

Méthodologie d'identification des invariants du traitement de l'information numérique

Étienne Vandeput

evandeput@ulg.ac.be, etienne.vandeput@unige.ch

CRIFA Université de Liège, IUFÉ Université de Genève

Résumé. Nos réflexions sur la didactique des progiciels, nous ont amenés à introduire brièvement et à deux occasions, la notion d'invariant du traitement de l'information numérique (TIN). Ce fut une première fois lors du colloque Didapro 2 (Vandeput & Colinet, 2006) et une seconde fois, à Didapro 3 (Poisseroux *et al.*, 2009). Depuis, nous avons maintes fois exploité cette notion dans des contextes de formation et avec des publics très différents : étudiants universitaires, étudiants de 3^e cycle, enseignants des Hautes Écoles, enseignants du secondaire. Si nous travaillons à l'identification des invariants du TIN avec les futurs enseignants, c'est essentiellement pour les aider à autonomiser leurs apprentissages dans le domaine des TIC et aussi pour les habituer à pointer de manière précise des « contenus à faire apprendre ». La compréhension de cette notion d'invariant ne va pas de soi. Ce document a pour objectif de la clarifier.

Mots-clés: didactique des TIC, invariants

Introduction

L'origine de cette démarche d'identification des invariants du traitement de l'information numérique remonte à une bonne quinzaine d'années. À cette époque, les aléas de la formation sur le terrain dans laquelle nous étions fortement engagés nous plaçaient souvent devant des équipements très variés, tant au niveau des ordinateurs, que des versions des logiciels qui les équipaient. Nous observions aussi que des participants s'inscrivaient parfois à plusieurs reprises à des formations dédiées à l'usage d'un même progiciel, sous prétexte que les versions de celui-ci changeaient.

Pour faire face à ces problèmes et prendre en compte ce constat, nous nous sommes efforcés de nous détacher des interfaces pour nous intéresser de plus près aux systèmes informatiques et à leurs fonctionnalités, observant que, pour des produits (logiciels) dédiés à une même tâche générique de traitement de l'information, un grand nombre de ces fonctionnalités étaient communes. Nous avons également fait le constat qu'à terme, un vocabulaire commun, spécifique à la fois à la tâche et au logiciel, semblait se dégager.

De là, notre intérêt pour l'identification de concepts et principes invariants à l'aide d'une méthodologie dont nous avons depuis testé l'efficacité, sur des progiciels

implémentant une grande variété de ces tâches génériques : le traitement de texte, la retouche d'image, la gestion d'une feuille de calcul, la recherche d'information, la communication par courriel, l'édition collaborative... et même, l'édition d'un portfolio.

Cet article se limite à l'explicitation de cette démarche et à son illustration. Il est clair que les invariants ne sont que des ingrédients pour la construction d'une séquence de cours ou de formation dont les objectifs devraient être de les faire maîtriser. Au-delà de cette démarche peut donc se dessiner une hiérarchie d'objectifs qui œuvrent à l'exercice de larges compétences, et que nous ne décrivons pas ici. Un exemple d'une telle hiérarchie est évoqué dans (Vandeput & Henry, 2011b).

Nous donnerons quelques exemples d'invariants au sein d'outils TIC qui implémentent certaines des tâches évoquées : édition et mise en page de texte, conception d'une feuille de calcul, retouche d'image... Bien sûr, il existe aussi des invariants plus transversaux. Ils concernent le traitement de l'information numérique en général et sont liés à la manière dont fonctionnent, globalement, les systèmes informatiques.

Un peu de (notre) vocabulaire

Trois mots ont été cités dans l'introduction et sont utilisés dans une acception précise que nous explicitons. Ces trois mots sont respectivement « tâche », « système » et « fonctionnalité ».

Tâche

Au sens où nous l'entendons, une **tâche** correspond à l'objectif global auquel un progiciel (logiciel ou programme pour produire) peut être associé et désigne, en quelque sorte, une activité très générale bien que ciblée : produire et mettre en page un texte, naviguer sur le Web à la recherche d'informations, communiquer par courriel, produire un document texte unique à plusieurs, collaborer à une édition collective... Cette définition est assez proche de la notion de tâche utilisée par Nogier (Nogier, 2005). Elle est également courante dans les ouvrages traitant de la conception des interfaces homme-machine (IHM).

