

Outils d'aide à la conception de rendus graphiques

Gilles Tabart ^{*,**}

* ENAC
7, av. Edouard Belin
31400, Toulouse, France
tabart@irit.fr

Stéphane Conversy ^{*,**}

** IHCS-IRIT
118 Route de Narbonne,
31400 Toulouse, France
stephane.conversy@enac.fr

Jean-Luc Vinot ^{**,**}

DSNA/DTI/R&D
7, av. Edouard Belin
31400, Toulouse, France
{vinot, athenes}@cena.fr

Sylvie Athènes ^{**,**}

RESUME

Cet article décrit un outil de spécification de propriétés visuelles pour des interfaces graphiques complexes. À partir d'exemples réels de scènes graphiques, nos outils proposent des interactions pour manipuler, structurer et contraindre les propriétés graphiques. Les propriétés graphiques contrôlées correspondent aux trois dimensions de la couleur : luminosité, saturation et teinte. Par rapport aux outils existants, nos outils permettent d'ajuster et de vérifier de façon plus précise et efficace le rendu des composants graphiques. Nous montrons les apports de nos outils par des cas d'utilisation appliqués à des problèmes concrets de paramétrage de rendus graphiques.

MOTS CLES : conception, rendu graphique, propriétés visuelles, couleurs, contraintes, outils.

ABSTRACT

This paper describes tools for the design and assessment of graphical rendering of complex interactive systems interfaces. More precisely, the tools help controlling graphical properties of actual sample scenes. The graphical properties pertain to the three dimensions of color: luminosity, chroma and hue. Our tools enable designers to set and check the graphical rendering properties in a more accurate and efficient way. We illustrate the benefits of our tools by giving actual scenarios of graphical rendering parameterization.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 User Interfaces.

GENERAL TERMS: .Design, Performance.

KEYWORDS: design, graphical rendering, color, visual properties, constraints, tools.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IHM 2009, 13-16 Octobre 2009, Grenoble, France
Copyright 2009 ACM 978-1-60558-461-4/09/10 ...\$5.00.

INTRODUCTION

Les interfaces graphiques des systèmes interactifs dont nous traitons dans cet article sont complexes et composées d'une multitude d'éléments visuels. Par exemple l'interface graphique d'un système de gestion du contrôle aérien (ATC) représente l'espace géographique d'un secteur de contrôle, traversé par des dizaines d'aéronefs, ainsi que les informations de trafic nécessaires aux contrôleurs pour accomplir leur tâche. Avec une tendance continue à l'augmentation de la densité des informations affichées, les éléments graphiques utilisés pour afficher ces informations sont portés par des formes de surface de plus en plus réduite. La multiplication des variables visuelles ou du nombre de leurs valeurs mises en œuvre engendre un nombre grandissant d'interactions entre éléments graphiques. Ces interactions visuelles sont de plus en plus difficiles à maîtriser par les équipes en charge du cycle de développement et les recommandations générales ne suffisent plus aux concepteurs d'interfaces. Ces derniers ne sont plus capables de garder à l'esprit l'ensemble des contraintes nécessaires à la réalisation d'un rendu cohérent, à la fois avec les attentes des utilisateurs, et avec les règles de design de rendu visuel.

La conception du rendu graphique des interfaces est une phase de conception nécessaire à la réalisation d'un système interactif correct (Figure 1). L'ignorer peut conduire à la conception et l'utilisation de systèmes ne respectant pas les propriétés d'observabilité et d'honnêteté. Alors que la conception graphique de l'IHM a pour but de construire des représentations visuelles adaptées à l'activité, la conception du rendu graphique de l'interface se préoccupe de la qualité de restitution des composants symboliques dans l'image IHM réalisée : bonne restitution perceptive (contrastes, interactions) des objets de visualisation présentés par le système graphique et efficacité de la restitution sémantique des signifiants graphiques implémentés. Cette qualité de restitution, sémiologique et sémantique, dépend tout particulièrement du choix et de la composition des objets graphiques à implémenter, de la compréhension de leurs propriétés perceptives et de leurs interactions ainsi que de l'exploration et du réglage fin des variables visuelles utilisées.

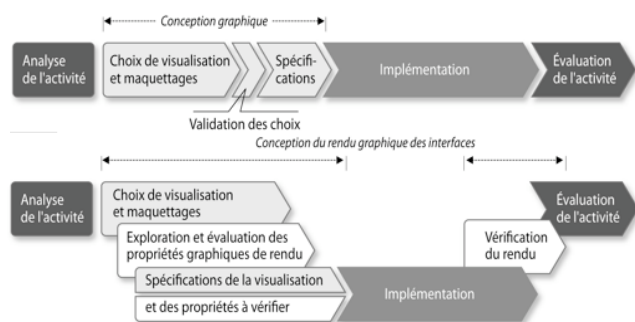


Figure 1 : Conception graphique classique et conception du rendu graphique

À l'image de la chaîne graphique des industries graphiques, la chaîne graphique de rendu désigne l'ensemble des disciplines (spécialistes et compétences) qui interviennent de la conception à la vérification d'un produit d'interface graphique. L'instrumentation (méthodes et outils) de cette chaîne graphique de rendu a pour but d'aider et de rendre efficace la conception graphique de l'interface. Intégrant les savoir-faire de plusieurs disciplines, elle inclut l'exploration de solutions possibles (design), l'intégration de connaissances sur les capacités humaines et l'évaluation expérimentale (psychologie expérimentale), l'influence des technologies matérielles et logicielles (informatique), et le processus de conception lui-même (design rationnelle, maintenance, informatique).

Cet article présente un outil faisant partie de la chaîne graphique de rendu. L'outil présenté permet la spécification des propriétés graphiques comme la couleur. Après avoir présenté l'état de l'art du domaine, nous expliquons le résultat d'un travail d'interview de spécialiste du design : ces résultats sont traduits en une liste de caractéristiques souhaitables pour un tel outil. Nous décrivons celui que nous avons conçu, en le mettant en scène au travers de scénarios d'utilisation issus d'exemples de problèmes réels.

ETAT DE L'ART

Il existe plusieurs types de travaux liés à l'aide à la conception graphique d'applications interactives et, dans la plupart d'entre eux, les concepteurs d'outils dédiés à la couleur utilisent les espaces colorimétriques de la CIE (Commission internationale de l'éclairage). En effet, en plus des espaces classiques de couleur RGB permettant d'adresser les couleurs des périphériques d'affichage en code « machine », il existe des espaces construits pour correspondre au système de la perception visuelle humaine et indépendants des matériels, comme les espaces CIE Lab ou CIE Luv [8] et leurs variantes LCHab et LCHuv. Dans les systèmes LCH, définissant la couleur selon 3 dimensions : Luminosité, Chroma (saturation), et Hue (teinte), les dimensions L, C et H sont orthogonales et perceptuellement uniformes.

