations, focusing on the automatic design of the colour schema for a presentation. We base the discussion in the context of the generation of a presentation about the art movement “De Stijl”.

Figure 1: Introduction page for a presentation on the art movement "De Stijl".

The user was browsing the collection and came across a Mondrian image resulting in a request for more information about the "De Stijl" movement. This request is interpreted, based on the user model, as the wish for a presentation that allows browsing through an evolving presentation. With evolving we refer to the concept of progression of detail that facilitates navigation based on a given weighted set of descriptors representing a story context on a micro level (next step in content exploration) as well as on a macro level (larger contextual units clustering content, such as classes of artefacts within an art movement), as described in [9].

Before we explain the colour design process in detail, as implemented in our system, we first provide a brief overview on the underlying assumptions with respect to colour semantics.

3. The semantics of colour

Our colour design process incorporates the two levels on which visual material can be experienced: optically (objectively, realistically) and cognitively (subjectively).

On the objective, thus perceptive, recognition and tonal level, colour operates purely in a functional way (see [22], p81). This means, for example, that making use of two out of roughly 20,000 colours, as are accessible to many viewers, establishes a difference between two information units without stating what the difference is. In other words, colour can be used as a mere functional element because it by itself does not have any initial meaning but does provide, based on its physical constraints, structure.

Colour is normally represented in the common colour systems, such as RGB, CMYK, CIE, YIQ, HSL, or HSV, in three dimensions, as described in Figure 2.

These systems facilitate the classification of colours in terms of hue, saturation, lightness, value or brightness [22]. The colour system used in our framework is HSL.

The subjective part of colour cognition addresses conceptual and emotional aspects. The conceptual level addresses semantic functionality, such as decoration (an aesthetic function) or label (a semantic function, such as distinguishing flags based on their colour scheme), or representation or imitation of

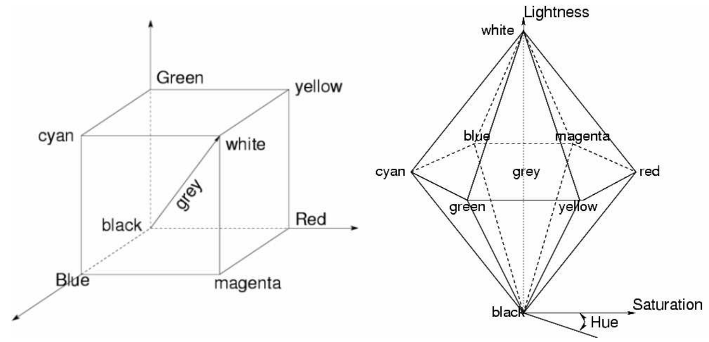

Figure 2: Three-dimensional colour space for RGB(left) and HSL(right)

reality (also a semantic function but here colour is used to represent nature, as blue is represented to characterize the sea in maps) [11, 22]. In order for colour to function in such a way, however, it must first be bound to a proper sign[1] such as an icon, symbol, or index[2]. Alternatively, colour may be bound to a verbal sign or a numerical value by proximity to form a code, as exemplified by Byrnes "The Elements of Euclid", where diagrams in primary colours and symbols replace the letter-coding of the native text (described on p84-87 in [22]).

Finally, colour functions on an emotional level, resulting in judgements assigning positive or negative weights to the presentation environment or the presented information itself. Some of those judgements are common cross-culturally, such as perceiving a colour as warm (cosy and inviting) or cold (distant and inhospitable), light (joyful) or dark (gloomy). Other scales of agreeable-disagreeable or attractive-unattractive use of colour combinations are rooted in culture. Examples for the culture dependency of colour use are the various and somehow contradictory colour harmony schemata [11, 22].

In the following section we explain in more detail how the different levels of colour functionality are applied in the automatic colour design process of our system.

4. An approach towards the automatic colour design

Our discussion starts at the point where the presentation engine has established the constraint-set on which the colour design process is based, as shown in the top line of Figure 3 on the next page.

4.1 Constraints driving the colour design process

In this section we give details about the constraint-set provided by the presentation engine. The required steps up to this point, such as the design of the discourse structure, the semi-automated generation of metadata and the content retrieval process [17, 16], as well as the automatic design of the spatial layout for retrieved material [10] are described elsewhere.

