La dimension sémantique, négligée de l'approche expérience-utilisateur

The semantic dimension, neglected by the user-experience approach

Yves Marcoux et Élias Rizkallah

Édition électronique

URL : <http://journals.openedition.org/edc/5418>

DOI : 10.4000/edc.5418

ISSN : 2101-0366

Éditeur

Université Lille-3

Édition imprimée

Date de publication : 1 décembre 2013

Pagination : 119-138

ISBN : 978-2-917562-10-9

ISSN : 1270-6841

Référence électronique

Yves Marcoux et Élias Rizkallah, « La dimension sémantique, négligée de l'approche expérience-utilisateur », *Études de communication* [En ligne], 41 | 2013, mis en ligne le 01 décembre 2013, consulté le 20 avril 2019. URL : <http://journals.openedition.org/edc/5418> ; DOI : 10.4000/edc.5418

YVES MARCOUX ET ÉLIAS RIZKALLAH

LA DIMENSION SÉMANTIQUE, NÉGLIGÉE DE L'APPROCHE EXPÉRIENCE-UTILISATEUR

Introduction et aperçu

Un truc pour remplir à l'ordinateur un formulaire qui n'est disponible que sous forme papier est d'utiliser une image numérisée du formulaire comme fond de page dans un document de logiciel de traitement de textes (Word, etc.), puis de positionner visuellement aux endroits appropriés les segments de texte voulus. Dans la version française de Microsoft Word 2010, on insère une image comme fond de page en :

1. sélectionnant l'onglet « Mise en page » ;
2. repérant le groupe « Arrière-plan de page » ;
3. activant l'item « Couleur de page » (illustré avec son infobulle à la Figure 1) ;
4. choisissant l'item « Motifs et textures... » au sein du sous-menu « Couleur de page » (illustré à la Figure 2) ;
5. activant l'onglet « Image » au sein de la fenêtre « Motifs et textures » ;
6. cliquant sur le bouton « Sélectionner une image... ».

Figure 1 : Item « Couleur de page » et son infobulle

Figure 2 : Sous-menu « Couleur de page »

Le nombre d'étapes peut sembler élevé, mais il est courant dans une interface aussi complexe. Ce qui est problématique est le fait qu'au moins deux étapes sont « cachées » derrière une façade langagière inexacte (donc mensongère), qui doit être « percée » par l'utilisateur.

Ces barrières sont le libellé et l'infobulle du bouton « Couleur de page », et le libellé de l'item « Motifs et textures... »¹. Rien ne laisse présager que d'autres choix sont possibles, en l'occurrence l'insertion d'une image. L'énorme infobulle, où toute la place pour apporter une précision serait disponible, n'ajoute rien.

La seule façon pour l'utilisateur de vaincre ces barrières est de *ne pas croire ce qui lui est présenté en texte clair* et de faire à certaines étapes de l'interaction des choix ne correspondant pas à ce qu'il veut faire.

Il semble que cet état de choses n'ait jamais été identifié comme problématique, du moins pas assez pour avoir été modifié au fil du temps. Il aurait été facile d'utiliser l'espace disponible dans l'infobulle de l'item « Couleur de page » pour préciser les autres choix possibles, mais cela n'a jamais été fait. Il semble donc acceptable – et accepté – qu'une composante textuelle dans une interface donne une information inexacte.

Malgré la préoccupation pour l'utilisateur manifeste dans l'évolution du design d'interfaces depuis une vingtaine d'années – de la conception de l'*Architecture de l'information* (AI ; Rosenfeld-Morville 1998, Morville-Rosenfeld 2006) à celle de l'*Expérience utilisateur* (UX ; Garrett 2002 & 2011) d'un produit –, les exemples de mauvais design et de faible utilisabilité sont légion et nous sommes tous confrontés régulièrement à la frustration devant des systèmes où l'interface est absconse, se contredit d'une étape à l'autre, ou nous ment de façon éhontée.

Comment en sommes-nous arrivés à considérer normal que, dans le cours d'une interaction, il faille ne pas croire ce qu'un élément d'interface est en train de nous dire en texte clair ?

Nous croyons qu'au moins un élément de réponse réside dans l'importance donnée aux tests utilisateurs en conception d'interface, aux dépens du développement d'outils d'aide au concepteur dans la

1 Selon la convention habituelle, les points de suspension n'indiquent pas la présence de choix additionnels, mais simplement que le dialogue se poursuivra dans une fenêtre subséquente.

partie *idéation* de la conception. Nous postulons qu'une représentation *sémantique* de l'interface en développement basée sur la *langue naturelle* permet au concepteur d'être plus efficace dans la phase d'idéation.