Du point de vue de l'utilisateur, le progiciel regroupe les actions possibles lui permettant de poursuivre divers objectifs précis de production liés à cette tâche. Dans une perspective d'élargissement de la réflexion, notons que lorsque le programme s'exécute à distance, on parle plutôt d'application, sans que le point de vue évoqué ne change fondamentalement.

Les progiciels et les applications portent souvent un nom en rapport avec la tâche qu'ils implémentent : traitement de texte, navigateur Web, courrielleur (logiciel ou application en ligne de courrier électronique), traitement de texte en ligne, moteur de wiki, moteur de blog...

La grande variété des programmes développés sur la toile rend plus difficile la catégorisation des applications. Sur le Web, les applications sont plus rapides à se développer que leurs concepteurs et leurs utilisateurs à leur trouver des noms.

Système

Ce que nous appelons un **système** est la représentation mentale qu'un utilisateur peut se donner de la combinaison « un ordinateur + un programme » et en particulier, d'un programme dédié à une tâche. Donc, lors de la réalisation d'une tâche donnée, l'utilisateur communique avec un système pour qu'elle soit exécutée. Comme la communication est forcément limitée, il a tout intérêt à comprendre comment ce système fonctionne.

L'intérêt de cette représentation, c'est de permettre à l'utilisateur d'associer un système différent, autrement dit qui « fonctionne » différemment, à chaque programme différent et/ou à chaque ordinateur différent qui l'exécute. Dans cette représentation, le possible passage rapide d'une application à une autre fait que l'utilisateur travaille avec un système susceptible de se transformer en quelques fractions de seconde et très régulièrement.

Fonctionnalité

La notion de système nous conduit tout naturellement à celle de **fonctionnalité**. Nous définissons les fonctionnalités d'un système comme les actions qu'il permet de réaliser dans le contexte de l'exécution de la tâche. Il y a des fonctionnalités spécifiques à la tâche (ex : aligner un paragraphe à droite en traitement de texte). Il y en a qui sont plus transversales (ex : enregistrer le produit de son travail sous forme d'un fichier de données). Certaines fonctionnalités ont un caractère automatique (copie de secours, historiques...). Elles ne nécessitent pas l'intervention de l'utilisateur, sauf pour leur programmation (choix de certains paramètres par défaut).

Méthodologie

Un système est, en quelque sorte, un outil pour l'utilisateur. Sa connaissance et sa maîtrise tiennent essentiellement dans les réponses à deux questions: « à quoi sert-il ? » et « comment fonctionne-t-il ? ». Autrement dit, « quelles en sont les fonctionnalités ? ». Ce sont ces questions simples qui sont à la base de la démarche.

Les réponses à ces questions vont, dans un premier temps, permettre d'énoncer des fonctionnalités que nous appelons **principes** de fonctionnement d'un système censé implémenter une tâche.

Nous verrons qu'il est difficile, voire impossible, d'énoncer ces principes sans évoquer des **concepts** qui sont également spécifiques à la tâche.

La pérennité de nos découvertes sera d'autant plus importante si ces principes et ces concepts sont communs aux progiciels ou aux applications qui implémentent une même tâche et peuvent donc être qualifiés d'**invariants**.

Motivation

Pourquoi faire la chasse aux invariants du traitement de l'information numérique ? S'exercer à **énoncer des principes**, et aussi essayer d'**identifier et de définir des concepts** peut paraître artificiel, car la démarche peut donner cette impression d'être déconnectée du contexte d'apprentissage.

Il est vrai que le résultat de cette quête est inexploitable en tant que tel dans un contexte d'apprentissage. Mais là n'est pas notre but premier. L'exercice nous oblige à être au clair par rapport au véritable contenu de notre discours. Formuler correctement une définition, c'est faire la preuve que l'on maîtrise ce dont on parle et que l'on sait pertinemment quelles représentations on veut créer dans l'esprit des apprenants.

Autre bonne raison de tenter cette conceptualisation : les curricula concernant les TIC évoluent peu et se cantonnent très souvent dans des procédures à « faire apprendre ». (Baron & Bruillard, 2001) constataient déjà que « *Les compétences énoncées apparaissent surtout sous forme de savoir-faire opératoires...* ».

