



HAL
open science

Vers une définition des spécificités des EIAH dédiés à l'évaluation pour l'application de recommandations ergonomiques

Stéphanie Jean-Daubias

► To cite this version:

Stéphanie Jean-Daubias. Vers une définition des spécificités des EIAH dédiés à l'évaluation pour l'application de recommandations ergonomiques. *Revue d'Interaction Homme Machine*, 2003, *Revue d'Interaction Homme-Machine*, 4 (1). hal-01573250

HAL Id: hal-01573250

<https://hal.science/hal-01573250>

Submitted on 8 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.

Vers une définition des spécificités des EIAH dédiés à l'évaluation pour l'application de recommandations ergonomiques

Stéphanie JEAN-DAUBIAS

LISI (Laboratoire d'Ingénierie des Systèmes d'Information)¹
Université Claude Bernard - Lyon 1
Bâtiment Nautibus (710)
43, bd du 11 novembre 1918
69622 Villeurbanne cedex, France

RÉSUMÉ : Les recherches en EIAH (environnements interactifs d'apprentissage avec ordinateur) font de plus en plus souvent appel à des problématiques d'IHM. Dans la plupart des cas, les résultats des recherches en IHM ne sont cependant pas directement applicables au domaine des EIAH. Cet article a pour objet de spécifier les différences qui séparent les EIAH et plus particulièrement les logiciels d'évaluation des connaissances, des logiciels « classiques ». Pour ce faire, nous prenons l'exemple de l'application des recommandations ergonomiques issues des recherches en IHM au logiciel PÉPI'TEST. Nous mettons en évidence les décalages qui existent entre ces recommandations et les spécificités du logiciel.

MOTS-CLEFS : Évaluation ergonomique, EIAH.

1 Introduction

Avec l'avènement des interfaces graphiques, les chercheurs en EIAH (environnements interactifs d'apprentissage avec ordinateur), 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 EIAH des logiciels interactifs « classiques », différences d'autant plus importantes lorsqu'il s'agit de logiciels dédiés à l'évaluation des connaissances. D'une part en effet l'élève n'est pas un utilisateur comme les autres, d'autre part l'évaluation des apprenants n'est pas une tâche comme les autres.

Dans cet article, nous tentons de spécifier ces différences qui séparent les EIAH des IHM classiques, en prenant l'exemple de l'application des critères ergonomiques proposés par Bastien et Scapin [Bastien & Scapin 1993] à PÉPI'TEST, un logiciel d'évaluation des connaissances des élèves.

Nous commençons par une présentation succincte de PÉPI'TEST et de l'évaluation de son interface. Nous précisons ensuite les particularités des EIAH dédiés à l'évaluation des connaissances avant d'indiquer dans quelle mesure nous avons dû adapter les critères ergonomiques choisis pour l'évaluation de l'interface de PÉPI'TEST. Nous tentons enfin de généraliser les adaptations proposées en étudiant leur champ d'application.

¹ Cet article présente des travaux qui ont été réalisés au Laboratoire d'Informatique de l'Université du Maine.

1.1 PÉPITEST

PÉPITE est un système d'aide au diagnostic de compétences, qui aide l'enseignant à évaluer les connaissances des élèves des classes de troisième et seconde de l'enseignement français (avec des élèves d'environ 15 ans) en algèbre élémentaire². Ce logiciel comporte trois modules :

- PÉPITEST (cf. ① Figure 1), le logiciel élève, propose les exercices aux élèves et recueille leurs réponses. Notons que ce logiciel n'est utilisé qu'une seule fois par les élèves.
- PÉPIDIAG (cf. ② Figure 1) analyse les productions des élèves, à partir des observables fournis par PÉPITEST, selon une grille d'analyse multidimensionnelle, en complétant une matrice.
- PÉPIPROFIL (cf. ③ Figure 1) établit les profils des élèves par analyse transversale de la matrice de diagnostic complétée par PÉPIDIAG et les présente aux enseignants.