Nous proposons aussi que le monde du design d'interfaces a perdu de vue que la qualité de l'expérience utilisateur est fortement tributaire du succès d'une *communication* entre concepteur et utilisateur, qui doit être porteuse de sens pour les deux. Nous postulons que d'appuyer le travail de conception sur une représentation *sémantique* de l'interface en développement basée sur la langue naturelle peut contribuer à la qualité de l'expérience utilisateur, en particulier pour les *nouveaux utilisateurs* (« first-time users »).

Cet article est un essai qui développe et tente de justifier les postulats ci-dessus, puis présente la *sémantique intertextuelle* (SI ; Marcoux 2006, Marcoux-Rizkallah 2009), basée sur la langue naturelle, et discute la possibilité de l'utiliser dans le processus de conception d'interfaces.

Après un bref argumentaire à l'effet que les tests utilisateurs ne suffisent pas pour le design d'interfaces, nous présentons un découpage du travail de création en cycles *idéation-tests*, que nous illustrons avec l'activité de programmation en génie logiciel, et qui sert de toile de fond à la formulation de nos postulats. Nous présentons ensuite la SI et arguons que de l'utiliser comme base du travail de design d'interfaces est susceptible d'apporter des bénéfices, en particulier d'éviter les cas d'interfaces « mensongères » du type relevé plus haut.

La portée de notre réflexion est celle des systèmes d'information de type « classiques » (et non ludiques ou artistiques), où l'utilisation vise à effectuer une transaction d'« affaires » au sens large.

Insuffisance des tests utilisateurs

Les *tests utilisateurs* consistent à évaluer la performance des artefacts déployés par des expérimentations réalisées avec un échantillon d'utilisateurs dans un contexte de tâches à accomplir réalistes. Dans sa forme originelle, l'idée de centrage sur l'utilisateur – fondamentale tant en AI qu'en UX – n'était qu'indirectement liée aux tests utilisateurs et s'appuyait plutôt surtout sur des principes généraux glanés au fil des ans par des praticiens d'expérience.

Par exemple, l'ouvrage de Norman (1988) consiste surtout en un catalogue d'exemples de bon et mauvais design, d'où sont tirés un certain nombre de principes. D'autres ouvrages, comme celui de Tufte (1995) sur le design de la visualisation (ou, avant lui, celui de Bertin, 1967), partagent la même structure générale. Ces ouvrages n'évoquent que très indirectement (quand ils le font) la notion de test utilisateurs.

Aujourd'hui cependant, les tests utilisateurs revêtent une telle importance en AI et en UX que certains auteurs, comme Krug (2010), les placent au premier plan, allant jusqu'à préconiser essentiellement, comme méthode de design, le mimétisme (reproduire ce que l'on voit ailleurs et qui semble bien marcher) et les tests utilisateurs. D'autres auteurs, cependant, sans nier leur importance, en soulignent les limites (Norman 2005, Molich 2013). On note en particulier que les tests ne peuvent jamais être exhaustifs et que si une dysfonction est constatée, ils ne fournissent pas d'indice direct sur sa cause ni sur la façon de la corriger.

Poussé à l'extrême, le principe de recours à des tests utilisateurs ne laisserait au concepteur aucune autonomie. Toute décision, si infime soit-elle, devrait être prise en évaluant par des tests utilisateurs toutes les options possibles, et en choisissant celle qui obtient le meilleur « score ». Or, la conception d'interface ne fonctionne pas de la sorte : en réalité, certains « bouts de chemin » doivent être parcourus sans test utilisateurs. Seulement à certains moments charnières (plus ou moins nombreux, selon le budget) a-t-on recours à des tests utilisateurs, un peu comme un auteur écrit, relit et amende son texte par lui-même, ne recourant à des lecteurs externes qu'à certains moments-clés.

Comment procède-t-on pour ces « bouts de chemin » pendant lesquels les décisions sont prises de façon autonome ? Les disciplines et la pratique du design offrent peu d'outils pour cette partie du travail de design, que nous appelons la phase *idéation*. Essentiellement, on propose au designer de s'en remettre à son inspiration, à son expérience, à son intuition, au mimétisme et à la consultation de recueils de principes généraux et de bons et mauvais exemples. Une des idées centrales de cet article est que la phase *idéation* peut aussi bénéficier d'une vision *sémantique* de l'artéfact en développement, et que la qualité de l'UX en est rehaussée par une communication sémantiquement adéquate entre concepteur et utilisateur.

La dyade théorie-test en création

Nous proposons de voir toute activité de création (et en particulier la conception d'interface) comme une succession de cycles « idéation-test ». La partie « idéation » du cycle est la phase de création proprement dite, au cours de laquelle le créateur construit ou élabore une hypothèse de ce que pourrait être sa création. Bien qu'intrinsèquement créatrice, cette phase est tout de même tributaire de nombreux intrants, notamment les différents « besoins » du créateur et des utilisateurs ciblés de la création.

