

Les transitions visuelles différenciées : principes et applications

Céline Schlienger¹

Pierre Dragicevic¹

Claire Ollagnon²

Stéphane Chatty¹

¹ IntuiLab

Prologue 1 - La Pyrénéenne

31672 Labège, France

{celine, dragice, chatty}@intuilab.com

² Intactile Design

20, rue du Carré du Roi

34000 Montpellier, France

ollagnon.c@intactile.com

RESUME

Pour servir de support à la conception d'animations dans les interfaces graphiques, nous introduisons et nous définissons le concept de *transition visuelle*. Nous utilisons ensuite ce modèle descriptif pour introduire une nouvelle technique d'animation : les *transitions visuelles différenciées*, qui permettent de véhiculer de l'information en jouant sur les transitions visuelles. Cette technique est illustrée sur des cas concrets avec le concours d'une animatrice professionnelle. Nous énonçons également des propriétés caractérisant les transitions visuelles différenciées, et abordons les principales questions relatives à leur conception.

MOTS CLES : animation, transitions animées, dessin animé, conscience de la situation.

ABSTRACT

In order to support animation design in graphics interfaces, we introduce the concept of *visual transition*. We then use this descriptive model to introduce a new animation technique: *differentiated visual transitions*, which allow conveying information by varying visual transitions. This technique is illustrated on concrete cases with a professional animator. We also introduce some properties of differential visual transitions and address design questions.

CATEGORIES AND SUBJECT DESCRIPTORS: H5.2 [Information Interfaces and presentation]: User Interfaces; H.5.1: Multimedia Information Systems - Animations.

GENERAL TERMS: Design, Theory.

KEYWORDS: animation, animated transitions, cartoon, situation awareness

INTRODUCTION

L'informatique moderne permet d'interagir avec des systèmes complexes et de grandes quantités de données. Ceci oblige d'une part à coder toujours plus d'information en restant lisible, d'autre part à assurer la « conscience de la situation » de l'utilisateur, c'est-à-dire sa compréhension des évolutions du système [13].

L'animation est souvent évoquée parmi les solutions à l'un ou l'autre problème. Elle peut tout d'abord être exploitée comme canal supplémentaire de présentation d'information [21]. Elle permet aussi de notifier des événements de façon plus ou moins intrusive [4, 9] et de confirmer les conséquences d'actions par des retours animés [17, 19]. Parfois ces animations expriment des changements immédiats en des changements visuels fluides : on parle alors de *transitions animées*. Il est régulièrement avancé que les transitions animées sollicitent moins les capacités cognitives de l'utilisateur pour maintenir sa conscience de la situation [6, 7, 14, 17, 26].

Toutefois, alors que le média graphique est bien maîtrisé, concevoir des animations pour les interfaces reste une tâche compliquée. Comment concevoir, avec les animateurs, des animations cohérentes avec les représentations graphiques ? Comment ne pas surcharger visuellement l'interface ? Quelles informations coder par des animations et comment ? Pour répondre à ces questions, nous allons tout d'abord introduire la notion de *transition visuelle*. Les transitions animées en sont des exemples. Nous utiliserons ensuite cette notion pour introduire une technique baptisée transition visuelle différenciée, ou transition différenciée.

Les *transitions différenciées* permettent de combiner dans une même opération visuelle les fonctions de transition et de présentation d'information. Cette approche repose sur la constatation suivante : bien que la plupart des animations de transition relient deux états par une trajectoire unique, plusieurs trajectoires sont possibles. Choisir une trajectoire plutôt qu'une autre suffit à véhiculer des informations utiles.

Dans cet article nous proposons une définition plus précise des transitions différenciées avant de montrer des exemples d'utilisation. Puis nous abordons les principales questions liées à la conception des transitions différenciées, et terminons en évoquant les travaux existants.

Réserver cet espace pour la notice de copyright

TERMINOLOGIE

Nous présentons ici une terminologie relative aux transitions différenciées, en nous appuyant sur une modélisation conceptuelle des transitions visuelles. La notion subjective d'animation [18] n'est pas évoquée ici, mais notre modèle s'applique aux animations de transition, qui constituent un cas particulier de transition visuelle.

Rendu

Nous supposons qu'une application graphique est constituée d'objets du domaine synchronisés avec une représentation visuelle. Chaque objet est caractérisé par un ensemble de propriétés dont chacune est *rendue* ou non. Une propriété est dite rendue si elle influe sur la représentation visuelle de l'objet.

On suppose que la représentation visuelle d'un objet peut se déduire de la valeur de ses propriétés rendues, par application d'une fonction R de *rendu statique*. Pour rendre compte des effets de transition, nous introduisons des fonctions T de *rendu transitoire* qui fournissent une représentation de l'objet variable au cours du temps.

Transitions

Nous nous limitons au cas d'un objet du domaine composé de deux propriétés p et q , et pour lequel :

- p est une propriété rendue dont la valeur passe instantanément de p_0 à p_1 au temps t_0 .
- q est une propriété non rendue.