A typical request rule looks like this:

```
design_module(  Design_task,
                Communication_preference,
                Constraint_list,
                Result_list).
```

[1]A sign usually consists of two distinguishable components: the signifier (which carries the meaning) and the signified (which is the concept or idea signified) [8]. The relationship between the signifier and the signified is arbitrary, which enables the creation of higher order sign systems.

[2]Icon: A sign that represents its object mainly through its similarity with some properties of the object, based on the reproduction of perceptual conditions.
Symbol: A sign with an arbitrary link to its object (the representation is based on convention or association).
Index: A sign that represents its object by an inherent relationship

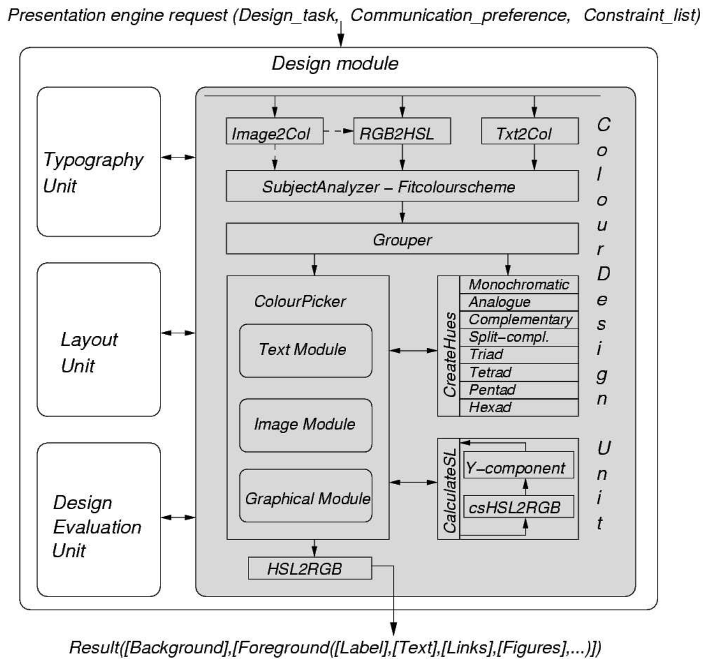

Figure 3: Detailed view on the colour component of the design module

With this rule the generation engine requests from the design engine a solution for a particular task based on the provided constraints. For the current example, the design-task is "colour_setting".

The request further determines a communication preference for the presentation that was already established in earlier design stages. The identified preference guides the graphic design process by suggesting which general design orientation the process should follow. In our environment we utilize the following three: **function**, **form**, **neutral**.

A preference for **function** determines that constraints based on importance weights are driving the design process. For the colour design process this might result in the dominant application of rules that compare colours based on contrast, as this effects the construction of legible text. On the other hand, favouring **form** will mainly employ rules that effect the appearance of colours, such for achieving pleasing appearances based on colour harmony schemata as described in colour theory [11, 22]. **Neutral** means that either way is acceptable. In our example, where the presentation is aimed at an educational presentation, a functional colour design process is required.

The `Constraint_list` contains the relevant restrictions for the current design task, namely "colour setting", established during previous generation stages performed by other generation modules. Such constraints supply the means to control the number of colours to be used and, even more important, the automatic allocation of colours to the various elements provided by the spatial layout of a page, as presented in Figure 4.

The structure of the `Constraint_list` introduces particular constraints related to the user, the domain and the layout, where each part represents in detail:

User (provided by the user module)
Physical_abilities that describes possible audio-visual deficiencies, such as colour blindness. Colour blindness makes it impossible to use red and green as structural distinguisher on the same page, as the user cannot distinguish between both colours. In our system colour blindness is characterised as `green_red`.

Domain (provided by the domain ontology and user module)