On peut donc résumer le travail à réaliser en disant qu'il s'agit de trouver des réponses pertinentes à la question « quoi faire apprendre? » d'un outil technologique dédié à une tâche donnée du traitement de l'information, cette question précédant logiquement une autre question à laquelle nous apportons par ailleurs (Vandeput & Henry, 2011a) des réponses contextualisées : « comment faire apprendre (ce *quoi*) ? ».

Méthode globale

Ce travail de réflexion ne peut être réalisé en une seule étape. Il demande d'être complété, relu, revu, corrigé maintes fois. Mais pour plagier La Fontaine dans « *Le laboureur et ses enfants* », la richesse est aussi et surtout dans l'obstination que l'on peut y mettre et qui conduit à une vision des choses à transmettre aux apprenants, qui soit toujours plus solide et rassurante.

Le point de départ est l'identification des principales fonctionnalités et donc, des principes de fonctionnement de l'outil. Pour l'essentiel, cette question n'est d'ailleurs pas très éloignée de la question des plus-values. Un tableur est utile parce qu'il recalcule l'ensemble des formules dès qu'une cellule du classeur a changé de contenu. Un traitement de texte redessine la mise en forme d'un texte dès que les paramètres d'un style utilisé dans le marquage du texte sont modifiés, etc. Nous décrivons, plus loin, différents moyens d'étoffer cette identification.

En énonçant chacune de ces fonctionnalités, on ne peut éviter d'évoquer des concepts. Les deux exemples qui précèdent mentionnent deux concepts importants que sont la formule et le style.

Dans un second temps, on cherche à établir des définitions didactiques de ces concepts. Autrement dit, à travers ces définitions, on tente de se fabriquer une représentation correcte de la manière dont le système fonctionne, une définition n'étant vraiment utile que si elle aide à le maîtriser davantage, et donc à le comprendre mieux. Quelles définitions utiles donner aux concepts de formule et de style ? Des réponses que nous apportons à cette question vont dépendre les séquences d'apprentissage à mettre en place et, dans la foulée, les représentations que les apprenants vont se construire.

On comprend que la recherche d'invariants n'est pas facile à mener si on se focalise sur un (le) seul outil (celui qu'on a l'habitude d'utiliser). Sans trop de surprise, la méthode principale pour parvenir à identifier des invariants consiste à examiner **plusieurs outils dédiés à la même tâche** (autrement dit, des produits de marques et/ou de versions différentes).

Pour cette « mise en commun », il convient de faire des choix pertinents. Si les outils de bureautique sont très « typés », les utilisateurs éprouvent plus de difficultés à distinguer certaines applications en ligne, qui peuvent donner cette impression d'être dédiées aux mêmes tâches alors qu'il n'en est rien. Par exemple, un moteur de wiki, un moteur de blog, une application de traitement de texte en ligne sont dédiés à des formes différentes d'édition collaborative. Ce sont des outils que les utilisateurs amalgament parfois.

Pistes

Avant de donner quelques exemples, citons quelques pistes à exploiter.

Assez paradoxalement, si l'objectif du travail sur les invariants est d'apprendre à se détacher des interfaces, celles-ci sont cependant utiles au travail d'identification. Elles sont composées d'objets d'interaction qui correspondent généralement à la mise en œuvre des fonctionnalités par le système. Même si leur design est légèrement différent, ces objets d'interaction se retrouvent généralement dans toutes les interfaces des progiciels dédiés à une même tâche. Ils constituent une première manière de reconnaître des fonctionnalités importantes.

Les menus de ces interfaces sont composés de rubriques dont les libellés sont généralement des noms ou des verbes. Même si c'est loin d'être une règle générale, les noms correspondent parfois à des concepts et les verbes cachent des fonctionnalités.

Les rubriques d'aide donnent généralement accès à des pages et des pages de documentation. Les titres des rubriques sont souvent assez proches de certaines fonctionnalités. Les développements de ces rubriques amènent souvent le lecteur à

(re)découvrir les termes précis d'un vocabulaire qui peut conduire à identifier des concepts importants.

D'autres pistes sont possibles et tout aussi intéressantes quand il s'agit de prioriser les apprentissages. Les pratiques déviantes observées chez les utilisateurs, la longueur et/ou la lourdeur de leurs démarches sont souvent des indices d'une mauvaise connaissance de certains concepts ou de certains principes. Une raison suffisante pour les identifier et tenter de les (re)définir clairement...

Illustration des techniques d'identification

Illustrons d'abord comment la chasse peut être menée. Nous verrons ensuite comment on peut passer à l'énoncé des principes et aux définitions des concepts.