Par exemple, l'objet est une jauge qui contient un liquide de viscosité q (q_i pour l'eau et q_j pour l'huile), et dont le taux de remplissage p passe brusquement de p_0 à p_1 . Ce changement peut se traduire visuellement :

- *Sans transition* (figure 1a): la mise à jour est effectuée par simple application de $R(p)$; la représentation visuelle de l'objet change donc instantanément. Par exemple, la hauteur du liquide de la jauge varie brusquement.
- Par *transition simple* (figure 1b): nous disposons, en plus de $R(p)$, d'une fonction de rendu transitoire $T(p_0, p_1, t-t_0)$ qui prend en entrée la variation de p et le temps écoulé depuis cette variation. T se substitue à R dans l'intervalle $[t_0, t_1]$. Par exemple, le liquide descend doucement en changeant de couleur.
- Par *transitions différenciées* (figure 1c): le cas est le même que précédemment mais la fonction de rendu

transitoire $T(q, p_0, p_1, t-t_0)$ dépend de q . Par exemple, la jauge se videra plus ou moins vite en fonction de la viscosité.

Lorsqu'il se produit une transition différenciée, nous dirons que p est la *propriété de référence* et q est la *propriété rappelée*.

Transitions pseudo-différenciées et renforcement

Intéressons-nous maintenant au cas où q est rendu, c'est-à-dire lorsque les fonctions de rendu statique et transitoire dépendent de q . Par exemple p est traduite par la position d'une icône et q par sa couleur.

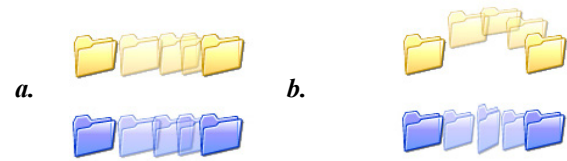


Figure 2 : Transition pseudo-différenciée (a) et renforcement (b) : ici la propriété rappelée est déjà codée par une couleur.

Sous cette hypothèse, les transitions simples deviennent des transitions différenciées. Dans la figure 2a, les transitions sont différenciées dans la mesure où le déplacement d'une icône jaune et le même déplacement d'une icône bleue sont deux effets visuels différents. Dans ce cas, nous parlerons cependant de transitions *pseudo-différenciées* afin de le distinguer des situations où d'autres dimensions visuelles comme la trajectoire spatiale sont également fonction de q (figure 2b).

Cette distinction est possible dans le cas où le rendu R est décomposable en deux rendus partiels R^p et R^q . Par exemple, R^p donne la position de l'icône en fonction de p et R^q sa couleur en fonction de q . Dans ce cas, chaque variation de p peut se traduire visuellement par :

- des *transitions pseudo-différenciées* (figure 2a) : le rendu statique est une combinaison de R^p et R^q , le rendu transitoire est une combinaison de $T(p_0, p_1, t-t_0)$ et de R^q .
- des *transitions différenciées* (figure 2b): même cas que précédemment, mais le rendu transitoire partiel $T(q, p_0, p_1, t-t_0)$ dépend de q . Dans ce cas particulier, on parlera de *renforcement* de transitions pseudo-différenciées.

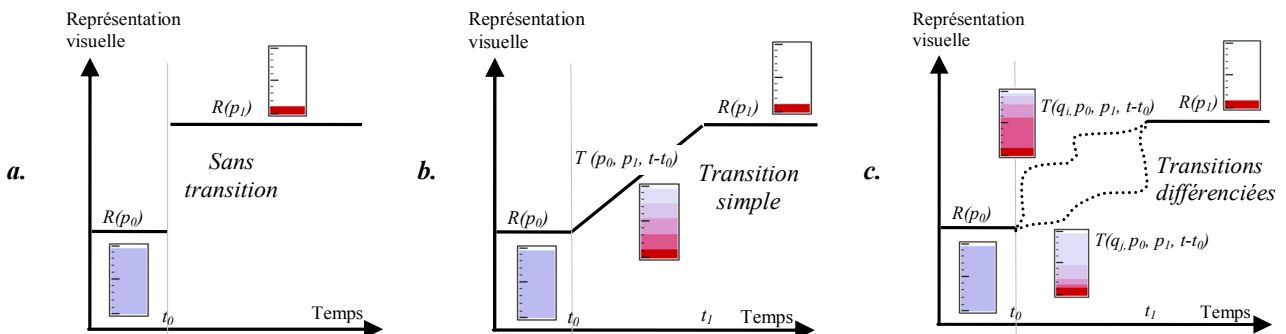


Figure 1 : trois types de transition visuelle pour une propriété p qui passe de p_0 à p_1 .

EXEMPLES DE TRANSITIONS DIFFERENCIÉES

Dans cette partie, nous illustrons les transitions différenciées par deux nouvelles techniques appelées *Graviticônes* et *Mobilistes*. La plupart des animations décrites ici peuvent être visualisées en Flash depuis le site <http://www.intuilab.com/transdiff>.

Les Graviticônes

Dans les bureaux virtuels, la copie d'un fichier s'accompagne parfois de l'affichage d'une barre de progression, et s'achève par l'apparition soudaine du fichier copié dans la fenêtre de destination (figure 3). Des efforts sont portés sur l'animation d'attente, mais peu sur la continuité visuelle de l'opération dans sa globalité. Par ailleurs, peu d'attention est prêtée à la prévention d'erreurs et à la conscience de la situation : au début de la copie, les informations sur le fichier et le transfert ne sont accessibles qu'à travers un examen attentif, et la copie terminée, les indications fournies sont maigres.

