Designing Graphical Elements for Cognitively Demanding Activities: An Account on Fine-Tuning for Colors

Gilles Tabart[1,3], Stéphane Conversy[1,3], Jean-Luc Vinot[2], and Sylvie Athènes[4]

[1] LII ENAC, 7, avenue Edouard Belin, BP 54005, 31055 Toulouse, Cedex 4, France
[2] DSNA, DTI R&D 7, avenue Edouard Belin, BP 54005, 31055 Toulouse, Cedex 4, France
[3] IHCS IRIT, 118, route de Narbonne, 31062 Toulouse Cedex 9, France
[4]EURISCO International, 23, avenue E. Belin, BP 44013 31028 Toulouse Cedex, France
tabart@cena.fr, stephane.conversy@enac.fr, vinot@cena.fr, sylvie.athenes@eurisco.org

Abstract. Interactive systems evolve: during their lifetime, new functions are added, and hardware or software parts are changed, which can impact graphical rendering. Tools and methods to design, justify, and validate user interfaces at the level of graphical rendering are still lacking. This not only hinders the design process, but can also lead to misinterpretation from users. This article is an account of our work as designers of colors for graphical elements. Though a number of tools support such design activities, we found that they were not suited for designing the subtle but important details of an interface used in cognitively demanding activities. We report the problems we encountered and solved during three design tasks. We then infer implications for designing tools and methods suitable to such graphical design activities.

Keywords: User interface, graphical rendering, graphical design, color design, design study, critical systems.

1 Introduction

Visualizations of rich graphical interactive systems are composed of a great amount of graphical elements. Perception of graphical elements is highly dependent on multiple interactions between visual dimensions such as color, area, shape etc. and display context such as type of screens and surrounding luminosity. Understanding these interactions involves multidisciplinary knowledge: psychophysics, human computer interaction, and graphical design. How can visualization designers make sure that they minimize the risk of confusion? How can they be sure that any modification done on a 20 years old system will not hinder the perception, and hence the activity, of the users? How to convince users and stakeholders? In general, how can they design, validate, check, assess, and justify their design?

This kind of questions has been addressed at the level of the design process for the functional core, with methods such as Rational Unified Process or with Design Rationale

T.C.N. Graham and P. Palanque (Eds.): DSVIS 2008, LNCS 5136, pp. 136–148, 2008.

tools [9], or at the level of code, using tools based on formal description of interaction, such as Petri Nets [1]. However, tools and methods to design, justify, and validate user interfaces at the level of graphical rendering are still lacking. A number of past studies addressed this problem, but their results did not quite apply to the specific kind of user interfaces we design: those that contain multiple, overlapping elements, the perception of which are very dependent on subtle details, and that users scrutinize during long periods of time in a demanding cognitive context. Good examples are the latest generation of jetliners, in which pilots interact with graphical elements on liquid crystal displays (LCD) to manage the flight, or air traffic controllers who rely mostly on radar views with multiple graphical elements, to space aircraft within safety distance. As these interactive systems are used in critical situation, the need for sensible, justified, and verified design is even more important.

In order to design such tools and methods, one must identify the relevant dimensions of the activity that they are supposed to support. This paper is a report of graphical design activities for interactive systems. We present our experience as designers during various design activities we conducted. We then discuss important considerations one has to take into account during such activities, or if one wants to design tools and methods to support it.

2 Related Work

Graphical design issues have been studied by organizations like W3C [16], FAA [7] and NASA [13]. They have established a batch of guidelines about UI graphical design and recommendations about common visual perception issues. Researchers in information visualization worked on efficient representations [5,17]. Graphical semiology introduced visual variables (size, value, color, granularity, orientation and shape) together with their ability to present nominative, ordered or quantitative data [2]. Brewer [3] proposed tools to help design harmonized color palettes for cartography visualizations. Lyons and Moretti analyzed current color design tools [11], and designed a tool for creating structured, harmonious color palettes [10]. We extensively used guidelines from NASA and Lyons & Moretti molecules approach. They help guide the design process, and help structure the colors used. However, NASA guidelines are short on precise guidelines with subtle but important rendering problems. In addition, the molecules tool does not provide much help for the kind of constraints and needs we had during the process.

