Interaction Homme-Machine : Points d'ancrage entre ergonomie et génie logiciel

Joëlle Coutaz
LGI-IMAG, BP 53, 38041 Grenoble Cedez 9
Tel. 76 51 48 54, fax : 76 44 66 75, email : Joelle.Coutaz@imag.fr

RESUME. La conception des systèmes interactifs telle qu'on la pratique en ergonomie et les processus de développement du génie logiciel s'effectuent le plus souvent de manière disjointe. Et lorsque la collaboration existe, la communication entre les équipes partenaires souffre de l'absence de support commun : modèles, notations et outils partagés ou compatibles font défaut. Dans cet article, nous proposons d'étudier les points d'ancrage privilégiés entre les pratiques de l'ergonomie et les méthodes du génie logiciel.

MOTS-CLES. Conception d'interface homme-machine, processus de développement, génie logiciel, ergonomie.

1. Introduction

La conception des systèmes interactifs telle qu'on la pratique en ergonomie et les processus de développement du génie logiciel s'effectuent le plus souvent de manière disjointe. Et lorsque la collaboration existe, la communication entre les équipes partenaires souffre de l'absence de support commun : modèles, notations et outils partagés ou compatibles font défaut. Dans cet article, je propose d'étudier les points d'ancrage privilégiés entre les pratiques de l'ergonomie et les méthodes du génie logiciel. Nous verrons en conclusion que l'interopérabilité et la compatibilité des notations conditionnent le succès de la collaboration entre ergonomes et informaticiens.

Pour les besoins de l'exposé, j'utiliserai le modèle en V qui structure de manière simple le processus de développement des logiciels. Celui-ci sera brièvement présenté en 3. En 4, je montrerai comment informaticiens et ergonomes peuvent collaborer au cours des étapes du modèle en V. Mais avant cela, j'aimerais dénoncer quelques écueils et tentations.

2. Ecueils et tentations

La technologie des IHM s'est enrichie récemment d'outils de développement efficaces. Parmi ces outils, nous relevons les générateurs d'interfaces, tels Interface Builder et XFaceMaker, qui permettent à moindre coût de créer des maquettes d'interface. Ces maquettes sont censées faire l'objet d'évaluations avec des utilisateurs représentatifs, d'être modifiées de manière itérative jusqu'à ce que les critères d'utilisabilité soient satisfaits.

Dans les faits, les moyens financiers et le temps dédiés à *l'Assurance Utilisabilité* font souvent l'objet de coupures radicales. Les évaluateurs sont les

Actes 01Design'95, Aspects Communicatifs en Conception, 4ème Table Ronde Francophone sur la Conception, 11-13 Janvier 1995, Autrans, pp.157-164

développeurs avec leur biais mais aussi leur manque de formation : connaissent-ils les critères d'utilisabilité et savent-ils définir des protocoles expérimentaux? Le plus souvent, le développeur fait appel à sa seule intuition et, fort des capacités de son générateur, croît concevoir une interface adaptée. C'est ce que j'appelle le syndrome de l'ABS. (L'ABS, qui évite le blocage des roues du véhicule, fausse l'estimation des distances de freinage et incite le conducteur à retarder l'action de freinage.) De même, les générateurs d'interface ne font pas du développeur un spécialiste en ergonomie!

La première tentation est donc d'éliminer de l'équipe de développement les ergonomes sous prétexte que l'on dispose d'un générateur d'interface. Le second danger est de croire que l'exploration sauvage des technologies nouvelles va permettre d'améliorer les interfaces. On évoque à grand coup publicitaire les Réalités Virtuelles, les systèmes multimédias mais aussi les interfaces multimodales. Si la puissance des machines, si les progrès des systèmes de reconnaissance et de synthèse permettent aujourd'hui d'augmenter les capacités sensori-motrices et représentationnelles des systèmes, il convient aussi de s'interroger sur leur adéquation aux besoins, aux objectifs et aux caractéristiques des utilisateurs. Certes, soyons ouverts à la créativité, imaginons de nouveaux paradigmes d'interaction, mais il faut en comprendre, voire prédire, l'apport et les limites. Sachons apprivoiser l'invention dans un cadre de développement.

