Quelle ergonomie pour le drag & drop ?

Auteur : Amélie Boucher
Date de publication : 02.11.2008

Article à retrouver sur :
www.ergolab.net/articles/ergonomie-drag-drop.php

Sommaire :

Introduction
1. Le drag & drop doit avoir une réelle utilité, augmenter l'efficience
2. La possibilité de drag & drop doit être visible
3. La destination doit être compréhensible et visible
4. Le drag & drop doit être facile
5. Le déplacement doit être visible
6. L'effet du drag & drop doit être visible
Conclusion

Avec la présence de plus en plus marquée des interfaces riches sur le web, on observe le transfert de modes d'interaction à l'origine réservés aux systèmes d'exploitation ou au monde logiciel. C'est le cas du drag & drop (en français, glisser - déposer).

Dans ce type d'interfaces, **l'utilisateur saisit un objet, puis le déplace pour l'amener à un autre endroit**.

Nous présenterons dans cet article les caractéristiques du drag & drop, ses avantages et inconvénients, mais surtout comment répondre à ces derniers pour profiter des promesses de ce mode d'interaction.

Le propos de cet article est de souligner que **le drag & drop ne doit être employé que s'il est utile pour l'internaute**, et non parce qu'il est possible de le faire techniquement. De plus, **une fois qu'il est présent, de nombreuses règles sont à appliquer pour qu'il soit utilisable**.

Notre réflexion intègre des explications et exemples issus du monde logiciel ou système, dans l'optique de questionner son utilité et son utilisabilité dans des interfaces web. La limite entre ces différents mondes étant parfois très fine, il paraît essentiel de souligner la manière dont le drag & drop peut être utilisé à plus ou moins bon escient selon le contexte.

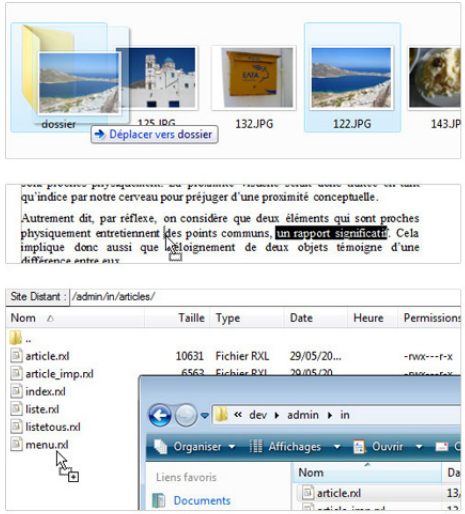
Introduction

Au quotidien, nous utilisons énormément le drag and drop, sans forcément nous en rendre compte.

C'est surtout le cas dans nos systèmes d'exploitation : déplacement de fichiers d'une fenêtre à une autre, redimensionnement de colonnes, éjection d'un DVD, etc. C'est aussi le cas dans de nombreux logiciels : glisser - déposer de blocs de textes dans un traitement de texte, glisser - déposer des fichiers pour les télécharger dans un logiciel de ftp, ré-organisation des palettes dans un logiciel graphique, etc.

Si l'on revient aux basiques, déplacer des objets dans des applications de dessin vectoriel, de mise en page ou de présentation, c'est aussi effectuer une action de drag & drop.

Ce type d'interface relève fortement de la **manipulation directe** : on agit directement sur l'objet que l'on souhaite traiter, sans passer par un intermédiaire symbolisant l'action que l'on souhaite effectuer sur cet objet. Mayhew dit à ce propose que dans ce type de mode d'interaction, "au lieu de le dire, les utilisateurs le font, simplement" (*instead of "saying it", users simply "do it"*, Mayhew, 1992, cf. lectures complémentaires).



Déplacer une image dans un dossier, déplacer un paragraphe de texte à un autre endroit de la page, uploader un fichier dans un logiciel de FTP, sont autant d'actions qui peuvent être effectuées par glisser - déposer.

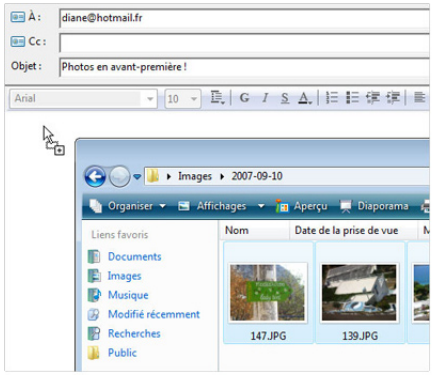
Il paraît logique de profiter du fait que les utilisateurs familiers avec l'informatique connaissent ce type de procédé. En effet, on peut ainsi observer des **courbes d'apprentissage très courtes** (étant donné que je sais exécuter le geste moteur sous-jacent au drag & drop, il suffit presque que je comprenne que l'on peut glisser - déposer des éléments pour avoir appris). Encore faut-il recourir au drag & drop de manière réfléchie.

Il existe de nombreux contextes dans lesquels le drag & drop devient **un réel outil d'optimisation des interactions**.

Comparez par exemple une interface d'ajout de fichiers attachés dans Gmail, et la même interface dans Windows Mail :



Exemple d'attachement de fichiers dans un e-mail sous Gmail : je joins mes 3 photos 1 à 1 (à noter: une extension Firefox existe qui permet de contourner ce système).



Exemple d'attachement de fichiers dans un e-mail sous Windows Mail : je joins mes 3 photos en 1 seule action.

Il est indéniable que le drag & drop a ici de grands bénéfices, surtout dans le cadre de multi-attachements de fichiers. Son utilisation demande toutefois de modifier complètement la procédure mentale consistant à joindre un fichier : dans un cas, l'utilisateur commence par déclarer l'action qu'il souhaite effectuer, dans l'autre il commence par l'objet qu'il souhaite joindre.

Remarquons que Windows Mail supporte à la fois l'entrée par l'action (« clic sur l'icône joindre un fichier -> ouverture de la fenêtre d'exploration du système ») et l'entrée par l'objet / les objets à attacher (« ouverture d'une fenêtre d'exploration -> glisser - déposer dans l'e-mail »).

Cet exemple vise à souligner que le drag & drop peut faciliter la réalisation de certaines actions. Cependant, il peut aussi la compliquer. Tout dépend du contexte et des choix qui sont effectués concernant l'ergonomie de ces interfaces.

Whiteside & al soulignent à cet égard que **le seul choix d'un style de dialogue** homme-machine (ex: le drag & drop) **ne peut pas être le déterminant principal de l'utilisabilité** d'une application.