3 Studies and Experience Feedback

In this section, we present three design tasks that we conducted. We redesigned interactive systems that support air traffic controllers. In order to understand the design process, we first set the context by briefly presenting air traffic control (ATC) and the three tasks we had to accomplish as designers. We then report on our experience.

3.1 Air Traffic Control Activity

All our tasks dealt with graphical design issues pertaining to the main French radar screen software used by the air traffic controllers. The software main goal is to display

three-dimensional aircraft positions as if seen from above. The air space is divided into "sectors": complex three-dimensional airspaces criss-crossed with various routes. Each sector is managed by a team of 2 controllers: the tactical controller, who monitors aircraft through the radar screen and give vocal orders to pilots through a radio link, and the planning controller who organizes flights arriving from neighboring sectors. Controllers rely on flight plans, requests by pilots, requests from other sectors, current weather and traffic conditions to manage the air traffic, making judgments about the most efficient and safe way for aircraft to proceed through the air space while keeping within safe distance from each other. Each controller faces a radar screen displaying the sector under his/her responsibility. Each aircraft is represented as an icon showing its current position and smaller icons showing a few of its past positions. The current position is linked to a label with the flight identifier, current speed and flight level. In accordance with the controller's preferred settings, each screen might have a different configuration (zoom level, pan, visible layers, etc).

In ATC, the graphical information displayed has a high level of criticality. A controller may hold the fates of several thousand people during his work shift and his judgment is based on well-established work practice, his experience, and his perception of the displayed information. Therefore, all information has to be coherently displayed, in a very accessible but not intrusive fashion in order to spare the cognitive resources.

3.2 Design Process

As the tasks are mostly concerned with designing colors, we present the approach we used in terms of color model, tools, and methods.

Color models, calibration, and tool
RGB is the color model used in graphic computer-cards for encoding color. RGB is based on additive syntheses of colors using 3 primaries: red, green and blue. Software developers often use this model to specify color. RGB is a "machine-based" model: it is difficult to manipulate, and hinders the structuring of color choices. Other color spaces, such as models proposed by the Commission internationale de l'éclairage (CIE, International Commission on Illumination) and specially CIE LCH(ab) are "human perception-based" model. We used the LCH color space for two reasons. As LCH is a mostly linear perceptual model, it allows predictable manipulations. Furthermore, the L (luminosity), C (saturation), and H (hue) dimensions are semantically known color dimensions which further structured design: it helps organize colors (and hence conceptual entities) with three mostly orthogonal dimensions. In the remaining of the paper, we call "color" the perceptual phenomena referring to a particular LCH or RGB combination (and not simply the hue). We applied a calibration process on each monitor we used, so as to minimize the effects of bad rendering chain settings. Furthermore, we defined a reference ICC profile [8], and used it while designing colors, so as to maximize consistency between design sessions.

During our tasks, we designed and used our own tool to choose and modify colors. The tool can import a set of colors, sort colors into group, and display them around a hue circle using the LCH model and the reference ICC profile. It also allows to express constraints with "molecules" of color [10], or to modify directly their hue, their level of saturation, or their level of luminosity (Fig. 1). We will not describe further this tool, as it is not the purpose of this paper and is only a draft of what should

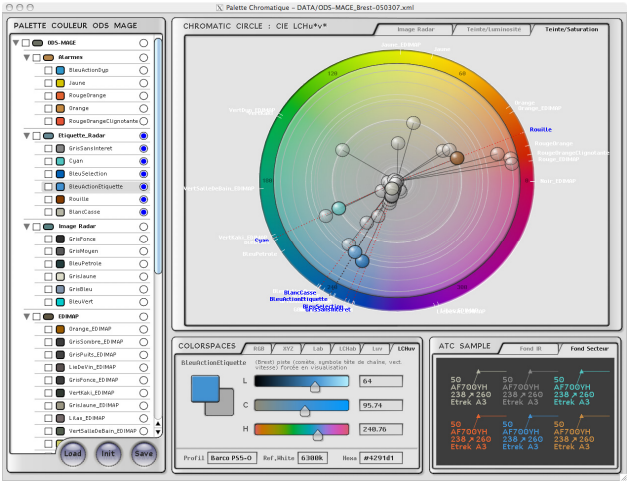

Fig. 1. The ad hoc tool with the color palette, color wheel, color spaces and color samples

become a genuine instrument. However, it helped us identify relevant aspects about the design activity and about desirable features of an efficient tool.

