



HAL
open science

Application de recommandations ergonomiques : spécificités des EIAO dédiés à l'évaluation

Stéphanie Jean

► **To cite this version:**

Stéphanie Jean. Application de recommandations ergonomiques: spécificités des EIAO dédiés à l'évaluation. 2000, pp.39-42. hal-01575854

HAL Id: hal-01575854

<https://hal.science/hal-01575854>

Submitted on 21 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

APPLICATION DE RECOMMANDATIONS ERGONOMIQUES :

SPÉCIFICITÉS DES EIAO DÉDIÉS À L'ÉVALUATION

STÉPHANIE JEAN

Laboratoire d'Informatique de l'Université du Maine
72085 LE MANS CEDEX 9 (FRANCE)
Stephanie.Jean@univ-lemans.fr

Résumé — Les recherches en EIAO (environnements interactifs d'apprentissage avec ordinateur) font de plus en plus souvent appel à des problématiques d'IHM. Les résultats des recherches en IHM ne sont cependant pas forcément directement applicables au domaine des EIAO. Cet article a pour objet de spécifier les différences qui séparent les EIAO et plus particulièrement les logiciels d'évaluation des logiciels "classiques". Pour ce nous prenons l'exemple du logiciel PÉPITEST auquel nous tentons d'appliquer des recommandations ergonomiques issues des recherches en IHM.

Introduction

Les chercheurs en EIAO (environnements interactifs d'apprentissage avec ordinateur), avec l'avènement des interfaces graphiques ne peuvent plus ignorer les problématiques d'IHM. Ils sont de plus en plus nombreux à chercher à utiliser les résultats de recherches en IHM pour valider les logiciels interactifs qu'ils conçoivent. Mais ces résultats sont rarement applicables tels quels en raison des différences qui séparent les EIAO des logiciels interactifs "classiques", différences d'autant plus importantes s'il s'agit de logiciels dédiés à l'évaluation. D'une part en effet l'élève n'est pas un utilisateur comme les autres, d'autre part l'évaluation n'est pas une tâche comme les autres.

Dans cet article nous tentons de spécifier ces différences en prenant l'exemple de l'application des critères ergonomiques proposés par Bastien et Scapin [3] au logiciel d'évaluation des connaissances des élèves PÉPITEST. PÉPITEST est l'interface de test d'un logiciel permettant d'évaluer les connaissances des élèves de 3^{ème} / 2^{de} de l'enseignement français en algèbre élémentaire. Elle propose un ensemble de 22 exercices aux élèves et récolte leurs réponses qui seront ensuite analysées et présentées à l'enseignant dans deux autres modules [6] [7]¹. Les exercices proposés sont très différents les uns des autres, ils donnent lieu à des réponses totalement ouvertes pour permettre à l'élève d'exprimer ses conceptions sans contrainte.

L'évaluation de l'interface de PÉPITEST

La démarche de conception adoptée – équipe de conception pluridisciplinaire et conception participative – impose le recensement précoce des critères et méthodes d'évaluation. Senach distingue deux dimensions principales pour l'évaluation, l'utilité du produit et son utilisabilité [8]. L'utilité s'intéresse à l'adéquation du logiciel aux objectifs de haut niveau du

client. L'utilisabilité concerne la capacité du logiciel à permettre à l'utilisateur d'atteindre facilement ses objectifs. En ce qui concerne PÉPITEST, l'utilisateur est l'élève dont l'objectif est de résoudre les exercices le mieux possible. Le client est la personne (l'enseignant) ou le système informatique chargé d'effectuer le diagnostic à partir des productions des élèves. L'utilisabilité du logiciel concerne la qualité de l'interface que nous évaluons d'une part selon des critères et recommandations ergonomiques (par exemple [3]) et d'autre part par des expérimentations du logiciel faites en classe. L'utilité concerne la capacité du logiciel à rendre compte du comportement de l'élève pour établir le diagnostic. De ce point de vue, le problème consiste à définir des tâches sur machine qui déterminent des observables équivalents à ceux du test papier - crayon. Concernant l'utilité, l'évaluation du logiciel consiste donc à préciser cette équivalence. Pour cette partie de l'évaluation, nous nous aidons des critères de validation plus spécifiquement adaptés aux EIAO, définis par Balacheff [2], la définition de ces critères de validation est détaillée dans [6].