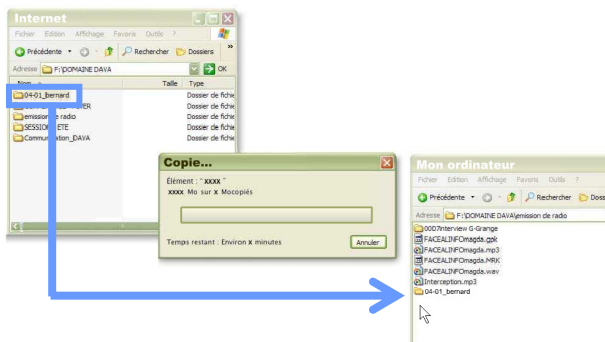


Figure 3 : copie d'un fichier

Les Graviticônes comblent ces lacunes par des transitions différenciées. Nous les illustrons par un scénario. Pour son état de l'art, Charles, chercheur en IHM, télécharge une vidéo. A l'initiation du transfert, l'icône de la vidéo se duplique, s'étire longuement (figure 4a) et est lentement « aspirée » par un tube (figure 4b) ; Charles en déduit que la vidéo est lourde et le transfert lent. Pendant le téléchargement, il trouve la même vidéo sur un autre site duquel il tente le téléchargement. La copie de l'icône s'étire autant mais est très rapidement aspirée par le tube ; il en déduit que le transfert est beaucoup plus rapide. Charles choisit donc d'interrompre le téléchargement précédent.

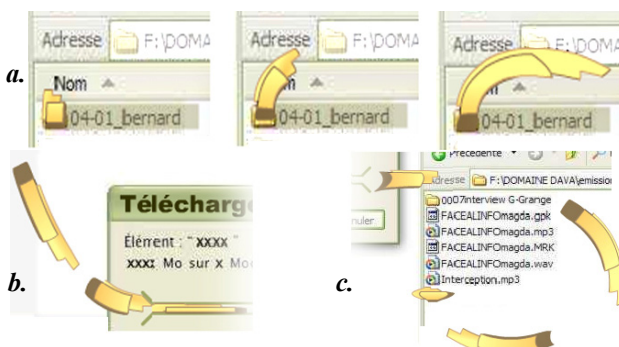


Figure 4 : téléchargement d'un fichier lourd (décollement, aspiration, expiration)

Charles veut ensuite copier son article et les documents associés sur une clé USB. Il commence par son article : l'icône est aspirée (figure 5a, 5b), rapidement « expirée », puis se positionne légèrement (figure 5c); la copie est rapide et le fichier de petite taille. Ensuite, il copie le répertoire contenant les articles de son état de l'art. Alors qu'il s'apprête à copier la vidéo, l'icône du répertoire s'écrase très lourdement (figure 4c). Il en déduit qu'il est très gros. Il décide alors de ne conserver que les articles indispensables afin de libérer de la place sur la clé USB pour sa vidéo.

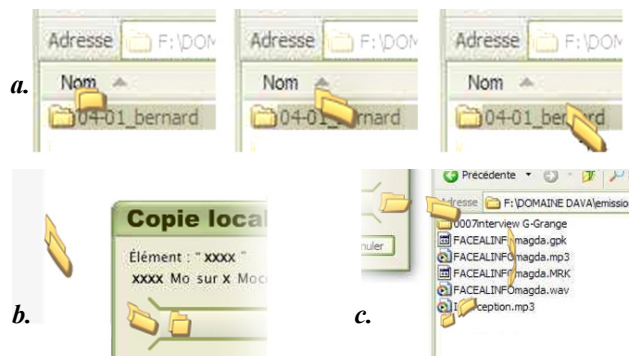


Figure 5 : copie d'un fichier léger (décollement, aspiration, expiration)

Les Graviticônes s'appuient sur un modèle de fichiers qui préserve la cohérence pendant la copie : en plus d'une *taille*, chaque fichier possède un *emplacement* variable qui peut être soit un répertoire soit un processus de transfert, et une *vitesse* de transfert. Des animations de transition sur l'*emplacement* (propriété de référence) permettent à la fois d'assurer la continuité visuelle et d'évoquer la *taille* et la *vitesse* (propriétés rappelées) par des déformations et des variations sur la durée de l'animation.

Les Mobilistes

Couramment employés dans les IHMs, les composants « liste » peuvent comporter des éléments hétérogènes, comme des fichiers et des dossiers. Ils peuvent également être réordonnés suite à un calcul ou un événement. Visuellement, ces modifications peuvent être exposées :

- *Sans transition*, comme dans les listes classiques. De par leur immédiateté, les changements d'ordre peuvent être difficile à interpréter [17].
- Par *transitions simples*, ou pseudo-différenciées lorsque chaque type d'élément possède un rendu différent. On peut animer de façon fluide des déplacements d'éléments, comme dans *DigiStrips* [17]. La nature des changements est plus claire : l'animation crée un lien continu entre l'état de départ et d'arrivée.
- Par *transitions différenciées*, par exemple en animant différemment les déplacements d'éléments selon leur type. La nature du réordonnement est visible, ainsi que les types des objets concernés. Il s'agit d'une technique de type *Mobiliste*.