La comparaison des interfaces

Les objets d'interaction de l'interface sont directement liés à des fonctionnalités. Leurs aspects graphiques, souvent proches, nous aident à les pointer du doigt.

Observez d'abord les deux copies d'écran qui suivent.



Figure 1: le navigateur Firefox



Figure 2: le navigateur Safari

Elles sont issues de deux navigateurs Web. De très nombreuses informations peuvent être extraites de ces deux images. En voici l'une ou l'autre.

On repère plusieurs objets d'interaction intéressants. La barre d'adresse fait de suite penser à la fonctionnalité d'encodage d'une URL, un concept fondamental. Les boutons fléchés « page précédente », « page suivante » traduisent une autre fonctionnalité invariante des navigateurs. Elle concerne une des manières d'exploiter un historique de navigation. De même le bouton ou l'icône en forme de flèche circulaire trahit le principe tout aussi invariant d'actualisation de la page courante.

Les rubriques des menus

Les rubriques des menus sont hélas peu intuitives, surtout aux yeux de l'utilisateur novice. Certaines d'entre elles peuvent cependant inciter le didacticien à rechercher, par ailleurs (dans l'aide, par exemple), si elles ne font pas référence à

des concepts ou à des principes. Notez qu'au bout de toute branche de l'arborescence d'un menu se cache une fonctionnalité.

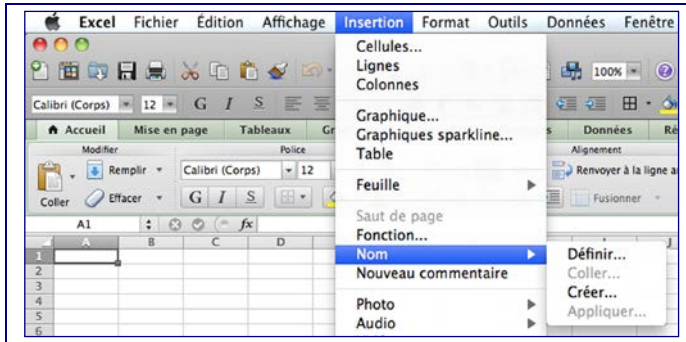


Figure 3 : un menu de l'interface du tableur Excel

Les mots « Cellules », « Lignes », « Colonnes » et d'autres mentionnent des concepts importants du tableur. Le sous-menu de la rubrique « Nom » signale des fonctionnalités intéressantes concernant la possibilité de nommer des entités.

L'aide

La difficulté liée à l'exploration de l'aide est semblable à celle des menus. Il faut pouvoir distinguer l'essentiel de l'accessoire. L'importance relative des rubriques concernant certains sujets peut être un indice pour identifier des concepts importants. On peut alors se référer aux descriptions des fonctionnalités qui les concernent.

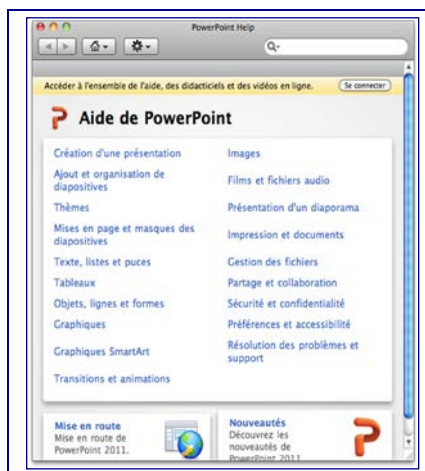


Figure 4: quelques concepts importants

On ne peut nier que « présentation », « diapositive », « objets », « masque » soient des concepts importants dont il faudra se fabriquer des définitions solides. Par ailleurs, des mots comme « transition », « animation » font penser à des fonctionnalités présentes dans tous les progiciels de cette catégorie.

Usages déviants

On pourrait en citer des dizaines : l'utilisation des marques de fin de paragraphe à des fins de réglage des espacements, l'insertion d'images de très grande taille (et donc de poids très élevé) dans des documents que l'on attachera à des courriels, des mises en forme sauvages ne tenant pas compte des possibilités de modélisation des documents, des encodages sans fin dans une feuille de calcul qui font fi des fonctionnalités de génération automatique...

Illustrons juste par un exemple.