Les méthodes d'évaluation que nous avons retenues pour la dimension utilisabilité sont les méthodes classiques utilisées pour la conception d'IHM : application des standards des interfaces graphiques, application des recommandations ergonomiques, soumission des prototypes à des jugements d'experts, à des tests informels d'acceptabilité auprès d'utilisateurs hors contexte d'utilisation, et enfin à des expérimentations contrôlées dans des classes.

L'application des recommandations ergonomiques, contrairement à ce que nous pensions, ne s'est pas faite sans problème. En effet, ces recommandations ne sont pas toujours adaptées à des environnements d'apprentissage. Ceci nous a amené à spécifier les particularités des EIAO et plus particulièrement de ceux dédiés à l'évaluation des connaissances.

Particularités des EIAO dédiés à l'évaluation

Dans les paragraphes suivants nous caractérisons les différences qui font des EIAO dédiés à l'évaluation un cas particulier d'interface graphique. Deux points majeurs les distinguent d'une interface classique : d'une part l'apprenant n'est pas un utilisateur comme

¹ Ce logiciel est le résultat de recherches effectuées dans le cadre d'une thèse de doctorat au Laboratoire d'Informatique de l'Université du Maine encadrée par Martial Vivet et Élisabeth Delozanne. Ce travail s'insère dans un projet pluridisciplinaire impliquant le LIUM en informatique et le DIDIREM de Paris 7 en didactique des mathématiques.

les autres ; d'autre part l'évaluation n'est pas une tâche comme les autres.

Deux types d'utilisateurs pour un EIAO

Contrairement au cas des interfaces classiques, il n'est pas trivial de désigner l'utilisateur dans le cas d'un EIAO. En effet, comme l'indiquent Dubourg et Teutsch [4], si l'utilisateur semble naturellement être l'élève, l'enseignant a lui aussi un rôle d'utilisateur. L'enseignant est en effet un utilisateur du logiciel en ce sens qu'il utilise le logiciel pour réaliser sa tâche : enseigner des notions à des élèves. En fait, l'enseignant peut être considéré comme le prescripteur du logiciel (au sens utilisé en économie de personne ayant une influence sur le choix des produits), l'élève en étant l'utilisateur final. Mais son rôle ne s'arrête pas là, il peut en effet également, si le logiciel le prévoit, utiliser le logiciel en tant qu'utilisateur final cette fois, pour préparer son utilisation par les élèves en l'adaptant à ses besoins, en le paramétrant, etc.

Pour résumer, un EIAO a deux types d'utilisateurs : les élèves, utilisateurs finaux et l'enseignant, à la fois prescripteur et utilisateur final.

L'élève n'est pas un utilisateur comme les autres

Si le rôle d'utilisateur de l'élève paraît être clairement désigné, l'élève n'est cependant pas un utilisateur de logiciel tout à fait comme les autres. Il est certes avant tout un utilisateur de dispositif informatique, en tant que tel il rencontre les mêmes difficultés qu'un utilisateur classique (par exemple de quelle manière effectuer telle tâche). Mais l'élève est aussi un utilisateur particulier puisque que, si l'objectif d'un utilisateur de logiciel classique est d'effectuer une tâche de la façon la plus efficace possible, l'objectif d'un élève comporte plusieurs niveaux : l'apprentissage (de la discipline enseignée et non de la manipulation du système), la résolution de problème, la réalisation de tâches. Et seuls ces deux derniers points peuvent avoir un équivalent pour un logiciel classique. L'utilisateur d'un logiciel classique n'a en effet généralement pas d'objectif de haut niveau dans l'utilisation même du logiciel, même si cette utilisation peut s'insérer dans des objectifs de plus haut niveau.

L'évaluation n'est pas une tâche comme les autres

En plus de ces spécificités des interfaces en EIAO, les EIAO dédiés à l'évaluation, en tant que logiciels d'évaluation, comportent une autre spécificité. En effet, que l'on se place au niveau de l'élève ou au niveau du logiciel, l'évaluation n'est pas une tâche comme les autres.

Au niveau de l'élève, l'évaluation peut induire des comportements différents des comportements habituels en EIAO : stress plus grand, hésitation à répondre lorsque la réponse n'est pas certaine (la propension de certains élèves à ne pas répondre plutôt que de risquer de proposer une solution erronée est d'autant plus grande dans le cadre d'une évaluation). Ces modifications du comportement de l'élève ont des conséquences au niveau du logiciel. Il faut par exemple que le logiciel prenne en compte la volonté qu'ont

certain élèves d'effacer une solution qu'ils ont proposé mais dont ils ne sont pas suffisamment sûrs. Pour PÉPITEST, nous avons veillé à ce que le contenu de tous les éléments de réponses soient totalement effaçables. Ceci est valable naturellement pour les cases à cocher dont c'est le fonctionnement normal, pour les zones de saisie, mais aussi pour les annotations faites aux représentations graphiques, pour les listes déroulantes, ainsi que pour les boutons radio alors que cela va à l'encontre de leur fonctionnement normal (elles ne sont normalement en effet pas effaçables).