Nous avons employé des Mobilistes dans le cadre d'un projet pour le Centre Expérimental d'Eurocontrol, qui réalise des travaux de recherche et de développement pour améliorer la gestion du trafic aérien en Europe.

Sur un aéroport, différents engins, appelés « mobiles » évoluent : des avions, des bus et d'autres véhicules pour le kérosène ou les plateaux repas. Pour se rendre sur leurs lieux d'intervention, ces mobiles sont amenés à traverser les pistes. Ils doivent alors demander une autorisation au contrôleur. L'autorisation peut être donnée en fonction de l'encombrement de la piste, de l'heure de traversée prévue et du type de mobile. Pour aider le contrôleur à gérer les traversées, les mobiles peuvent être présentés sous forme de liste, triée par heure de traversée, affichant les informations suivantes :

Identifiant	Tâche	Type de mobile	Heure de traversée
AF6128	Taxiing		10.10

Figure 6 : rendu d'un élément de type avion

Un événement tel qu'une panne de bus peut entraîner un re-calcule des heures de traversée. Il est important pour le contrôleur d'être averti des réordonnements et du type des mobiles concernés. Par exemple, s'il s'apprête à autoriser un avion à décoller juste après la traversée prévue d'un bus et qu'il apprend que ce bus est retardé, il doit se raviser pour avancer le départ. Or l'attention du contrôleur est la plupart du temps concentrée vers l'extérieur. Par exemple, il doit s'assurer que l'avion est au point d'arrêt avant de l'autoriser à décoller. Il est donc important qu'à ce moment-là l'interface attire son attention sur le fait qu'un réordonnement a lieu, et si possible, qu'il concerne un bus.

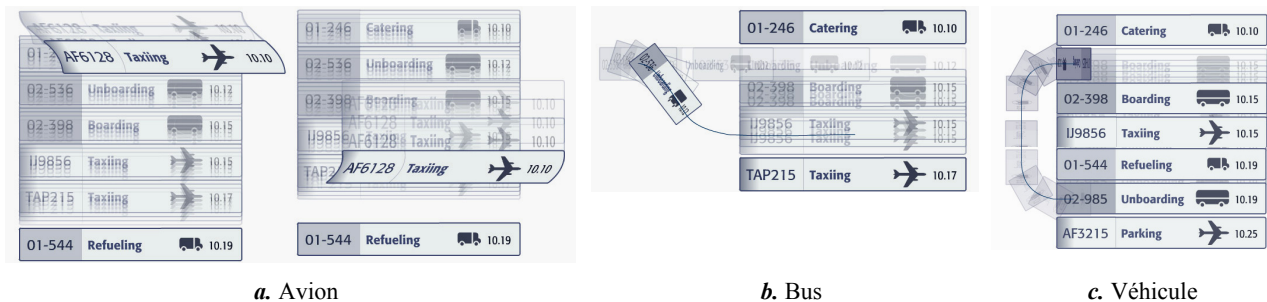


Figure 7 : transitions différenciées sur des déplacements dans une Mobiliste.

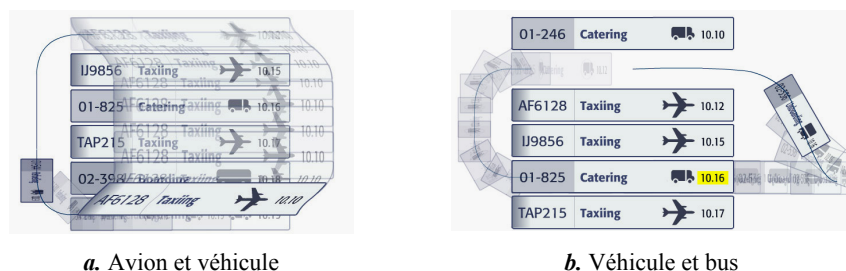


Figure 8 : combinaisons de transitions différenciées sur des inversions dans une Mobiliste.

Pour visualiser les changements, nous employons des transitions différenciées de *renforcement*. Le type de mobile est rappelé par un chemin et une déformation caractéristique. Ainsi, un avion va « survoler » la liste par la droite (figure 7a), un bus va se compresser, « faire une marche arrière » vers la gauche puis s'insérer à sa position (figure 7b), et un véhicule va se compacter puis faire un demi tour par la gauche jusqu'à sa position (figure 7c).

Si plusieurs mobiles sont concernés par le changement, les transitions sont combinées par application de temporisations et par modification des chemins si nécessaire. Par exemple, pour une inversion de deux mobiles, le mobile le plus haut dans la liste part en premier, puis l'élément qu'il remplace amorce son déplacement (figure 8a). Pour l'inversion d'un véhicule et d'un bus, la trajectoire du bus est modifiée (figure 8b).

CONCEVOIR DES TRANSITIONS DIFFERENCIÉES

Dans cette partie, nous proposons des pistes méthodologiques pour concevoir des transitions différenciées. Nous introduisons d'abord des propriétés souhaitables pour ces transitions, sur la base du modèle conceptuel décrit précédemment. Nous proposons ensuite un ensemble de règles de conception en nous appuyant sur ces propriétés.