Ainsi, de bons (ou mauvais) choix de conception peuvent contrecarrer les défauts (ou annuler les bénéfices) potentiels d'un style de dialogue (ref. dans Mayhew, 1992, cf. lectures complémentaires).

La suite de cet article détaille dans quels cas le drag & drop s'avère le plus pertinent, et les bonnes pratiques à appliquer pour exploiter ses bénéfices. Nous allons ainsi voir que 3 grandes dimensions sont capitales : utilité, visibilité et feedback.

1. Le drag & drop doit avoir une réelle utilité, augmenter l'efficience

Pour que le drag & drop soit pertinent, encore faut-il qu'il aide l'internaute. Or, on remarque que ce mode d'interaction est souvent utilisé comme un gadget dans les sites web. Autrement dit, les responsables du site décident d'intégrer cette fonctionnalité parce qu'ils la trouvent "cool", "sexy", "innovante"... Dans ce genre de cas, on part d'une possibilité technique pour la plugger sur un site web, sans se soucier de son intérêt sur le site en question.

Le premier point à évaluer, c'est donc l'utilité du drag & drop. Il est indéniable que ce dernier permet, dans certains contextes, d'optimiser les interactions avec le site. Cependant, il peut aussi les alourdir.

Sur le web, on distingue aujourd'hui **2 grandes types d'objectifs** pour lesquels est utilisé le drag & drop. Le déplacement de l'objet est toujours nécessaire, puisqu'il appartient à la définition même du mode d'interaction (par "drag", on sous-entend forcément un déplacement). Cependant, il est utilisé à des fins différentes :

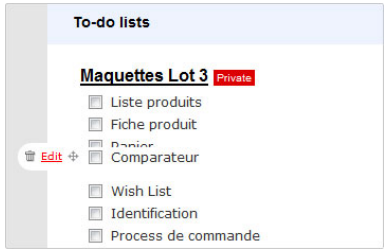
» Soit pour **lancer une action** sur un objet (ex. ajout au panier). Dans ce cas, **le déplacement = un moyen pour lancer une action**. L'objet "référent" reste à son emplacement initial, mais il peut être dupliqué symboliquement.

» Soit pour **déplacer un objet** (ex. passage d'un groupe à un autre, ordonnancement de listes, etc.). Pour ce type d'objectifs, **le déplacement = la finalité**. On n'a qu'une instance de chaque objet.
On remarque en général que **le drag & drop est particulièrement adapté pour supporter ces actions de pur déplacement** d'un item. Il est alors très utile puisqu'il mime parfaitement l'action désirée. Le déplacement est par définition un type d'action pour lequel il semble logique de saisir l'objet pour l'amener ailleurs (comme dans le monde réel).

Illustrons par des exemples ces deux manières d'exploiter le glisser - déposer, et leurs implications en termes d'utilisabilité.

1.1. Exemple positif : efficience optimisée

Commençons par une application positive, où le drag & drop démontre toute son utilité en termes d'efficience utilisateur :



La fonction de drag & drop est très intéressante pour l'ordonnancement de listes.

Ici, un exemple sur l'interface de BaseCamp, où les items peuvent être glissés - déposés aux emplacements désirés. Ce procédé permet d'optimiser l'action d'ordonnancement au regard d'un comportement html plus classique consistant à indiquer un chiffre correspondant à l'emplacement souhaité, ou à déclarer l'élément à déplacer, un emplacement de destination, puis à lancer l'action en cliquant sur un bouton.

Source : www.basecamp.com

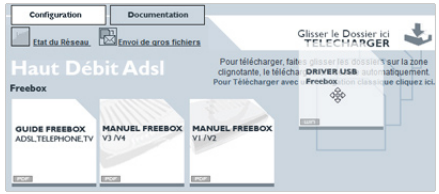
Comme c'est le cas dans cet exemple, le drag & drop est intéressant lorsqu'il permet à l'utilisateur d'**atteindre plus simplement ou plus rapidement son objectif**.

Attention, le simple fait d'utiliser le drag & drop pour faciliter des actions de déplacement d'objets n'implique pas que ce soit le mode d'interaction le plus adapté. Le niveau d'efficience obtenu est en effet très dépendant de la visibilité directe de l'élément à déplacer, ainsi que de sa destination.

Passons à un exemple où le drag & drop vient complexifier ou gêner l'atteinte des objectifs utilisateur.

1.2. Exemple négatif : efficience diminuée

Les contextes où l'utilisation d'un système de drag & drop est inappropriée sont nombreux sur le web. C'est le cas dans l'exemple suivant, où le drag & drop complique la réalisation de la tâche :

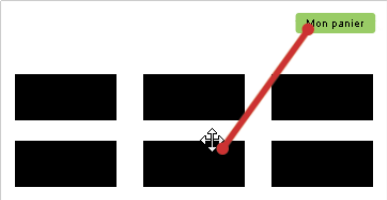


La fonction de drag & drop peut compliquer la réalisation d'une action.

Ici, un exemple sur l'ancienne interface du site de Free, où les manuels d'aide doivent être téléchargés en utilisant le glisser - déposer. Ce procédé s'avère plus complexe qu'un clic sur un lien "Télécharger". En effet, l'internaute doit d'abord comprendre le fonctionnement de l'interface en analysant la page et en lisant la légende. Ensuite, il doit procéder au drag & drop, ce qui n'est pas forcément évident pour tous les internautes et nécessite plus d'actions qu'un simple clic (cf. schéma ci-dessous).

Source : ancien site www.free.fr

Dans le cas d'un système de glisser-déposer d'éléments uniques, voici ce que le style de dialogue implique par rapport à une interface plus classique :



Décomposition de l'action effectuée en drag & drop : clic sur l'objet, parcours de la distance objet / récepteur, dépôt de l'objet sur le récepteur.
Plus la distance objet / récepteur est longue, plus l'action est longue à effectuer, voire compliquée d'un point de vue moteur.
Avantage : l'interface est légère visuellement.



Décomposition de l'action effectuée via un bouton : clic sur le bouton.
Inconvénient : l'interface est plus lourde visuellement (il faut répéter le bouton autant de fois qu'il y a d'objets).

En fait, dans la plupart des exemples de lancement d'actions par un drag & drop, **on oublie l'efficience** (clé de l'utilisabilité) **en ne réfléchissant qu'à l'efficacité** (base de l'utilisabilité). Certes, une interface d'ajout au panier me permet d'ajouter un article au panier, mais à quel coût utilisateur en termes de difficulté et de temps de réalisation ?