La partie « test » consiste à soumettre l'hypothèse de création à une situation réaliste d'utilisation, de façon à voir si elle se comporte de façon escomptée. L'importance relative des deux parties du cycle (idéation et test) et le nombre de cycles dans la réalisation d'une création spécifique variera en fonction de plusieurs facteurs : la tradition disciplinaire du domaine de création, le contexte de réalisation (calendrier, budget, ressources disponibles, etc.), la façon particulière de travailler du créateur, entre autres. Dans certains cas extrêmes la partie test pourrait être inexistante : ce serait le cas, par exemple, d'un artiste qui crée une œuvre qu'il soumet au public sans aucune forme de « validation » ou d'« édition » préalable (un musicien improvisant à la volée et en direct un passage musical serait dans cette situation). La partie idéation, pour sa part, ne semble pas pouvoir être absente, du moins dans les cas qui nous intéressent, où il y a création délibérée de la part d'un ou de plusieurs humains².

Une création complexe peut être divisée en « morceaux » (ou « modules ») qui seront chacun l'objet d'un processus de création individuel comportant plusieurs cycles idéation-test, indépendamment des autres morceaux.

Aide à l'idéation

De quoi le créateur dispose-t-il comme aide à l'idéation ? D'abord, de sa compréhension des besoins des utilisateurs visés par la création. Certainement, il mettra aussi à profit son inspiration, son expérience, son intuition, sa connaissance de ce qui existe déjà et des recueils

2 Ici et plus loin nous parlons « d'un » créateur, mais un groupe peut bien sûr travailler collectivement à une création ; c'est même une situation très courante.

de bons et mauvais exemples. Mais, au-delà de ces évidences, nous suggérons que tout créateur a besoin d'une « théorie » lui permettant de « prévoir » le mieux possible le comportement de sa création en cours d'utilisation.

Dans certaines spécialités du génie, des théories prédictives suffisamment puissantes peuvent orienter précisément l'idéation. Par exemple, pour la construction d'un pont, toute la conception est faite sur la base de théories concernant la résistance des matériaux. Les calculs de dimension sont basés sur ces théories, qui servent également aux simulations du comportement du pont. Ce n'est qu'après la construction que l'on peut procéder à des « tests utilisateurs », qui consistent à faire passer des véhicules.

Un exemple : vérification formelle et tests de programmes

En génie logiciel, le processus de programmation peut être envisagé comme l'élaboration d'un artéfact – le programme en développement – destiné à être interprété selon un formalisme spécifique : le langage de programmation utilisé (l'interprétation est effectuée ici par un environnement matériel automatisé). La « théorie » se résume donc au fait que l'artéfact en développement sera interprété selon la définition du langage de programmation.

La dichotomie théorie-tests est ici constituée des deux techniques que sont la *vérification formelle de programmes* et les *tests de programmes*. En gros, la vérification formelle des programmes consiste à déduire les propriétés d'exécution d'un programme en analysant sa structure (les instructions qu'il contient) sur la base d'une définition formelle du langage de programmation utilisé. En principe, si l'on arrive à prouver mathématiquement que tel programme a bel et bien le comportement voulu en toutes circonstances, alors cette preuve établit de façon irréfutable le bon fonctionnement du programme.

Toute attrayante que soit cette voie, elle a ses limites : 1) pour tout langage de programmation suffisamment puissant, le problème de prouver un comportement donné pour un programme est extrêmement complexe, souvent même indécidable (i.e. non résoluble de façon algorithmique) ; 2) la confiance que l'on peut avoir en une telle preuve de bon fonctionnement s'appuie sur l'hypothèse que tout l'environnement d'exécution du programme fonctionnera de façon non seulement irréprochable (sans dysfonction matérielle), mais

aussi rigoureusement conforme à la définition formelle du langage. Or, cette hypothèse ne peut habituellement pas être démontrée, en tout cas pas mathématiquement. Ainsi, dans la pratique, une preuve mathématique de fonctionnement correct d'un programme est relative au bon fonctionnement d'autres mécanismes que le programme lui-même.

Pour cette raison, on a également recours aux tests de programmes. Il s'agit de soumettre à un programme nouvellement créé un jeu de données (ou de situations) pour lequel on connaît le bon extrant (e.g. comportement) et de vérifier que le programme exhibe cet extrant. Le programme est utilisé comme une boîte noire entre les intrants et les extrants. L'opération est répétée pour un grand nombre de situations réalistes. Si un mauvais résultat est observé, on sait que le programme fonctionne mal au moins dans cette situation spécifique. Si le programme passe tous les tests, la confiance en son bon fonctionnement dans *tous* les cas augmente.