Combien d'utilisateurs n'éprouvent aucune difficulté à réaliser la mise en page qui suit et qui n'est pourtant pas extrêmement sophistiquée. Elle peut correspondre à celle d'un lexique, d'un glossaire, d'une liste de définitions.

<p> Lorem ipsum : Lorem ipsum dolor sit amet, consectetur adipiscing elit. In fermentum pulvinar varius. Cras eu risus et augue interdum pharetra id at velit. Vivamus ac tortor ante. Quisque in nisi quis velit imperdiet eleifend a in nisi. Integer leo neque, ornare quis aliquet at, dignissim vel odio. Duis lacinia fringilla erat non interdum. Quisque turpis lectus, vestibulum nec bibendum nec, malesuada nec enim. Nullam venenatis tincidunt erat nec iaculis. </p>
<p> Duis varius : Duis varius dignissim velit, eget porta urna hendrerit at. Fusce rutrum, eros quis mattis cursus, ipsum nunc condimentum eros, vel lacinia orci lacus nec nulla. Sed dapibus leo vel metus pharetra at sodales nulla sagittis. Etiam felis nunc, eleifend in pharetra nec, varius sed odio. Phasellus vulputate vestibulum nisl, in imperdiet odio euismod sit amet. Vivamus viverra semper bibendum. Mauris fermentum ultrices magna quis consectetur. </p>

Figure 5: texte à mettre en page

Voici la solution que beaucoup d'utilisateurs choisissent. Pour la première ligne, pas de problème : encodage de l'expression à définir, usage d'une tabulation ! C'est ensuite que les choses se corsent. Pour les lignes suivantes, ils n'ont guère d'autres recours que d'utiliser à nouveau la tabulation. En procédant de la sorte, des marques de tabulation sont insérées en plein milieu du paragraphe. Malheur à l'éditeur du texte s'il vient à y rajouter ou à y supprimer des parties.

Si l'usage d'une marque de tabulation s'impose à la première ligne, ce sont les concepts de **retrait de paragraphe** et de **retrait de première ligne** qui sont à mettre en évidence ici.

Exemples de principes invariants

Pour vous convaincre de la pertinence de la démarche, nous les avons choisis dans des progiciels implémentant plusieurs types de tâches différentes. Nous venons d'illustrer brièvement comment on pouvait les détecter. Pour les tâches citées et les programmes qui les implémentent, les principes énoncés sont invariants. En gras, les concepts qu'ils évoquent le sont tout autant.

Tableau 1. Progiciels et principes organisateurs

Tâche	Progiciel/application	Principe (« le système permet de... »)
Mettre en page un texte	Traitement de texte	appliquer un style prédéfini au paragraphe dans lequel se trouve le point d'insertion
Naviguer sur le Web à la recherche d'information	Navigateur Web	enregistrer les URL des pages web comme signets pour la navigation future
Traiter une image matricielle	Retouche d'image	recadrer (rogner) une image matricielle
Produire un texte	Traitement de texte	insérer une marque de fin de paragraphe à la position du point d'insertion
Exploiter une feuille de calcul	Tableur	évaluer une formule contenant des variables , des constantes , des opérateurs et des fonctions
Concevoir un produit multimédia	Présentation assistée	suite au déroulement d'un événement , faire apparaître, disparaître, évoluer un objet sur une diapositive en produisant un effet particulier
Produire un document texte à plusieurs	Traitement de texte en ligne	partager un document entre plusieurs utilisateurs possédant un compte sur le site de l'application et définir leur rôle en leur accordant des permissions
Communication synchrone	Messagerie instantanée	bloquer, supprimer, renommer un contact
Exploiter une feuille de calcul	Tableur	réévaluer (recalculer) automatiquement une formule contenue dans une cellule

Tâche	Progiciel/application	Principe (« le système permet de... »)
Naviguer sur le Web à la recherche d'information	Navigateur Web	conserver, dans un historique les URL des pages Web consultées

Comme vous pouvez le constater, il n'y a guère de **principe** qui ne charrie son lot de **concepts**. Nous donnons, au point suivant, quelques définitions de ceux-ci.

Observez aussi que l'énoncé des principes nous oblige à une certaine rigueur et à une correction du langage si nous voulons éviter les mauvaises représentations.

Définitions de concepts

Nous reprenons les définitions de quelques-uns des concepts cités au point précédent. Comme pour l'énoncé des principes, l'intérêt de l'exercice consiste à clarifier sa propre compréhension du concept avant d'imaginer l'enseigner à travers une séquence d'apprentissage. La même rigueur est donc de mise.