Plus précisément, lorsque je ne peux pas effectuer de **multi-sélection**, il y a peu de chances que le drag & drop soit plus efficient qu'un simple bouton d'action. Lorsque la multi-sélection est possible, le drag & drop a d'autant plus de chances d'optimiser l'efficience qu'il y a d'objets sélectionnés simultanément.

Cela explique que le drag & drop a une plus grande vocation à être utilisé dans des **interfaces à forte récurrence**, que leurs utilisateurs utilisent fréquemment (c'est notamment le cas des logiciels web, des outils, des sites web "applicatifs"), dans lesquelles ils traitent beaucoup d'objets, et bien sûr uniquement **lorsqu'il facilite la tâche**.

Il semble que plus le web évolue, plus il se destine à accueillir un grand nombre de nos applications. De nombreux sites web sont de véritables outils, et non seulement des "bases de connaissances" dans lesquelles on se contente de circuler. Le drag & drop peut alors être envisagé, mais avec les précautions qui s'imposent. Pour conclure sur ce point, on considèrera que **le drag & drop est utile quand il permet à l'utilisateur d'être efficient**.

Notons que si la fonctionnalité du drag & drop est imposée, mais que vous remarquez qu'elle pose des problèmes d'efficience, il est nécessaire de **la doubler par une alternative plus classique**.

C'est par exemple ce que fait Panic pour l'ajout au panier :



Sur le site de Panic, on peut ajouter un produit au panier soit par glisser - déposer, soit plus classiquement par clic sur un bouton.
Ainsi, on ne risque pas d'empêcher les internautes d'ajouter des produits à leur panier, ou de leur compliquer la tâche.

Source : www.panic.com/goods

Pour finir, ce n'est pas parce que le système de drag & drop peut augmenter l'efficience qu'il l'augmente effectivement. Comme nous l'avons déjà évoqué, **le style de dialogue n'est qu'un potentialisateur**. Ainsi, en fonction des choix d'interface, le drag & drop peut être plus ou moins difficile à utiliser, et, de fait, augmenter ou diminuer l'efficience de l'internaute.

La suite de cet article s'attache donc à décortiquer les bonnes pratiques à mettre en place pour profiter des bénéfices du drag & drop, lorsqu'on est certain que c'est le mode d'interaction adapté.

2. La possibilité de drag & drop doit être visible

Certaines des fonctionnalités reposant sur le drag & drop nous paraissent plus intuitives que d'autres. Nous avons vu que les actions de déplacement sont celles pour lesquelles il semble le plus naturel de pouvoir glisser - déposer des objets. Pour les autres types d'action, on peut optimiser l'interface pour aider l'internaute à prendre en main le système.

Il faut en effet ajouter aux critères inhérents à la fonction elle-même les **choix d'interface, qui vont jouer sur le niveau d'opacité du drag & drop** pour l'utilisateur. Les internautes ont besoin de deux connaissances afin d'avoir l'idée d'utiliser un système de glisser-déposer : savoir qu'ils peuvent glisser-déposer des objets, et savoir quels objets peuvent être glissés-déposés. Les formats de présentation, terminologies et comportements au survol peuvent aider les internautes dans ce sens. Il est capital de les optimiser, surtout lorsqu'on n'offre pas d'autre alternative que le drag & drop pour réaliser l'action.

2.1. Savoir que l'on peut glisser-déposer des objets

Un des défauts fréquents du drag & drop est que les utilisateurs ne voient pas qu'il est possible. Dans ce type de situation, on perd alors toute la puissance attendue de ce mode d'interaction, puisque les utilisateurs sont face à une impasse.

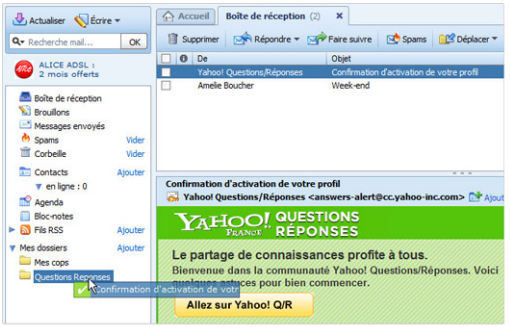
Ce point pose la question de l'apprentissage : une interface reposant exclusivement sur des procédures de glisser - déposer doit nécessairement être découverte pour en exploiter le potentiel. Elle doit donc, par son apparence et / ou son fonctionnement, **inciter l'internaute à comprendre ce qu'il peut faire** ou ce que l'on attend de lui. L'enjeu consiste donc à ce qu'il comprenne le rapport entre l'interface de départ et ses possibilités.

À l'inverse d'une interaction sur la base de menus ou de boutons, où l'action est représentée par des mots, l'action effectuée lors d'un drag & drop est représentée par des éléments moins explicites :

» Soit uniquement par l'action de déplacement (je déplace un objet pour le mettre à un autre endroit, l'objectif de l'action, c'est simplement le déplacement)
» Soit par une caractéristique symbolique du récepteur (je déplace un objet dans une corbeille, cela signifie que je le jette).

Lorsque l'interface **mime quelque chose que les utilisateurs connaissent déjà** dans le monde logiciel, cela pose peu de problèmes. Le plus souvent, ils essaient par réflexe de faire ce qu'ils font déjà sur d'autres interfaces comparables.

C'est par exemple le cas sur Yahoo! Mail, où je peux déplacer un message en le glissant dans un des dossiers en colonne de gauche (comme sur un logiciel classique):



Yahoo! Mail permet d'effectuer des actions sur les messages via un système de drag & drop. Les internautes le font spontanément lorsqu'ils ont déjà utilisé le glisser - déposer sur un référent logiciel agencé de la même manière en termes d'interface.

Notons que des relais plus classiques (cases à cocher + boutons) permettent d'effectuer les mêmes actions.

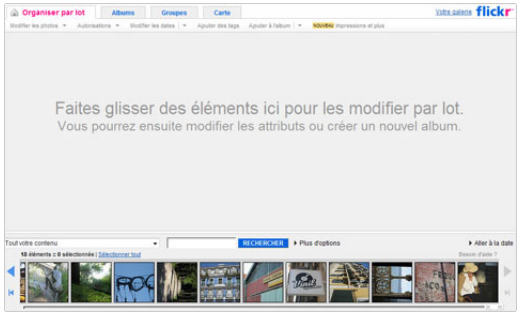
Source : http://fr.mail.yahoo.com

Dès que l'on sort d'une interface mimant un existant et qui peut donc capitaliser sur les habitudes des utilisateurs, cela devient plus compliqué. Pour que l'internaute comprenne qu'il existe une possibilité de drag & drop, le plus efficace aujourd'hui reste encore d'**être le plus explicite possible**.

