Techniques d'Évaluation Ergonomique

- Who are you ?
- I am Oz, the Great and Terrible,
said the little man, in a trembling voice.

Lyman Frank Baum
"The Wonderful Wizard of Oz"

Techniques d'Évaluation Ergonomique

5.1. Introduction

Dans le cadre des systèmes interactifs mono-utilisateurs, l'intérêt de l'évaluation ergonomique des interfaces homme-machine s'affirme progressivement. En ce qui concerne les systèmes multi-utilisateurs et bien que [Grudin 1989] ait montré l'importance cruciale de l'évaluation dans la conception des systèmes multi-utilisateurs, peu de méthodes et d'outils sont disponibles. En revanche, on trouve dans la littérature de nombreuses études de cas de conception de systèmes multi-utilisateurs qui relatent souvent une phase d'évaluation. La diversité des types de systèmes multi-utilisateurs, la complexité des interactions qu'ils permettent et les nouvelles approches de conception qu'ils exigent sont à la source de ce manque de méthodes d'évaluation généralisables.

Vis-à-vis de notre structuration en principes, propriétés et techniques présentée au chapitre 1, l'évaluation intervient sur l'axe remontant. Elle permet de s'assurer que le système spécifié ou construit vérifie les propriétés souhaitées. Nous examinons dans ce chapitre ce que les méthodes d'évaluation des systèmes mono-utilisateurs peuvent apporter à l'évaluation des systèmes multi-utilisateurs. Nous présentons d'abord l'ensemble de ces méthodes de façon structurée afin d'analyser leur pertinence pour les systèmes multi-utilisateurs. A partir de cette analyse, nous illustrons l'utilisation d'une méthode prédictive fondée sur User Action Notation (UAN), puis nous nous concentrons sur l'évaluation expérimentale. En nous appuyant sur l'observation du travail des ergonomes lors de tests d'utilisabilité, nous avons établi un cahier des charges pour un laboratoire d'utilisabilité informatisé. NEIMO (Nouvelle Evaluation des Interfaces par le Magicien d'Oz) est une plate-forme informatique d'observation des utilisateurs et de simulation par Magicien d'Oz qui répond à ce cahier des charges.

5.2. Définitions

Avant de caractériser les méthodes d'évaluation des systèmes mono-utilisateurs, nous rappelons les définitions des termes "évaluation ergonomique" et "utilisabilité". Nous discutons la validité de ces définitions pour les systèmes multi-utilisateurs.

5.2.1. Définitions usuelles

L'*évaluation ergonomique* consiste à estimer la conformité des performances effectives avec les performances désirées du système [Dowell 1989]. Le "système" recouvre ici le couple utilisateur-système informatique. Pour les systèmes multi-utilisateurs,

l'imprécision de cette définition est immédiate : faut-il considérer un utilisateur donné et le système informatique, ou bien l'ensemble des utilisateurs et le système informatique ? Cette question n'a pas de réponse claire dans la littérature et nous verrons que les deux approches sont justifiables. Pour contourner l'ambiguïté de cette définition, nous préférons considérer que l'évaluation ergonomique est concernée par l'étude de l'utilisabilité d'un système informatique.

Nous avons défini au chapitre 4 des propriétés d'utilisabilité d'un système qui affinent la notion d'utilisabilité. Cependant, ce point de vue inspiré de l'étude de la qualité du logiciel doit être confronté à l'approche ergonomique de l'utilisabilité.

L'*utilisabilité* telle que la définit la norme ISO 9241 est "l'efficacité et la satisfaction avec laquelle des utilisateurs définis peuvent réaliser des buts définis dans un environnement particulier"[1]. [Shackel 1991] est plus précis : il définit l'utilisabilité d'un système comme "la capacité, en termes de fonctionnement humain, d'être utilisé facilement et efficacement par une classe d'utilisateurs définie, recevant une formation et une aide définies, pour réaliser une classe de tâches définies, dans le cadre d'une classe définie de scénarios prenant en compte l'environnement"[2]. A partir de cette définition générale, les mêmes auteurs donnent une série de critères opérationnels quantifiables : efficacité, facilité d'apprentissage, flexibilité et attitude. Ce dernier critère est constitué de deux aspects : l'acceptabilité en termes de "coût humain" (fatigue, inconfort, frustration, et effort individuel) et la satisfaction de l'utilisateur. Notons que, contrairement aux trois autres, le critère "attitude" n'est que très partiellement couvert par nos propriétés du chapitre 4.