Les tests de programmes ont aussi leurs faiblesses : 1) le nombre de données ou situations différentes qu'un programme peut avoir à traiter est en général prohibitif, et ne sont donc pas toutes testées, ce faisant on n'a jamais l'équivalent d'une *preuve* de bon fonctionnement du programme ; 2) si le programme exhibe un mauvais résultat, aucun indice quant à la cause de la dysfonction n'est fourni.

Dans la pratique, la meilleure approche est d'utiliser une combinaison de vérification formelle et de tests. Du côté de la vérification formelle, il est rare qu'on aille jusqu'à formuler une preuve mathématique complète du bon fonctionnement d'un programme, mais la définition formelle précise du langage est toujours disponible et est utilisée de façon essentielle dans toute activité de programmation. Utiliser la définition formelle du langage de programmation pendant l'élaboration d'un programme et ensuite effectuer des tests extensifs sont donc deux éléments essentiels de la démarche génie logiciel.

La définition formelle d'un langage de programmation prend toujours la forme d'une spécification de la *sémantique* des différentes *constructions syntaxiques* disponibles dans le langage, i.e. de leur « effet » sur le comportement des programmes. Cette spécification sémantique peut être de différents types (*opérationnel*, *dénotationnel* ou *axiomatique*), mais elle est essentielle au travail d'idéation en programmation.

La définition du formalisme de lecture comme aide à l'idéation

Nous postulons que dans de nombreuses activités de création, comme dans l'exemple que nous venons de discuter, une des « théories » les plus utiles pour éclairer la phase idéation est l'hypothèse que l'artéfact sera interprété selon un certain formalisme, ou système sémiotique, prédéfini.

Il faut bien sûr réaliser que, de la même façon qu'en génie logiciel le fonctionnement correct du matériel d'exécution du programme n'est toujours qu'une hypothèse, toute présomption sur le système sémiotique utilisé pour décoder une création ne peut être qu'une hypothèse, puisque rien n'empêche des lectures immensément variées et imprévues d'être appliquées sur n'importe quel artéfact, quelles qu'aient été les intentions de son créateur.

Il n'en demeure pas moins que l'hypothèse d'une lecture basée sur un certain système sémiotique est une technique extrêmement utilisée dans plusieurs domaines de la création.

La création littéraire

En création littéraire, l'hypothèse d'une lecture basée sur une certaine langue permet à un auteur de se relire, d'amender son texte ; bref, d'assumer alternativement les rôles d'auteur et de lecteur dans des cycles très rapprochés, sans avoir à constamment demander l'avis de lecteur externe (ce qui revient à effectuer un « test d'utilisateur »). Nous sommes bien conscients que l'essor des réseaux tend à faciliter le recours à des lecteurs externes, mais nous prétendons qu'il reste une part importante de création autonome, pour laquelle le créateur assume lui-même les quelques premières itérations de « tests utilisateur », en « se mettant dans la peau du lecteur ». Autrement dit, le créateur arrive à intégrer à l'intérieur même de la phase idéation un mécanisme du même type que le test utilisateur, mais sans recourir à de vrais utilisateurs externes.

Le design UX

En contraste avec le scénario de création littéraire que nous venons de postuler, une des premières mises en garde dans la majorité

des livres portant sur l'AI ou l'UX est l'idée qu'il n'est d'aucune utilité pour le concepteur d'une interface de tenter de « se mettre dans la peau » d'un utilisateur pour évaluer son système³.

D'une part, nous pensons qu'en dépit des mises en garde, les concepteurs recourent couramment à cette façon de faire. D'autre part, nous croyons qu'effectivement, cette technique a une efficacité mitigée avec les systèmes sémiotiques *non langagiers*, par exemple, iconiques ou graphiques (ce qui amène les experts à condamner la technique pour la conception d'interface). Elle semble mieux fonctionner pour les créations en langue naturelle et en langages formels (e.g. programmation informatique).

Systèmes sémiotiques langagiers

Nous appelons *systèmes sémiotiques langagiers* ceux dotés de définitions syntaxique et sémantique « fortes », i.e. faisant l'objet d'un consensus social fort. En première approximation, nous entendons par là l'existence d'entités socialement reconnues s'occupant publiquement de la définition du système sémiotique : organisme d'état, métiers bien établis et reconnus (lexicographe, etc.), organisme de normalisation. Dans le cas de la langue française, notamment, des organismes d'état comme l'*Office québécois de la langue française* au Québec jouent ce rôle. Pour les langages de programmation, ce sont des comités de normalisation idoines au sein d'ISO ou d'organismes similaires.