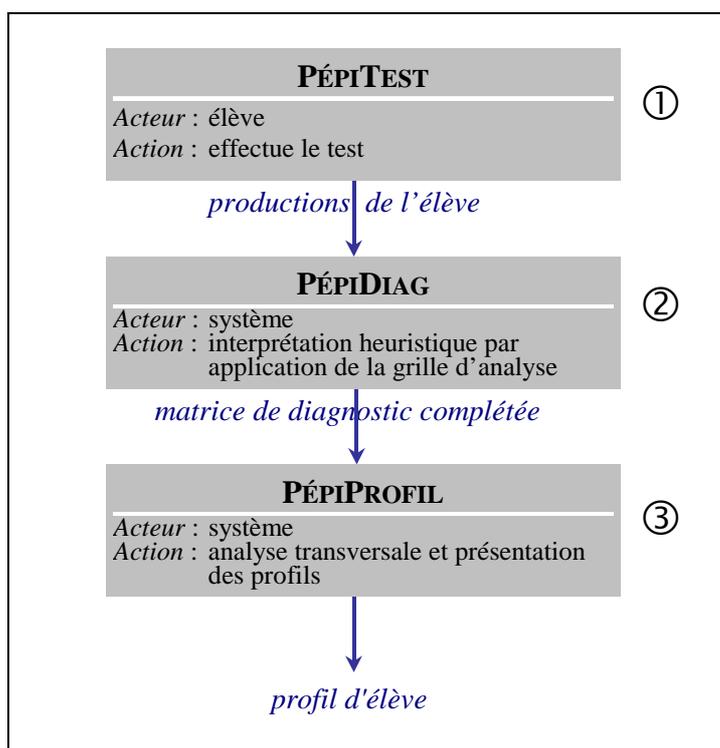


Figure 1 : L'architecture de PÉPITE.

PÉPITE est entièrement développé. Il a été testé à plusieurs reprises en classe (avec au total une centaine d'élèves). Il est également utilisé en formation des maîtres.

PÉPITEST propose un ensemble de 22 exercices aux élèves et rassemble leurs réponses. Les exercices proposés sont très différents les uns des autres, ils donnent lieu à des réponses

² 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 (LIUM) encadrée par Martial Vivet et Élisabeth Delozanne. Ce travail s'insère dans le projet PÉPITE : une recherche pluridisciplinaire impliquant le LIUM en informatique et le laboratoire DIDIREM de l'Université Paris 7 en didactique des mathématiques.

S. Jean-Daubias, « Vers une définition des spécificités des EIAH dédiés à l'évaluation »

totallement ouvertes pour permettre à l'élève d'exprimer ses conceptions sans contrainte. On pourra trouver une description plus détaillée de PÉPI TEST dans [Jean 2000a] et [Jean et al. 1999].

PÉPI TEST propose tout d'abord un écran de présentation et un écran d'identification de l'élève. De tels écrans peuvent paraître anodins, ils ont cependant un rôle important dans le test : l'objectif de l'écran de présentation est de dédramatiser le test en expliquant à l'élève le but du test ; l'écran d'identification a quant à lui pour but d'acquérir des informations concernant l'élève, mais c'est surtout l'occasion pour l'élève de commencer à prendre en main le logiciel.

PÉPI TEST comporte ensuite 36 écrans (la Figure 2 montre un exemple d'écran). Chacun de ces écrans est composé d'un classeur à onglets qui contient les exercices (un exercice par onglet), d'un menu proposant en particulier les différents outils proposés à l'élève (calculatrice, cahier de brouillon, outil de représentation graphique, etc.) et d'une palette permettant de produire des expressions algébriques et d'annoter des figures. Un système de repérage visuel permet par ailleurs à l'utilisateur de distinguer l'exercice courant, les exercices consultés, les exercices non encore consultés et les exercices marqués par l'élève comme étant terminés (cf. Figure 2).