3. Le modèle en V

Le modèle en V [McDermit et al., 84] convient au développement de systèmes dont la finalité est bien cernée. Pour les systèmes à risque, on adoptera le modèle en spirale [Boehm, 88] qui conduit, par itérations et incréments, de l'analyse à une solution satisfaisante. Ici, nous retenons le modèle en V pour sa simplicité mais aussi pour sa ressemblance avec certaines méthodes de développement déjà ancrées en ergonomie [Dowell et al., 89].

La figure 1 montre la structuration du processus de développement en étapes. La pente descendante du V couvre les étapes de réification progressive du système depuis la définition des besoins jusqu'à son incarnation logicielle. Sur la pente ascendante, à chacune des étapes de réification correspond un ensemble de tests qui permettent de vérifier et/ou de valider l'étape en regard du code produit. De manière générale, les tests sont effectués en séquence une fois le codage réalisé, mais ils doivent être spécifiés en même temps que leur correspondant dans la pente descendante. Parfois, certaines vérifications ou validations sont effectuées lors de la descente, avant même le codage. Le modèle en V autorise ainsi des retours arrière limités.

L'analyse des besoins permet d'établir, en relation avec le client, les services requis du système et les contraintes de développement. Cette activité donne lieu à un cahier des charges qui sert de document contractuel entre le maître d'œuvre et le client.

La conception du système consiste à définir une solution matérielle et logicielle qui répond à l'analyse des besoins et aux contraintes. Cette activité produit des spécifications externes. Dans le cas qui nous intéresse, ces spécifications décrivent l'interface utilisateur du système.

La conception globale constitue l'étape préliminaire orientée vers la mise en œuvre logicielle : nous quittons l'espace de conception de l'IHM pour pénétrer dans

l'espace logiciel réservé aux informaticiens. La conception globale donne lieu à un dossier d'architecture générale.

La conception détaillée décrit les choix algorithmiques, précise la signature des procédures et des fonctions, définit les structures de données les plus pertinentes et spécifie les protocoles de communication. C'est le monde privilégié de l'analyste programmeur. Ici, l'ergonome n'a plus sa place! Il n'est plus consulté alors que des décisions logicielles dirigées par des considérations techniques peuvent avoir un impact néfaste sur l'utilisabilité.

Puis intervient le *codage* dans un langage de programmation avant d'entreprendre les *tests unitaires*. Ces derniers permettent de vérifier que les composants modulaires du système répondent chacun à leurs spécifications.

Les *tests d'intégration* servent à vérifier que les modules réalisés indépendamment interagissent correctement.