Utiliser les mots semble donc un des meilleurs moyens d'y parvenir, que ces derniers soient affichés en permanence ou à des moments choisis du flux d'interaction.

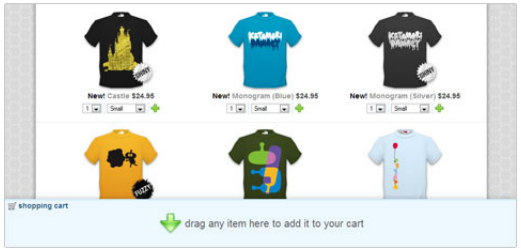
2.1.1. Message adossé au récepteur

L'aide explicite permettant de suggérer la possibilité de glisser - déposer des objets est souvent liée à un **état particulier du récepteur**, notamment **lorsqu'il n'a pas encore été utilisé**. En voici 2 exemples :



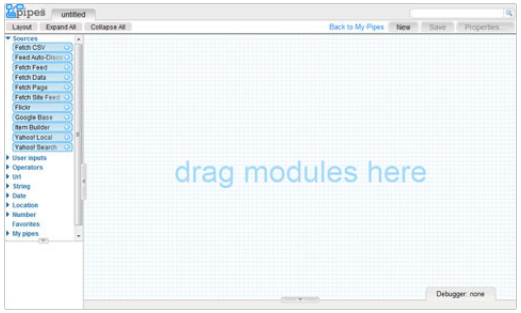
Lorsqu'on arrive sur l'organiseur par lot de Flickr pour la première fois, un message incitatif nous aide à comprendre la manière dont on peut utiliser cet outil.

Source : www.flickr.com



Même chose chez Panic, où l'interface du panier vide explique qu'il faut déposer des articles à cet endroit pour les ajouter au panier.

Source : www.panic.com/goods



Et enfin sur Pipes, où l'on invite l'internaute à glisser des modules dans l'espace de travail.

Source : http://pipes.yahoo.com

Lorsqu'on utilise ce principe, il faut veiller à **adapter la visibilité du message à l'importance de la fonction** : plus le drag & drop est nécessaire pour que l'internaute puisse utiliser le service, plus le message accompagnant le récepteur doit être visible.

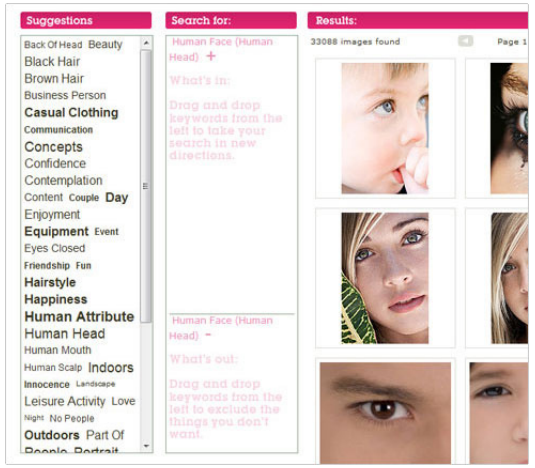
Le message d'incitation peut aussi être affiché **au survol du récepteur**. C'est par exemple le cas dans l'interface suivante :



Chez SFR Mes Contenus, le survol des différents récepteurs lorsqu'ils sont vides donne lieu à l'affichage de messages d'incitation.

Source : www.sfr.fr/communiquer/mes-contenus

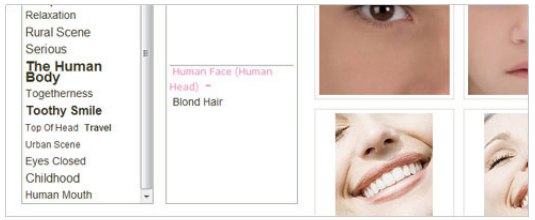
Notons qu'**une fois que le drag & drop a été utilisé une première fois**, donc qu'il est appris, **l'interface peut redevenir plus « light »**, et moins forte en termes d'incitation. C'est par exemple le cas sur l'outil Catalyst de Getty Image :



Un message d'incitation accompagne les récepteurs lorsque ces derniers sont vides.

(Soulignons d'ailleurs qu'il n'est pas assez visible pour les utilisateurs qui ne connaissent pas le système).

Source : www.gettyimages.com/Catalyst



Lorsque j'ai déjà utilisé le drag & drop, ce message disparaît : j'ai compris comment fonctionne le système, je n'ai donc plus besoin d'autant de guidage.

Enfin, le message adossé au récepteur peut aussi venir accompagner dynamiquement l'internaute, et le supporter dans ses tentatives, comme le fait Flickr dans l'exemple ci-dessous :



Dans l'organiseur par lot de Flickr, si je commence à glisser une photo dans l'organiseur, la galerie est totalement remplacée par un message qui me confirme qu'en déposant la photo à cet endroit, je peux la supprimer de l'organiseur.

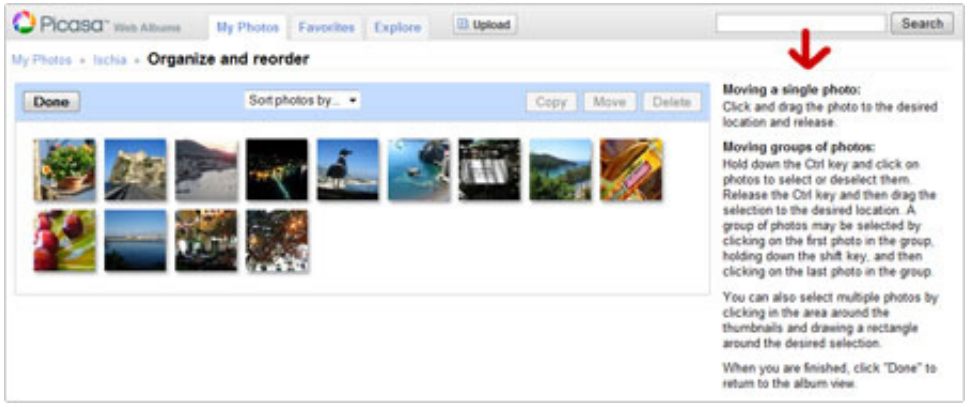
Notons que ce type de comportement n'est possible que parce que les deux seules actions qui peuvent être effectuées sont de déplacer des photos de la galerie à l'organiseur par lot, et inversement.