Pourquoi, donc, la technique d'alterner entre modes écriture et lecture en phase d'idéation fonctionnerait-elle moins bien avec des systèmes non langagiers qu'avec des systèmes langagiers ? Nous irons à nouveau d'un postulat : nous pensons que c'est la disponibilité de définitions syntaxique et sémantique *fortes* mais également *externes* tant au créateur et aux éventuels lecteurs/utilisateurs qui donne l'avantage aux systèmes langagiers.

Notre postulat pourrait être formulé comme suit : l'existence d'une définition consensuelle forte et externe du système sémiotique permet à un créateur de plus facilement s'affranchir de ses propres idiosyncrasies perceptuelles et cognitives lorsqu'il passe en mode

3 Pour une remise en question de l'efficacité d'une telle mise en garde voir Molich (2013).

« lecture » de sa création. Selon nous cet affranchissement n'est pas autant possible dans le cas de systèmes sémiotiques dont la syntaxe et la sémantique sont faiblement ou mal définis.

Constructions structurelles signifiantes

Selon ce postulat, la situation qui permettrait le plus le travail autonome du concepteur d'interface serait celle d'une interface entièrement en langue naturelle (les langages formels étant bien sûr exclus). Or, cela n'est souvent ni possible, ni souhaitable pour de nombreuses raisons : petitesse des dispositifs de navigation, inadéquation avec le profil cognitif des utilisateurs, etc.

Les interfaces modernes utilisent des systèmes sémiotiques textuels, visuels, sonores, multimodaux, etc. Ces systèmes intègrent des signifiants de base (mots, icônes, sons élémentaires, etc.), mais aussi des propriétés de présentation et des relations de *positionnement relatif* (tant géométrique que temporel) entre signifiants de base pour véhiculer du sens. Nous entendons par *construction structurelle signifiante* (ou simplement *construction structurelle*) d'un système sémiotique une propriété distinctive de présentation (e.g. la graisse en typographie) ou un positionnement relatif particulier (e.g. la mise en exergue typique des titres dans un texte), applicable à un ou plusieurs autres signifiants et *porteur de sens dans ce système*⁴. Les constructions structurelles correspondent *grosso modo* à des balises dans une représentation des créations (ou productions sémiotiques) exprimée en un langage de balisage de type XUL, HTML ou XML en général.

Notons qu'il est très rare que soient expliqués précisément (dans l'interface elle-même ou dans sa documentation) tant les signifiants de bases non-textuels que les constructions structurelles signifiantes du système sémiotique sous-jacent à une interface. Ces choses sont habituellement gardées implicites (parfois même pour les créateurs) et on suppose que les utilisateurs les apprendront à force d'utiliser le système. Ces systèmes sémiotiques ne peuvent donc pas être considérés comme faisant l'objet d'un consensus fort, ni donc comme langagiers.

4 Nous désignons par le vocable « signifiant » tant les signifiants de bases que les constructions structurelles signifiantes.

Aide à l'idéation pour la conception d'interface

Nous croyons qu'il est possible de définir des systèmes sémiotiques non langagiers de telle sorte que les créateurs puissent travailler en mode « se mettre dans la peau de l'utilisateur » avec un niveau d'efficacité s'approchant de celui observé avec les systèmes langagiers.

Normalisation

Chaque concepteur a son propre « répertoire » personnel de signifiants de base et de constructions structurelles correspondant à ce qu'il considère être des « conventions tacites » émergeant au sein de diverses collectivités d'utilisateurs. Néanmoins, on assiste à une « normalisation spontanée » des signifiants pour certains types d'interface. Une telle normalisation et ses bénéfices sont discutés par Moore (2012) pour les icônes de navigation dans les interfaces mobiles.

Serait-il envisageable d'externaliser et de systématiser ce travail de recension actuellement réalisé individuellement et en parallèle par les concepteurs ? Il s'agirait d'un travail analogue à celui des lexicographes des différentes langues naturelles où on en arriverait à des systèmes de signifiants non langagiers standards munis d'une sémantique forte et externe aux créateurs et aux utilisateurs. Ces systèmes pourraient éventuellement se comporter comme des systèmes langagiers et permettre aux créateurs d'être plus efficaces dans les phases d'idéation autonome du processus de création.

Bénéfices possibles d'une approche sémantique

Même en présence d'éventuels répertoires plus ou moins « officiels » de signifiants, deux problèmes subsisteraient : 1) De tels répertoires ne pourraient être exhaustifs car il y aura toujours des cas où la création de nouvelles façons de faire sera nécessaire ; (2) Comment décrire sémantiquement les signifiants répertoriés de façon à ce que le répertoire procure au signifiant la sémantique « forte » recherchée ?