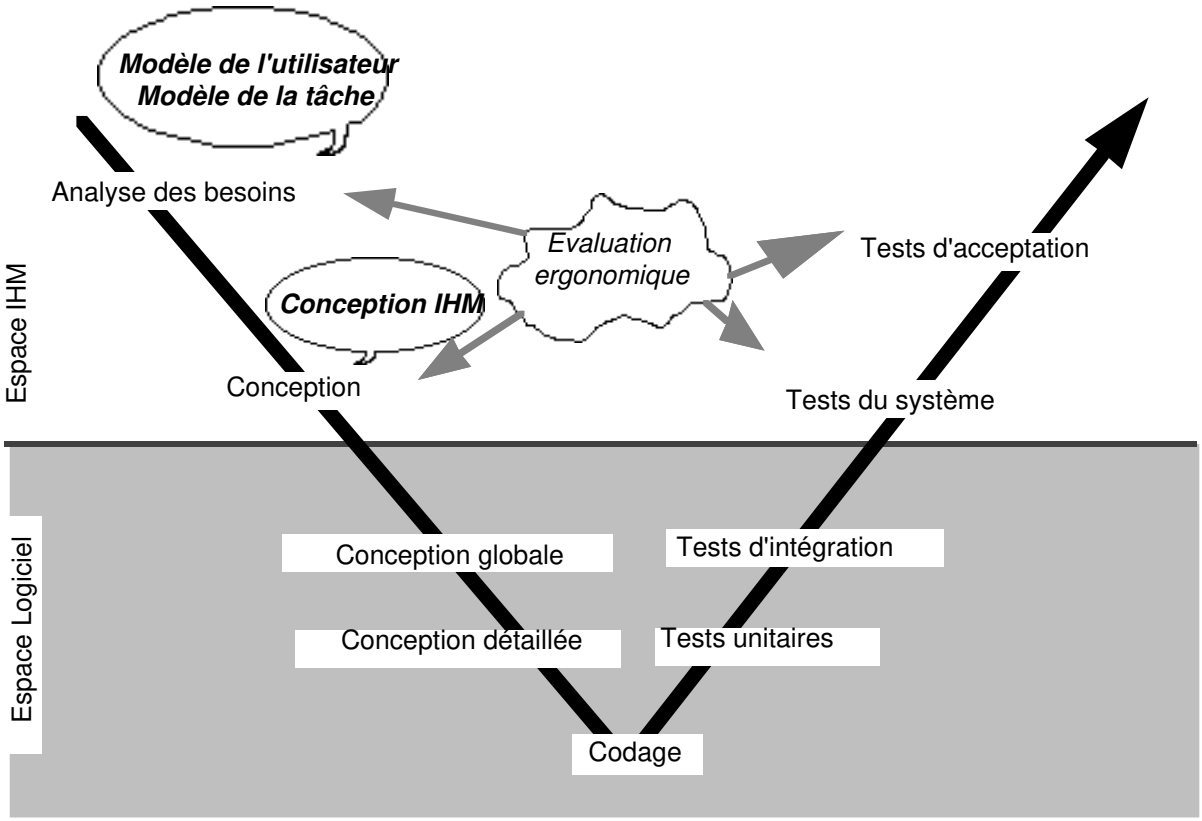

Figure 1 : *Les étapes du modèle en V.*

Les tests du système servent à vérifier que les éléments de la solution exprimés dans le dossier de spécifications externes sont présents. Ici, nous pénétrons à nouveau dans un espace commun à l'ergonome et au développeur.

Enfin, *les tests d'acceptation* permettent de vérifier que les besoins exprimés dans le cahier des charges sont couverts. C'est le dernier contrôle avant la livraison du produit.

4. Les points d'ancrage

Nous avons identifié quatre points d'ancrage entre l'ergonomie et le génie logiciel : la définition des besoins fonctionnels, l'établissement du plan qualité, les tests d'évaluation de l'utilisabilité, l'établissement des spécifications externes. Nous reprenons les trois premiers points.

4.1.Besoins fonctionnels et cahier des charges.

La structure du cahier des charges répond généralement à une norme. Par exemple, la norme AFNOR regroupe à la fois la définition et la spécification des besoins. On entend par "définition des besoins" la présentation du "contexte" et une "description générale" du problème. La spécification est une "expression détaillée" du problème. Quelle que soit la norme adoptée, il faut distinguer les besoins fonctionnels c'est-à-dire les services attendus du système informatique, des besoins non fonctionnels et notamment les qualités requises et les contraintes.

En ergonomie, les besoins fonctionnels sont identifiés via des méthodes aujourd'hui bien documentées. A ce titre, nous relevons Diane [Barthet, 88] et Muse [Whitefield et al., 91] (analyse du domaine, études de systèmes existants, interviews, etc.). Alors que ces méthodes centrent l'analyse sur l'utilisateur et sa tâche, les informaticiens ignorent la tâche pour ne modéliser que les services que le système devra rendre. Il convient de ne pas sauter les étapes : il faut au préalable représenter explicitement l'ensemble des tâches à réaliser, puis dans un second temps, identifier celles qui devront migrer vers le système. De même, les informaticiens ignorent les critères de fréquence d'une tâche, de complexité, de relations temporelles, c'est-à-dire autant d'enseignements que les représentations formelles informatiques tels SADT et OMT, devraient véhiculer.

Là, nous constatons une perte d'information entre les notations utilisées par les ergonomes et celles adoptées par les informaticiens. Nous observons aussi une différence d'appréhension du problème lorsque l'ergonome raisonne en terme de tâche alors que les méthodes avancées de modélisation en génie logiciel prône le tout-objet. Comment dans ces conditions, transférer un modèle orienté tâche en une conception logicielle dirigée par les objets?