Source : www.flickr.com

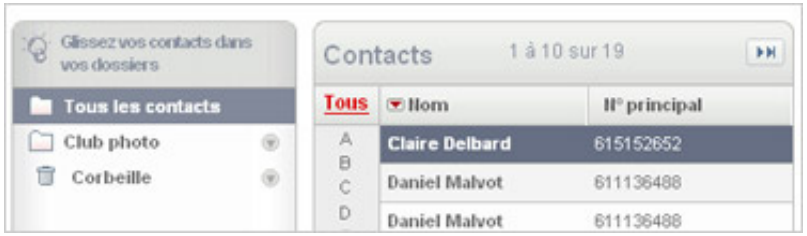
2.1.2. Aide permanente

Pour inciter l'internaute à comprendre qu'il peut glisser - déposer des éléments, on peut aussi utiliser un message d'aide permanent. Il faut alors que ce dernier soit suffisamment fort et bien placé pour être vu par l'internaute.

Cependant, ce besoin de grande visibilité entre en compétition avec les autres éléments de la page. Les défauts de ce type de message sont donc souvent qu'ils doivent nécessairement respecter un compromis (visibilité du message vs force visuelle), et que, de ce fait, ils perdent la puissance d'un message très fort.



Sur Picasa, toute la colonne de droite est réservée à un message permanent attirant l'attention sur la possibilité de glisser - déposer.
On en profite pour fournir des explications pour la multi-sélection.
Source : http://picasaweb.google.com



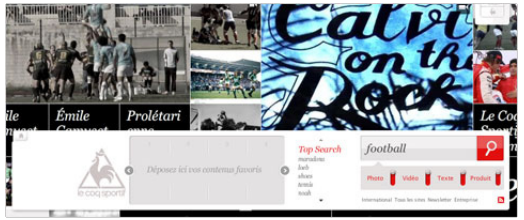
Sur l'outil Mes Contacts de SFR, un message permanent m'indique que je peux glisser mes contacts dans des dossiers.
Source : www.sfr.fr/communiquer/mes-contacts

Notons que si la question de la visibilité est critique, elle soulève celle du **modèle d'interaction unique** : si je ne peux atteindre mon objectif que par un glisser - déposer, mais que je ne le vois ou que je ne le comprends pas, j'échoue.

Il peut alors être utile de fournir une **alternative au drag & drop** qui permette d'effectuer la même action de manière plus explicite (en général, sous la forme d'une décomposition d'actions représentée par des liens ou boutons - pour reprendre l'exemple de Yahoo! Mail, je peux déplacer un message par drag & drop directement dans un dossier, ou cliquer sur le bouton déplacer, puis choisir le dossier qui m'intéresse). En outre, cela permet en général de disposer d'un modèle plus classique et, de fait, plus accessible.

Si le drag & drop n'est proposé que sur une fonction complémentaire, il est certes moins important que la fonction soit décrite très explicitement afin de guider l'internaute sur la voie de l'apprentissage. Cependant, rappelons qu'une interface qui accompagne bien l'internaute a aussi un **plus grand pouvoir suggestif**, et peut ainsi maximiser le taux d'utilisation de certaines fonctionnalités.

Ainsi, dans l'exemple suivant, si je m'aperçois de la possibilité de glisser - déposer mes contenus préférés dans une "boîte à favoris", cela peut me donner envie d'utiliser la fonction alors que je n'y aurais pas forcément pensé de moi-même.



Sur le site du Coq Sportif, l'importance du message "Déposez ici vos contenus favoris" a une valeur suggestive en plus d'une valeur explicative.

Source : www.lecoqsportif.com

Se rendre compte de la possibilité de drag & drop, cela peut aussi passer par le fait de se rendre compte que certaines classes d'objet sont déplaçables.

2.2. Savoir ce qui peut être déplacé

Avant de commencer à détailler cette recommandation, notons qu'elle doit être adaptée au contexte. En effet, la visibilité de la possibilité de drag & drop (cf. point précédent) peut compenser un manque de visibilité directe de chaque objet qui peut être déplacé.

Par exemple, si le récepteur indique bien que l'on peut y déposer des choses, et qu'il n'y a qu'un type d'objets potentiellement déplaçables dans l'interface, ces objets peuvent se passer d'un symbole indiquant qu'il peuvent être déplacés. Dans le cas contraire, il faut faire passer le message à l'internaute via les objets eux-mêmes.

L'action de **déplacer un objet d'un endroit à un autre** est une bonne **métaphore** de ce que l'on effectue dans le monde réel. Cependant, l'interface informatique pose la problématique de savoir communiquer à l'utilisateur ce qui est (ou non) déplaçable. Alors que dans le monde physique, tout objet que l'on voit est palpable, saisissable, actionnable, ce n'est pas le cas à l'écran.

Dans une interface informatique, ce qui va nous aider à différencier un objet que l'on peut saisir d'un objet que l'on ne peut pas saisir est lié à la notion d'affordance perçue.

2.2.1. La notion d'affordance perçue

Les affordances d'un objet, au sens large, ce sont les possibilités d'action qu'un individu peut entretenir avec cet objet. Ce concept tel que proposé par Gibson dans les années 50 s'applique aux objets du monde physique.

Norman popularise cette notion en l'appliquant au design d'interfaces, et en l'adaptant aux besoins de la discipline. Il précise en effet après avoir d'abord utilisé le simple terme d'affordance que le plus approprié consiste à parler d'**affordances perçues**, référant aux **possibilités d'action *suggérées* par l'objet au vu de ses caractéristiques de forme** (et non plus aux possibilités d'action réelles).

Sur le web, si je vois un lien ou un bouton, le libellé (ou au pire, le contexte) vient m'indiquer ce qui m'attend si j'actionne l'objet. Dans le cas du drag & drop, on peut considérer que ce type d'interaction n'offre pas d'affordance perçue automatique.

Autrement dit, puisque tout le signifiant est contenu dans l'action, rien ne me dit **a priori** que je peux effectuer cette action. Il faut donc **accompagner l'internaute** pour qu'il accède à la **découverte** et donc à l'utilisation potentielle du système.

2.2.2. Visibilité du caractère déplaçable

Pour éviter qu'un utilisateur ne voie pas la possibilité de réaliser des actions en drag & drop, il faut **augmenter l'affordance perçue de déplacement** des objets déplaçables. Cependant, cela nécessite d'être très vigilant. En effet, la traduction graphique de cette recommandation peut vite mener à une interface très chargée visuellement.