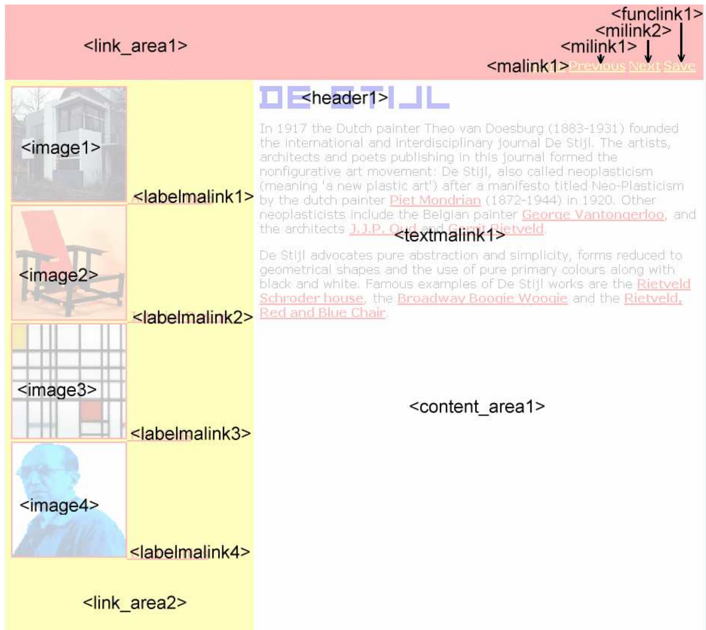

Figure 4: Graphical representation of the spatial layout structure before the colour design process.

Domain_colour that is a list of colours that are related to the domain the content is about or to particular colours required by the content environment the user is using, for example corporate identity colours. In our example the system might pick the primary colours used in the domain ontology for describing the "De Stijl" colour scheme [red, blue, yellow].

Layout (provided by the Layout unit of the Design module)
Presentation_elements, which is a structured list of all areas available and the various information units, combined in them. The list reflects on the one hand the spatial layout through ordering the various presentation areas according to their size (the largest area, which is with respect to colour the most dominant, comes first). On the other hand it also provides information about the importance of each information item according to its role within the discourse structure, qualified by an associated importance weight. Information items in this context can be: text, text containing links (textmalink), header, label, label as link (labelmalink), links for navigating on a micro level, such as moving from a definition to an example (milink), links for navigating on a macro-level, such as jumping from the architecture section to the painting section (malink), functional links, such as search, print, or save (funclink), images, etc.

An instantiated Constraint_list is described in Figure 5.

The educational basis of the communication goal of our example presentation is represented in the high importance value of the textmalink element within content_area1. Moreover, as the navigation style of the presentation is set to be "evolving", the importance for macro- and micro-navigation is reflected in the high importance value of the various navigation links.

The `Result_list` determines the output of the colour design module, namely the description of areas and included elements with the corresponding colour allocations. The presentation engine then processes the outcome to generate the final presentation of that page.

Based on the given description of the input parameters of the colour design module we can now demonstrate how the module solves the established constraints and creates the colour scheme for the presentation.

```
constraint_list(
[green_red,[red,blue,yellow],
[[area:[content_area1:1.0],elements:[textmalink1:1.0,header1:1.0]],
[area:[link_area2:0.9],elements:[labelmalink1:0.8,labelmalink2:0.8,
labelmalink3:0.8,labelmalink4:0.8,image1:0.6,image2:0.6,image3:0.6,
image4:0.6]],
[area:[link_area1:0.4],elements:[malink1:0.3,milink1:0.8,
milink2:0.8,funclink1:0.8]]]]).
```

Figure 5: Example of an instantiated Constraint_list

4.2 The generation of the overall colour design

The design module, as described in Figure 3, covers various design aspects, such as typography, layout or colour-design. In addition to those rather specialised units within the design module, the Design Evaluation unit provides high-level rules, which are used to guide the design process or to evaluate intermediate results as a form of sanity check. Such rules might state, for example, with respect to the colour design process that:

A. If `Communication_preference`
   = `function => work on area elements`
   = `form => work on area`
   = `neutral => work on mix`

B. If there are multiple background areas, begin assigning the one with the largest size first and continue in descending order.

C. The number of colours used ≤ 6.

Rule A determines whether the emphasis of the colour design is oriented towards the functionality of the information elements (foreground and structure) or rather on a look and feel, emotionally oriented presentation (emphasis on background and contrast).

Rule B considers the overall importance of mass in form of space within colour design. Larger areas are dominant with respect to perception and thus are more influential for the design process than smaller areas. This particular problem will be discussed later in this article in more detail. Note, there are other rules related to space that consider the shape of an area and the depth of colour in colour combinations (foreground background relationship) (see [11], pp. 120-124, 144 149).

Rule C provides the maximum number of colours to be used. Tests described in the literature showed that too many and different colours on a screen attract the users attention to the colours themselves and distract from the content [6]. We experimented with a range from 4 up to 8 colours and found that the system could work most efficiently with 6. More user testing is required, though, to finalise this rule.