Dans leurs travaux, Lyons et Moretti ont comparé différents types d'outils de sélection de la couleur [15]. Ils proposent aux programmeurs d'interfaces des outils

pour spécifier les couleurs d'une interface en se basant sur des « molécules harmoniques » [16], créées de manière automatique en fonction de la « distinguabilité » requise et de la taille des composants à colorer. Du fait de leur « automatisation », ces outils ne laissent cependant pas toute liberté à un utilisateur de contrôler finement le choix des couleurs.

La NASA propose un outil pour choisir les couleurs d'interfaces complexes [21]. Cet outil, basé sur un modèle LCH, propose des matrices permettant de prévisualiser le résultat des choix de couleurs fond/forme ainsi qu'un système de comparaison des luminosités des couleurs choisies. Cependant les couleurs ne sont pas adressables finement (uniquement par paliers trop importants) et l'utilisateur ne peut pas modifier la taille et la position des exemples (seulement leur forme). De plus, une fois les couleurs choisies, il n'existe plus aucun lien entre elles, ce qui contraint le concepteur à manipuler les couleurs séquentiellement et individuellement.

On trouve sur le web des outils¹ destinés aux designers et qui reprennent le principe d'harmonie entre teintes du « cercle chromatique » décrit par exemple par Itten [12]. L'outil d'Adobe propose de créer une palette allant jusqu'à 5 couleurs manipulables par des ronds colorés placés sur un cercle chromatique. Le designer peut lier ces ronds à différentes structures harmoniques prédéfinies et les faire varier. Les coordonnées des couleurs sont affichées dans différents espaces colorimétriques.

Des travaux de recoloration d'images [9] ou de palette interactive pour les artistes [19] s'appuient aussi sur ces principes d'harmonie des teintes. Barbara J. Meier a proposé de résoudre la problématique de choix des couleurs pour les interfaces utilisateurs à l'aide d'un système expert [18] se basant sur des règles reconnues de design des couleurs. Le choix des couleurs se fait après avoir renseigné le système sur la nature des éléments devant apparaître dans l'interface ainsi que sur les interactions (spatiales et temporelles) entre ces éléments. Cette approche plutôt « Intelligence Artificielle » est intéressante car des problématiques communes aux nôtres sont soulevées dans les phases de construction du système expert. Cependant cette approche conduit à des résultats trop approximatifs par rapport au paramétrage fin nécessité par des interfaces critiques.

PHASE D'ANALYSE

Durant les activités de conception d'interfaces critiques, ou d'interventions « correctrices » dans lesquelles nous avons été impliqués par le passé, nous avons mené des phases de conception de rendu. Nous avons peu à peu inventé des méthodes et conçu des outils ad hoc pour

¹www.kuler.adobe.com ou colorshemedesigner.com

structurer cette conception de rendu. À partir de cette analyse, nous avons tiré des enseignements sur les services qu'une méthode instrumentée doit fournir pour mener efficacement l'activité de conception de rendu [22]. Cette section résume les enseignements les plus pertinents par rapports aux outils présentés dans l'article présent.

Concevoir avec des exemples réels et contrôlables

Dans la plupart des outils de conception graphique proposant de choisir et régler des couleurs, ces choix sont faits hors contexte sur des formes ni choisies ni modifiables. Il en résulte souvent des couleurs inadéquates lorsqu'elles sont transposées sur le système final. Un système adapté doit permettre l'inclusion d'exemples réels, dont les variables visuelles sont mises à jour (en cours d'interaction) dès lors que le concepteur en modifie les caractéristiques.

Encourager l'exploration

Afin de maximiser la qualité de la conception, il est nécessaire d'explorer et de comparer des solutions alternatives sans ralentir le processus en passant pas des changements de configuration coûteux. Ainsi il faut permettre une exploration aisée, qu'elle concerne un élément ou plusieurs, en s'appuyant par exemple sur les principes de « chemin parallèle d'exploration » [23].

Encourager l'expression des contraintes

Nous avons constaté l'importance d'exprimer et de réifier les contraintes qui sous-tendent la conception. En effet, il est difficile de se rappeler des contraintes pendant les phases d'exploration. Des contraintes exprimées de façon mathématique pourraient vérifier que des modifications de valeurs d'une propriété ne violent pas une contrainte respectée auparavant. Cependant, il est parfois difficile d'exprimer des contraintes, que ce soit de façon graphique, formelle ou même prosaïque... Dans ce cas l'expression des contraintes serait plutôt de conserver une notation textuelle du problème et des choix réalisés.

Exprimer et structurer les couleurs

Comme nous l'avons dit dans l'état de l'art, le modèle LCHab, associé à des moniteurs calibrés [11], semble être un bon outil pour exprimer les couleurs. Cet espace de couleur permet d'effectuer des manipulations dont les résultats sont perceptuellement prévisibles. Il permet de plus de concevoir des familles ou des hiérarchies de couleur selon une des dimensions L, C, ou H. Cependant, le modèle a ses limites. Par exemple, le gamut du modèle de couleur du système de rendu (RGB) ne permet pas d'atteindre toutes les couleurs proposées par LCHab ni de les régler finement. Il est donc nécessaire de disposer d'un outil de réglage fin utilisant et visualisant les limites du modèle d'affichage final. A l'opposé, certaines manipulations de valeur conduisent à des résultats surprenants dus à des imperfections du modèle. Il peut par exemple survenir des déviations de teinte lors de la

modification de la luminosité de certaines couleurs. Un outil approprié doit donc fournir plusieurs manières d'exprimer les couleurs et les contraintes associées, notamment pour tenir compte des couleurs nommées.

Instrumenter l'ensemble de l'activité de conception

Nous avons observé que notre activité ne consistait pas seulement à élaborer une configuration finale de variables visuelles. Ainsi, c'est l'explicitation et l'expression des contraintes qui a permis aux utilisateurs de redéfinir leur problème, de discuter des solutions, et de favoriser leur acceptation. Un outil adapté ne doit donc pas seulement instrumenter l'activité de conception, mais aussi celle de *design rationale* [4] [13].