Context of the design

During the first task, we designed colors directly in the control room. We had to work on specific displays that were installed in control centers in order to design with real activity conditions in mind. In addition, we had to take the controllers' opinions into account and iterate with them to reach an agreement and validate our work. As previously said, a control position comprises two screens. We kept an image of the old configuration on one display and applied modifications to the other so that we could compare the results of the transformation and discuss them with the controllers. We also displayed the old configuration on the old CRT monitors to compare between color renderings. Using an actual configuration also allowed us to check if looking at the screen from different visual angles did not influence too much color perception.

The colors were then translated to RGB and inserted in the radar view configuration file, in which color names are matched with their RGB hexadecimal code, e.g. `(name "Orange") (value 0xd08c00)`. When drawing a graphical element, the software refers to colors by their name e.g. `ConflitEtiquette#N_Foreground: MC#Orange#NColorModel`. Using this indirection, designers can share the same color between different elements. For example, when an alarm has to be applied both to a radar track and to an information panel (Fig. 2}, a designer can tag these two elements with the named color *orange*. Thus, if the hexadecimal value of orange color is modified, all orange elements will be changed. The configuration scheme is a way for the designers to structure color-coding. As such, it makes the task of configuring the radar view easy, and enables the system to accommodate unexpected changes or important security fixes. For the two other tasks, we worked on our computer on which we imported the palette to be changed.

Fig. 2. Two elements: same color code but not identically perceived

3.3 Design Activities Study

Our team includes a graphic designer, an experimental psychologist, and two HCI specialists. The tasks we present are real-world tasks: they are part of an industrial process, as changing such systems must follow precise steps. We were then constrained in the amount of modifications we were able to recommend.

First task: updating a global color design
Our task was to adapt the color settings of the main radar view software. presents the interface: the control panel on the left side, the main radar view in the middle, and the flight lists on the right side. The left panel present manifold options for choosing pan and zoom level or slices/layers of the sector to displayed, for example. On the main view, different areas are represented in the background with different colors, while 1 pixel wide lines represent flight routes. Flights current and past positions are represented by 3 to 5 pixel wide squares. A tag with textual information about the flight (callsign, level, speed etc.) is linked to the shape with a 1 pixel wide line. The right panel is reserved for alarms and a list of flights. Selected flight information is displayed at the bottom of this panel.

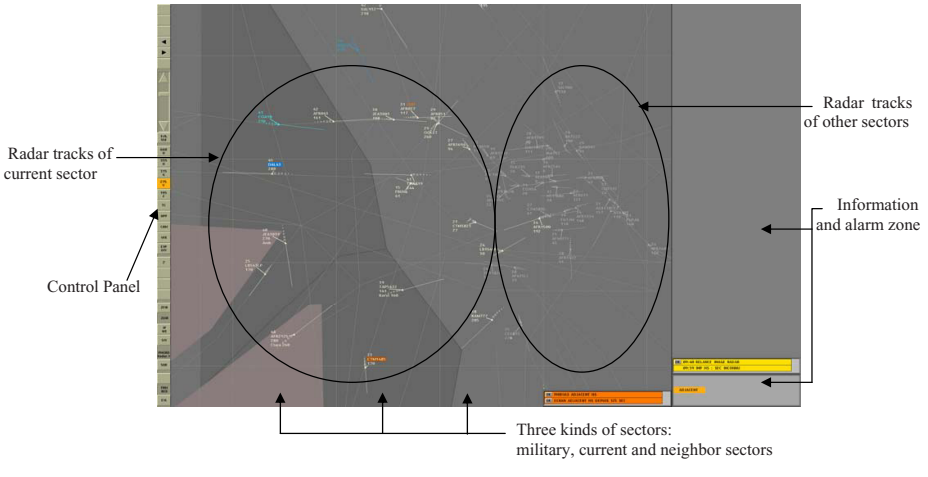

Fig. 3. The main visualization for air traffic control

We had to adapt the color settings because the system evolved both technologically and functionally. The CRT displays on which the application runs have been replaced with LCD displays. Color rendering on LCD differs from CRT: they are more saturated, while the beam is narrower. The difference in rendering completely changes the overall appearance of the visualization. Furthermore, the CRT displays are square while LCD displays have a 16:10 ratio, which changes the proportional amount of the different graphical elements. Beside hardware evolution, the activity regularly evolves, with the addition of new functionalities, new control procedures or new sector arrangements. This results in stacked modification, with no real global design.