Propriétés des transitions différenciées

Observabilité. L'observabilité est la capacité d'un système à rendre perceptibles à l'utilisateur les variables d'état internes [1]. Les interfaces utilisant des transitions différenciées comportent deux niveaux d'observabilité :

- l'*observabilité statique*, qui concerne toute propriété rendue
- l'*observabilité transitoire*, qui ne concerne que la propriété rappelée lors d'une transition différenciée.

Honnêteté. Par rapport à l'observabilité, l'honnêteté suppose en outre que le rendu de l'état du système est correctement interprété par l'utilisateur [20]. Nous pouvons identifier deux types de « malhonnêteté » :

- la *malhonnêteté de transition*, inhérente aux transitions simples et différenciées qui montrent une évolution de la propriété de référence qui n'a pas réellement lieu dans le système
- la *malhonnêteté de rappel*, dans le cas où des animations mènent à des interprétations imprécises ou erronées sur la valeur de la propriété rappelée.

Continuité. Bien qu'elles ne rendent pas bien compte de la réalité perceptive de la continuité visuelle, il est utile d'introduire deux propriétés de continuité reposant sur la notion de rendu :

- la *continuité* d'une transition simple ou différenciée caractérise la fluidité de cette transition lorsqu'elle se déroule dans le temps. Elle est liée à la continuité de la fonction de rendu transitoire selon le temps, et à son bon raccord avec le rendu statique : $T(p_0, p_1, 0) = R(p_0)$ et $T(p_0, p_1, t_1 - t_0) = R(p_1)$, voir figure 1.
- la *continuité du rappel* pour une transition différenciée caractérise la fluidité visuelle de cette transition lorsqu'on fait uniquement varier la propriété rappelée. Elle est liée à la continuité de la fonction de rendu transitoire $T(q, p_0, p_1, t - t_0)$ selon q .

Etapes préliminaires

En amont de la conception visuelle des transitions différenciées se pose la question de leur utilisation pertinente, et celle de l'approche de conception appropriée. Nous les abordons ici.

Quand utiliser des transitions différenciées ? Dès l'analyse de la tâche, il est important de se demander quelles informations sont pertinentes pour l'utilisateur, et à quels moments [20]. L'*observabilité statique* doit être vérifiée pour toutes les informations qui sont pertinentes à tout moment : ces informations feront partie des propriétés rendues.

Ensuite, l'analyse de la tâche a dû permettre d'identifier les moments où d'autres informations sont importantes. Dans la phase de conception, on peut choisir de rendre ces informations *observables statiquement*, ou bien de façon *transitoire* lorsque le moment coïncide avec une modification d'informations déjà visibles. Il s'agit par exemple du cas où une action de l'utilisateur ou un événement du système produit un retour visuel : on pourra alors choisir de rappeler les informations utiles en remplaçant ce changement visuel instantané par une transition différenciée. Ce choix sera d'autant plus pertinent que la discontinuité visuelle est importante.

En outre, on peut choisir de *renforcer* des informations déjà visibles mais encore plus utiles au moment étudié : c'est le cas par exemple dans les Mobilistes, où l'on renforce l'information sur le type de mobile par une transition différenciée. Le renforcement est également indiqué

lorsqu'une transition visuelle réduit temporairement la lisibilité de l'information (un texte, par exemple).

Quelle approche choisir ? S'il existe potentiellement de nombreuses manières de concevoir et de spécifier des transitions différenciées, elles s'inscrivent toutes dans l'une ou l'autre de ces deux approches : l'approche *par cas* et celle *par paramétrisation*.

L'approche *par cas* consiste à produire un nombre fini de variantes d'animations, une par valeur rappelée ou par domaine de valeurs si ce nombre est trop grand. Cette approche se prête bien aux propriétés rappelées qui possèdent un petit nombre de valeurs, et permet de concevoir avec soin chaque animation.

L'approche *par paramétrisation* consiste à paramétrer l'animation en fonction de la propriété rappelée. Elle peut être employée si le nombre de cas est trop grand. Elle permet également de jouer plus finement sur les subtilités de l'animation dans le cas d'une propriété continue, mais laisse moins de liberté créative que la précédente.

Concevoir des transitions différenciées efficaces

S'il est difficile de donner des règles exhaustives pour concevoir des transitions différenciées efficaces, quelques éléments peuvent cependant servir de guide.

Jouer sur les trajectoires. Les trajectoires possibles pour une transition sont liées à la manière dont est rendue la propriété de référence. Deux types de situations peuvent être distinguées : soit cette propriété est codée de manière non spatiale (par une couleur, par exemple), soit elle est codée de manière spatiale (par une position, par exemple).

Dans le cas d'une représentation non spatiale, le rappel peut s'effectuer par des variations sur la dynamique de la transition : vitesse, effets d'accélération et de décélération, oscillations. Pour les transitions sur des propriétés qui conduisent au déplacement de l'objet ou de l'une de ses parties, s'ajoutent à ces effets les variations sur les trajectoires spatiales qui peuvent être combinés avec des déformations locales.

Les transitions sur les propriétés non spatiales peuvent également être enrichies par des déplacements et des déformations (vibrations, pulsations,...). En effet, les trajectoires spatiales sont visuellement les plus discriminantes [4] et offrent les moyens d'expression les plus riches pour les transitions différenciées. Quant aux déformations, elles peuvent enrichir le mouvement et constituer une illustration métaphorique rappelant efficacement la propriété rappelée [19].