Il faut donc adapter le niveau d'affordance perçue de déplacement au contexte : si l'on dispose déjà d'un moyen "général" de faire découvrir la possibilité de drag & drop à l'internaute (cf. le point abordé précédemment), le caractère déplaçable de chaque objet ne doit pas nécessairement être exagéré.

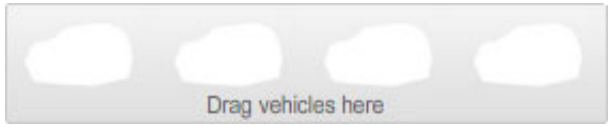
La solution la plus courante consiste à **afficher le caractère déplaçable au survol de l'objet**. Cela présente l'avantage de donner l'information, tout en évitant de charger l'interface visuellement.

L'objectif est que l'internaute comprenne que les objets déplaçables sont différents des objets non déplaçables. Dans l'idéal, **il faut qu'il comprenne qu'il peut les déplacer**. Plusieurs choses peuvent l'y aider :

» Utiliser une icône ou une forme de curseur indiquant conventionnellement la possibilité de glisser - déposer (un pointeur en forme de croix ou de main serrée, cette dernière étant plutôt utilisée pour les interfaces où l'on déplace le focus).

» Utiliser du texte pour indiquer ce que l'on peut glisser - déposer (par exemple, sur le site de General Motors, le libellé "*drag* ***vehicles*** *here*" permet de comprendre que ce sont les véhicules qui sont déplaçables).

» Éventuellement, **design du récepteur**. Par exemple, toujours sur le site de General Motors, on voit légèrement des silhouettes de voitures dans le récepteur. Cela facilite la compréhension du mécanisme d'ensemble.



Source : www.gm.com

3. La destination doit être compréhensible et visible

3.1. L'effet symbolisé par la destination doit être compréhensible

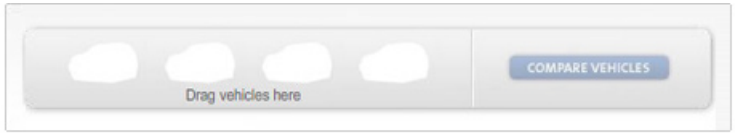
Pour optimiser un système de glisser - déposer, il faut faciliter sa compréhension par l'utilisateur : comprend-on la signification du drag & drop d'un objet d'un endroit à un autre? **ce que cela permet de faire ?** L'effet du drag & drop doit être prédictible, le plus proche des attentes ou besoins de l'utilisateur.

3.1.1. Information donnée avant l'action

Les **messages adossés aux récepteurs** doivent aider dans ce sens. On peut remarquer qu'ils sont en général plus efficaces ou immédiatement compréhensibles quand ils sont en deux parties, présentant à la fois l'action physique à effectuer et la conséquence qui s'ensuivra :

Exemple : "déposer **pour** ajouter au panier"
Exemple sur Flickr : "Faites glisser des éléments ici **pour** les modifier par lot."
Exemple sur Panic : "Drag any item here to add it **to** your cart."
Exemple sur Getty Images : "Drag and drop keywords from the left **to** exclude the things you don't want."

Dans le cas où l'action ne se fait qu'a posteriori sur l'ensemble des objets contenus dans le récepteur, le **design de ce récepteur** peut aussi aider à comprendre la relation entre le groupe d'objets déposés, et une action qui peut être lancée sur l'ensemble. C'est par exemple le cas pour la comparaison de véhicules sur le site de General Motors :

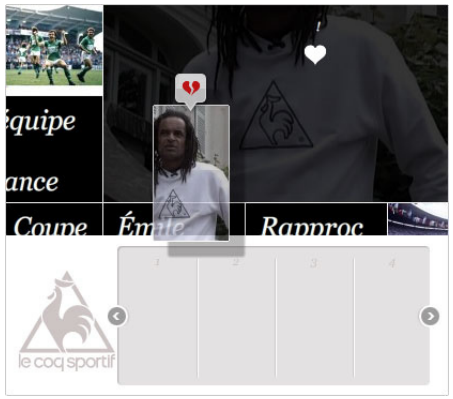


Source : www.gm.com

3.1.2. Information donnée après l'action

La conséquence du drag & drop peut aussi être **affichée en cours d'interaction**, même si le message a alors **moins d'impact**. Notamment, il ne peut pas suggérer de commencer à glisser un objet.

Ce type de message peut toutefois **accompagner l'internaute dans ses actions de manière assez fine** et pertinente. Pour illustration, cet exemple sur le site du Coq Sportif :



Sur le site du Coq Sportif, si je commence à glisser un de mes objets favoris hors du récepteur "mes favoris", comme si je voulais l'en sortir, on affiche une icône représentant un cœur brisé.

Cela a du sens, puisque les objets favoris sont indiqués par une icône de coeur. On traduit donc symboliquement l'action qui est en train d'être menée par l'utilisateur.

Source : www.lecoqsportif.com

3.2. La destination doit être visible

Dans une interface utilisant le drag & drop, on doit voir qu'un objet est déposable à un ou à plusieurs endroits. De manière plus précise, on doit pouvoir accéder à deux types d'informations, selon le moment de l'interaction :

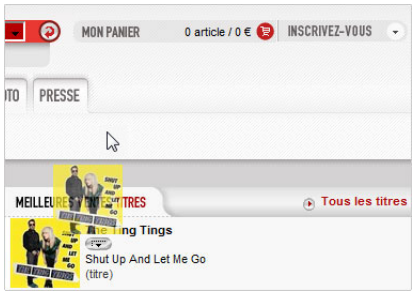
» Savoir (à l'avance) où on peut déposer
» Savoir si on peut dropper ce qu'on est en train de déplacer.

3.2.1. Savoir (à l'avance) où on peut déposer

En général, dans les interfaces utilisant le drag & drop pour déplacer des objets, la question de savoir où on peut déposer est peu importante. Dans ce type de cas, les utilisateurs comprennent de manière intuitive où ils peuvent déposer des objets, puisque la destination représente exactement l'endroit où ils veulent voir l'objet (exemples : déplacement de rdv dans un calendrier, ordonnancement de listes, etc.).

Lorsque déposer un objet à un endroit n'est qu'un moyen de lancer une action (ex: ajout au panier), il est important de faire plus d'efforts en termes de guidage pour l'internaute. En général, les **messages adossés aux récepteurs** jouent le rôle d'indicateurs. Ils ont en effet d'abord pour fonction initiale d'indiquer l'endroit où les objets peuvent être déposés.