These rules are required to organise the different steps within the colour design process, as described in Figure 3, which can be separated into:

- identification of available colours;
- detection of the colour scheme to be used;
- instantiation of the process order for the colour assignment;
- colour assignment.

These steps are described in more detail in the following sections.

Identification of available colours

The first step of the colour design module is to identify the potential number of colours to be used. For that the SubjectAnalyzer (see Figure 3) evaluates the `constraint_list` on areas and the information units contained in them. For the example provided in Figure 5 the SubjectAnalyzer comes up with 8 colours, namely three colours for the areas, three colours for the different sorts of links (mi-, ma- and funclinks), one colour for the text part of the textmalink field and one colour for the header.

However, high-level rule C states that the overall number of colours should not exceed a total of 6. Thus the SubjectAnalyzer re-evaluates the constraint-set by trying to group units. Merging the three types of links into one presentational group provides the required result and thus the number of colours to be used is established as 6.

Additionally, the SubjectAnalyzer identifies already selected colours, provided by the `Domain_colour` element of the Constraint_list [red, blue and yellow]. Note,in our implementation the colours have already been transformed to actual HSL values using the "Txt2Col" transformation module (see top in Figure 3), which transforms textual colour representations into the numeric expression in HSL format of the responding fundamental colour. For example red is transformed into `hsl[0,255,128]`.

At this stage it is known that not more than 6 colours need to be associated and that three of them are known.

Detection of the colour scheme

The next step is to identify the colour scheme used to assign colours. In our system four different schemata are implemented, namely achromatic, monochromatic, analogue and complementary (including split complementary), where each scheme represents a particular rule set for colour use.

The schemata are inspired by the work of Tufte [23, 22] and Itten [11]. Both theories, though somewhat contradictory, are relevant because of the point of view of the underlying analysis, where Tufte aims to provide the best display of information and Itten is interested in a general theory of colour synthesised out of objective physical laws and their applicability in art.

Tufte's approach is oriented towards a minimal set of colours with a subtle distinction between them. Within a three-dimensional colour space, such as the HSL space described in Figure 2, this leads towards the identification of adjacent colours and results in three possible harmony schemata:

Achromatic: lacks all colours. Here only black, white and all shades of grey in between can be used.

Monochromatic: uses only a single hue. Differences are created by adjusting the saturation and lightness, which establishes different shades and tints.

Analogue: uses colours that are next to each other within the HSL space. The distance between colours is restricted to the hues being used and the distinction between warm and cold colours, which should not be mixed.

Itten, on the other hand, suggests a harmony scheme that is based on contrast. He developed a constructive theory of colour design that combines seven kinds of contrast, such as the contrast between hues, light and dark colours, cold and warm colours, or the contrast between the saturation of colours. With respect to harmony, Itten facilitates a scheme, the complementary scheme, which is based on the contrast of complementary colours in combination with the effect of the changing behaviour of colours depending on the surrounding colours. This scheme establishes systematic relationships between colours that are capable of serving as a basis for composition (see [11], p. 118 119).

Complementary: is based on the use of two different hue's opposite each other in the HSL space. The aim is to use cold and dark colours for the background and the contrasting colour for the foreground.

The simple complementary scheme can be extended, to allow for greater flexibility, into a split complementary (3 hues distributed unevenly within the colour space), triad (3 hues distributed evenly within the colour space), a tetrad (4 hues), a pentad (using an equatorial tetrad scheme and adding black and white), and a hexad scheme (a tetrad scheme including black and white or six different hues).

Applying our implemented schemata to the example scenario, the SubjectAnalyzer now tries to establish the scheme used to pick and allocate colours for the presentation, based on the potential number of and the already existing colours.

The selection process applies the approach of growing complexity to support the time constraints of the overall presentation generation. Thus, the algorithm tries to apply the achromatic scheme first, as it provides the smallest rule set. However, it fails because this scheme does not support colour. The monochromatic scheme is rejected because it only allows a single hue, whereas the domain colours already provide three different hues. The analogue scheme cannot be selected out of several reasons, such as

- the covered distance between the two outermost domain colours (red and blue) is too large (the angle between red and blue over yellow is larger than 120° in the HSL space),
- cool and warm colours would be mixed (blue with red or yellow),
- the distance between red and blue over yellow potentially allowed the use of green as additional colour, which is not possible, as the user has extreme deficiencies in distinguishing between red and green.