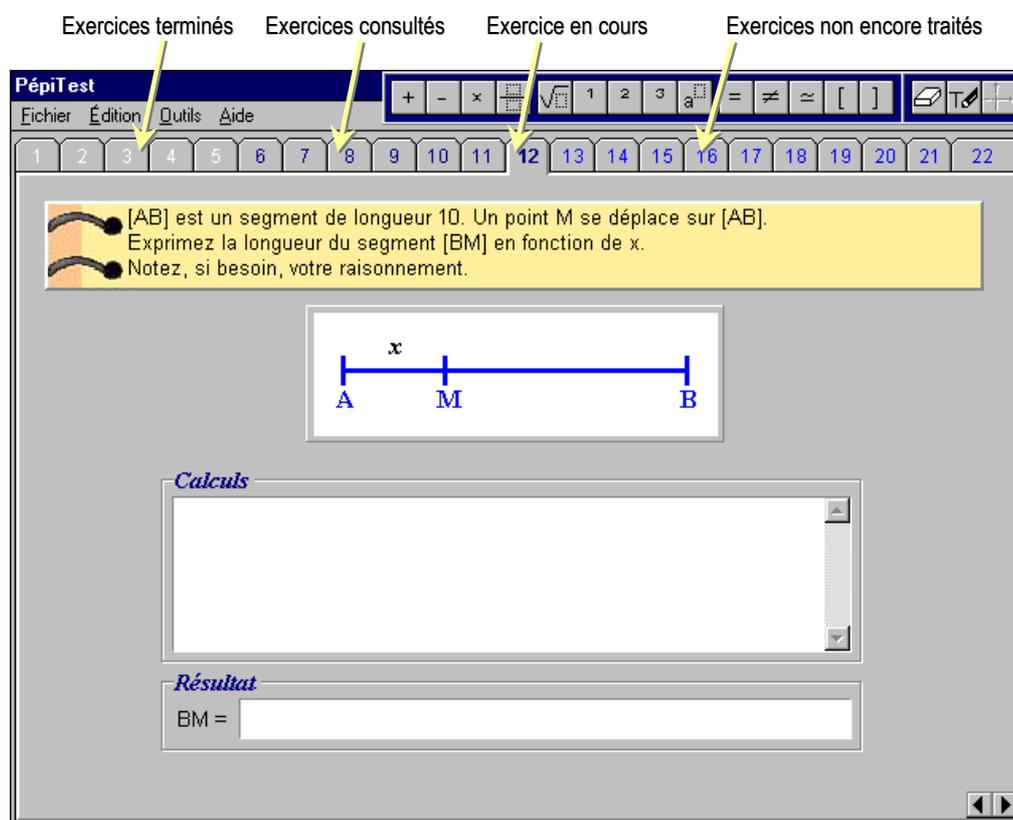


Figure 2 : Un exemple d'exercice de PÉPI TEST.

Au cours de la session, certaines actions de l'élève (utilisation des outils par exemple) sont mémorisées. À la fin de la session, les réponses de l'élève sont enregistrées dans un fichier afin d'être analysées par PÉPIDIAG.

1.2 L'évaluation de l'interface de PÉPITEST

La conception de PÉPITE est le résultat d'un travail pluridisciplinaire. C'est pourquoi pour PÉPITEST, nous avons adopté une démarche de conception à la fois itérative et participative qui impose le recensement précoce des critères et méthodes d'évaluation. Nous avons repris la distinction de Bernard Senach désignant deux dimensions principales pour l'évaluation de l'interface : l'*utilité* du logiciel qui s'intéresse à l'adéquation du logiciel aux objectifs de haut niveau du *client* et son *utilisabilité* qui concerne la capacité du logiciel à permettre à l'utilisateur d'atteindre facilement ses objectifs [Senach 1993].

En ce qui concerne PÉPITEST, l'*utilisateur* est l'élève dont l'objectif est de résoudre les exercices le mieux possible. Le *client* est le système informatique chargé d'effectuer le diagnostic à partir des productions des élèves (c'est-à-dire PÉPIDIAG, cf. © Figure 1), mais aussi plus globalement l'enseignant qui souhaite obtenir les résultats du diagnostic de PÉPITE.