On the one hand, we had to hold the perceived color constant while moving from CRT and LCD display. On the other hand, we had to harmonize color palettes between configurations from five air traffic control centers. Each one has its own color palette, due among others to traffic particularities. This specific task may seem trivial (changing colors); but to achieve it, we had to modify almost all colors of the application, and a lot of questions and problems were raised.

Second task: organizing flights into categories

The second task was to add new colors to an existing color palette. This requirement came from a new need in approach control activity. Controllers doing "approach control" regulate air traffic around airport areas. They needed to distinguish three categories of flights around Paris, those concerning Orly airport, those concerning Roissy airport, and in transit flights. They also needed to separate flights into two flows, e.g. Orly or "Orly-associated" airports. Together with users, a team of engineers had previously designed and installed a palette with three named colors ("green", "pink", "blue"). We had to harmonize the palette, while keeping identifiable colors.

Third task: redesign of an interface

The third task consisted in the entire redesign of the prototype of a future radar view. We were less constraint by historical constraints, and freer to test original configurations. Even though this task is still in progress, it has brought some valuable information.

3.4 Design Accomplishment and Teaching

This section presents a description of our work as designers. The description is organized around the similar issues we encountered during the three tasks. For each issue, we describe our goals, the constraints driving our choices and the solutions we eventually chose.

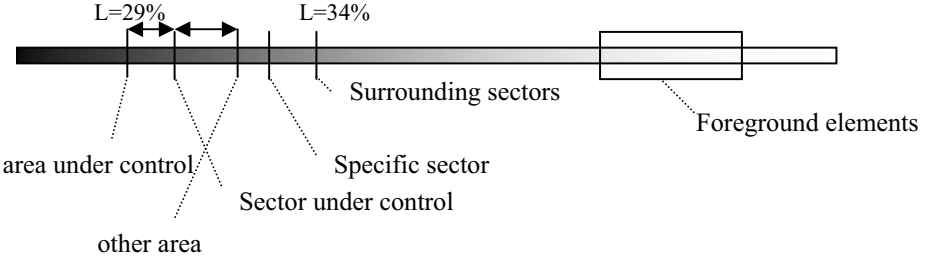

Fig. 4. Representation of the graphical elements' luminosity

Information visibility, luminosity and background

The first issue concerns information visibility (am I able to see an entity?), discriminability (am I able to differentiate between two entities?), identifiability (am I able to identify an entity among a known set of entities?) and legibility (how easily read is the text?). We first worked on luminosity. All the colors we designed are achromatic: there is no perceptible hue information. In the LHC model, it is implemented by setting color saturation to null.

Luminosity difference enables separation of juxtaposed and layered objects. Sectors are large, uniform surface juxtaposed on the background. Controllers must discriminate and identify them so as to see if a flight is about to enter or leave their controlled sector. Layered objects include sectors (background), routes, beacons, and flights (foreground). Routes and beacons must be visible, while flights must pop out and be legible.

We first designed sectors luminosities, since they end up acting as the background for most objects, and foreground colors can only be set according to the background. Fig. 4 shows the resulting distribution of the gray luminosities of the different sectors: a gray for the controlled sector, another gray for the surrounding sectors, and a last gray for a special area. The sector under control is the darkest: this sector is the most important for the controllers, and flights should be maximally visible here. The surrounding sectors are thus lighter. In bi-layered sector, controllers have to distinguish between two areas in the sector under control. We spread apart two grays around the gray of the sector under control. The second area luminosity is farther from the controlled area luminosity than the surrounding sector luminosity because it is more important to identify the controlled area than the others. However, the second layer gray must also be different enough from the surrounding sector gray. The four grays are very close in luminosity (range L=5%).