Tableau 2. Tâches et concepts

Tâche	Concept
Mettre en page un texte	<i>Style prédéfini</i> : ensemble de paramètres de mise en forme et en page connus du système. Il existe notamment des styles de paragraphe et des styles de caractère . <i>Paragraphe</i> : bloc de caractères délimité par des marques de fin de paragraphe .
Mettre en page un texte	<i>Point d'insertion</i> : symbole graphique (généralement en forme de « i ») qui indique à quel endroit du document le prochain caractère frappé au clavier sera inséré.
Naviguer sur le Web à la recherche d'information	<i>URL</i> : chaîne de caractères désignant toute ressource pouvant être requise par le navigateur. L'écriture de cette chaîne doit respecter une syntaxe rigoureuse. L'URL se décompose en plusieurs parties dont le protocole de communication , le nom du serveur et de son domaine , la localisation et le nom de la ressource sur ce serveur. <i>Page Web</i> : page affichée par un navigateur après réception d'un ou de plusieurs fichiers lui permettant de réaliser la mise en page. Le fichier principal est généralement un document HTML . Les autres fichiers sont des images , des scripts d'animation de la page, une feuille de style ...

Tâche	Concept
	<i>Signet</i> : URL mémorisée par le navigateur sur demande de l'utilisateur. Le signet est parfois appelé favori . Les signets s'organisent en dossiers. Certains peuvent être placés directement sur une barre des signets .
Traiter une image matricielle	<i>Image matricielle</i> : ensemble de pixels colorés disposés en bloc rectangulaire.
Mettre en page un texte	<i>Marque de fin de paragraphe</i> : caractère non imprimable créant la limite entre un paragraphe et le suivant.
Exploiter une feuille de calcul	<i>Formule</i> : expression syntaxique construite à partir de constantes (valeur d'un type donné), variables (références de cellule ou de bloc de cellules contenant des valeurs de types donnés), d' opérateurs et de fonctions .

La liste peut être étoffée en observant que certains des concepts qui précèdent sont, eux aussi, définis à partir d'autres concepts.

À ce stade de notre travail, ces définitions n'ont pas la prétention d'être consensuelles. Elles précèdent notre démarche d'enseignant dans notre travail de construction de séquences d'apprentissage. Nos propres définitions des concepts (en tant que formateur ou enseignant) doit laisser deviner comment nous allons nous organiser pour les faire apprendre. Bien que nous ne puissions les utiliser comme telles dans notre enseignement, ces définitions sont liées à des objectifs très précis. On pourra vérifier dans (Henry, 2011) que les activités autant que les exercices proposés aident l'apprenant à poursuivre de tels objectifs en lien direct avec les invariants.

Contrexemples

De mauvais exemples sont utiles, à la fois pour illustrer la complexité de la démarche, et pour comprendre en quoi ils peuvent véhiculer de mauvaises représentations. C'est souvent la formulation qui fait défaut, mais aussi l'usage de termes qui ne sont pas les termes propres. C'est ce que nous allons tenter de montrer. Chaque fois que c'est le cas, une procédure de déconstruction-reconstruction est nécessaire (Giordan, 2002).

Les exemples qui suivent ont été produits par des étudiants dans le cadre d'un cours de didactique. Ils montrent que nos représentations ou celles que nous véhiculons sont parfois loin d'être parfaites.

Voici d'abord de mauvaises formulations de principes. Les mots ou expressions posant problème sont mis en évidence.