Privilégier la continuité. La représentation continue et la persistance des objets d'intérêt [23, 20] sont des propriétés essentielles qui doivent être également vérifiées

lors des transitions [26, 17]. De manière générale, l'impression de cohérence et de réalité (solidity) produite par un objet animé est fortement altérée par des *discontinuités de transition*, telles que des mises à jour lentes, des apparitions et disparitions magiques, ou des images-clés trop espacées [6].

S'il est important d'assurer la *continuité de transition*, la *continuité de rappel* est également souhaitable si la propriété rappelée varie rapidement et de manière continue. Dans une approche *par cas* cependant, il est préférable de figer la propriété rappelée lors des transitions, afin d'éviter des *discontinuités de rappel* potentiellement perturbatrices pour l'utilisateur.

Rester honnête. Pour prévenir les *malhonnêtetés de transition*, les animations doivent être conçues de manière à ce qu'elles ne soient pas confondues avec des évolutions réelles du système. Dans les Mobilistes par exemple, la nature métaphorique des animations est claire (figure 7).

L'approche *par cas* pour les transitions différenciées peut également susciter deux formes de *malhonnêteté de rappel*. Si chaque cas réunit plusieurs valeurs possibles de la propriété rappelée (discrétisation), l'utilisateur peut surestimer la précision de la valeur rappelée. D'autre part, si la propriété rappelée est variable mais figée pendant les transitions (voir section précédente), l'utilisateur peut se tromper sur les moments où se produisent les changements.

De manière générale, les *malhonnêtetés de transition* et de *rappel* peuvent être réduites si elles sont rendues explicites. Par ailleurs, elles sont d'autant plus tolérables qu'elles sont temporaires, c'est-à-dire que les transitions sont de courte durée.

Exploiter les apparitions et disparitions. L'apparition et la disparition d'objets peuvent être vues comme des évolutions d'une propriété codant leur visibilité. Il est alors conseillé d'appliquer des transitions d'arrivée et de départ lors de ces changements, pour donner une réalité plus tangible à l'objet [6]. Par ailleurs, jouer sur les arrivées et les départs permet de fournir des informations supplémentaires sur l'objet à des moments cruciaux de son cycle de vie [17].

Bien combiner les transitions. Une difficulté potentielle dans la conception de transitions différenciées est la combinaison d'animations, en particulier lorsque plusieurs transitions se déclenchent simultanément : par exemple, lorsqu'une Mobiliste subit une modification importante. La technique des Mobilistes montre comment les transitions peuvent être enchaînées ou décalées pour éviter la superposition de mouvements et offrir une meilleure lisibilité, et comment des modifications peuvent être apportées aux trajectoires spatiales pour éviter les chevauchements (figure 8).

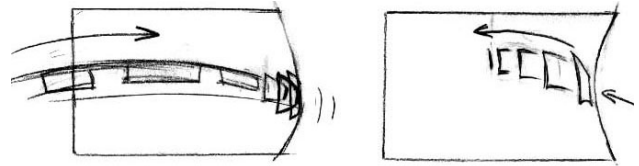


Figure 9 : story-board d'animatrice illustrant un effet d'apparition « fracassante » d'un objet

Le rôle de l'animateur. Les règles de conception pourront difficilement se substituer à l'expertise d'un animateur professionnel pour la conception d'animations efficaces, en particulier lorsqu'il s'agit de prendre en compte des contraintes aussi complexes que celles résultant des transitions différenciées. En effet, si les graphistes maîtrisent des méthodes éprouvées par les artistes depuis des siècles leur permettant de produire des visuels efficaces [10], les animateurs maîtrisent quant à eux les techniques du dessin animé et les effets associés [29, 6]. Ils peuvent ainsi illustrer et proposer les meilleurs effets en fonction de la propriété à rappeler, et appliquer des effets de type dessin animé pour renforcer une métaphore (figure 9).

Travailler avec un animateur. La collaboration avec les designers dans la conception et le développement de systèmes interactifs nécessite des méthodes qui tiennent compte de leur spécialité [22], et des environnements de prototypage et de développement qui intègrent leurs outils [10]. Mais les solutions logicielles pour incorporer dans des applications réelles des animations créées avec des outils auteur sont limitées. Une piste prometteuse est celle des techniques d'interpolation avancées, qui peuvent permettre de produire des transitions paramétrées à partir de cas-clés illustrés par l'animateur dans des storyboards ou des maquettes Flash [11]. L'environnement de développement devrait également permettre à l'animateur d'ajuster ses transitions après avoir les avoir testées en contexte dans l'application finale.

TRAVAUX ANTERIEURS

En dehors de leurs emplois purement narratifs (présentations multimédia, EAO), les animations ont été utilisées à diverses fins dans les IHM [3]. Trois types d'application sont en rapport avec notre contribution : les animations de transition, les animations de notification, et les animations de manipulation directe. Nous les évoquons ici. Puis, nous décrivons les travaux de modélisation des animations, et enfin les guides de conception.