This example highlighted a problem with the possibilities of choosing a color in a relatively small range. This issue comes from imposed constraints about gray and from the fact that the 8bit-per-channel RGB color space used by the system is poor; it does not contain enough values to express all the shades of a color range. On a machine color model, grays are made by mixing equally R, G and B. Thus, between white and black there are only 254 possible grays. Furthermore, we precisely tuned the set of grays by incrementing or decrementing RGB values one by one, as the conversion between LCH and RGB was not precise enough. We had to work with the system color space instead of the perceptual color space.

Some graphical objects must be more than simply visible. For example, alarms must grab attention when they are displayed. Even though other visual dimensions such as animation help grab the user's attention, we chose to separate them from background or others grays elements with one additional color dimension, the saturation. Indeed, alarms have specific hues that reflect the emergency level. We gave alarms object a high level of saturation to accentuate the discrimination from background objects.

Some areas, such as military zone, can be considered as an "alarming" area. To differentiate them from civil area and give them an alarming appearance, we decided to slightly color them with a reddish gray.

Confidence and comfort

The global image must be harmonious: even if it is difficult to formally quantify it, the satisfaction resulting from using a good-looking image nevertheless matters. Moreover, it improves the controllers' confidence into the system. For example, the planning controller typically configures the zoom level to have a global view of future flights arriving in his sector. However, for narrower sectors, a lot of gray flights not under this controller's responsibility become visible on both sides of the current sector, because the new screen has a 16/10 ratio. These flights tend to raise the object density of the image too much. The global scene perception is spoiled and controllers are less confident in their ability to analyze the image. This resulted in uncomfortable situations, where controllers were afraid to miss an important event, and felt obliged to constantly check the image. This issue never arose with square screens.

Global comfort of the scene is also an issue when designing alarms. On the one hand, alarms must interrupt the user and be remembered, so they are intrinsically not comfortable. On the other hand, if an alarm comes to persist on the screen (e.g. the controllers have seen the alarm but they have to finish some other actions first, or because no action allows the controllers to get rid of them), it should not hinder the controllers'activity. In order to increase comfort with such persistent alarms, we had to decrease their saturation level, and make them less "flashy".

Categorizing and ordering graphical objects

One important point in the design process is categorizing and ordering objects. In the second task, we had to group flights in categories and flows. The three main categories had color hues that had been decided in a past design: green, pink and blue. The controllers proposed a separation into two sub-flows. They designed a solution by using various color dimensions, which resulted in heterogeneous colors. We worked with the LCH color space in order to homogenize the design choices. We set apart the three hue angles by 120 degrees and we distributed the sub-flows around each main hue. In order to see the results and finely tune the design, we built an image containing 6 examples of the exact shapes to be colored. We embedded this image in the tools we used, as can be seen at the bottom left of Fig. 1.

We tried to match the conceptual hierarchy with the perceptual hierarchy. For example, the two kinds of flights displayed match their relative importance for the controller. Flights that the controller has currently in his charge are represented in a bright color and others, controlled by neighboring sectors, are in a darker gray.

Alarms are also graduated: according to their importance, they have a certain hue and saturation level. We had to conform to alarm hierarchy and cultural color habits (such as red for danger).

Surface does matter: perception and software design limitation

We observed that surface influences perception: according to the surface on which a color is applied, the perception of this color is different. For example, we designed a color for small/medium size military sectors. The color is a gray with a hint of red (which name is "lie de vin"). We later used the color palette in another control center, embedding a larger military zone. When this same color was applied to this surface, the reddish gray seemed too saturated (i.e. too red). We had to decrease the saturation in order to make sectors look grayer when they are big, but still keep a distinctly reddish nuance when they are smaller. Fig. 2 shows a second example with two elements

displayed with the same alarm color code. The first element is a 1 pixel wide text; the second one is the background of an information panel. Due to surface and/or pixel arrangement, the same orange color applied to both these graphical elements does not appear to be the same when a text or a background.