Finally, among the available complementary schemata a split complementary scheme is chosen (red and yellow grouped on one side, blue on the other within the three dimensional colour space), as it facilitates the use of the established colours.

Instantiation of the process order for the colour assignment

The next step in the colour design process is to establish the order in which the different areas and their elements should be coloured. The order is of importance, as assignments for the most relevant area should be based on the largest colour set (see rule A in Section 4.2). Established assignments will then act as further constraints in the ongoing colour picking process. The Grouper, as presented in Figure 3, performs this task.

There is one essential assumption on which the performance of the Grouper is based, namely that areas usually provide the background within a presentation, whereas the associated elements supply everything that is layered on top of it (foreground). The Grouper can now either pay special attention to the size of areas (form), to the importance weights for the associated information units (function), or a mix of both (neutral).

However, emphasising does not mean neglecting other relevant elements. For example, as the physical size of an area specifies its importance for the feel of a presentation (the larger an area is the more dominant its colour will behave), it has to be considered even though the main orientation of the presentation is functional. Thus, in our experimental system we work with three different ways of calculating the importance of an area, each taking both the size of the area and the contained information elements into account.

I_{AT} represents the overall importance of the area, I_A stands for the importance weight of the area with respect to content, P represents the size of the area provided by the reverse position value in the `constraint_list`, and I_E stands for the weight of an individual presentational unit.

We are aware that the current formulae do not fully represent the complex relationship between the foreground elements themselves, or all aspects of the relationship between foreground and background, especially when it comes to various layers of foreground elements. However, trials with changing numbers of areas (max 5), associated elements (max 20) and various levels of importance (in the range between 0.1 and 1.0) showed that the formulas establish a sensible hierarchy of areas, where areas are organized in decreasing order of importance value. Note, it does not necessarily mean that

$$I_{AT_{function}} = \frac{(I_A \times P) + (\frac{\sum_{i=1}^{n} I_{E_i}}{n}) \times 2}{2}$$

$$I_{AT_{form}} = \frac{P + \frac{\sum_{i=1}^{n} I_{E_i}}{n}}{2}$$

$$I_{AT_{neutral}} = \frac{(I_A \times P) + \frac{\sum_{i=1}^{n} I_{E_i}}{n}}{2}$$

the order is different to the one already provided by the `constraint_list`, which is only based on the size of areas.

As the `communication_preference` of the ongoing example is set to `function`, the Grouper uses the formula IATfunction, resulting in the following order of areas: Content_area1, Link_area1, Link_area2.

Once the order of areas is established the colour design unit starts with the assignment of colours to the most important area, i.e. Content_area1.

Colour assignment

The relevant unit for this task is the ColourPicker as shown in Figure 3.

Colour assignment colour_area1

The ColourPicker first separates background from foreground elements.

As functionality is more important, the ColourPicker tries to identify the colour for the foreground elements in `content_area1`, i.e. the `textmalink1` and the `header1` element. As both elements are of type text, the TextModule of the ColourPicker attempts to establish the element colours using those colours that already available: namely red, blue and yellow.

It is important to notice that the TextModule establishes simultaneously the background colour of this area too, as both colours depend on each other. The emphasis in the colour design process lies on the foreground.

The TextModule uses the domain specific colours in their HSL notation and tries all possible colour combinations with respect to all foreground and background groupings, where each combination is assigned with a legibility value (LV) based on

- the relative brilliance factor $(Y_{colour1} - Y_{colour2})$
- the contrast between both colours based on their classification as light and dark colours (light colours should be used as background and dark colours for the foreground on the screen [3])
- the contrast between both colours based on their classification as cold and warm colours (cool colours should be used as background and warm colours for the foreground)
- the complementary contrast between both colours.

The calculated LV lies between 0.0 and 1.0 and the higher the value, the better the combination.

Note, the above approach does not use colour ranges as input. The reason for not allowing that is that usually the provided colours have a particular value that must not be altered, such as the domain colours for a particular style or art movement or corporate identity colours. If the colour design module chose freely, ranges would be useful but we do not explore into this direction.

For our example the combination of a yellow background and a red foreground provides the highest LV (0.633). However, the LV does not reach the threshold of 0.8 that is required by the DesignEvaluation unit to accept a colour combination in the function mode.