CONCEVOIR UN OUTIL DE CONCEPTION

À partir des constatations faites dans la partie précédente, nous avons entrepris la réalisation d'un outil de support à la conception et à la vérification du rendu graphique des interfaces. La démarche de conception utilisée pour réaliser ces prototypes est basée sur les principes de conception participative avec l'utilisation de scénarios, des phases de brainstorming, de prototypage papier et de prototypage vidéo [17]. Dans cette section, nous décrivons une partie de ces outils.

Le premier module de cet outil permet la création d'exemples de scène graphique et le second permet de contraindre les propriétés graphiques de couleurs des exemples créés. Nous décrivons brièvement les aspects techniques liés à la conception de ces outils : toolkits et méthodologie utilisées. Nous expliquons ensuite les principes du premier module « création d'exemples » avec le modèle conceptuel sous-jacent. Pour finir, nous présentons en détail les interactions et fonctionnalités de l'outil de contrainte des couleurs.

Module de création d'exemples

Réaliser des briques graphiques élémentaires.

Comme nous l'avons précisé dans la partie précédente, pour concevoir un rendu, il est important de travailler avec des exemples réels et contrôlables, appropriés à l'interface cible, plutôt que sur des exemples simplistes comme des rectangles colorés. Ainsi, nous avons conçu un outil de construction d'exemples à partir d'ensembles de « briques graphiques » élémentaires. Les briques sont présentées dans la première partie (à gauche) de l'outil (cf. Figure 2 a). L'utilisateur peut charger ou modifier des briques déjà existantes, ou en créer de nouvelles. Ces briques sont des représentations « orientées rendu » des objets de l'interface, des éléments visuels de composition d'une future scène graphique à paramétrer. Les briques sont définies et enregistrées sous la forme de représentations structurées à dominante vectorielle. L'intérêt d'utiliser un outil d'édition externe (type Inkscape) réside dans le fait que nous disposons ainsi d'un logiciel existant, et disposant de nombreuses fonctionnalités de création graphique. Cependant, il est important pour l'outil que

nous proposons, de permettre la mise à jour dynamique du fichier source, ou la possibilité de modifier, en interne, ces briques au niveau atomique pour faire varier la couleur, la taille ou la forme d'un sous composant. Les exemples sur lesquels le concepteur travaille sont ainsi « réels » et homogènes avec l'affichage ciblé (intégration des profils ICC), exhaustifs et contrôlables. Les briques graphiques sont composées au sein de scènes exemples. Pour cela il crée une nouvelle scène dans laquelle il dispose les briques graphiques par *drag & drop*. L'utilisateur contrôle la disposition de sa scène, en ajustant la position et la profondeur (hiérarchie) des briques graphiques la composant. Il peut supprimer des briques élémentaires à la scène ou en rajouter soit depuis la bibliothèque de composants, soit en copiant ou en clonant une brique déjà présente dans la scène. Dans le cas d'un clonage, et uniquement dans ce cas, les propriétés graphiques des sous composants de la brique clonée seront dépendantes de celles de la brique mère, une modification sur l'une d'entre elles entraînera une modification à l'identique sur l'autre. L'utilisateur peut créer et disposer des scènes différentes afin de les comparer. Il peut de la même manière que pour les briques copier ou cloner des composants entre deux scènes, ou une scène entière existante afin de la dériver pour faciliter l'exploration du design. Le partage de couleur par clonage permet à un concepteur de constater immédiatement l'impact d'une modification de cette couleur sur tous les éléments de la scène exemple.

Propriétés graphiques. Une liste des couleurs présentes sur la scène ouverte est tenue à jour sur le côté droit (Figure 2a) de l'outil. Sur cette liste figurent les couleurs uniques de chacune des briques de la scène. Par exemple pour une brique avec 3 éléments verts et 2 autres blancs, il y aura seulement deux couleurs dans la liste : du vert et du blanc. Si une autre brique contenant du blanc est ajoutée, une autre couleur blanche sera rajoutée sur la liste. Lors d'un clonage de brique au sein d'une même scène, et à cause de la relation de dépendance décrite ci-

avant, aucune nouvelle couleur ne sera ajoutée à la liste des couleurs sur la droite. En utilisant le même principe de partage des couleurs que celui du clonage de briques élémentaires, certains éléments peuvent partager la même couleur sans qu'il n'y ait d'ajout dans la liste. Si l'utilisateur veut ajouter une couleur à un sous élément d'une brique (qu'elle soit unique, la copie ou le clone d'une autre), il peut sélectionner ce sous élément et faire une action « new » sur sa couleur afin de faire apparaître une nouvelle couleur dans la liste (qui sera initialement de valeur équivalente).

Module de contrainte des propriétés graphiques
Explorer les paramètres des propriétés graphiques.

Les couleurs de la scène active, présentées sous forme de liste dans la partie propriétés de l'outil, sont affichées sous forme de pastilles dans l'espace colorimétrique (Figure 2b). Celui-ci utilise par défaut l'espace CIE LCH(ab) [8], « compatible » avec la perception visuelle humaine et permettant un contrôle « intuitif » au travers de 3 dimensions structurantes et partageables de la couleur : luminosité, saturation et teinte. Chacune des 3 dimensions est matérialisée par un axe, linéaire (pour L et C) ou radial (pour H), sur lequel sont positionnées des pastilles. Il y a donc 3 pastilles par couleur présente dans l'exemple : une sur L, une sur C et une sur H. Lors d'un pointage d'une pastille colorée sur un des axes, les 3 pastilles correspondant à cette couleur sont mises en évidence sur chacun des 3 axes (pastille verte sur la figure 2 b). En déplaçant ces pastilles colorées sur leur axe respectif, l'utilisateur modifie les valeurs des paramètres de la couleur et donc de la couleur d'interface résultante (RGB). Cette modification est immédiatement répercutée sur les briques de la (ou des) scène(s) exemple(s) concernée(s) (Figure 2). Ainsi, pendant l'exploration des valeurs de la couleur et par une interaction de type *drag & drop* dans l'espace des propriétés, le concepteur visualise et compare, en temps réel et en contexte sur la scène exemple, la modification du rendu de l'interface.



Figure 2 : Module de création de scènes exemples (a) et module de contrainte des propriétés (b).

De la même manière qu'à partir de la scène exemple le concepteur peut vouloir créer une nouvelle couleur sur un élément graphique, il peut ici vouloir « splitter » une pastille pour créer une nouvelle couleur qu'il associera ensuite à un élément graphique ou à un sous ensemble d'éléments. À l'inverse, il est possible de réunir deux pastilles colorées en une seule ce qui signifie qu'une couleur devient le clone d'une autre.