Même si le récepteur n'indique pas explicitement qu'il est une destination pour l'action de drag & drop, il peut prendre une **apparence particulière quand un objet est saisi**, aidant à comprendre que les deux peuvent entrer en relation.



Sur le site de Virgin Mega, lorsque je commence à déplacer un article, le panier "clignote" pour indiquer son rôle de récepteur.

Source : www.virginmega.fr

Notons qu'il faut **éviter d'exagérer cette mise en avant des destinations**, surtout lorsqu'elles sont multiples, car la charge mentale et visuelle peut alors être très gênante.

3.2.2. Savoir si on peut déposer ce qu'on est en train de déplacer

Lorsque je suis en train de glisser un objet, **toute zone survolée candidate au dépôt doit être highlightée**.

Ce **feedback visuel** est très important, et permet à l'utilisateur d'utiliser le drag & drop de manière plus confortable.

3.2.2.1. Indiquer la possibilité plutôt que l'interdiction

Nous recommandons d'**indiquer où un objet est déposable, bien plus qu'où il n'est pas déposable**. En effet, cette indication peut être mal interprétée par les utilisateurs. Du reste, il n'en ont pas besoin. En effet, à partir du moment où l'on indique les destinations permises, il va de soi que lorsque l'indicateur n'est pas affiché, c'est que l'on ne peut pas déposer l'objet.

Cette recommandation vise à **limiter le nombre de signifiants** pour alléger la charge mentale liée au traitement de l'action de glisser - déposer. Elle est encore plus importante lorsque l'indicateur de « non-déposabilité » prête à confusion. Regardez cet exemple d'une application Flex permettant de glisser des éléments d'une liste à une autre :



Lorsque je survole une zone où je ne peux pas déposer mon item, une icône s'affiche. Or, cette dernière indique conventionnellement la suppression, et m'interroge donc sur ce qui peut se passer si je lâche l'item à ce moment précis : Va-t-on supprimer l'item ?

Source : http://www.adobe.com/devnet/flex/quickstart/adding_drag_and_drop/#list

3.2.2.2. Bonus : donner un aperçu de l'effet

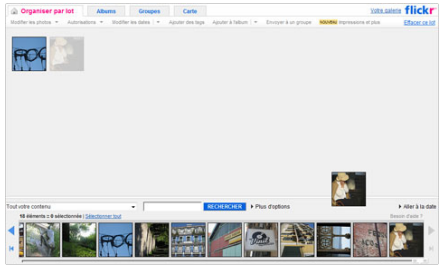
On peut optimiser le simple feedback visuel par une information plus précise. Bien sûr, cela doit être réservé aux applications où cela a du sens.

La plupart du temps, cela consiste à donner un aperçu de ce que donnera l'interface si l'on procède au dépôt. C'est notamment ce que font Netvibes et Flickr :



Sur Netvibes, on me donne une indication précise de l'endroit où je peux dropper, mais aussi une représentation exacte de la "place" que prendra l'objet à cet endroit de l'écran.

Source : www.netvibes.com



Chez Flickr, on me montre 2 choses : un duplicata de l'image que je déplace, et une représentation transparente de son état "prévu" lorsque je l'aurais déposée : on m'indique ainsi l'endroit où va être placée mon image.

En prime, cela peut m'aider à comprendre que je n'ai pas besoin d'aller la mettre tout en haut, ce qui réduit la difficulté motrice et le temps d'exécution du glisser - déposer.

Source : www.flickr.com

4. Le drag & drop doit être facile

Une des difficultés du drag & drop réside dans le fait qu'il implique des actions motrices plus compliquées qu'un simple clic. Cela touche les novices comme les plus expérimentés des utilisateurs, même s'ils n'éprouvent pas le même type de difficultés.

Si on rappelle la **loi de Fitts**, qui énonce qu'***une cible est d'autant plus rapide à atteindre qu'elle est proche et grande***, on peut alors trouver des moyens d'optimiser le drag & drop.

4.1. Plus le récepteur est proche, plus le drag & drop est facile

Pour commencer, le drag & drop sera d'autant plus facile que la destination sera proche de l'objet. Sur le web, on recommande donc de **positionner le récepteur dans un panneau flottant**, qui vient se poser sur l'interface et reste au même endroit quelque soit la position du curseur dans la page.

Cela explique pourquoi le drag & drop est plus facile dans l'exemple du site de Panic (panier toujours disponible en bas de l'écran) que dans celui de Virgin Mega (le panier, situé dans le header, disparaît si je scrolle dans la page). On remarque d'ailleurs que lorsque le récepteur n'est pas directement disponible à l'écran, les utilisateurs hésitent à lancer une action de drag & drop.

Faites-en l'expérience : effectuer une opération de drag & drop dans un système d'exploitation est **quelque chose qui se prépare** : en effet, les deux espaces de travail (celui d'où je prends l'objet cible, et celui où je souhaite le déposer) doivent être **visibles simultanément à l'écran**, sans quoi il me sera impossible de procéder au glisser-déposer.

De micro-ajustements sont possibles (notamment dans le cas de hiérarchies de dossiers, où le survol d'un dossier peut permettre de l'ouvrir, pour déposer l'objet dans un sous-dossier. C'est notamment le cas dans Outlook). Cependant, ces occasions de contourner la nécessaire simultanéité du drag and drop sont rares.

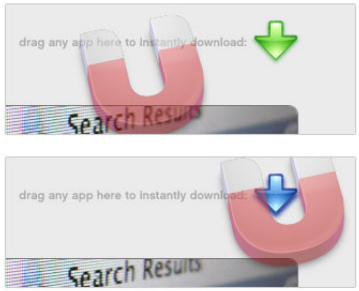
Notons que quelques pistes de recherche explorent la voie d'un **drag & drop par étapes** (sélection, pause, drop), pour limiter ces difficultés liées aux distances virtuelles entre point de départ et point d'arrivée.

4.2. Plus le récepteur est grand / flexible, plus le drag & drop est facile

La deuxième partie de la loi de Fitts nous aide à recommander de prévoir des zones de réception suffisamment grandes pour faciliter le drag & drop.

4.3. Plus le récepteur est logique, plus le drag & drop est facile

De manière complémentaire, la zone de réception doit sembler naturelle à l'internaute.
Ci-dessous, un contre-exemple sur le site de Panic :



Sur le site de Panic, le message indique "drag here", mais la véritable zone de réception est limitée à la flèche.

On induit donc l'utilisateur en erreur. Il aurait été préférable de rendre l'ensemble de la zone (message + flèche) active.
En prime, cela aurait permis d'augmenter la taille de la zone de réception, respectant par là-même la seconde partie de la loi de Fitts.