4.2. Le plan qualité

Le plan qualité, qui fait généralement partie du cahier des charges, régit les tests d'évaluation de la pente ascendante du V. Il précise la qualité requise, les métriques et les moyens de mesure. McCall fut le premier à caractériser la qualité en termes de facteurs et de critères [McCall, 77]. Parmi ces facteurs, nous relevons *l'utilisabilité*. D'après McCall, l'utilisabilité se mesure au moyen de deux critères : la facilité d'apprentissage et l'opérabilité. Il est clair que les pratiques de l'ergonomie cognitive ne peuvent se suffire de propriétés aussi générales.

En ergonomie, l'évaluation de l'utilisabilité s'appuie sur des principes comme ceux de Smith et Mosier [Mosier et al., 86] dont l'exploitation nécessite expertise et art du compromis. Scapin [Scapin, 90] et Nielsen [Nielsen, 89] proposent des catalogues simplifiés plus accessibles. Toutefois ces règles, qui ne sont pas formalisées, laissent la place à l'interprétation [Whitefield et al., 91]. Elles ne sont pas transférables telles quelles au non spécialiste.

Pour répondre à la généralité de McCall et à l'absence de formalisation en ergonomie, nous avons décliné la notion d'utilisabilité en un ensemble de propriétés précises, comme l'observabilité, l'honnêteté, la redondance, l'équivalence, etc. Ces propriétés sont pertinentes à la fois pour les tests d'ergonomie et la conception logicielle [Abowd et al., 92]. Nous avons déjà démontré leur intégration dans les modèles conceptuels d'architecture tel PAC-Amodeus [Nigay et al., 93]. Ces modèles, qui concernent l'étape de conception globale du système, jouent le rôle de charnière entre la conception ergonomique et la conception logicielle. Il est donc important qu'ils véhiculent les "bonnes propriétés". Nous travaillons actuellement à l'exploitation de ces propriétés en évaluation.

4.3. L'évaluation de l'utilisabilité

Comme le montre la figure 1, l'évaluation ergonomique peut se pratiquer de manière prédictive dès la remise des spécifications externes ou bien de manière expérimentale, voire dès la spécification des besoins. Dans les deux cas, génie informatique et ergonomie peuvent utilement collaborer.

Actuellement nous cherchons à appliquer nos propriétés à l'évaluation prédictive. Par exemple, nous dirons que le principe d'observabilité est transgressé si, dans les spécifications de l'interface utilisateur, la précondition qui régit l'exécution d'une tâche ne peut s'exprimer en termes perceptibles par l'utilisateur. Ces mêmes propriétés peuvent également servir à structurer les évaluations expérimentales dont les plans de test peuvent être structurés selon les recommandations du génie logiciel : pour chaque critère, identifier le domaine de valeurs, les performances observées, le coût (ici, il convient de considérer les coûts conatifs, cognitifs et affectifs que le génie logiciel passe sous silence), etc. On trouvera dans [Balbo, 94] des recommandations sur les méthodes d'évaluation à recruter en fonction du projet, de l'étape, des ressources, etc. Nous résumons ici les point essentiels.

La figure 2 présente de manière synthétique les dimensions de notre espace de choix. Ces dimensions représentent cinq catégories de préoccupations : les moyens humains, les ressources matérielles, les connaissances requises, les facteurs situationnels, et la nature des résultats fournis.

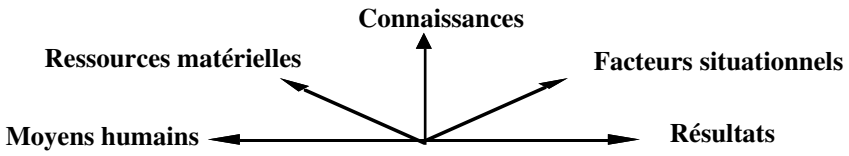

Figure 2 : *Choix de méthodes d'évaluation ergonomique.*

Les moyens humains désignent l'ensemble des acteurs impliqués directement ou indirectement dans l'activité d'évaluation. Nous distinguons trois classes d'intervenants : les sujets, les évaluateurs spécialistes et une catégorie qui regroupe tous les autres acteurs (les développeurs, les concepteurs d'interface, les clients, les agents commerciaux, etc.). Les acteurs peuvent appartenir à l'organisation chargée du développement (il s'agit alors d'acteurs *locaux*), ou bien sont recrutés à *l'extérieur*. Cette distinction a de l'importance dans l'évaluation des coûts du recrutement. Les facteurs situationnels et notamment le type de projet, l'enveloppe budgétaire et le planning conditionneront l'origine et le nombre de sujets et de spécialistes qu'il sera possible de retenir. Pour la dernière catégorie d'acteurs (c'est-