Tableau 3. Mauvaises formulations et propositions de correction

Tâche	Progiciel/application	Principe (« le système permet de... »)
Mettre en page un texte	Traitement de texte	<p>Le système permet de définir une nouvelle section...</p> <p><i>Nous conseillons : « Le système permet d'insérer un saut de section... ». Le saut de section est un caractère non imprimable. Il s'insère comme tout autre caractère.</i></p> <p>Le système permet d'insérer un en-tête et un pied de page.</p> <p><i>Nous conseillons : « Le système permet de paramétrer les zones d'en-tête et de pied de page. ». En effet, ces zones existent sans qu'il y ait besoin de les insérer. On a juste la possibilité de modifier leurs paramètres si on le souhaite.</i></p>
Produire un texte à plusieurs	Traitement de texte en ligne	<p>...permettre au propriétaire d'inviter des collaborateurs à travailler dans le document.</p> <p><i>La formulation « Le système est capable de permettre... » est particulière. N'oubliez pas que c'est le système qui « fait ». Dites donc, par exemple : « Le système permet d'inviter des collaborateurs choisis par le propriétaire du document ou par tout collaborateur en ayant le droit. ».</i></p>
Exploiter une feuille de calcul	Tableur	<p>...effectuer des opérations au travers d'opérateurs et de fonctions.</p> <p><i>La formulation est floue. Que signifie « au travers » ? Il est préférable de dire « ...évaluer une formule dont l'expression inclut des opérateurs et des fonctions. ».</i></p>
Concevoir un produit multimédia	Présentation assistée	<p>...faire passer d'une diapositive à une autre de manière automatique ou manuelle.</p> <p><i>À nouveau, que fait le système ? Il affiche une diapositive. Disons donc : « afficher une nouvelle diapositive lorsqu'un événement se produit. ». Cela vaut d'ailleurs la peine de s'attarder sur la notion d'événement qui correspond, soit à un clic de l'utilisateur, soit à une période de temps qui s'est écoulée.</i></p>

Tâche	Progiciel/application	Principe (« le système permet de... »)
Traiter une image matricielle	Retouche d'image	<p>...modifier les pixels d'origine avec des outils particuliers : brosses, pinceaux, pot de couleur, gomme, etc.</p> <p><i>Le système ne modifie pas les pixels, mais leur couleur. Dès lors : « ...modifier la couleur de certains pixels grâce à divers outils de dessin (brosse, pinceau,...) ».</i></p>

Il ne s'agit pas d'ergoter, ni de jouer sur les mots. L'usage des termes propres est un paramètre important qui conditionne sérieusement la compréhension. Par exemple, l'expression « insérer un saut de section » laisse plus volontiers à penser que le saut de section est un caractère non imprimable et qu'il peut donc être effacé comme un autre caractère. L'usage du mot « définir », au contraire, peut laisser penser qu'il est possible de modifier cette définition.

Certains enseignants confrontés à la recherche d'invariants confondent « principe » et « concept ». La principale ambiguïté consiste à appeler « concepts » ce qui cache des fonctionnalités. En effet, le critère consistant à associer un concept à un nom lorsqu'il est utilisé dans le vocabulaire du progiciel ou de l'application n'est pas sans exceptions. Quand nous parlons de concept, nous parlons d'une notion qu'il est important de développer dans un contexte d'apprentissage. Ainsi, dans le cadre du traitement d'une séquence audio, « lecture » n'est pas vraiment un concept, car cette notion demande peu d'explications. En revanche, « *Le système permet de lire une séquence audio.* » est évidemment un principe, une fonctionnalité.

Pour le reste, la précision et la correction du langage restent de mise.

Voici, pour compléter, des définitions de concepts qui posent problème, selon nous. Les mots ou expressions posant problème sont également mis en évidence.

Tableau 4. Mauvaises définitions et propositions de correction

Tâche	Concept
Traitement du son	<p>Lecture, pause, stop, enregistrer (REC), avancer, reculer...</p> <p><i>Il s'agit essentiellement de fonctionnalités, même si ces dernières sont parfois formulées avec des noms. « Le système permet de mettre en pause un fichier audio en cours de lecture. ».</i></p>

Tâche	Concept
Produire un texte	<p>Un texte est une suite de caractères imprimables, d'espaces et de retours à la ligne.</p> <p><i>Cette définition laisse croire que l'espace n'est pas un caractère. Or elle (espace : n.f. lorsque il désigne le caractère) est considérée comme telle par le traitement de texte. De même, l'expression « retour à la ligne » fait davantage penser à un effet de mise en page. On parle plutôt de marque de fin de paragraphe et de saut de ligne manuel qui sont aussi considérés par le système comme des caractères non imprimables.</i></p>
Naviguer sur le Web à la recherche d'information	<p>Rafraîchissement dans le sens d'une actualisation...</p> <p><i>Il y a ici confusion entre un concept et un principe. « Le système permet de renvoyer, au serveur, la requête ayant permis l'affichage de la page Web courante. ».</i></p>
Traiter une image matricielle	<p>Couleur : fréquence de la lumière</p> <p><i>Si on ne peut qualifier cette réponse de complètement fausse, elle ne nous aide guère dans un contexte d'enseignement des TIC.</i></p> <p><i>Couleur : attribut du pixel ; la couleur se définit à travers différents modèles. Le plus connu, qui est aussi celui qui convient le mieux aux dispositifs d'affichage, est le modèle RVB.</i></p>