Animations de transition

Jusqu'à-là confiné aux applications de présentation et de montage vidéo, le concept d'animation de transition est devenu populaire dans les interfaces [2]. Historiquement employée pour l'animation de programmes et d'algorithmes [16], elle a également été exploitée dans des systèmes Post-WIMP tels que les documents et les tableurs fluides [7, 14]. Chang a été le premier à propo-

ser l'emploi de techniques de dessin animé pour rendre les transitions plus convaincantes [6]. Mais ces approches n'exploitent pas la multiplicité des transitions possibles. Des exemples de transitions différenciées existent cependant, comme dans les mouvements humanisés des pièces du jeu BattleChess [15]. DigiStrips emploie également des apparitions et disparitions différenciées pour les menus, afin de rappeler l'objet sur lequel ils s'appliquent [17].

Alarmes

Un usage apparenté des animations concerne la notification d'événements, et en particulier les alarmes. Bien que l'informativité d'une alarme soit traditionnellement faible [28], on a montré qu'il était possible de jouer sur les signaux visuels (taille, forme, fréquence) pour transporter des informations supplémentaires sur la nature de l'alarme et sa cause [9]. Si la notion de « notifications différenciées » n'est donc pas nouvelle, celle de transitions différenciées en constitue un cas particulier qui a reçu peu d'attention.

Manipulation directe animée

L'animation a également été appliquée à la manipulation directe. En s'inspirant de techniques de dessin animé, Thomas a proposé d'accentuer le réalisme des objets manipulés en leur attribuant de l'élasticité et de l'inertie [26]. Mais il s'intéresse peu à la différenciation, si ce n'est pour expliciter les contraintes physiques entre les objets. Ce type de différenciation a néanmoins été utilisé comme moyen d'enrichir et faciliter la coordination dans les collecticiels à manipulation directe : par exemple, la trajectoire empruntée par un document lors d'un échange peut évoquer un degré d'urgence [24].

Modélisations

Les études sur la perception d'animations reposent sur des modèles très partiels de l'animation, chacune se concentrant sur quelques paramètres isolés [27]. A l'inverse, beaucoup d'auteurs ont cherché à fournir des modèles de programmation de l'animation en étendant les modèles graphiques à objets avec des chemins géométriques [25], des contraintes temporelles [12] ou des flots de données [8]. Mais ces modèles ont une couverture soit trop restreinte soit trop vaste pour affirmer qu'ils donnent un modèle de l'animation. La cause en est que l'animation est un phénomène perceptif et cognitif et non un concept informatisable [18]. C'est pourquoi, ayant besoin ici d'un modèle générateur, nous avons préféré opter pour une autre voie, en étendant la modélisation des représentations graphiques de données proposée par Bertin [5], sans chercher à modéliser l'animation.

Règles de conception

Il existe peu de règles de conception pour les animations. Lorsqu'il s'agit de capter l'attention, un corpus limité de règles peut se déduire des études sur l'efficacité des dimensions isolées de l'animation [27, 9]. Mais dès lors que les animations deviennent plus complexes pour se voir attribuer un rôle communicatif et métaphorique, les connaissances deviennent plus vagues. On peut néan-

moins citer des principes de réalisme physique tels que la solidité, l'exagération, le renforcement [6] et la fluidité [26]. Un certain nombre de techniques sont en outre exposées dans des ouvrages destinés aux animateurs professionnels, mais elles concernent essentiellement l'animation de personnages [29].

CONCLUSION

Dans cet article, nous avons introduit et défini les transitions visuelles différenciées comme nouveau moyen de coder de l'information dans une transition en jouant sur les trajectoires visuelles. Des exemples ont permis d'illustrer l'utilisation de transitions différenciées dans deux types d'application : les applications asynchrones tels que les systèmes de contrôle et les collecticiels où il est nécessaire d'améliorer la conscience de la situation [9], et les applications synchrones où il est utile d'améliorer le retour visuel sur les actions de l'utilisateur [17]. Nous avons ensuite proposé des pistes pour concevoir des transitions différenciées efficaces, en énonçant des règles pratiques qui intègrent des propriétés telles que l'observabilité, l'honnêteté et la continuité, et en évoquant l'importance du rôle de l'animateur professionnel.

Outre le caractère affectif [6] et esthétique des transitions différenciées conçues par des animateurs, celles-ci peuvent être exploitées comme un nouveau moyen de réduire la densité des informations affichées. En effet, avec l'informatique moderne la quantité d'informations et de tâches que doit gérer un utilisateur augmente exponentiellement. Les transitions différenciées autorisent une approche où seules les informations toujours indispensables sont rendues visibles, et d'autres informations pertinentes sont rappelées à des moments clés. Dans la mesure où ce rappel exploite des variations visuelles existantes, il est très économe en bande passante. La densité d'informations affichées peut ainsi être diminuée efficacement sans nécessité de navigation dans les données, ou d'ajout de nouvelles modalités comme la modalité sonore.

Les transitions différenciées peuvent également être un moyen d'améliorer la conscience de la situation [13]. En effet, l'utilisation d'animations, notamment en vision périphérique, améliore la *perception* de changements [4]. Par ailleurs, en codant ou en renforçant des informations pertinentes pour une tâche en cours, les transitions différenciées peuvent aider l'utilisateur à *comprendre* ces changements.