L'approche que nous nous apprêtons à esquisser – la *sémantique intertextuelle* – propose une façon de décrire la sémantique des signifiants d'un système sémiotique comportant des signifiants de base et des constructions structurelles. En ce sens, elle pourrait constituer une piste de solution au problème (2) ci-dessus. Mais en fait, nous croyons que sa plus grande vertu réside dans ce qu'elle peut apporter comme *point de vue* dans le travail de conception de *nouveaux signifiants* et donc dans ce qu'elle propose comme piste de solution au problème (1).

La sémantique intertextuelle

L'idée de base de la *sémantique intertextuelle* (Marcoux 2006, Marcoux-Rizkallah 2009) est simple : il s'agit essentiellement de décrire *en langue naturelle*⁵ le sens des différents signifiants d'un système sémiotique pour des interfaces.

L'idée d'utiliser une langue naturelle comme représentation n'est pas anodine ; elle vient directement de notre postulat selon lequel les systèmes langagiers (ce que sont les langues naturelles, suivant notre définition) sont mieux adaptés à l'alternance écriture-lecture qui, selon nous, permet au concepteur de travailler plus efficacement dans les phases autonomes d'idéation du processus de création.

On distingue les cas d'un signifiant de base et d'une construction structurelle. Un signifiant de base (icône, son simple, mot ou expression) est représenté par un équivalent en langue (naturelle). Normalement, dans le cas d'un mot ou d'une expression linguistique, la représentation sera le signifiant lui-même.

Une construction structurelle – positionnement relatif ou propriété de présentation – est pour sa part représentée par une « phrase à trous » *en langue naturelle*, phrase qui s'instancie, dans une utilisation spécifique, en remplaçant les marqueurs de place (les trous) par les éléments liés par la structure correspondant à la construction. (Pour une propriété de présentation, il n'y a qu'un seul « trou ».)

Avec une telle phrase à trous, le sens d'une utilisation spécifique d'une certaine construction est donné *en langue naturelle* par le résultat de ces remplacements. Il est donc possible de « calculer »

5 Dans la plupart des utilisations de la SI à ce jour, il s'agit en fait de langue « quasi- » naturelle, mais par simplicité, nous parlerons toujours ici de langue naturelle.

automatiquement ce sens, à partir de la phrase à trous correspondant à la construction et des différents éléments auxquels la construction est appliquée. Un exemple simple (tiré de Rizkallah-Marcoux 2012) est présenté à la Figure 3.

Figure 3 : Exemple d'interface

Concentrons-nous sur le rectangle identifié « b » dans la Figure 3. Nous avons ici sept signifiants de base distincts ; deux iconiques : le carré vierge, le carré coché (non illustré dans l'exemple) et cinq mots ou expressions. Le sens attribué au carré vierge est « Je ne sélectionne pas » et celui attribué au carré coché « Je sélectionne ». Deux positionnements relatifs sont perceptibles, assez clairement pour que nous les supposions perçus par l'utilisateur : une juxtaposition horizontale (entre les cases à cocher et les mots) et un empilement vertical (des différentes lignes). À la première de ces constructions structurales (la juxtaposition horizontale) nous faisons correspondre la phrase à deux trous « identité », i.e. celle qui juxtapose simplement ses deux « arguments », en respectant l'ordre. À la seconde (l'empilement vertical), nous faisons correspondre une phrase à trous « conjonctive », soit celle qui unit ses différents arguments par la conjonction « et ». Le sens du rectangle « b » est donc au final un passage en langue naturelle⁶ du genre : « Je ne sélectionne pas *Book* et Je ne sélectionne pas *Corporate report* et... ».

6 La capitalisation fautive est un exemple des raisons nous amenant à parler en général de langue « quasi- » naturelle.

Le sens calculé d'une instance de construction structurelle pourrait en principe être présenté tant au créateur en phase idéation qu'à un lecteur peu familier avec une construction présente dans une interface donnée. Pour le créateur en mode idéation, ce sens calculé jouerait un peu le rôle d'une vérification de première ligne que son utilisation de la construction en question véhicule bien le sens qu'il voulait faire véhiculer par l'interface. Pour le lecteur peu familier avec la construction structurelle, ce sens peut servir d'outil d'apprentissage lui permettant de devenir peu à peu suffisamment familier avec la construction pour pouvoir la comprendre directement, sans avoir recours au sens « développé » en langue naturelle.