Le fait de placer ces pastilles sur un espace « perceptuel » permet au concepteur de visualiser directement certaines propriétés et de catégoriser ainsi les objets graphiques. Par exemple, il peut spécifier des hiérarchies et organiser les couleurs [5] en ensembles distincts de pastilles sur l'axe de la luminosité ou sur l'axe de teinte, ce qui favorise une première idée de la répartition et des groupements dans ses choix de couleurs. D'autre part le fait de travailler sur des espaces « perceptuels » permet de s'assurer que les écarts de distance physiques selon les dimensions L, C ou H sont bien répercutés perceptuellement, et permettent aux concepteurs de matérialiser graphiquement les différents types de perceptions définies par Bertin [2]. La perception sélective sera traduite par un écart important entre pastilles ou groupe de pastilles ; la perception ordonnée par la mise dans un ordre des pastilles sur un des axes de l'outil ; et la perception quantitative en les ordonnant tout en prenant en compte un rapport dans l'écart entre pastilles.

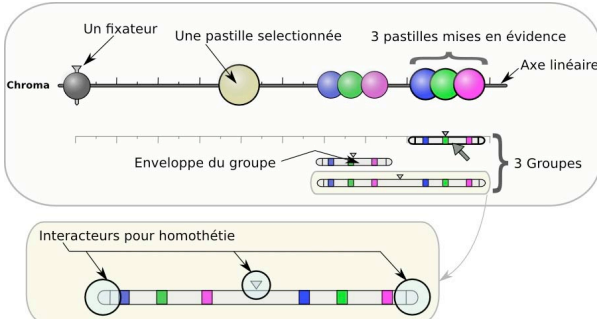


Figure 3 : Légende des éléments appartenant à un axe linéaire avec plusieurs pastilles et groupes.

En outre, plus la scène exemple est complexe et plus il est important d'avoir une vision globale de l'expression des couleurs (avec les harmonies et les dominantes de couleur) et de structurer les choix des couleurs. Ainsi la représentation structurée des couleurs en pastilles sur chacun des axes de la couleur présentée sur l'outil permet de visualiser la forme globale des « molécules », de détecter les espaces « vides » ou les concentrations de couleurs, et plus finement de comparer visuellement les différents écarts et régularités qui peuvent se dégager et qui sont des indicateurs importants de l'organisation possible de ces couleurs. Comme les modifications sur les positions des pastilles impactent immédiatement le rendu graphique des scènes exemples, le concepteur peut

comparer facilement et sur les exemples réels ses divers choix de design.

Atteindre les valeurs RGB possibles. Comme nous l'avons déjà précisé, nous utilisons le modèle de couleur CIE LCHab dont le gamut théorique est beaucoup plus large que celui de l'espace « écran » RGB. Il faut cependant que les concepteurs spécifient des couleurs affichables par le périphérique sous la forme de codes RGB. Pour les assister dans cette tâche, l'outil de contrainte propose lors de la manipulation isolée d'une pastille sur un axe d'afficher les valeurs atteignables par un dégradé de couleur à partir de la couleur de la pastille actuelle [10]. D'autre part, lors de modifications sur une pastille isolée entraînant le changement d'au moins un digit du code RGB, des graduations sont affichées au besoin si le concepteur veut visualiser ces changements.

De plus lors de la manipulation directe d'une pastille, des coordonnées RGB sont toujours mises à jour et affichées textuellement dans trois champs R, G et B afin de permettre au concepteur d'avoir en permanence une vue « coordonnée machine » de la couleur manipulée. Il existe en effet des problèmes très précis que seules des manipulations de l'espace colorimétrique du système, donc au bit RGB près, permettent de résoudre [22].

Les pastilles étant manipulées par *drag&drop*, il est parfois difficile d'atteindre une valeur précise (LCH ou RGB). Pour remédier à ce problème nous proposons au designer de sélectionner une pastille et d'y renseigner directement une valeur absolue (par exemple un angle de teinte) par l'intermédiaire d'un champ texte. Le designer peut également effectuer grâce au clavier des déplacements relatifs et incrémentaux des pastilles. Par exemple, on augmente ou diminue la valeur d'une pastille sélectionnée de digit en digit. Cette opération est aussi applicable aux groupes de pastilles définis ci-après.

Pose de contraintes

L'outil d'aide à la conception doit permettre de catégoriser et d'ordonner des objets graphiques mais aussi de conduire à l'expression de contraintes entre les propriétés de ces objets. Ces contraintes sont en fait la traduction d'exigences de haut niveau (en rapport avec l'activité supportée), plus ou moins explicitement exprimées dans le cahier des charges ou d'exigences issues de règles de conception connues du concepteur, en contraintes de bas niveau (affichage). Les contraintes de bas niveau sont des relations exprimées formellement entre les propriétés graphiques des variables visuelles, par exemple des écarts chiffrés entre valeurs des propriétés graphiques d'éléments composant une scène.

Nous avons réifié [3] ces contraintes en un ensemble d'objets interactifs, manipulables directement: une contrainte de «groupement» avec les opérations associées, une contrainte de «fixation» et une contrainte de «séparation». À ces contraintes s'ajoute un objet particulier «repère» qui permet de faciliter l'exploration visuelle ou définir des contraintes de façon relative et un objet «annotation» qui permet de commenter les contraintes réifiées.

Notion de groupe de propriétés graphiques. La première contrainte permet l'association de pastilles colorées dans un groupe. Un groupe traduit une propriété qui est ici une relation d'association logique entre plusieurs pastilles. Cette réification [3] de lien logique en un lien graphique a un lien plus ou moins fort avec les aspects fonctionnels de l'objet graphique qu'elle colore. Par exemple, un groupe pourra réifier un lien entre les luminosités des fonds de secteur de contrôle ou encore entre les saturations des éléments d'alarme, ou encore un groupe englobant toutes les teintes de la scène exemple. Le designer sélectionne des pastilles pour un groupe par clic ou par une interaction de type lasso. Dès que plus de deux pastilles sont sélectionnées, un groupe apparaît au-dessus de l'axe des pastilles (ou à l'extérieur de l'axe des teintes). Ce groupe correspond à un rectangle allongé dont les dimensions sont comprises entre les valeurs min et max des pastilles sélectionnées (Figure 3).