à-dire les intervenants autres que les sujets et les spécialistes), il est important de s'interroger sur leur niveau de culture en matière d'assurance qualité (au sens large du génie logiciel) mais aussi sur leur attitude vis-à-vis des principes de l'ergonomie. A fortes réticences doit correspondre, nous semble-t-il, une technique d'évaluation légère!

Les ressources matérielles recouvrent tous les moyens physiques impliqués dans l'activité d'évaluation. Nous convenons de faire la distinction entre les ressources servant d'instruments de capture et les ressources matérielles faisant l'objet d'une évaluation. Les ressources matérielles pour la capture incluent les questionnaires et interviews, les régies vidéo (éventuellement portatives) ou laboratoires spécialisés, les ordinateurs. Toutes ces ressources peuvent être caractérisées par un coût mais aussi par la *portée*, la *nature* et le *niveau d'abstraction* des données qu'elles permettent de recueillir. Le génie logiciel a un rôle à jouer dans la capture automatique par ordinateur.

Actuellement la capture par ordinateur permet d'enregistrer des actions physiques de type clavier-souris et au mieux la manipulation d'objets simples dans le cadre de tâches élémentaires [Hammontree et al., 92]. Lorsque la manipulation d'objet est prise en compte, seuls sont repérables les objets prédéfinis des boîtes à outils (par exemple, les menus et les boutons de Motif). En conséquence, la capture automatique ne couvre que les phénomènes de bas niveau d'abstraction et de plus, pour un niveau d'abstraction donné, présentent des ruptures de portée : les actions physiques ne sont pas toutes repérées (et notamment les métacomportements verbaux et gestuels) et les objets manipulables ne sont pas tous observés (et notamment les objets construits à façon au-dessus des bibliothèques graphiques). Si l'objectif est une évaluation fiable, ces limitations exigent de pratiquer des tests complémentaires. Le projet NEIMO de notre équipe vise à combler ces lacunes [Salber et al., 93].

Les connaissances requises pour mettre en application une méthode sont de deux sortes: les premières ont trait au coût cognitif d'accès à la méthode, la seconde concerne les descriptions nécessaires en entrée au bon fonctionnement de la méthode : modèle de l'utilisateur, modèle de la tâche, spécifications externes de l'interface, définition de scénarios, etc. Considérant le coût d'accès aux méthodes, toutes les approches heuristiques nécessitent une formation en ergonomie cognitive. Si l'évaluateur dont on dispose n'a pas cette compétence, il est sans doute préférable pour la fiabilité des résultats de s'orienter vers un modèle prédictif théorique par exemple le modèle Keystroke [Card et al., 83] (sous réserve que ce niveau d'abstraction soit suffisant pour l'étude) ; ou encore appliquer les recommandations de Nielsen qui permettent de pratiquer une "ingénierie de l'utilisabilité à faible coût" [Nielsen, 89] ou s'inspirer des questionnaires structurant des "cognitive walkthrough" [Lewis et al., 90]. L'intégration du savoir heuristique dans les générateurs d'interface limiterait les effets du syndrome de l'ABS.

Les résultats qu'une méthode d'évaluation est capable de fournir se caractérisent selon le contenu, la qualité, et le support de restitution. Le contenu s'étudie selon deux volets orthogonaux : le niveau d'abstraction des informations (niveaux tâche, syntaxique, lexical et physique) et le type d'information (par exemple, le nombre d'erreurs et degré de gravité, nombre de recommandations transgressées, temps d'exécution et d'apprentissage, degré de satisfaction de l'utilisateur, traces, etc.). Sous l'en-tête "qualité" nous regroupons des critères généraux qui dénotent la robustesse des résultats de la méthode d'évaluation : sensibilité de l'information aux hypothèses (en particulier GOMS est sensible à l'hypothèse du comportement expert), concision de l'information qui joue sur les temps d'analyse (comme contre-

exemple, citons le manque de concision des enregistrements audiovisuels), inspectabilité de l'information qui permet au spécialiste de parcourir les résultats et de comprendre les phénomènes.