Cependant, beaucoup plus que dans la possibilité de *calculer* automatiquement le sens des différentes instances de construction présentes dans une interface, nous croyons que le principal bénéfice de l'approche SI réside dans le fait qu'elle amène le concepteur à envisager sa création en termes de sens en langue naturelle et non comme phénomène exclusivement perceptuel ou cognitif. Le créateur a conscience qu'il doit exister une certaine *syntaxe* liant les différents éléments de son interface pour que l'utilisateur puisse en tirer du sens. Cette conscience l'amène à utiliser les signifiants dans son interface (qu'ils soient plus ou moins formellement normalisés ou qu'ils soient du propre cru du créateur) de façon à ce qu'ils soient naturellement décodables par l'utilisateur, via des règles reposant ultimement sur la langue naturelle.

Nous croyons que ces bénéfices sont particulièrement utiles à un *nouvel utilisateur* (« *first-time user* ») qui n'est pas déjà familier avec l'interface. Il est indéniable qu'un apprentissage spontané de la part de l'utilisateur a lieu lors des premières utilisations d'une interface. Surtout avec les interfaces ultra-compactes courantes, il y a forcément au départ un travail d'exploration à tâtons où l'utilisateur doit « deviner » le fonctionnement de l'interface.

Nous postulons que si les signifiants ont été utilisés en pensant à comment ils peuvent être interprétés selon la mécanique de la SI (équivalence en langue naturelle pour les signifiants de base et phrase à trous pour les constructions structurelles), alors leur sens peut être plus facilement (et correctement) « deviné » par un premier utilisateur. Plus précisément, nous postulons que, face à un signifiant inconnu, un utilisateur formule des hypothèses de sens possibles, et cherche à en renforcer une. Nous pensons qu'une des premières formes d'hypothèses formulées par un utilisateur dans une telle phase

d'exploration est d'associer un sens selon la mécanique proposée par la SI (équivalence en langue naturelle pour les signifiants de base et phrase à trous pour les constructions structurelles). Les hypothèses de ce genre, nous suggérons, ont l'avantage de pouvoir être validées selon des règles externes à l'utilisateur et au créateur : celles de la langue naturelle.

Un des effets particuliers que nous escomptons de l'approche SI est que les concepteurs prennent pour acquis que les mots présents dans l'interface (et il y en a encore beaucoup) seront *d'abord* interprétés linguistiquement par l'utilisateur, avant toute autre hypothèse d'interprétation. Considérer que les mots n'ont plus leur valeur linguistique conduit, selon nous, à des aberrations du type relaté en début d'article.

Un autre exemple est le contresens flagrant induit par l'utilisation du mot « via » dans l'interface de Facebook (Figure 4), lorsque ce dernier veut signifier qu'un article partagé par un utilisateur provient en fait d'un autre utilisateur (cette faille se retrouve également dans l'interface anglaise). L'article est (incorrectement) attribué à « Utilisateur-A *via* Utilisateur-B » ; en réalité, il nous parvient de l'Utilisateur-B *via* l'Utilisateur-A. Nous croyons que ce genre de laxisme dans la conception d'interfaces peut avoir des effets pernicioeux bien au-delà d'« affaiblir » les consensus émergents autour des systèmes sémiotiques sous-jacents, et aller même jusqu'à affaiblir la langue naturelle. Dans le cas précis du « via » de Facebook, nous ne serions pas surpris que certains utilisateurs soient maintenant convaincus que le sens du mot « via » est en fait l'inverse de son sens réel.

Figure 4 : Utilisation du mot « via » dans Facebook

Une méthode de conception

Voyons maintenant comment les idées de la SI présentées ci-dessus peuvent être intégrées dans une méthode de conception. Une façon naturelle serait d'introduire la langue naturelle dans le

processus même de design. Il ne s'agit évidemment pas de rédiger l'interface en langue naturelle. Ce serait irréaliste pour plusieurs raisons déjà invoquées.

Une possibilité est qu'au début du processus de conception, le concepteur établisse *en langue naturelle* une version « de référence » des dialogues système-utilisateur que l'interface doit pouvoir soutenir. Cette version demeure disponible tout au long de l'élaboration de la création, où elle sert de référence pour « valider » les décisions de conception concrète de l'interface, laquelle doit en quelque sorte traduire dans un nouveau système sémiotique la version originelle en langue naturelle.

Dans certains cas, la traduction peut être presque immédiate. C'est le cas, par exemple, de l'interface de définition des préférences système du logiciel de gestion intégrée de bibliothèques Koha<<http://koha-community.org/>>, dont un extrait est présenté à la Figure 5.