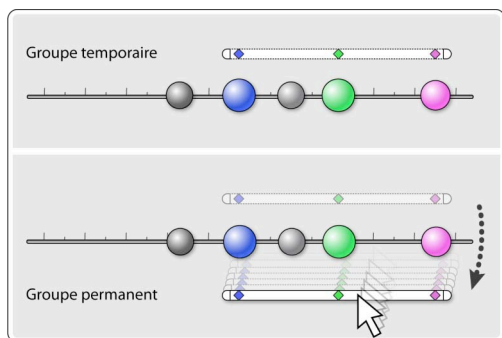


Figure 4 : Transformation d'un groupe temporaire en groupe permanent.

Certains groupes ne nécessitent pas d'être permanents, ce sont des groupes transitoires qui permettent au concepteur d'effectuer temporairement des opérations sur plusieurs pastilles à la fois. Il s'agit alors d'une commodité d'utilisation, par exemple pour répartir toutes les pastilles sur un axe, ou pour déplacer grossièrement une partie des pastilles. Si le groupement est relatif aux fonctions des objets liés à ces pastilles, le designer le rendra permanent pour conserver les informations de design liées à ce groupe. Le designer peut faire passer un groupe d'un état transitoire à un état permanent par une interaction d'ajout : touche p ou par drag&drop allant du dessus au-dessous de la barre de l'axe des pastilles (Figure 4), ou de l'extérieur à l'intérieur pour l'axe de teintes. De la même façon, un

groupe devenu obsolète ou hérité d'une copie de scène, peut être « rompu » (suppr ou drag&drop vers une zone extérieure aux groupes) si besoin. À côté de chaque axe, un bouton permet de rappeler le dernier groupe transitoire. Cela évite à l'utilisateur de devoir refaire la dernière sélection de pastilles sur cet axe.

Opérations sur les groupes. Comme définit ci-dessus, l'utilisateur peut grouper des pastilles entre elles, temporairement ou non, et effectuer des actions et opérations sur ces dernières. Trois interacteurs de redimensionnement sont disposés : deux de part et d'autre du groupe et un autre au-dessus du groupe. Parmi les actions possibles figurent des actions de positionnement, comme le déplacement simultané des pastilles sur un axe, une homothétie (étirement ou réduction, Figure 5), qui conserve le rapport des écarts internes entre pastilles (ce rapport peut ensuite être figé au sein d'un groupe). L'utilisateur peut aligner les pastilles sur une des extrémités du groupe, ou sur le « barycentre », ou sur le centre du groupe. Il peut aussi répartir des pastilles de manière égale sur un espace donné, soit sur un axe en entier soit entre deux séparateurs. Le fait qu'une pastille colorée soit groupée avec d'autres n'empêche pas l'action sur un élément de ce groupe (interaction non modale), et la position, l'ordre interne des pastilles ou la taille du groupe restent dynamiquement mis à jour par la manipulation d'un ou plusieurs de ses éléments internes.

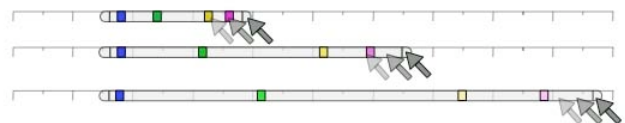


Figure 5 : Opération d'homothétie sur un groupe

Notion de fixateur. Le troisième type de contrainte, appelé fixateur, permet de fixer/défixer la position absolue d'une pastille (Figure 3). L'action de fixation est polymorphe [3] ainsi, elle peut être utilisée sur un groupe où l'action de fixer inversera alors tous les états de fixation des éléments du groupe ou encore sur un séparateur pour lequel il fixera la position.

Notion de repère. Un objet particulier « repère » peut être activé dans les représentations de propriétés graphiques. Il permet de structurer, visualiser ou exprimer des contraintes. Ce repère est tout d'abord un guide visuel positionné par le concepteur pour construire et ordonner les propriétés. Par exemple, un repère définit sur l'axe des luminosités un seuil au-dessous duquel devraient être disposées toutes les couleurs de fond. Il permet aussi de visualiser des zones: ainsi, une succession de repères sur une des dimensions de la couleur délimitera des espaces de valeurs. Par exemple, une zone (figure 9) entre deux repères positionnés sur l'axe des teintes définira les valeurs acceptables ici « la zone des alarmes ».

Ces zones servent de gabarit et sont affichées ou cachées sur demande.

Notion de molécule harmonique. Un autre mécanisme de composition s'ajoute à ces types de contraintes : une contrainte « harmonique » permet de prendre en compte sur l'axe des teintes des notions perceptives plus esthétiques. Le designer graphique peut vouloir utiliser ou définir une contrainte « harmonique » pour régler les propriétés de couleur des composants graphiques de la scène. Cette notion de contrainte harmonique est issue du domaine des Arts avec un usage très important par les peintres et plasticiens du cercle chromatique (Newton) et des théories de l'harmonie des couleurs (Chevreul, Goethe, Itten [12]) qui définissent les propriétés de composition et d'harmonie de couleurs « complémentaires », « semi-complémentaires » et « analogues » issues du cercle chromatique. Le designer graphique utilise habituellement ces concepts et techniques pour créer un nuancier de couleurs harmoniques qu'il met en œuvre dans sa composition. Dans l'outil, cette harmonie des complémentaires correspond à utiliser un groupe sur l'axe des teintes (radial) comme structure géométrique régulière particulière [14] entre les pastilles lui appartenant (Figure 6).

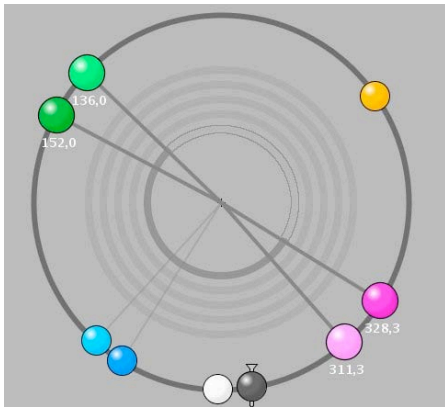


Figure 6 : L'axe des teintes avec plusieurs pastilles et groupes, un groupe de 4 couleurs est mis en évidence.

Comparaison entre groupe linéaire et radial.