Source : www.panic.com/goods

4.4. Plus la zone préhensible est étendue / logique, plus le drag & drop est facile

Enfin, le drag & drop sera d'autant plus facile que l'objet sera facilement préhensible.

Dans la plupart des cas, il faut donc veiller à prévoir que l'ensemble de l'objet puisse donner lieu à une action de saisie (et non uniquement une "poignée").

5. Le déplacement doit être visible

Une interface de drag and drop ne peut être réussie que si le système fournit un **feedback informant l'utilisateur** sur le fait **qu'il est bien en train de déplacer un élément**. Cela permet à ce dernier d'éviter une erreur (du type "je crois déplacer un objet alors que je l'ai mal saisi, donc que je ne déplace rien"), d'avoir un retour en temps réel sur l'efficacité de son action de déplacement, et enfin de pouvoir contrôler avec précision son action en vue de la dépose de l'objet à sa destination.

Pour indiquer le fait qu'un élément est en train d'être déplacé, on peut recourir classiquement à deux types de représentations.

5.1. Représenter l'objet déplacé en tant que tel

Lorsqu'on utilise ce procédé, l'objet peut être représenté sous plusieurs formes :

» **Le même objet en transparence**, en tout cas de manière dégradée. Cela permet de comprendre l'état « instable » de l'objet, et de continuer à avoir une vue de l'interface située en-dessous.

» **Le même objet en plus petit**, sous forme de vignettes. C'est par exemple ce qu'utilise General Motors sur son site :

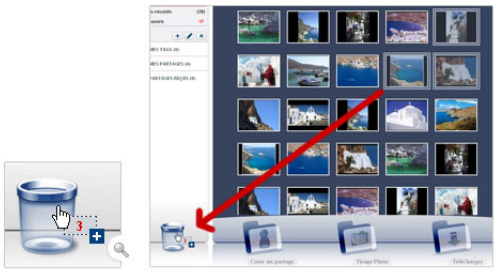


La voiture en cours de déplacement est représentée par son duplicata en miniature.
Source : www.gm.com

5.2. Utiliser un représentant symbolique de l'objet

L'objet **peut aussi être représenté sous la forme d'un symbole**. Cela permet de limiter l'intrusion de l'objet déplacé dans l'interface de base, et la charge visuelle liée au déplacement de l'objet. Tant que je n'ai pas relâché le bouton de la souris, l'objet reste à sa place initiale.

C'est par exemple le cas lorsqu'on utilise un filet horizontal pour représenter une ligne d'une liste, un rectangle pour représenter un bloc de texte en déplacement dans un traitement de texte, ou un chiffre pour représenter le nombre d'éléments en train d'être glissés :



Sur l'application SFR Mes Contenus, les items en déplacement sont figurés par une incrémentation de chiffres.

Les éléments de départ ont une apparence particulière, soulignant leur caractère "sélectionné".

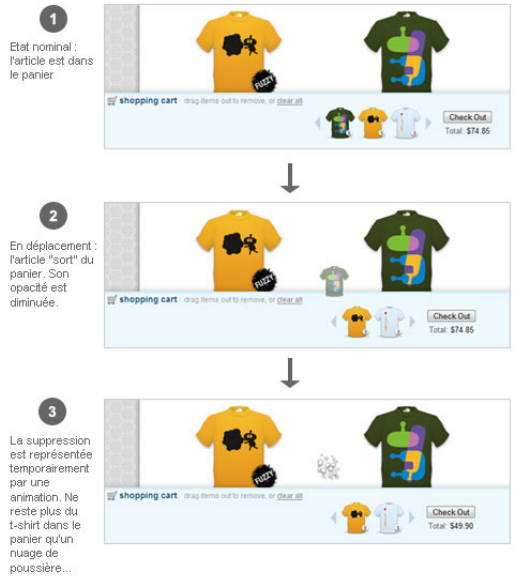
Source : www.sfr.fr/communiquer/mes-contenus

6. L'effet du drag & drop doit être visible

La dernière grande recommandation consiste à optimiser la visibilité de la conséquence du drag & drop lorsqu'il a été réalisé. Cette recommandation est valable pour les interfaces où le drag & drop sert à déplacer une "instance" d'un élément (ex: ordonnancement d'une liste vs drag panier) plutôt qu'à déplacer un élément. En effet, dans ce dernier cas, l'effet du drag & drop est auto-explicite.

Pour souligner l'effet du drag & drop, on doit utiliser soit un **duplicata de l'objet**, soit un **représentant symbolique** (une incrémentation de chiffres dans un micro-panier par exemple).

On peut y ajouter un **effet temporaire** pour représenter l'action qui vient d'être effectuée. Voyons ainsi comment Panic représente l'effet consécutif au glisser - déposer d'un article du panier vers la liste :



Source : www.panic.com/goods

On peut observer un comportement similaire chez Flickr:



Source : www.flickr.com

Conclusion

L'ensemble des recommandations évoquées dans cet article visent à **profiter des avantages indéniables que présentent les interfaces de drag & drop, tout en limitant les défauts** que nous leur connaissons.

Soulignons une dernière fois qu'il est préférable de ne pas recourir au drag & drop « pour le drag & drop », mais plutôt parce que ce mode d'interaction nous permet de simplifier voire d'optimiser le travail de l'utilisateur.

Il est évidemment acceptable de vouloir exploiter le côté séduisant, innovant, dynamique du drag & drop. Cependant, il faudra alors penser à le doubler d'un mode d'interaction plus classique, ou répondre strictement à chacune des recommandations citées dans cet article. L'idée étant de ne pas compliquer la tâche de l'internaute sous prétexte d'avoir voulu utiliser du drag & drop !

Pour en savoir plus

» Usability by Hand, AJAX and Efficiency, Thomas Baekdal, Baekdal.com (2006).

» Drag ‘n Drop is Invisible To Users, Jared Spool, User Interface Engineering (2005).

» When to use drag & drop (some informal research results), Leisa Reichelt, Disambiguity (2006).

» Drag and Drop Design Pattern Library, Yahoo! Developer Network

» Principles and Guidelines in Software User Interface Design, Deborah J. Mayhew, Prentice Hall PTR (1991). Chapitre 9 : Dialog Styles : Direct Manipulation.

» Boomerang: suspendable drag-and-drop interactions based on a throw-and-catch metaphor, Kobayashi, M. and Igarashi, T. (2007). In Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology (Newport, Rhode Island, USA, October 07 - 10, 2007). UIST ‘07. ACM, New York, NY, 187-190