Commentaires et conclusion

La méthode d'identification des invariants demande de la patience, car la démarche est itérative et qu'elle inclut un travail de clarification des représentations de celui qui l'applique. En même temps, elle nécessite une prise de recul qui s'avère difficile si on ne peut oublier, pour un moment, le travail avec les apprenants. Ceux-ci ne sont pas absents de la réflexion. C'est à eux qu'on pense (même si c'est de manière « générique ») lorsqu'on tente de définir un concept ou de formuler un principe, car on sait que l'on travaille à « ce qu'il faudra faire apprendre ». Dans d'autres disciplines plus « stabilisées », ce travail n'est généralement pas à faire, étant le fruit d'un consensus établi de longue date.

Notre ambition, en écrivant ce papier, est d'apporter une solution partielle à l'absence de consensus sur les contenus à faire apprendre à propos de l'usage des outils TIC courants. Faut-il d'ailleurs parler d'absence de consensus ou d'absence de réflexion ? Si on admet que les TIC constituent une discipline (avec ses savoirs, ses savoir-faire), il nous semble que le travail de transposition didactique qui doit permettre l'identification des « savoirs à enseigner » (Chevallard, 1991; 1994) est curieusement absente des débats concernant l'apprentissage de l'usage des TIC. La réflexion sur les invariants essaie modestement de pallier cette absence.

Au-delà de cette étape, la définition des « savoirs enseignés » passe par la définition d'objectifs liés à la maîtrise de ces invariants dans une mesure à déterminer, en fonction du public concerné, et à travers des activités et une évaluation tout aussi adaptées. Le projet *visaTICE* dont il est question par ailleurs (Vandeput & Henry, 2011a) est un exemple de réalisation de cette deuxième étape.

Bibliographie

- Baron, G.-L., & Bruillard, E. (2001). Une didactique de l'informatique ?. *Revue Française de Pédagogie*, n°135, avril-mai-juin 2001, 163-172.
- Chevallard, Y. (1991). *La transposition didactique*. La pensée sauvage. Grenoble.
- Chevallard, Y. (1994). Les processus de transposition didactique et leur théorisation, Contribution à l'ouvrage dirigé par G. Arsac, Y. Chevallard, J.-L. Martinand, Andrée Tiberghien (éds), *La transposition didactique à l'épreuve*. La Pensée sauvage, Grenoble, p. 135-180.
- Giordan, A. (1998). *!*. Belin. Nouvelle édition 2002.
- Henry, J. (2011). *visaTICE : des objectifs, une didactique, des matières*. Communication proposée au Colloque International DIDAPRO 4, Université de Patras, Patras, 24-26 octobre 2011.
- Nogier, J.-F. (2005). *Ergonomie du logiciel et design Web*. Dunod. Paris.
- Poisseroux, J., & al. (2009). TactIC pour une intégration réussie des technologies en Haute École. In Baron, G.L., Bruillard, E., Pochon L.O., *Informatique et progiciels en éducation et en formation. Continuités et perspectives*. INRP, Coll. « Technologies nouvelles et éducation ».
- Vandeput, E., & Colinet, M. (2006). Utiliser le tableur en toute autonomie. Deuxièmes journées francophones de didactique des progiciels. Neuchâtel. 28-30 septembre 2005. In Pochon, L.O., Bruillard, E. & Maréchal, A. (eds). *Apprendre avec les progiciels. Entre apprentissages scolaires et pratiques professionnelles*. Neuchâtel, IRDP. Lyon, INRP.
- Vandeput, E., & Henry, J. (2011a). *visaTICE : se mesurer aux TIC et se former sous le regard d'un coach*. Communication proposée au Colloque International DIDAPRO 4. Université de Patras, Patras, 24-26 octobre 2011.
- Vandeput, E., & Henry, J. (2011b). *visaTICE : Construction d'un dispositif de formation et de certification de la maîtrise des TIC par les élèves des 3^e degrés de transition*. Projet de recherche soutenu par la Communauté française de Belgique. Rapport interne, non publié. Université de Liège. CRIFA.