As the result the TextModule falls back on the default colour for text, namely black, which now also sets the foreground colour for that area. The next colour to be assigned is the background colour.

Within the split complementary scheme, the complementary colour for black in the HSL space is white.

Once both colours are picked the DesignEvaluation unit performs a final test on lightness and contrast required for achieving "legibility". In fact, the extreme contrast between black and white should be avoided, as it can create extra shimmering effects that can strain the eye. Thus, the contrast-smoothing rule is applied, which adds 10% of each colour to the other, resulting in two grey values that still give the impression of being essentially black and white. However, both originally chosen colours will be added to the colour selection (i.e. red, blue, yellow, white and black = 5).

As the text body also contains links the TextModule assigns the colour for the link next. The aim is to distinguish both link and text from each other. The goal is to find among the depicted colours the one that provides the largest contrast to both the foreground and the background. In our example the best distance within the HSL colour space to black and white is provided by red. Thus, red will be assigned to the link. Note, the link colour comes with two values. One represents a non-visited link and the other a link that was already visited. The TextModule applies one of the split-item strategies for assigning two colours to a link. This rule diminishes the saturation of the established colour by 50%, which provides a sufficient difference to present the two states of a link.

Finally the ColourDesign unit has to assign a colour value to the "header". The reasoning is similar to the one described for the process of colouring links. The result is that header is assigned to blue because it provides a larger distance to white (background) than yellow.

At the end of the colour design process for the content area we have the following situation (text in brackets is added for better readability of this document):

Result_list:			
Background:	content_area1:	hsl[0,0,230]	(white-grey)
Foreground:	content_area1:		
	textmalink1:	hsl[0,0,25]	(black-grey)
	link:	hsl[0,255,128]	(red)
		hsl[0,128,128]	(0.5 red)
	header:	hsl[240,255,128]	(blue)

Colour assignment link_area2

The next area to be coloured is Link_area2, as described in Figure 4. As the process will use already established colour allocations, the ColourPicker takes the colour for link as given. The ImageModule considers images as being of colour various. This is a default within our current implementation, as the problem of matching the background with visual material such as images or a video area, are not fully implemented yet. Details about the approaches taken are described in section 5.

Thus, the final decision to be made is concerned with the allocation of the background colour of this area. As the colour needs to be distinguished from the neighbouring area (white) but has to allow the legibility of the links (red) the TextModule decides on yellow, mainly because

- the combination of blue as background and red as foreground result in a too small LV (0.178) to be acceptable and
- the contrast difference between yellow and white is big enough to allow a differentiation between areas ($Y_{white} - Y_{yellow} > 25$).

At the end of this part of the colour design process the system has made the following allocations:

Result_list:			
Background:	content_area1:	hsl[0,0,230]	(white-grey)
	link_area2:	hsl[60,255,128]	(yellow)
Foreground:	content_area1:		
	textmalink1:	hsl[0,0,25]	(black-grey)
	link:	hsl[0,255,128]	(red)
		hsl[0,128,128]	(0.5 red)
	header:	hsl[240,255,128]	(blue)
	link_area2:		
	link:	hsl[0,255,128]	(red)
		hsl[0,128,128]	(0.5 red)

Colour assignment link_area1
The final area to be designed is Link_area1, as described in Figure 4. Equivalent to the steps described in section 4.2 the system considers the colours for links set to red.

With the three not assigned colours, namely red, blue, and black, the TextModule is not in the position to resolve the general contrast constraints for background colouring applicable in the split complementary colour scheme. Hence, the ColourDesign unit backtracks and tries to apply another colour scheme to solve this problem. The choice is the analogue scheme, as it mainly addresses the combination of colours that are next to each other within the HSL colour space.

The first approach to assign blue to the background fails because the general rule applies that two dark colours (here blue and red) should not make up back- and foreground. The same applies to black.

The next option for the system is to find an area where

- the information unit link is used and
- a background colour is allocated that is within the neighbouring range of one of the colours within the analogue scheme.

As a result the system retrieves the allocations for the `link_area2`. However, the combination yellow (background) and red (foreground) cannot be used as no contrast between the two areas can be achieved (in fact the contrast is 0).