We have been able to accommodate the problem in the first example with a single color. But it proved to be impossible in the second example: we had to design two colors. It follows that the configuration file is not as structured any longer: if one decides to change the orange in the future, one has to change two colors instead of one. This matter is linked to the use of the indirect method for coding colors that we presented above. With a simple indirect color-coding scheme, there is no means to accommodate for differences in color perception due to the amount of surface. This example shows that the coding method can hinder the controller's activity: there is a risk that a color is not identified as corresponding to a particular state, or that two elements cannot be associated through their color.

Another issue concerns very small elements like one pixel-wide lines or glyphs. When we applied low saturated colors to such elements, their hues did not come out very well. These observations can be explained by the fact that, with this kind of small elements, some pixels may end up being isolated on a background color. They are thus "eaten" by background colors and lose some of their properties [14].

Human subjectivity: naming color, acceptability opinions.
The next issue is about color perception properties. In the LCH color space we used to organize colors. L, C and H dimensions are supposed to be orthogonal, i.e. if a designer changes a color along a single dimension, the perception of the other dimensions should not change. However, if some colors can be modified in saturation or luminosity without losing their essence (think of light or dark blue), some colors cannot be easily modified without impacting perception of hue. Red for example is identified as such only for a medium luminosity level, otherwise it is identified as ochre/brown with low luminosity, and pink with high luminosity. We experienced this problem when we tried to lower the saturation of alarms, because they were too sharp: when we applied the modification, the element was not perceived as red any longer, but as ochre, which completely disabled its identification as an alarm. We had to change both saturation and hue to keep a color identifiable as red. This phenomenon shows that colors cannot be modified automatically, or at least without precaution.

A related issue concerns the naming of colors. In their activity, controllers use color name so as to identify graphical elements. For example, they use the name of the color to refer to a particular flight status instead of referring to the status itself, as in "can you check the bathroom green airway?". In such circumstances, if a color has to be modified, it must be kept recognizable and identified as the same named color to accommodate historical use.

Human subjectivity is also an issue. For example, there is a large diversity of opinions about the saturation thresholds between a comfortable color and an uncomfortable one. This depends on human perception and sensation but also on the hue value. Furthermore, opinions vary in time, because of habituation or fatigue: the same person can disagree with a design choice at some time, and then agree with it later.

Display context

The perception of colors is dependent on the type of monitor. Nowadays, controllers use multiple screens: a radar view, but also a list of flights view, displayed on an almost horizontal screen under the radar view. The colors used on this screen must match the colors used on the radar view, as some of them allow elements to be grouped. However, even after calibration, it proved to be difficult to get exactly the same colors on both screen. For example, there were situations where up to four different blues were displayed on the screen. All four colors were very close in terms of LCH. The problem was worse when we took into account the second screen: we had to spread apart further the hue of each blue so as to allow recognition and association within the two screens. However, we did not explore further the problem, as our assignment was only to work on the main radar view. Fortunately, there are other contextual information that allow the controllers to discriminate between the status reflected by the colors. Nevertheless, this problem should not be overlooked.

The temperature of the display also influences perception. For example, we changed the saturation and hue of a slightly colored gray from C=3% to C=2.92% and from H=156° to H=206° to make the values coherent with other colors. We did it offline, and to our surprise, when users saw the new result, they said it was too colored. We learned three things. First, a 50° modification of hue with saturation as low as 3% is noticeable (and hindering). Second, offline modifications are harmful, even if based on sensible reflection made by an experienced graphical designer. Third, this is another example that shows that specifying a gray with R=G=B is harmful, because it does not take into account every parameter that influences color rendering and perception.

A lasting, iterative activity

Even though it is possible to roughly describe the workflow we used (design luminosity first, then saturation, then small objects), the actual activity was done in an iterative manner. Besides, as any design activity, the tasks took us some time to accomplish.

First, we had to fix problems introduced by our own new settings: it was difficult to know the impact of a modification, to remember the dependencies between constraints, and to check every possible problem all along the process. Furthermore, we had to explore several configurations, going back and forth between intermediate solutions, which was not an easy task to do with the tools we were using. Besides, designing needs maturation and understanding of the context, for both the designers and the users. For example, designing the right warning orange required the designer to really integrate the conditions of apparition and the context of use of such orange. Regular discussions around the examples and the tools really helped designers and users to achieve a successful result. Finally, designing a color palette is highly subjective. This is not to say that users do not know what they want, but diversity between users, fatigue due to hours of design, changing context conditions etc. make the design subject to unexpected modifications, at best local, at worst global. As designers, we had to react accordingly. For example, in the first task, we worked with users so as to get their feedback and fix problems as soon as possible. After one day of designing, we had a new palette that was satisfying to both the users and the designers. When we came back the following day, the users found that the new configuration made the image too uncomfortable because it was too luminous. We had to lower the luminosity of each color one by one to fix this problem.