Le support de restitution est aussi un ingrédient important dans la présentation des résultats. Nous distinguerons entre la présence et l'absence de support automatisé. Aujourd'hui, la plupart des outils informatisés assurent l'inspectabilité du recueil des données comportementales mais vont rarement au-delà de la détection automatique des problèmes. De plus, les problèmes détectés relèvent des niveaux d'abstraction les plus bas et s'appuient uniquement sur des informations quantitatives objectives. Le génie logiciel a là encore un rôle à jouer pour fournir les bons outils d'aide à l'analyse.

5. Conclusion

En ergonomie comme en génie logiciel, il existe des pratiques, des modèles, des méthodes et des notations qui correspondent à des besoins bien ciblés. Il convient donc de les respecter. Mais il convient aussi de dénoncer l'absence d'ouverture et d'identifier les points de rencontre. C'est l'objet de cet article. Une fois les points d'ancrage définis, il faut définir les moyens de communiquer. Les notations, qui servent de véhicule pour la pensée, sont toutes indiquées. Mais si l'on veut conserver les besoins d'expression et les pratiques de chaque discipline, ces notations doivent être compatibles, c'est-à-dire compréhensibles et transférables sans perte d'information : un cahier des charges intéressant pour une recherche pluridisciplinaire.

6. Références

Abowd G., Coutaz J., Nigay L., 1992, Structuring the space of interactive system properties. Proceedings of the IFIP TC2/WG2.7 Working Conference on Engineering for Human-Computer Interaction, Ellivuori, Finland, Elsevier Publisher, pp..

Balbo S., 1994, *Evaluation ergonomique des Interface Utilisateur : Un pas vers l'automatisation*, Thèse de doctorat de l'Université Joseph Fourier.

Barthet M.F., 1988, *Logiciels interactifs et ergonomie, modèles et méthodes de conception*, Editions Dunod Informatique.

Boehm B. W., may 1988, A spiral Model of Software Development and Enhancement, IEEE Computer.

Card S.K. , Moran T.P., Newell A., 1983. *The psychology of Human Computer Interaction*. Lawrence Erlbaum Associates.

Dowell J., Long J., 1989, Towards a Conception for an Engineering Discipline of Human Factors. *Ergonomics*, 32, 1513-1535.

Hammontree M.L., Hendrickson J.J., Hensley B.W., 1992, Integrated data capture and analysis tools for research and testing on graphical user interfaces. Proceedings of the CHI'92 Conference, ACM Press, p 431-432.

Lewis C., Polson P., Wharton C., Rieman J., 1990, Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. Proceedings CHI'90, ACM, p 235-242.

McCall J., 1977, *Factors in Software Quality*; General Electric Eds, 1977.

McDermid J., Ripkin K., 1994, *Life Cycle Support in the ADA environment*, Cambridge University Press.

Nigay L., Coutaz J., 1993, A design space for multimodal interfaces: concurrent processing and data fusion". *INTERCHI'93* Proceedings, Amsterdam, p 172-178.

Mosier J.N., Smith S.L., 1986, Application of guidelines for designing user interface software, Behaviour and Information Technology, Vol. 5, no. 1, p 39-46.

Nielsen J., 1989, Usability Engineering at a Discount. *Designing and Using Human Computer Interfaces and Knowledge-Based Systems*, Salvendy & Smith Eds., Elsevier North Holland, p 394-401.

Salber D., Coutaz J., 1993, Applying the Wizard of Oz Technique to the Study of Multimodal Systems. East/West Human Computer Interaction, Moscow. L. Bass, J. Gornostaaev, C. Unger Eds. Springer Verlag Publ., Lecture notes in Computer Science, Vol. 753, 1993, p 219-230.

Scapin D., 1990, Des critères ergonomiques pour l'évaluation et la conception d'interfaces utilisateurs. Acte du XXVI Congrès de la SELF, Montréal.

Whitefield A., Wilson F., Dowell J., 1991, A framework for human factors evaluation".*Behaviour and information technology*, vol. 10, no. 1, p 65-79.