L'*utilité* de PÉPITEST 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 exercices sur ordinateur qui conduisent à la production par les élèves, d'observables équivalents à leurs productions sur papier - crayon. Concernant l'utilité, l'évaluation du logiciel consiste donc à préciser cette équivalence. Pour cette partie de l'évaluation de PÉPITEST, nous nous aidons des critères de validation plus spécifiquement adaptés aux EIAH, définis par Nicolas Balacheff [Balacheff 1994], la définition de ces critères de validation est détaillée dans [Jean 2000a]. Pour résumer, cette partie de la validation de PÉPITEST comporte deux étapes : vérifier que les observables obtenus à partir du logiciel sont représentatifs du comportement de l'élève et vérifier que ces observables permettent de construire des profils qui correspondent à l'image que l'enseignant a des connaissances des élèves.

L'*utilisabilité* de PÉPITEST concerne la qualité de l'interface. 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 (les standards Windows), application de critères et recommandations ergonomiques (par exemple [Bastien & Scapin 1993]), 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 ne s'est pas faite sans problème. En effet, ces recommandations ne sont pas toujours adaptées à des environnements d'apprentissage. Ceci nous a amenée à spécifier les particularités des EIAH et plus particulièrement de ceux dédiés à l'évaluation des connaissances.

2 Particularités des EIAH dédiés à l'évaluation des connaissances

Dans les paragraphes suivants, nous caractérisons les différences qui font des EIAH dédiés à l'évaluation des connaissances, un cas particulier d'interface graphique. Trois points distinguent les EIAH d'une interface classique : d'une part il existe deux types d'utilisateurs pour EIAH ; d'autre part l'apprenant n'est pas un utilisateur comme les autres ; de plus, l'évaluation des connaissances n'est pas une tâche comme les autres.

2.1 Deux types d'utilisateurs pour un EIAH

Contrairement au cas des interfaces classiques, il n'est pas trivial de désigner l'utilisateur dans le cas d'un EIAH. En effet, comme l'indiquent Xavier Dubourg et Philippe Teutsch [Dubourg & Teutsch 1997], mais également Olivier Hù [Hù 2000], 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 EIAH a deux types d'utilisateurs : les élèves, utilisateurs finaux et l'enseignant, à la fois prescripteur et parfois utilisateur final.

2.2 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, si l'objectif d'un utilisateur de logiciel classique est d'effectuer une tâche de la façon la plus efficace possible, les objectifs d'un élève sont non seulement la résolution de problème et la réalisation de tâches (qui ont un équivalent dans l'utilisation d'un logiciel classique), mais également l'apprentissage (de la discipline enseignée et non de la manipulation du système). 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.

2.3 L'évaluation des connaissances n'est pas une tâche comme les autres

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

Au niveau de l'élève, l'évaluation des connaissances peut induire des comportements différents des comportements habituels en EIAH : 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 des connaissances). 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 certains élèves d'effacer une solution qu'ils ont proposée mais dont ils ne sont pas suffisamment sûrs. Pour PÉPI'TEST, 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 des connaissances vient du fait que l'utilisation du logiciel 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 des connaissances 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 des connaissances 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 des connaissances 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 EIAH dédiés à l'évaluation des connaissances. C'est ce que nous détaillons dans la section suivante.

3 Application des recommandations ergonomiques

Nous discutons dans cette partie de la prise en compte des recommandations ergonomiques pour des EIAH dédiés à l'évaluation des connaissances en prenant l'exemple de PÉPITEST et en nous référant aux critères proposés par Christian Bastien et Dominique Scapin pour l'INRIA [Bastien & Scapin 1993]. Ces règles ne peuvent en effet pas être toutes appliquées telles quelles étant données les spécificités de ces interfaces par rapport aux interfaces classiques. Ces particularités nous obligent à adapter les critères ergonomiques appliqués habituellement. Dans la suite, pour chaque critère donné dans [Bastien & Scapin 1993], nous indiquons dans quelle mesure nous appliquons ce critère (partie annoncée par le symbole =) et dans quelle mesure nous devons l'adapter (partie annoncée par le symbole ≠). Notons que ces adaptations ont été faites dans l'optique de l'évaluation formative de PÉPITEST. Notre but n'était pas de proposer de nouveaux critères. C'est la raison pour laquelle les modifications faites ne sont pas générales mais adaptées au cas de PÉPITEST, et n'ont pas fait l'objet d'une validation.