As the importance of link_area1 is ≤ 0.5 (see Figure 5) and both red and yellow are in the domain set, the ColourPicker retracts to a shortcut rule that allows the reverse application of colours under these circumstances, resulting in an allocation for red to the background and yellow to the foreground. As the importance of this area is ≤ 0.5, the ColourPicker rules out further functional evaluation. This is important, as the ColourDesign unit does not try to fill the still open 6^{th} colour-slot.

The end result of the colour design provides the allocations as can be seen on the next page:

The result list is returned as an answer to the request that was described earlier at the end of section 4.1. The transformation steps that generate the final presentation, e.g. generating a SMILE presentation, are described elsewhere [24]. Note, the established colour allocations will be used during the ongoing presentation as input for further alterations, as required.

5. Evaluation of our approach

The described process of colour design, the established rule sets and their theoretical foundation, all implemented in Eclipse Prolog and using the RDF-based domain ontology provided by the Rijksmuseum in Amsterdam, are best regarded as a platform that demonstrates the feasibility of our approach on the automated colour design within the generation of multimedia presentations. The colour unit of

Result_list:			
Background:	content_area1:	hsl[0,0,230]	(white-grey)
	link_area2:	hsl[60,255,128]	(yellow)
	link_area1:	hsl[0,255,128]	(red)
Foreground:	content_area1:		
	textmalink1:	hsl[0,0,25]	(black-grey)
	link:	hsl[0,255,128]	(red)
		hsl[0,128,128]	(0.5 red)
	header:	hsl[240,255,128]	(blue)
	link_area2:		
	link:	hsl[0,255,128]	(red)
		hsl[0,128,128]	(0.5 red)
	link_area1:		
	link:	hsl[60,255,128]	(yellow)
		hsl[60,128,128]	(0.5 yellow)

the design module is in the position to generate, depending on the constraint set, user adapted pages with overall acceptable colour representations.

We are aware of the fact that the current system needs further elaboration with respect to detail. Further work is required on such aspects as the relation on a micro-level between colour and links, colour and typography, colour and other visual information units, foreground and background colour, and more variations on how to choose between colour schemata.

For example, the relation between a coloured background and images as foreground has not yet been fully explored. The approach we intend to follow is to apply various colour descriptions of images, such as the average colour of an image based on the combination of histogram peaks or the average colour of the outer image boundary, as the basis for the background colouring process.

Another relevant problem is the expression of various states and functionalities of a link through colour. This is a by far more complex process than the design module is currently able to solve (for details see [14] and [25]). The most interesting question here is how to adapt our environment to work with an additional virtual colour space that contains those potential colours that might be presented at a later presentation stage because particular functionality of an information unit might not be required all the time.

In this context the question of colour ranges becomes interesting too. In our current approach we work with strict HSL values due to domain constraints. If the system were allowed to alter the range of given colours we would gain in flexibility but have to pay with a growing complexity in the constraint solving. Exploring these questions is part of our ongoing research.

The important question we would like to answer, however, is not so much directed towards the scalability of our approach, but rather what it means to facilitate the system with resources to synchronise the various efforts of balance on the diverse micro-level during the evolving presentation design process. Here we are in particular interested in the question how to allow the system to decide autonomously when to overrule a functional decision with a formal solution or the other way round because the result serves the current communication goal better.

We already introduce mechanisms to address this problem, such as

- adaptation of colour balance schemata based on functional or formal requirements
- importance weights allowing to swap between functional and formal presentation strategies if necessary
- providing sets of rules that increase the constraint space by applying aesthetic solutions to functional problems or the other way round (see for example the use of saturation variance in the determination of link colours).

These preliminary results, however, do not solve the problem per se but they focus our research on determining which level of autonomy can or should be achieved in automatically generated multimedia presentations. Further research is needed to determine how valid our hypothesis is that the automated generation of dynamic presentations can be improved by providing a balance between form and function.

6. Conclusion

In this paper we argue that multimedia presentation generation has to be able to balance the functional aspects of a presentation, that address the information needs of the user, and its aesthetic form. We demonstrated our approach using automatic colour design for which we integrate relevant aspects of colour theory. We also showed the ability of our system to change the emphasis between functional and aesthetical presentation techniques to solve presentation constraints. We did not provide a definition of the relative importance of form versus function, but sought to explore the roles of subjective elements in the generation process.

The description and its evaluation showed the feasibility of our approach on the automated generation of functional and aesthetically balanced multimedia presentations but also determined the open questions still to be solved, such as scalability of the approach by increasing constraint complexity due to larger presentation detail, or the synchronisation of various efforts to balance form and function on the diverse micro-level during the evolving presentation generation process.