Au niveau du logiciel, la spécificité la plus marquante de l'évaluation vient du fait que son utilisation est généralement unique, il n'y a donc pas de séance de prise en main possible (ou très limitée dans le temps). L'utilisation du logiciel doit être d'autant plus facile et intuitive.

On pourrait penser que ces spécificités de l'évaluation par rapport aux tâches classiques des utilisateurs sont prises en compte dans les recherches sur la modélisation de l'utilisateur. Mais là encore il y a des différences entre modélisation de l'utilisateur et modélisation de l'élève. En modélisation de l'utilisateur, l'évaluation est faite au cours de l'utilisation, souvent sans que l'utilisateur en soit averti, son but étant de faciliter le travail de l'utilisateur. En modélisation de l'élève, l'objectif est d'évaluer les connaissances et / ou les compétences de l'élève, cet objectif pouvant même être l'objectif final du système comme dans PÉPITE. L'évaluation a donc plus d'importance lorsque l'utilisateur est un élève. Désignée comme objectif principal de l'utilisation du système, elle peut être explicite, et même plus contraignante pour l'utilisateur.

Les spécificités exposées plus haut, ont des conséquences sur l'adaptation des règles ergonomiques au cas des EIAO dédiés à l'évaluation. C'est ce qui est détaillé dans les paragraphes suivants.

Application des recommandations ergonomiques

Hû et Trigano proposent déjà dans [5] une adaptation des critères ergonomiques aux logiciels multimédia pédagogiques. Nous discutons dans ce paragraphe de la prise en compte des recommandations ergonomiques pour des EIAO dédiés à l'évaluation en prenant l'exemple de PÉPITEST et en nous référant à [3]. Ces règles ne peuvent en effet pas être toutes appliquées telles qu'elles étant donné les spécificités de ces interfaces par rapport aux interfaces classiques. Ces particularités nous obligent à adapter les critères ergonomiques appliqués habituellement.

Guidage

“ Le guidage est l'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur (messages, alarmes, labels, etc.), y compris dans ses aspects lexicaux. Quatre sous-critères participent au guidage : incitation, groupement / distinction entre items, feedback immédiat et lisibilité. ” [3]

= Le guidage est particulièrement important dans le cas de PÉPITEST. Chaque élève ne fera le test qu'une seule fois, il ne peut donc pas y avoir de séance de prise en main du logiciel. L'élève est guidé par les consignes, par la structuration de l'écran (groupement / distinction entre items par la localisation et le format, lisibilité), par des modifications de l'aspect du curseur, etc. Le guidage facilite ainsi l'apprentissage de l'utilisation du logiciel.

≠ Cependant, l'objectif étant l'évaluation des connaissances, l'élève ne doit pas être guidé quant au type de réponse qu'il doit fournir (unité, longueur des données, etc.). Le guidage doit donc concerner exclusivement l'interface, et ne doit jamais donner d'indication liée au contenu.

Charge de travail

“ Le critère charge de travail concerne l'ensemble des éléments de l'interface qui ont un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs et dans l'augmentation de l'efficacité du dialogue. Deux sous-critères participent au critère charge de travail : brièveté (qui inclut les critères concision et actions minimales), et densité informationnelle.” [3]

= Le critère de limitation de la charge de travail (concision, actions minimales et densité informationnelle) est pris en compte dans PÉPITEST. Les exercices trop longs sont par exemple scindés en plusieurs parties afin de ne pas afficher trop d'informations sur un même écran.

≠ Le respect de ce critère est cependant en partie problématique dans notre contexte. Certaines fonctionnalités logicielles mises en place pour faciliter l'utilisation (par exemple le glisser - lâcher) peuvent détourner l'attention de certains élèves sur des aspects de manipulation d'interface au détriment des activités mathématiques ce qui peut perturber le diagnostic comme l'indique Artigue concernant la tâche de linéarisation des expressions algébriques [1]. De plus, la densité informationnelle de certains écrans est assez forte. Mais d'une part l'élève est aidé par la structuration de l'écran qui reste stable d'un exercice à l'autre, il sait donc où trouver l'énoncé, où et comment taper sa réponse. D'autre part, dans les exercices concernés, les informations supplémentaires apportées à l'élève lui sont utiles, il les accepte donc facilement, de plus elles apparaissent progressivement (seulement dans la deuxième partie de l'exercice par exemple).