3.1 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. »

[Bastien & Scapin 1993]

= 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.

3.2 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. »

[Bastien & Scapin 1993]

= 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 Michèle Artigue concernant la tâche de linéarisation des expressions algébriques [Artigue 1990]. 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). Ce dernier point peut être retrouvé dans tout dispositif nécessitant la présence simultanée de nombreuses informations à l'écran.

3.3 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. »

[Bastien & Scapin 1993]

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

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

3.4 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. »

[Bastien & Scapin 1993]

= 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 de l'élève, 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).

3.5 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. »

[Bastien & Scapin 1993]

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.

3.6 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. »

[Bastien & Scapin 1993]

= 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 (énoncé, zone de calcul, de résultat, etc.).

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.

3.7 Signifiante des codes et dénominations

« Le critère signifiante 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. »

[Bastien & Scapin 1993]

= Les dénominations et les codes utilisés dans PÉPI TEST 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.

3.8 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. »

[Bastien & Scapin 1993]

= 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 PÉPI TEST.

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 PÉPI TEST), le critère se réfère aux problèmes de transfert de tâches d'un environnement papier – crayon à un environnement informatisé que nous avons traité dans [Jean 2000a] et [Jean 2000c]. Pour les logiciels pédagogiques et plus encore dans le cas de l'évaluation des connaissances, cette question du transfert est cruciale. Dans le cas de PÉPI TEST, la cohérence (à défaut de totale équivalence) entre les deux environnements a fait l'objet d'attentions particulières.

4 Vers une généralisation de ces adaptations

L'adaptation que nous avons faite des critères de Christian Bastien et Dominique Scapin est certes liée aux spécificités de PÉPI TEST, elle est toutefois en partie généralisable. Dans cette partie, nous tentons de mettre en évidence l'intérêt des adaptations que nous avons faites des critères ergonomiques utilisés pour évaluer PÉPI TEST, pour évaluer des logiciels au champ d'application plus large que PÉPI TEST. Pour cela, nous traitons séparément les spécificités de PÉPI TEST : apprentissage, évaluation des connaissances et utilisation unique du logiciel. Pour chacune de ces trois spécificités, nous indiquons quels critères ergonomiques (toujours en nous référant [Bastien & Scapin 1993]) sont particulièrement importants et quels critères doivent au contraire être adaptés.

4.1 Apprentissage

PÉPI TEST est un EIAH, même s'il est dédié à l'évaluation des connaissances et non directement à l'apprentissage, il s'agit toutefois d'un logiciel s'insérant dans une situation d'apprentissage. Cette spécificité est sans doute celle qui éloigne le plus PÉPI TEST des IHM classiques. Les adaptations que nous avons dû faire des critères ergonomiques en raison de cette caractéristique ne peuvent être réutilisés que dans d'autres EIAH.

Le point le plus important pour nous, concerne le critère de guidage. Dans le cadre d'un environnement d'apprentissage, le *guidage* doit porter uniquement sur l'utilisation du logiciel et pas sur le contenu. Par ailleurs, dans un EIAH, la *charge de travail* peut être volontairement importante en raison de choix didactiques ou pédagogiques. Enfin, le critère de *compatibilité* concerne la possibilité pour l'apprenant de mettre en œuvre avec le logiciel des compétences acquises en environnement papier – crayon et vice - versa. Il n'implique pas uniquement des questions d'IHM, mais également de didactique.

4.2 Évaluation des connaissances

Pour tous les dispositifs d'évaluation des connaissances, qu'ils soient ou non liés à une situation d'apprentissage, le critère de *contrôle explicite* est particulièrement important puisque que le logiciel doit laisser l'initiative à l'apprenant. De même, la *signification des codes et dénominations* doit être étudiée avec soin afin de ne pas laisser d'ambiguïtés qui pourraient perturber l'apprenant. Quant au critère de *gestion des erreurs*, il nécessite une précision. On doit en effet distinguer les erreurs conceptuelles sur lesquelles le logiciel ne doit en aucun cas intervenir, des erreurs de manipulation du logiciel, qui doivent, comme dans tout dispositif informatique, être évitées et le cas échéant, réversibles.