Les groupes et interactions réalisables sur les axes linéaires sont également possibles sur l'axe radial. Ils sont simplement adaptés en faisant boucler le début et la fin d'un axe linéaire pour le transformer en radial. Ainsi, les groupes transitoires sont rendus permanents en les faisant traverser l'axe jusqu'à la zone des groupes. Seules quelques spécificités apparaissent dues au fait que l'axe circulaire n'a ni début ni fin « logique » et qu'il propose intrinsèquement des propriétés de type « harmonie » définies ci avant. Par exemple, les groupes sur l'axe des teintes étant circulaires, une interaction permet de déterminer quelles sont les pastilles de début et de fin du groupe en question. Le designer fait simplement glisser par *drag&drop* la zone (trait creux) séparant le début de la

fin d'un groupe entre les nouvelles pastilles de début et de fin. (Figure 6). En plus d'être réifié par une enveloppe allant du début à la fin du groupe, un groupe radial est aussi réifié par les rayons allant du centre de l'axe jusqu'aux pastilles appartenant à ce groupe.

Annotations. Le dernier type d'outil ne permet pas de réifier des contraintes mais d'annoter les actions effectuées. Pour cela le designer crée un nouveau commentaire/ annotation, qu'il peut attacher à n'importe quel objet de l'interface : une pastille, un groupe, un séparateur, un ou des repères ; ou même sur une action comme un split de pastille. Cette fonctionnalité est très importante car elle permet de spécifier à quelle contrainte du cahier des charges correspond la contrainte réifiée sur nos outils. Les annotations permettent aussi de signaler des contraintes ou des problèmes non exprimables par notre ensemble d'outils. Par exemple une « couleur nommée » ne peut pas être spécifiée par nos outils, même par des repères/zones sur un axe car c'est une contrainte subjective et dépendant de plusieurs dimensions de la couleur. Enfin ces annotations ont une utilité pour conserver la trace du processus de choix des paramètres graphiques. Cela est utile au designer autant pour se remémorer les critères de ses choix (pour le design rationale [13]) que pour justifier ses choix auprès d'autres personnes. Le designer a le choix d'afficher ou de cacher les annotations sur les éléments, ainsi qu'une liste des actions précédemment effectuées maintenue à jour avec les annotations associées.

Un peu de technique. Le premier outil a été réalisé en s'appuyant sur la toolkit graphique perl/tk/zinc[7], le second est codé en Java et s'appuie sur la bibliothèque SwingStates [1] qui rajoute les machines à états à la toolkit Java Swing. Les deux parties du logiciel communiquent par des messages au format texte transitant par le bus logiciel Ivy [5]. Ainsi, les modifications faites dans la partie de création de scènes exemples sont immédiatement répercutées dans la partie de contrainte. Par exemple, l'ajout d'une brique à une scène crée une pastille si besoin. À l'inverse une modification sur une pastille de la partie contrainte sera immédiatement répercutée sur les surfaces affichées concernées.

SCENARIOS D'UTILISATION

Pablo est designer graphique et il doit répondre à divers problèmes de paramétrage de rendus graphiques de systèmes interactifs critiques. Pour cela, il dispose en plus de son expérience de designer de savoirs empiriques et théoriques comme des connaissances sur les « bonnes propriétés de la couleur » [12][20], des connaissances issues du domaine de la visualisation d'information [24], ou de la colorimétrie [11][8]. Il dispose également des outils que nous venons de présenter.

Interaction couleur/surface : « Oranges alarmes. »

Dans son cahier des charges, le designer doit paramétrer une couleur d’alarme « orange » qui sera appliquée à des éléments de formes et de surfaces différentes. Pour cela, il crée une scène exemple (Figure 7) avec une couleur orange alarme qu’il lie à la fois à une zone textuelle sur une première brique « piste radar » et plus loin au fond d’une autre brique « zone d’information sur les alarmes ». Les 2 orange sont liés (clones) donc représentés une seule fois dans la liste des couleurs. Sur l’outil de contrainte, le designer veut régler plus finement cet orange mais il s’aperçoit rapidement que la différence de forme et de surface est trop importante entre les deux types d’éléments comportant cet « orange alarme ». Il décide alors de « spliter » la pastille en deux et de régler différemment leurs saturations. Il groupe ces deux pastilles et les annote afin de transmettre cette information aux responsables des spécifications



Figure 7 : 2 scènes exemples mettent en œuvre la même couleur orange : en aplat de couleur à gauche, comme couleur de l’élément texte « ALRT » à droite.

« Échelle des luminosités »

Le designer commence en général son travail de conception par le choix des fonds et par le réglage global des luminosités. Pablo a déjà fait un travail sur les luminosités d’une nouvelle scène graphique comportant un grand nombre d’éléments. Il s’aperçoit cependant qu’il a oublié de placer un élément qui doit forcément être parmi les éléments les moins lumineux de la scène. Sur l’axe des luminosités, cette zone est déjà très encombrée par des pastilles. Afin de ne pas remettre en cause le travail déjà fait sur les autres pastilles, il crée un groupe comportant toutes ces pastilles déjà paramétrées. Le ratio interne du groupe sera conservé pour toute opération de translation ou d’homothétie sur ce groupe. Afin de créer un espace pour la nouvelle pastille, il peut donc choisir de translater légèrement ce groupe vers les luminosités plus élevées (Figure 8) ou d’étirer le groupe à l’aide des interacteurs situés de part et d’autre (Figure 5). Pour un exemple similaire mais avec un besoin de place au centre d’un groupe de pastilles existantes, Pablo peut utiliser l’interacteur de redimensionnement homothétique situé au-dessus de chaque groupe.

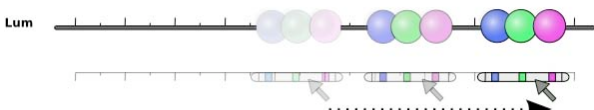


Figure 8: Opération de translation sur un groupe.

Dans un exemple similaire à celui des « orange alarme », Pablo doit effectuer le réglage d’une couleur très légèrement saturée (un gris coloré légèrement teinté de rouge). Cette couleur sera toujours appliquée au même type d’objet graphique, une surface de « secteur à circulation limitée ». Cependant cette surface sera perçue différemment selon la configuration du logiciel (petit ou grand secteur) ou selon le niveau de zoom appliqué à cette surface. Dans certains cas, celle-ci sera perçue comme contenant trop de rouge et l’inverse elle sera perçue comme trop grise (désaturée) dans d’autres. Ne pouvant spécifier autant de couleur qu’il existe de configuration ou de niveau de zoom, Pablo crée sur l’axe de la saturation une zone à l’aide de repères. Cette zone est la plage des valeurs autorisées sur cet axe pour la pastille de la couleur à paramétrer. Il annote alors la pastille et la zone repérée pour enrichir les spécifications.