Ces notions sont replacées dans un cadre plus vaste par [Nielsen 1994] qui considère l'utilisabilité comme un sous-ensemble d'une notion plus large : l'acceptabilité globale du système. L'acceptabilité est définie comme "la question de savoir si le système satisfait les besoins et les demandes des utilisateurs et des autres personnes potentiellement concernées, telles que les clients des utilisateurs et les dirigeants. L'acceptabilité globale d'un système informatique est une combinaison de son acceptabilité sociale et de son acceptabilité pratique". Nielsen propose les critères suivants pour quantifier l'utilisabilité d'un système : facilité d'apprentissage, efficacité, facilité de mémorisation, faible taux d'erreur et possibilités de réparation des erreurs, satisfaction des utilisateurs ("le système

1 "The effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in particular environments." (Norme ISO 9241, Part 11 : Ergonomics requirements for office work with VDTs - Guidance on usability specification and measures).

2 "[the usability of a system or equipment is] the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and user support, to fulfil the specified range of tasks, within the specified range of environmental scenarios."

doit être "agréable" à utiliser, les utilisateurs l'apprécient"). Scapin retient des critères similaires [Scapin 1990].

Comme on le voit à travers ces définitions, l'utilisabilité est encore un concept mal cerné, même si ces définitions présentent des points communs. En fait, ces différences apparentes révèlent des différences de point de vue : du point de vue technique de la norme ISO jusqu'à une perspective sociale plus large proposée par Nielsen. De notre analyse, nous retenons les points suivants :

- l'utilisabilité se traduit par des métriques qui permettent de quantifier et évaluer l'utilisabilité d'un système. Schackel est le plus précis et identifie des paramètres mesurables, par exemple en termes de pourcentage de variation pour la flexibilité. Toutefois, du point de vue d'un concepteur de système, ces critères sont trop généraux. Les propriétés du chapitre 4, qui caractérisent directement les aspects du système pertinents pour l'étude de l'utilisabilité, sont applicables dès les étapes amont de la conception. Les critères exposés ci-dessus sont plutôt orientés vers l'évaluation expérimentale. L'évaluation prédictive ne peut pas être guidée efficacement par ces critères centrés sur l'expérimentation. Nous précisons les différences entre ces pratiques ergonomiques dans le paragraphe suivant.

- Les trois définitions de l'utilisabilité exposées ci-dessus prennent en compte la satisfaction de l'utilisateur. Ici encore, il s'agit d'un critère qui ne peut être évalué qu'expérimentalement. L'évaluation prédictive ne peut fournir d'indications sur la façon de satisfaire ce critère.

- Seul Nielsen donne de l'utilisabilité une vision large incluant les aspects sociaux. Même si elle semble ouvrir une boîte de Pandore en prenant en compte dans l'utilisabilité l'avis de toutes les personnes ayant rapport de près ou de loin avec le système informatique, cette définition est mieux adaptée aux systèmes multi-utilisateurs. En particulier, elle permet d'intégrer la dimension sociale qui est essentielle pour ce type de système.

- Les définitions ci-dessus mettent en évidence la nécessité de spécifier le plus précisément possible les utilisateurs du système et les tâches à réaliser. Cette précision semblera évidente pour les lecteurs familiers avec l'ergonomie. Rappelons toutefois qu'il s'agit d'un prérequis indispensable à toute évaluation ergonomique. Moins connu mais cité dans les trois définitions, l'environnement doit également être spécifié avec précision. Le contexte de travail impose en effet des contraintes qui auront une influence sur l'utilisabilité. Il est intéressant de

remarquer qu'avec l'informatique mobile, cette caractéristique devient plus difficile à déterminer avec précision et un logiciel prévu pour être utilisé dans un contexte de bureau peut aussi être utilisé dans un environnement beaucoup plus contraignant mais plus difficile à cerner. Un exemple révélateur en est donné par [Tognazzini 1991] qui étudie l'utilisation des ordinateurs portables dans les avions de ligne. Il préconise un ensemble de règles, en particulier sur la prévention des erreurs dans l'utilisation des menus et du trackball, en argumentant que l'espace réduit et les vibrations rendent les manipulations moins précises.

Comme nous l'avons vu au chapitre 4, l'utilisabilité peut aussi être affinée en un ensemble de propriétés. Ces propriétés peuvent servir de critères pour l'étude de l'utilisabilité. Nous remarquons que nos propriétés recouvrent les critères de Schackel ou de Nielsen, exception faite du critère de satisfaction de l'utilisateur. Comme nous allons le voir, le choix des critères est guidé par la démarche ergonomique, qui elle-même est influencée par un ensemble de facteurs caractérisant le système et le contexte de développement.