Figure 5 : Définition des préférences système dans Koha

Même si cette interface n'a pas été développée en appliquant la SI (en tout cas, à notre connaissance), elle illustre bien le genre de résultat qui peut être obtenu par son application. La prépondérance de la clarté sémantique sur les autres aspects de l'interface est évidente, et il nous semble que l'UX risque d'être meilleure que si l'accent avait été mis sur le graphisme. Un des effets de l'approche SI est de se permettre de recourir, lorsque cela est possible et acceptable, à une interface d'une grande clarté sémantique, au prix d'un poids textuel plus important.

Un exemple peut-être plus typique du résultat d'une telle démarche (appliquée à l'interface originelle montrée à la Figure 3) est présenté à la Figure 6 (tirée de Rizkallah-Marcoux 2012). Le lecteur est référé à Rizkallah-Marcoux 2012 pour les détails de comment ce résultat est obtenu.

Figure 6 : *Exemple d'interface résultant de la SI
(sans et avec limite enclenchée)*

Conclusion et travail futur

Bien des recherches empiriques seraient évidemment nécessaires pour établir ou réfuter les différents postulats formulés tout au long de cet article. Il s'agirait ensuite d'affiner ou de modifier le modèle du processus de création sous-jacent aux postulats en fonction des résultats.

À notre avis, travailler dans les lignes d'une intégration de la langue au processus de conception améliorerait la communication sémantique entre un système et son utilisateur au moment de l'interaction. En particulier, cela permettrait de reconnaître que les mots présents dans une interface (et il en reste, malgré que plusieurs mouvements en conception d'interface prônent leur élimination plus ou moins systématique) opèrent de façon linguistique et qu'en conséquence, on devrait se soucier de la façon dont ils interagissent linguistiquement avec les autres éléments d'interface. Selon nous, l'intégration de la langue naturelle dans le processus de conception est en soi une manière de rétablir une définition plus large et plus proche de son sens originel de l'architecture de l'information.

YVES MARCOUX

GRDS – EBSI –

Université de Montréal

ÉLIAS RIZKALLAH

Centre d'ATO, Sociologie –

UQAM

BIBLIOGRAPHIE

- Bertin J.**, (1967), *Sémiologie graphique : les diagrammes, les réseaux, les cartes*, Paris, Mouton.
- Garrett J. J.**, (2002), *The Elements of User Experience: User-Centered Design for the Web*, Peachpit Press.
- Garrett J. J.**, (2011), *The Elements of User Experience: User-Centered Design for the Web and Beyond*, Berkeley (CA), New Riders.
- Krug S.**, (2000), *Don't make me think! A common sense approach to Web usability*, Indianapolis (Ind.), New Riders.
- Krug S.**, (2010), *Rocket surgery made easy: the do-it-yourself guide to finding and fixing usability problems*, Berkeley (CA), New Riders.
- Marcoux Y.**, (2006), *A natural-language approach to modeling: Why is some XML so difficult to write?*, Actes de la conférence Extreme Markup Languages 2006, août 2006, vol. 14 : <http://conferences.idealliance.org/extreme/html/2006/Marcoux01/EML2006Marcoux01.html>, date de la dernière visite : 10 octobre 2012.
- Marcoux Y. & Rizkallah É.**, (2009), *Intertextual semantics: A semantics for information design*, Journal of the American Society for Information Science and Technology, vol. 60, n° 9, pp. 1895-1906.
- Molich R.**, (2013), *The top usability testing myths* : <http://www.creativebloq.com/web-design/usability-testing-myths-2132991>, date de la dernière visite : 10 septembre 2013.
- Moore J.**, (2012), *The Semantic, Responsive Navicon*, Smashing Magazine : <http://mobile.smashingmagazine.com/2012/10/08/the-semantic-responsive-design-navicon>, date de la dernière visite : 8 octobre 2012.
- Morville P. & Rosenfeld L.**, (2006), *Information architecture for the World Wide Web* (3rd éd.), Sebastopol (CA), O'Reilly.
- Nielsen J.**, (1993), *Usability engineering*, Boston, Academic Press.
- Nielsen J.**, (2006), *Prioritizing Web usability*, Berkeley (CA), New Riders.
- Norman D. A.**, (1988), *The psychology of everyday things*, New York, Basic Books.
- Norman D. A.**, (2005), Human-centered design considered harmful, *Interactions*, vol. 12, n° 4, pp. 14-19.
- Rizkallah É. & Marcoux Y.**, (2012), *Interface Deconstruction-Reconstruction Using Intertextual Semantics: Rationale and Example*, Actes de la conférence IADIS International Conference Interfaces and Human Computer Interaction (IHCI) 2012, pp. 217-222.
- Rosenfeld L. & Morville P.**, (1998), *Information architecture for the World Wide Web* (1st éd.), Cambridge, Sebastopol, CA: O'Reilly.
- Tufte E. R.**, (1995), *Envisioning information*, Cheshire (Conn.), Graphics Press.