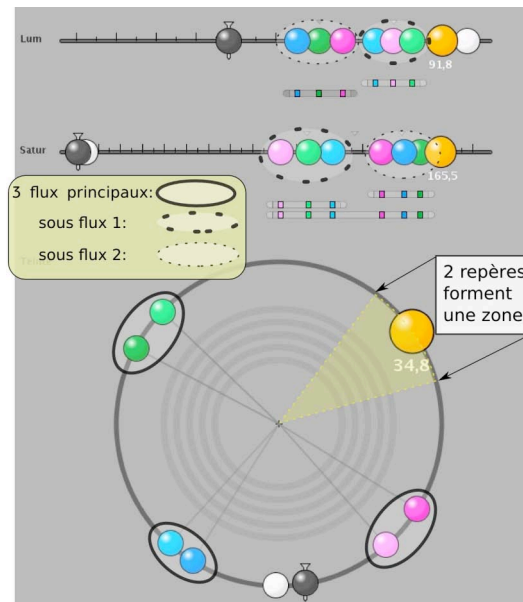


Figure 9: Choix de design finaux pour « Paris Flux »

« Paris flux ».

Dans son cahier des charges, Pablo doit régler les couleurs pour des flux d’avions. Il doit discriminer 3 flux à dominantes de teintes prédéfinies par les utilisateurs, rose, verte et bleue. Chaque flux associe deux sous flux visuellement distincts. Pablo sait qu’il a 3 dimensions (L, C et H) à sa disposition. Comme il doit séparer chaque flux par leur teinte « nommée » (rose, vert, bleu), il va se servir de légers écarts de teinte couplés à des écarts de luminosité et de saturation pour « typer » et séparer les sous flux. Après avoir créé dans le premier module (Figure 2 a) une scène exemple comprenant des avions appartenant aux 3x2 types de flux, le designer crée des groupes permanents sur chacune des dimensions, par sous flux sur L et C et par flux sur H (Figure 9). Sur l’axe des teintes, les pastilles devraient être placées sur une structure harmonique et séparées d’environ 120° selon le cahier des charges, mais Pablo sait qu’il doit réserver un pan de l’axe des

teintes aux alertes et alarmes (du jaune au rouge). Il crée donc une zone repère à ne pas remplir. De plus, comme il sait que l'espace LCH n'est pas parfait (non orthogonalité au niveau des bleus), il modifie légèrement la structure harmonique pour éviter les déviations de teinte. Cette adaptation n'aurait pas été possible sur certains des outils existants présentés dans l'état de l'art, puisque ceux-ci se limitent souvent à proposer des structures harmoniques prédéfinies, non personnalisables. Ce dernier exemple est un cas particulièrement délicat à résoudre, car la différence de couleurs doit être suffisamment grande pour permettre l'identification immédiate de chaque sous-flux, et suffisamment petite pour permettre la fusion des paires de sous-flux en leur flux respectif. Les connaissances du designer l'aident à trouver et à explorer des solutions et les outils apportent un support nécessaire à la réflexion puis à la prise de décision. Comme les contraintes sont accessibles de manière directe, cela permettra à Pablo de concevoir ou modifier ces choix avec les utilisateurs et de justifier plus facilement ses choix aux décideurs. Lors d'une itération avec les contrôleurs, Pablo a appris que pour des raisons liées au métier les couleurs des sous flux proposées étaient trop saturées. Les utilisateurs ont proposé sur l'outil de rectifier le choix initial du designer en abaissant leurs contrastes. Nos outils servent alors de plateforme d'échange entre le designer et les utilisateurs. D'autre part, les décideurs pourront « lire » les contraintes beaucoup plus facilement que sur un tableau de codes RGB. Enfin, Pablo sait que les annotations qu'il a faites assureront la pérennité de ses choix et qu'il pourra toujours retrouver la logique de construction de ses rendus graphiques.

La superposition des groupes est importante car elle permet de créer et de visualiser une représentation mentale des contraintes. Par exemple, le designer peut créer le groupe des fortes et des faibles luminosités. Cela lui permet de passer rapidement d'un groupe à un autre et de voir les relations se dessiner notamment sur l'axe des teintes. Le designer peut résoudre des problèmes complexes en traduisant la problématique de haut niveau en une construction de contraintes de plus bas niveau puis en les manipulant directement. L'effort de structuration que le concepteur devait faire auparavant dans sa tête peut être traduit sur nos outils, manipulé et visualisé à chaque instant. La charge cognitive du designer est ainsi réduite et il peut se concentrer sur l'exploration des possibles et voir émerger des solutions.

CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons décrit des outils d'aide à la spécification des paramètres graphiques pour des interfaces graphiques ayant des rendus complexes. À partir d'exemples réels de scènes graphiques, nos outils proposent des interactions pour créer, manipuler, grouper et visualiser des contraintes sur les propriétés

graphiques de la couleur au travers de ses trois dimensions : luminosité, saturation et teinte. Nos outils permettent d'agir sur des pastilles colorées placées sur des axes correspondants à ces différentes dimensions de la couleur. Nous avons détaillé dans l'article un ensemble d'actions et d'interacteurs permettant de manipuler et contraindre ces pastilles. Par rapport aux outils existants, nos outils permettent d'ajuster et de vérifier de façon plus efficace le rendu des composants graphiques. En effet, au travers des scénarios nous avons montré que les designers graphiques sont capables grâce à nos outils de traduire des contraintes de haut niveau issues de cahiers des charges en des contraintes de bas niveau exprimées par des relations entre variables visuelles. Ces relations sont exprimées sur des pastilles placées sur chacun des axes de la couleur par le biais de réifications en groupes. Des actions et manipulations sur ces groupes ainsi que la pose de repères et d'annotations permettent de préciser ces contraintes. Plus important, ce travail sur la structure et sur ces axes de propriétés aide le designer à structurer une pensée globale de ces propriétés pour mieux formaliser, spécifier et justifier un espace de solution.