Besides addressing these problems in our current research we are currently working on the problem of facilitating the system to decide autonomously when to overrule a functional decision with a formal solution or the other way round because the result would better supports the current communicational goal.

7. Acknowledgements

Part of the research described here was funded by the Dutch national ToKeN2000/CHIME project. The authors wish to thank their colleagues Jacco van Ossenbruggen and Lloyd Rutledge for useful discussions during the development of this work. The authors would also like to thank Kees van der Meer from the Technical University of Delft, the supervisor of the second author, for interesting insights into the process of physiological colour perception.

References

1. E. André, J. Müller, and T. Rist. *WIP/PPP: Knowledge-Based Methods for Fully Automated Multimedia Authoring*. London, UK, 1996.
2. J. Bateman, J. Kleinz, T. Kamps, and K. Reichenberger. Towards Constructive Text, Diagram, and Layout Generation for Information Presentation. *Computational Linguistics*, 27(3):409–449, September 2001.
3. D. Bauer and C. Cavonius. *Ergonomic aspects of visual display terminals*, chapter Improving the legibility of visual display units through contrast reversal, pages 137–142. Taylor and Francis, 1980.
4. 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.
5. P. Brusilovsky. Adaptive Hypermedia. *Journal on User Modeling and User-Adapted Interaction*, 11:87–110, 2001.
6. K. Czajka. Design of interactive and adaptive interfaces to exploit large media-based knowledge spaces in the domain of museums for the fine arts. Master's thesis, University of Applied Science Darmstadt, June 14, 2002.
7. A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc., 1999.
8. U. Eco. *A Theory of Semiotic*. The Macmillan Press, 1977.
9. M. M. G. Davenport. ConText: Towards the Evolving Documentary. In *ACM Multimedia '95, Proceedings*, pages 377–378, November 1995.
10. J. Geurts. Constraints for Multimedia Presentation Generation. Master's thesis, University of Amsterdam, 2002.
11. J. Itten. *The Art of Color*. John Wiley and Sons, Inc., 2002.
12. T. Kamps. *Diagram Design : A Constructive Theory*. Springer Verlag, 1999.
13. G. Kress and T. V. Leeuwen. *Reading Images: the grammar of visual design*. Routledge, 1996.
14. H. W. Lie, B. Bos, and R. Caillau. *Cascading Style Sheets: Designing for the Web (2nd Edition)*. Addison-Wesley Pub Co, 1999.
15. S. Mccloud. *Understanding Comics*. Kitchen Sink Press, 1994.

16. F. Nack and L. Hardman. Denotative and Connotative Semantics in Hypermedia: Proposal for a Semiotic-Aware Architecture. *New Review of Hypermedia and Multimedia*, 7:7–37, 2001.

17. F. Nack, M. Windhouwer, L. Hardman, E. Pauwels, and M. Huijberts. The Role of High-level and Low-level Features in Style-based Retrieval and Generation of Multimedia Presentations. *New Review of Hypermedia and Multimedia*, 7:39–65, 2001.

18. J. Nielsen. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2001.

19. L. Rutledge, J. van Ossenbruggen, L. Hardman, and D. C. 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.

20. A. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. Ontology-based Photo Annotation. *IEEE Intelligent Systems*, 16(3):66–74, May-June 2001.

21. A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval: the end of the early years. *IEEE trans.*, pages 1349–1380, 2000.

22. E. R. Tufte. *Envisioning Information*. Graphics Press, 1999.

23. E. R. Tufte. *Visual Explanations*. Graphics Press, 2000.

24. 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, ACM Press.

25. H. Weinreich, H. Obendorf, and W. Lamersdorf. The Look of the Link — Concepts for the User Interface of Extended Hyperlinks. In *Proceedings of the 12th ACM conference on Hypertext and Hypermedia (HT01)*, pages 19–28, University of Aarhus, Århus, Denmark, August 14-18, 2001.

26. L. Weitzman and K. Wittenburg. Automatic presentation of multimedia documents using relational grammars. In *Proceedings of the second ACM international conference on Multimedia '94*, pages 443–451, San Francisco, October 15 - 20, 1994.

27. R. Williams. *The Non-Designer's Design Book*. Peachpit Press, 1994.