4 Implication for Design

In this section, we sum up the experience we gained during our tasks. We identify the relevant dimensions to take care of, when designing tools and methods to support graphical elements design.

Design with actual, controllable examples
Actual color design tools allow control of color dimensions and checking of the results on a square displaying the resulting color [11]. However, to really design a color, we had to configure the application with the newly designed color, and check it in an actual scene, in our case a radar view. This takes time and prevents an efficient iteration loop. In our ad-hoc tools, we tried to solve this problem by embedding a sample of the flights that were supposed to be organized in flows and sub-flows. This allowed us not only to check the results, but also to completely change the way we handled designing, as we could test multiple solutions quickly, and adjust swiftly and precisely each color. In fact, color-design tools should use an imaging model, not a color model as they do today [6].

Design with multiple examples at once
An object may be involved in multiple situations. For example, when designing the color of a flight, we had to take into account all the backgrounds over which it could be displayed. This forced us to go back and forth between different configurations of the application. Thus, a color tool should not only embed controllable examples, but it should also allow an easy switching between examples (either by juxtaposing them, or progressively disclosing them).

The global scene is important
We highlighted the importance of designing on real scene samples. However, it is important to keep in mind that these samples are only parts of a global graphical scene. All individual elements build up the perception of the global scene, and global rendering is the only mean to check the global comfort of the UI. Inversely, the global scene influences the perception of a single element. In order to experience these interactions, a designer must work on real scenes, and not just approximate or simplistic ones.

Foster explorative design
Making a design successful requires exploring and comparing alternative solutions. Our tools hinder exploration, as they require to save the configuration and to relaunch the application, to compare with early designs. Fortunately, we could use two screens to compare our designs with the configuration currently in use in control centers: this scheme must be generalized to any intermediary configuration, whether it concerns a single element, or a set of elements. Sideviews is an example of such style of design [15].

Foster constraints expression
We also noticed the importance of expressing constraints and reifying them. During the design phases, remembering all constraints is difficult. Actually, color molecules implement a kind of constraints, enforced with graphical interactions [10]. Such graphical constraints would have made group settings easier: it would have allowed us to lower

the luminosities of several elements at once. In addition, constraints expressed with formulas would check that a change of a parameter does not violate a previously fulfilled constraint. However it is sometimes difficult to express constraints, either graphically, or even prosaically: the constraints between the sectors gray are complex, and a tool that would enforce them would be too cumbersome to use.

Expressing and structuring colors
The LCH model, together with calibrated displays, is the right tool to express color. The LCH color space allows for predictable manipulations and structured design. However, when designing very precise values, the resolution of the machine color model hinders tuning. We were obliged to tune the final RGB values to find the right set of gray level for background. A right tool would facilitate expressing and manipulating the structured relationships between colors while at the same time allowing small adaptations using the final color model.

Even if based on the perceptual system, the LCH model is not perfect. The dimensions are mostly orthogonal, but not perfectly orthogonal. The LCH model does not allow for modifications that would guarantee that a named color is still perceived as the same. Color expression and constraints must take into account the specificities of named colors, and provide suitable interaction to help designers manipulate them.

Not just about design: integrate all purposes
During our design activities, we found that our task was not only to reach a final palette, but also to help users express their needs, to help us justify our choices and convince users, and to help accept the new settings. In the justification phase, by giving quantitative arguments, constraints would enable to argument for the choice eventually made. A list of constraints would also act as a proof that criterions required by a specification document are respected, and would help define an experimental plan to experimentally assess the design choices [12].

A tool to help designing should not be used only once, but also as an instrument that would accompany the configured system all along its lifetime. Actually, the tool itself would play the role of the configuration file of the target application. Such a tool would reify the design choices and justifications and help designers understand and respect past constraints that led to a particular design. As such, it would serve as a design rationale tool, and would extend the notion of active design documents [9, 4].