5.2.2. L'évaluation d'un système multi-utilisateur

Comme nous l'avons remarqué plus haut, un système multi-utilisateur peut être vu au moins de deux façons. Soit le système est vu comme l'ensemble de ses parties et l'on peut mener une étude de l'utilisabilité du système pour un utilisateur donné et les tâches que réalise cet utilisateur. Soit le système est considéré dans son ensemble et l'on mène une évaluation globale du système en considérant l'ensemble des utilisateurs et les tâches globales qu'ils réalisent. Ces deux approches ne sont sûrement pas exclusives et sont en fait deux points de vue complémentaires sur un même problème. Pascal écrivait justement : "Je tiens pour impossible de connaître les parties sans connaître le tout, non plus que de connaître le tout sans connaître particulièrement les parties."

Pour réconcilier ces deux points de vue dans le cadre de l'évaluation ergonomique, nous introduisons la distinction faite par [Senach 1990] entre utilité et utilisabilité. L'utilité caractérise l'adéquation fonctionnelle du système : permet-il à l'utilisateur d'atteindre ses objectifs de travail ? L'utilisabilité concerne l'adéquation de l'interface homme-machine : le logiciel est-il facile à apprendre et à opérer ? Ramenée aux systèmes multi-utilisateurs, cette distinction nous aide à fixer des objectifs pour l'évaluation. L'étude de l'utilité permet d'évaluer le système multi-utilisateur dans son ensemble ; l'étude de l'utilisabilité permet d'évaluer le système du point de vue d'un utilisateur donné. Nous définissons maintenant ce qu'est l'utilité d'un système multi-utilisateur.

Pour un système multi-utilisateur, l'utilité a deux facettes : d'abord, elle caractérise l'adéquation fonctionnelle du système pour chaque utilisateur. En ce sens, cette approche de l'utilité est proche de celle pratiquée avec les systèmes mono-utilisateurs. La différence réside dans les tâches à étudier : en effet, comme l'explicite le modèle du trèfle (présenté au chapitre 1), les systèmes multi-utilisateurs introduisent de nouvelles tâches ayant trait à la coordination et à la communication. On peut citer MERMAID, un système de conférence audio/vidéo [Watabe 1990] comme exemple d'étude de l'utilité d'un système multi-utilisateur. Les auteurs notent par exemple que l'utilisation de la voix seule rend la communication plus difficile lorsque plus de quatre utilisateurs qui se connaissent peu communiquent à l'aide du système. Les auteurs remarquent aussi que le télépointage est en général exclusivement utilisé par la personne qui a la parole. Cette étude exploite des données obtenues par observation de l'utilisation du système et présente des résultats qualitatifs. En l'absence de modèles théoriques de la communication et de la coordination et de règles permettant de les appliquer à l'étude des systèmes multi-utilisateurs, l'approche expérimentale est privilégiée.

La deuxième facette de l'utilité d'un système multi-utilisateur est son adéquation fonctionnelle à la tâche globale réalisée par l'ensemble des utilisateurs. Cette utilité "globale" ne peut être déduite de l'étude de l'utilité du système pour chacun des utilisateurs mais requiert de considérer le système et le groupe d'utilisateurs dans leur ensemble (le tout n'est pas simplement la somme des parties). [Brothers 1990] présente une méthode expérimentale pour évaluer l'utilité globale. Le système décrit, ICICLE, est un environnement multi-utilisateur d'inspection de code. Les auteurs suggèrent d'étudier l'utilité du système en comparant différentes inspections du même code, avec ou sans ICICLE. Ils proposent des métriques permettant une évaluation quantitative de l'utilité, telles le nombre de *bugs* découverts ou des métriques mesurant la qualité du logiciel produit, ainsi que des mesures du temps requis par l'inspection et de la productivité.

[Boersma 1994] relate une évaluation d'un système de workflow pour l'accord de prêts bancaires. Ici encore, l'approche est expérimentale et repose sur des métriques comme le temps de traitement des dossiers de prêts, le nombre d'erreurs dans les décisions d'accord des prêts, et le gain de productivité.

Ces deux exemples choisis pour leur représentativité mettent en évidence l'utilisation de deux classes de métriques. Les métriques de la première classe expriment la qualité du travail produit : nombre d'erreurs, qualité du résultat, par exemple. La deuxième classe recouvre des mesures de temps d'exécution des tâches et de productivité. On retrouve ici la traditionnelle distinction entre bénéfice (représenté par la première classe) et coût (deuxième classe). Le système doit permettre d'augmenter la qualité du résultat de la tâche

tout en réduisant le coût nécessaire à son accomplissement. Cette notion est bien sûr également présente dans les systèmes mono-utilisateurs.