4.3 Utilisation unique

L'utilisation unique d'un logiciel n'est pas forcément liée à une évaluation des connaissances, certains dispositifs, comme certaines bornes interactives, peuvent également être utilisés une seule fois. Dans ce cas, le *guidage*, la *signification des codes et dénominations* et *l'homogénéité et la cohérence* sont particulièrement importants pour faciliter la prise en main du système par l'utilisateur. Quant à la prise en compte de l'expérience de l'utilisateur, elle ne peut concerner que l'expérience de dispositifs informatiques, et non celle du logiciel étudié.

5 Conclusion

Les problématiques de l'IHM sont de plus en plus souvent intégrées par les chercheurs en EIAH. 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 EIAH dédié à l'évaluation des connaissances, la nécessité d'une adaptation des résultats des recherches en IHM aux spécificités des EIAH.

Nous sommes toutefois consciente que la méthode empirique employée dans le cas présenté ne suffit pas : nous pensons que seuls des travaux pluridisciplinaires IHM – EIAH devraient permettre de définir des critères de validation spécifiques aux interfaces des EIAH.

6 Références

- [Artigue 1990] Artigue M., *Analyse des processus d'enseignement en environnement informatique*, Actes de l'Université d'été informatique et enseignement de la géométrie, IREM de Toulouse, 1990, pp. 125-150.
- [Balacheff 1994] Balacheff N., *Didactique et intelligence artificielle*, in N. Balacheff et M. Vivet, *Didactique et intelligence artificielle*, La pensée sauvage éditions, 1994, pp. 7-42.
- [Bastien & Scapin 1993] Bastien C. et Scapin D., *Critères ergonomiques pour l'évaluation des interfaces utilisateurs*, Rapport Technique n°156, INRIA, juin 1993.
- [Dubourg & Teutsch] Dubourg X. et Teutsch P., *Interface Design Issues in Interactive Learning Environments*, IFIP WG 3.3 Working Conference, Human-Computer Interaction and Educational Tools, Sozopol, mai 1997.
- [Hù 2000] Hù O., *Quelques réflexions sur l'évaluation ergonomique et le contexte pédagogique*, Actes des Rencontres Jeunes Chercheurs en IHM 2000, mai 2000, pp 95-66.
- [Hù, Trigano & Crozat 2000] Hù O., Trigano P. et Crozat S., *Une aide à l'évaluation des logiciels multimédias de formation*, Sciences et techniques éducatives, Numéro spécial "Communication Homme-Machine & Apprentissage", Hermès, 2000.
- [Jean 2000a] Jean S., *PÉPITE : un système d'assistance au diagnostic de compétences*, Thèse de doctorat de l'Université du Maine, 21 janvier 2000.
- [Jean 2000n] Jean S., *Application de recommandations ergonomiques : spécificités des EIAH dédiés à l'évaluation*, Actes des Rencontres Jeunes Chercheurs en IHM 2000, mai 2000, pp 39-42.
- [Jean 2000c] Jean S., *Manipulation des expressions algébriques en environnement informatisé*, Actes des journées "Environnements informatiques de calcul symbolique et apprentissage des mathématiques", Commission inter-IREM Math Informatique, juin 2000.
- [Jean et al. 1999] Jean S., Delozanne E., Jacoboni P. et Grugeon B., *A Diagnosis based on a Qualitative Model of Competence in Elementary Algebra*, AI-ED 99, Le Mans, 1999, pp. 491-498.
- [Senach 1993] Senach B., *L'évaluation ergonomique des interfaces homme - machine*, in J.-C. Sperandio éditeur, *L'ergonomie dans la conception des projets informatiques*, Octares éditions, 1993, pp. 69-122.