Contrôle explicite

“ Le critère contrôle explicite concerne à la fois la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions. Deux sous-critères participent au contrôle explicite : actions explicites et contrôle utilisateur.” [3]

= L'objectif d'utilisation étant le diagnostic, les réponses sont totalement libres, l'élève exerce un contrôle explicite sur le système. Comme dans l'environnement papier - crayon l'élève peut changer d'exercice et modifier ses réponses quand il le souhaite, sauter des questions et revenir en arrière.

≠ Cette liberté totale de l'utilisateur va plus loin que la notion de contrôle utilisateur.

Adaptabilité

“ L'adaptabilité d'un système concerne sa capacité à réagir selon le contexte, et selon les besoins et préférences des utilisateurs. Deux sous-critères participent au critère adaptabilité : flexibilité et prise en compte de l'expérience de l'utilisateur.” [3]

= La prise en compte du critère d'adaptabilité se traduit d'une part par la flexibilité et d'autre part par la prise en compte de l'expérience de l'utilisateur. Concernant la flexibilité, dans PÉPITEST l'élève peut répondre de différentes façons à une même question (par exemple en utilisant ou non un outil graphique, en utilisant la palette de termes ou en tapant lui-même les phrases). Concernant la prise en compte de l'expérience de l'utilisateur, pour les élèves disposant d'une expérience préalable en informatique, PÉPITEST propose des outils classiques de type copier - coller.

≠ Le logiciel étant utilisé une seule fois pendant la scolarité, la prise en compte de l'expérience de l'utilisateur ne concerne pas l'utilisation des particularités du logiciel mais seulement celle, plus générale, d'une interface graphique (utilisation du copier - coller ou du glisser - déplacer par exemple).

Gestion des erreurs

“ Le critère gestion des erreurs concerne tous les moyens permettant d'une part d'éviter ou de réduire les erreurs, et d'autre part de les corriger lorsqu'elles surviennent. Les erreurs sont ici considérées comme des saisies de données incorrectes, des saisies dans des formats inadéquats, des saisies de commandes avec syntaxe incorrecte, etc. Trois sous-critères participent à la gestion des erreurs : protection contre les erreurs, qualité des messages d'erreurs et correction des erreurs.” [3]

Concernant la gestion des erreurs, distinguons deux types d'erreurs :

- les erreurs dans l'utilisation du logiciel,
- les erreurs au sens de réponses erronées (erreurs conceptuelles).

= Les erreurs dans l'utilisation de PÉPITEST sont limitées étant donné la simplicité des interactions logicielles (erreur dans la manipulation des onglets). Elles sont la plupart du temps récupérables. Il faut cependant noter que la fonctionnalité “ Annuler ” indispensable dans une interface graphique (dans le cas de l'effacement accidentel d'une zone de saisie par exemple) n'a pas été implantée.

≠ Concernant le second type d'erreurs, le but de PÉPITEST étant l'évaluation des connaissances, le logiciel de test ne doit absolument pas identifier et encore moins proposer de corriger des erreurs conceptuelles, ce qui fausserait naturellement le diagnostic : aucune protection contre ce type d'erreurs n'est envisageable. Par contre l'élève peut à tout moment modifier ses entrées.

Homogénéité / cohérence

“ Le critère homogénéité / cohérence se réfère à la façon avec laquelle les choix de conception de l'interface (codes, dénominations, formats, procédures, etc.) sont conservés pour des contextes identiques, et sont différents pour des contextes différents. ” [3]

= Le respect de la cohérence et de l'homogénéité dans l'interface a fait l'objet d'une attention particulière en ce qui concerne l'organisation spatiale des informations, la stabilité de l'écran et l'utilisation des dispositifs physiques de contrôle. Ce point est capital étant donné la diversité des exercices. Si chaque exercice est différent des autres, l'élève doit cependant pouvoir en identifier rapidement les différents éléments.

Le fonctionnement de la gomme est également caractéristique de la prise en compte du critère homogénéité / cohérence dans PÉPITEST. La gomme permet en effet d'effacer le contenu de toutes les zones renfermant des informations (zone de saisie, mais aussi annotations de graphiques, boutons radio, etc.). Son fonctionnement est donc le même dans tous les cas à ceci près que le curseur est légèrement différent si on tente d'effacer du texte ou des annotations : il s'agit de la même action, mais dans un contexte légèrement différent.