De plus, l'utilisation d'exemples réels et contrôlables permet aux concepteurs de percevoir immédiatement l'efficacité des explorations de leurs choix de design. Cela permet de vérifier qu'aucune interaction visuelle ne vient biaiser la conception du rendu. Par exemple, il est possible de détecter des interactions entre surface et couleur des éléments graphiques (exemple des oranges). Notre expérience nous a par ailleurs appris que le design des couleurs ne peut avoir lieu et n'a de sens que sur des périphériques d'affichages calibrés et dont les profils ICC [11] sont pris en compte. Même si ces aspects ne sont pas détaillés dans cet article, les designers doivent y accorder une grande attention et les outils que nous proposons doivent clairement indiquer quels profils sont utilisés afin de respecter la chaîne colorimétrique allant du codage à la perception des couleurs.

Le fait que les éléments et actions de nos outils soient annotables permet au concepteur de décrire des contraintes de haut niveau non directement exprimables, mais cela permet surtout d'impacter les phases futures du cycle de vie du rendu graphique. En effet à court terme et après avoir été spécifié, le système interactif sera codé puis évalué. Dans ces phases, les annotations permettront de conserver la logique des choix à respecter lors du codage et permettront de savoir quelles propriétés perceptives doivent être validées. À plus long terme, le système interactif correspondant à ce rendu pourra évoluer ou être reconçu, il sera alors important de retrouver quels choix de conception de rendu graphique avaient été faits et pour quelles raisons. Ainsi ces outils d'aide à la conception ne sont pas simplement une aide aux programmeurs mais bel et bien aux concepteurs d'interfaces dans leur ensemble,

puisque'ils permettent à la fois de garder une continuité entre les phases de spécification et de codage des rendus (sans le goulet d'étranglement classique entre expert graphique et programmeur) mais aussi de justifier ces choix aux décideurs et de planifier les futures évaluations et re-conceptions de ces rendus.

BIBLIOGRAPHIE

1. Appert, C. and Beaudouin-Lafon, M. *SwingStates: adding state machines to the swing toolkit*. In Proceedings of UIST '06. ACM, New York, NY, 2006, pp. 319-322.
2. Bertin, J. *Sémiologie graphique: Les diagrammes - Les réseaux - Les cartes*. Editions de l'École des Hautes Études en Sciences, 31 janvier 1999, 1070 p.
3. Beaudouin-Lafon, M. and Mackay, W. E. *Reification, polymorphism and reuse: three principles for designing visual interfaces*. In Proceedings of the Conference AVI (Palermo, Italy). AVI '00. ACM, New York, 2000, pp. 102-109.
4. Boy, G. A. *Active design documents*. In proceedings of the 2nd Conf. on Designing interactive Systems: Processes, Practices, Methods, and Techniques. Amsterdam, 1997, pp. 31-36.
5. Brewer, C. *Designing Better Maps: A Guide for GIS Users*. Redlands, Calif.: ESRI Press. 2005, 203 p.
6. Buisson, M., Bustico, A., Chatty, S., Colin, F., Jestin, Y., Maury, S., Mertz, C., and Truillet, P. *Ivy: un bus logiciel au service du développement de prototypes de systèmes interactifs*. In Proceedings of IHM'02, Poitiers, France, Ed. IHM '02, vol. 32. ACM, New York, NY, November 26 - 29, 2002, pp. 223-226.
7. Chatty, S., Sire, S., and Lemort, A. *Vers des outils pour les équipes de conception d'interfaces post-WIMP*. In Proceedings of IHM (Namur, Belgium, August 30 - September 03, 2004). IHM 2004, vol. 386. ACM, New York, 2004, pp. 45-52.
8. CIE, CIECAM02, 159:2004, *A Colour Appearance Model for Colour Management System*.
9. Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., and Xu, Y. *Color harmonization*. ACM Trans. Graph. 25, 3, Jul. 2006, pp. 624-630.
10. Douglas, S. A. and Kirkpatrick, A. E. *Model and representation: the effect of visual feedback on human performance in a color picker interface*. ACM Trans. Graph. 18, 2, Apr. 1999, pp. 96-127.
11. ICC.1:2004-10 Specification (Profile version 4.2.0.0) *Image technology colour management : Architecture, profile format, and data structure*, International Color Consortium, 2004.
12. Itten, J., *Art de la couleur*. Édition Kunst Der Farbe., Ed. Dessain et Tolra, 196, 155 p.
13. Lacaze, X., Palanque, P., Barboni, E., Navarre, D. *Design Rationale for Increasing Profitability of Interactive Systems Development*. Rationale Management in Software Engineering, 2005, pp.182-197.
14. Lyons, P. and 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, HCI, and Measurement Systems, 18-20 July 2005, pp. 80-85.
15. Lyons, P. and Moretti, G. *Nine tools for generating Harmonious Colour Schemes*. In Proceedings of APCHI 2004, Rotorua.
16. Lyons, P., Moretti, G., and Wilson M. *Colour Group Selection for Computer Interfaces*. Human Vision and Electronic Imaging V. San Jose, USA: SPIE - The International Society for Optical Engineering, 2000, pp. 302-313.
17. Mackay, W. E., Rutzer, A. V., and Janecek, P. *Video artifacts for design: bridging the Gap between abstraction and detail*. In Proceedings of DIS'00 (NY City, USA, August 17 - 19, 2000). D. Boyarski and W. A. Kellogg, Eds. DIS '00. ACM, New York, 2000, pp. 72-82.
18. Meier, B. J. *ACE: a color expert system for user interface design*. In Proceedings of the 1st Annual ACM SIGGRAPH Symposium on User interface Software (Alberta, Canada, October 17 - 19, 1988). UIST '88. ACM, New York, 1988, pp. 117-128.
19. Meier, B. J., Spalter, A. M., and Karelitz, D. B. *Interactive Color Palette Tools*. IEEE Comput. Graph. Appl. 24, 3, May. 2004, pp. 64-72.
20. Munsell, A. H. *A Pigment color System and Notation*. The American Journal of Psychology 23, 1912, pp. 236-244.
21. NASA Color Usage Website. (2004): <http://colorusage.arc.nasa.gov> (visité en mai 2009).
22. Tabart, G., Conversy, S., Vinot, J. L., Athènes, S. *Designing graphical elements for cognitively demanding activities: an account on fine-tuning for colors*. In Proceedings of DSVIS 2008, LNCS, Springer Verlag, 2008, pp 136-148.
23. Terry, M. and Mynatt, E. D. *Side views: persistent, on-demand previews for open-ended tasks*. In Proceedings of the 15th Annual ACM Symposium on UIST (Paris, France, October 27 - 30, 2002). UIST '02. ACM, New York, 2002, pp. 71-80.
24. Ware, C. *Information Visualization: Perception for Design*. (2nd Edition) Morgan Kaufman. December 2004, 435 p.