Cet article reste cependant essentiellement exploratoire et des expérimentations formelles doivent être réalisées pour valider ou invalider les apports des transitions différenciées. Dans le cadre du projet ANIMS, nous préparons actuellement des expérimentations sur les Mobilistes pour évaluer les apports du codage d'informations non visibles, du renforcement des informations déjà visibles, et des effets des transitions différenciées sur la conscience de la situation.

REMERCIEMENTS

Cette étude a été en grande partie réalisée dans le cadre du projet ANIMS (Animations and Sounds in Air Traffic Control User Interfaces) avec le Centre Expérimental d'Eurocontrol.

BIBLIOGRAPHIE

1. Abowd G.D., Coutaz J., Nigay L., *Structuring the Space of Interactive System Properties*, Actes de EHCI'92, Elsevier, pp. 113-128
2. Apple, Mac OS X Tiger. <http://www.apple.com/>
3. Baecker, R., Small, I., *Animation at the interface*, In B.Laurel, *The Art of Human-Computer Interface Design*, Addison-Wesley, New York, 1990.
4. Bartram, L., *Moving icons: detection and distraction*, Actes d'Interact'01.
5. Bertin, J., *La sémiologie graphique*, Editions Mouton, Gauthier-Villars, Paris, 1967.
6. Chang, B.-W., Ungar, D. *Animation: From Cartoon to User Interface*, Actes de UIST'93, pp. 45-55
7. Chang, B.-W., Mackinlay, J.D., Zellweger, P.T., Igarashi, T., *A Negotiation Architecture for Fluid Documents*, Actes de UIST'98, pp. 123-132
8. Chatty, S. *Réaliser des interfaces homme-machine animées*, Thèse de doctorat de l'Université Paris Sud, 1992.
9. Chatty, S., Athènes, S., Bustico A., *Votre attention s'il vous plaît. Eléments d'un espace de conception de signaux visuels*, Actes d'IHM'99, pp.17-24
10. Chatty, S., Sire, S., Vinot, J.-L., Lecoanet, P., Lemort, A., Mertz, C., *Revisiting visual interface programming : creating GUI tools for designers and programmers*, Actes de UIST'04, pp. 267-276
11. Dragicevic, P., Chatty, S., Thevenin, D., Vinot, J.L., *Artistic Resizing: a technique for rich scale-sensitive vector graphics*, Actes de UIST'05, pp. 201-210
12. Duisberg, R.A. *Animation using temporal constraints: An overview of the Animus system*, Human-Computer Interaction, 3(3), 1987-1988, pp 275-307
13. Endsley, M. R., *Toward a theory of situation awareness in dynamic systems*, Human Factors Journal, Vol. 37(1), pp. 32-64.
14. Igarashi, T., Mackinlay, J.D., Chang, B.-W., Zellweger, P.T., *Fluid Visualization of Spreadsheet Structures*, Actes de Visual Languages'99, pp. 118-125
15. Interplay, Battle Chess, <http://www.interplay.com/games/product.asp?GameID=325>
16. Kerren, A, Stasko, J., *Algorithm Animation, Introduction de Software Visualization*, State of the Art Survey, LNCS 2269, Springer, 2002, pp. 1-15.
17. Mertz, C., Chatty, S., Vinot, J.-L., *Pushing the limits of ATC user interface design beyond S&M interaction: the DigiStrips Experience*, Actes d'ATM'00
18. Mertz, C., Chatty, S., *Intactile Design, State of the art on sound and animation in Human Machine Interfaces*, Rapport de recherche TECH04-017, Eurocontrol, 2004.
19. Norman, D.A., *The Psychology of Everyday Things*, Basic Books, New York, 1988, p.23
20. Salber, S, Coutaz, J., Decouchant, D., Riveill, M., *De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la Communication Homme-Homme Médiatisée*, Actes d'IHM'95, pp. 27-34.
21. Saulnier, A., *La perception du mouvement dans les systèmes de visualisation d'informations*, Actes de la conférence IHM'05, pp. 185-192
22. Schlienger, C., Valès, S., Chatty, S., *Une expérience de conception et de prototypage d'interfaces évoluées dans le milieu industriel*, Actes d'IHM'04, pp. 165-172
23. Schneiderman, B., *Direct Manipulation: A Step Beyond Programming Languages*, IEEE Computer, 16(8), 1983, pp. 57-69.
24. Sire, S., *La collaboration directe. Un paradigme d'interaction pour le travail collaboratif assisté par ordinateur*. Thèse de doctorat de l'université Toulouse 1, 2000.
25. Stasko, J.T., *The Path-transition Paradigm: a Practical Methodology for Adding Animation to Program Interfaces*, Journal of Visual Languages and Computing, 1990, pp. 213-236.
26. Thomas, B.H., Calder, P., *Applying Cartoon Animation Techniques to Graphical User Interfaces*, ACM Transactions on Computer-Human Interaction, Vol. 8, No. 3, September 2001, pp198-222.
27. Ware, C., *Information Visualization: Perception for Design*, Morgan Kaufmann, San Francisco, 2000.
28. Wickens, C., *Engineering Psychology and Human Performance*, 2nd Ed. Harper Collins, New York, 1992.
29. Williams, R. *Techniques d'animation pour le dessin animé, l'animation 3D et le jeu vidéo*, Eyrolles, 1998.