Signification des codes et dénominations

“ Le critère signification des codes et dénominations concerne l'adéquation entre l'objet ou l'information affichée ou entrée, et son référent. Des codes et dénominations “ signifiants ” disposent d'une relation sémantique forte avec leur référent. ” [3]

= Les dénominations et les codes utilisés dans PÉPITEST sont ceux choisis en collaboration par les didacticiens des mathématiques et les informaticiens. Le codage propre à l'interface se réduit aux icônes utilisées dans les boutons de la barre d'outils.

Compatibilité

“ Le critère compatibilité se réfère à l'accord pouvant exister entre les caractéristiques des utilisateurs (mémoire, perceptions, habitudes, compétences, âge, attentes, etc.) et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part.

De plus, la compatibilité concerne également le degré de similitude entre divers environnements ou applications. ” [3]

= Le critère de compatibilité concerne en particulier l'adaptation du logiciel au niveau des utilisateurs. Ceci a été fait en tenant compte des connaissances minimales en informatique des élèves amenés à utiliser PÉPITEST.

Concernant la similitude entre les diverses applications utilisées par l'élève, elle est assurée par l'application des standards des interfaces graphiques.

≠ Concernant le degré de similitude entre les divers environnements de travail de l'utilisateur élève (papier - crayon et ordinateur avec PÉPITEST), le critère se réfère

aux problèmes de transfert de tâches d'un environnement papier – crayon à un environnement informatisé que nous avons traité dans [6]. Pour les logiciels pédagogiques et plus encore dans le cas de l'évaluation des connaissances, cette question de transfert est cruciale. Dans le cas de PÉPITEST, la cohérence (à défaut de totale équivalence) entre les deux environnements a fait l'objet d'attentions particulières.

Conclusion

Les problématiques de l'IHM sont de plus en plus souvent intégrées par les chercheurs en EIAO. Mais cette intégration ne se fait pas sans problème du fait des spécificités des logiciels que conçoivent ces derniers. Nous avons montré dans cet article, au travers de l'exemple de l'application de recommandations ergonomiques à un EIAO dédiés à l'évaluation des connaissances, la nécessité d'une adaptation des résultats des recherches en IHM aux spécificités des EIAO.

La méthode empirique employée dans le cas présenté ne suffit cependant pas, nous pensons que des travaux pluridisciplinaires IHM – EIAO devraient permettre de définir des critères de validation spécifiques aux interfaces des EIAO.

Références

- [1] Michèle Artigue, *Analyse des processus d'enseignement en environnement informatique*, Actes de l'Université d'été informatique et enseignement de la géométrie, IREM de Toulouse, 1990, 125:150.
- [2] Nicolas Balacheff, *Didactique et intelligence artificielle*, in N. Balacheff et M. Vivet, *Didactique et intelligence artificielle*, La pensée sauvage éditions, 1994, 7:42.
- [3] Christian Bastien et Dominique Scapin, *Critères ergonomiques pour l'évaluation des interfaces utilisateurs*, RT n°156, INRIA, juin 1993.
- [4] Xavier Dubourg et Philippe Teutsch, *Interface Design Issues in Interactive Learning Environments*, IFIP WG 3.3 Working Conference, Human-Computer Interaction and Educational Tools, Sozopol, mai 1997.
- [5] Olivier Hu et Philippe Trigano, *Proposition de critères d'aide à l'évaluation de l'Interface Homme-Machine des logiciels multimédia pédagogiques*, IHM'98, 10^{èmes} journées Francophones sur l'Interaction Homme-Machine, Nantes, septembre 1998.
- [6] Stéphanie Jean, Élisabeth Delozanne, Pierre Jacoboni et Brigitte Grugeon, *Conception, réalisation et évaluation d'interfaces en EIAO : l'exemple de PÉPITE*, Actes des 5èmes journées EIAO de Cachan, Hermès, 1997, 37:48.
- [7] Stéphanie Jean, Élisabeth Delozanne, Pierre Jacoboni et Brigitte Grugeon, *A Diagnosis based on a Qualitative Model of Competence in Elementary Algebra*, AI-ED 99, Le Mans, 1999, 491:498.
- [8] Bernard Senach, *L'évaluation ergonomique des interfaces homme - machine*, in J.-C. Sperandio éditeur, *L'ergonomie dans la conception des projets informatiques*, Octares éditions, 1993, 69:122.