5 Conclusion and Perspectives

In this paper, we reported about our experience as designers of colors for graphical elements. We showed that interaction between visual dimensions and display context makes the design very dependent on small details. We reported how we handled various technical, cultural, and perceptual constraints. Based on this experience, we devised a set of implications for designing future instruments to support graphical design activities.

Notwithstanding the specificity of cognitively demanding ATC activities where even the smallest detail is important, the set of implications for design we devised should be of interest in other contexts. For example, web design requires defining a palette, but for a design to be coherent and harmonious, the same concerns that we expressed here should be taken into account. The features of the tool we envision

would be of the same usefulness, whether as a design tool, as a design rationale tool, or as an evaluation tool.

References

1. Barboni, E., Navarre, D., Palanque, P., Bazalgette, D.: PetShop: A Model-Based Tool for the Formal Modelling and Simulation of Interactive Safety Critical Embedded Systems. In: Proceedings of HCI aero conference, Seatle, USA (2006)
2. Bertin, J.: Sémiologie graphique: Les diagrammes -Les réseaux - Les cartes (Broché) 1070 pages Editeur: Editions de l'Ecole des Hautes Etudes en Sciences janvier 31 (1999)
3. Brewer, C.A.: Color Use Guidelines for Mappingand Visualization. In: MacEachren, A.M., Taylor, D.R.F. (eds.) Visualization in Modern Cartography, ch. 7, pp. 123–147. Elsevier Science, Tarrytown (1994)
4. Boy, G.A.: Active design documents. In: proceedings of the 2nd Conf. on Designing interactive Systems: Processes, Practices, Methods, and Techniques. Amsterdam (1997)
5. Card, S., Mackinlay, J., Shneiderman, B.: Information Visualization Readings in Information Visualization:Using Vision to Think, pp. 1–34. Morgan Kaufman, San Francisco (1998)
6. A Colour Appearance Model for Colour Management Systems: CIE CAM 2002, CIE 159, 2004 (2004)
7. Federal Aviation Administration (FAA). Human factors design standard (HFDS), HF-STD-001 (March 2008), http://hf.tc.faa.gov/hfds
8. Specification ICC.1:2004-10 (Profile version 4.2.0.0) Image technology colour management : Architecture, profile format, and data structure, International Color Consortium (2004)
9. Lacaze, X., Palanque, P., Barboni, E., Navarre, D.: Design Rationale for Increasing Profitability of Interactive Systems Development., Rationale Management in Software Engineering, pp.182-197 (2005)
10. Lyons, P., Moretti, G.: Incorporating Groups into a Mathematical Model of Color Harmony for Generating Color Schemes for Computer Interfaces. In: Proceedings of the 2005 IEEE conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, 18-20 July 2005, pp. 80–85 (2005)
11. Lyons, P., Moretti, G.: Nine tools for generating Harmonious Colour Shemes. In: Masoodian, M., Jones, S., Rogers, B. (eds.) APCHI 2004. LNCS, vol. 3101. Springer, Heidelberg (2004)
12. Mackay, E.W., Appert, C., Beaudouin-Lafon, M., Chapuis, O., Du, Y., Fekete, J.D., Guiard, Y.: Touchstone: exploratory design of experiments. In: Conference on Human Factors in Computing Systems, pp. 1425–1434 (2007)
13. NASA Color Usage (2004) (March 2008), http://colorusage.arc.nasa.gov
14. Tabart, G., Athènes, S., Conversy, S., Vinot, J.L., Effets des Paramètres Graphiques sur la Perception Visuelle : Expérimentations sur la Forme, la Surface, l'Orientation des Objets et la Définition des Ecrans. In: IHM 2007 (2007)
15. Terry, M., Mynatt, D.E.: Supporting experimentation with Side-Views. Communications of the ACM 45(45), 1006–1008 (2002)
16. Techniques For Accessibility Evaluation And Repair Tools, W3C workink draft (2000), http://www.w3.org/TR/AERT#color-contrast
17. Ware, C.: Information Visualization: Perception for Design, December 2004, 435 pages, 2nd edn. Morgan Kaufmann, San Francisco (2004)