La distinction utilité/utilisabilité nous a permis de mieux cerner ce que représente l'évaluation pour les systèmes multi-utilisateurs. Pour examiner comment les techniques d'évaluation existantes peuvent être appliquées à ces systèmes, il nous faut caractériser ces techniques et réfléchir à leurs intégration dans le processus de développement d'un système. Pour tenir compte des pratiques actuelles de conception des systèmes multi-utilisateurs et des pratiques d'évaluation ergonomique, et pour replacer notre travail dans le cadre de ces pratiques, nous pensons utile de présenter les conceptions de l'interaction homme-machine de Long et Dowell.

5.3. Les démarches en IHM et en ergonomie

Long et Dowell distinguent trois types de démarches en interface homme-machine (IHM) : artisanale, sciences appliquées et ingénierie [Long 1989]. Nous présentons pour chacune de ces trois démarches la façon dont elles s'appliquent à l'ergonomie.

La démarche artisanale repose sur un savoir empirique, acquis par l'expérience. Ce savoir est informel et souvent ambigu, donc difficile à transférer parce que sujet à interprétation. Les règles ergonomiques élaborées expérimentalement, telles les recommandations de Smith et Mozier [Smith 1986], sont un exemple typique de connaissance utilisée dans la démarche artisanale. La pratique de la démarche artisanale rappelle l'approche exploratoire qui repose sur le prototypage rapide : une maquette est développée sans spécifications préalables et est évaluée immédiatement. Ce cycle prototypage-évaluation est ensuite réitéré à partir des informations fournies par l'évaluation.

La démarche sciences appliquées s'appuie sur un savoir élaboré à partir d'une théorie scientifique. Mais comme pour la démarche artisanale, le savoir doit être interprété pour être appliqué. La technique du "cognitive walkthrough" est un représentant de cette démarche [Lewis 1991]. Dans la pratique, le cycle de base est constitué des trois phases : spécification partielle, mise en œuvre, et évaluation. Des retours arrière après évaluation permettent de compenser les incertitudes liées à l'ambiguïté du savoir et d'ajuster les spécifications.

La démarche ingénierie est une approche scientifique rigoureuse. Le savoir est formalisé donc non ambigu et transférable. Les propriétés et leur formalisation par [Dix 1993] relèvent de cette approche. En pratique, le cycle de développement comporte :

spécification complète, réalisation, évaluation. L'évaluation mène éventuellement à une modification des spécifications et à un nouveau cycle.

Ces différentes démarches sont appliquées par Long et Dowell aussi bien à la composante informatique qu'à la composante ergonomie de la discipline IHM. Nous pensons que cette vue de l'IHM doit être étendue car l'on retrouve aussi ces trois démarches dans la composante psychologie et dans les composantes sciences humaines. Nous pensons aussi que la vue monolithique de l'IHM que présentent Long et Dowell se trouve aujourd'hui contredite par la pratique. Long et Dowell voient l'ergonomie et l'informatique évoluant de concert, adoptant en même temps l'un des trois types de démarche. En fait, et c'est d'autant plus vrai lorsque l'IHM doit intégrer de nouvelles disciplines comme les sciences humaines, chacune des disciplines partie prenante de l'IHM évolue indépendamment et suit l'une des démarches identifiées par Long et Dowell.

Comme le remarque [Balbo 1994], l'approche ingénierie est une vue idéale vers laquelle devrait tendre l'ergonomie. La rigueur scientifique qui la caractérise garantit que la méthode est généralisable, réutilisable, transférable et testable. Mais les démarches utilisées en réalité sont plus proches de la démarche artisanale ou de la démarche sciences appliquées. Dans les domaines où la théorie scientifique n'est pas encore suffisamment mature, comme pour les systèmes multimodaux ou multi-utilisateurs, la démarche artisanale est largement privilégiée. Notre approche fondée sur l'utilisation des propriétés vise à développer les premières bases d'une ingénierie de la conception ergonomique des systèmes multi-utilisateurs. Mais étant donné l'état actuel encore limité des connaissances théoriques applicables aux systèmes multi-utilisateurs, une approche plus concrète est aussi nécessaire. Nous présentons plus loin dans ce chapitre l'utilisation des propriétés (approche ingénierie) et l'évaluation expérimentale (approche artisanale). Mais il nous faut d'abord situer les techniques d'évaluation existantes et les lier aux processus de développement couramment utilisés pour les systèmes multi-utilisateurs.

5.4. Taxinomies

Nous exposons dans ce paragraphe une taxinomie des techniques d'évaluation afin de situer les techniques que nous allons présenter par la suite. Nous proposons aussi une classification des processus de développement.

5.4.1. Taxinomie des techniques d'évaluation

Une taxinomie classique des méthodes d'évaluation distingue les méthodes prédictives des méthodes expérimentales. Une méthode prédictive ne nécessite pas la présence de

l'utilisateur alors qu'une approche expérimentale repose sur l'observation d'utilisateurs. Cette classification peut être précisée comme le montre la figure 5.1.

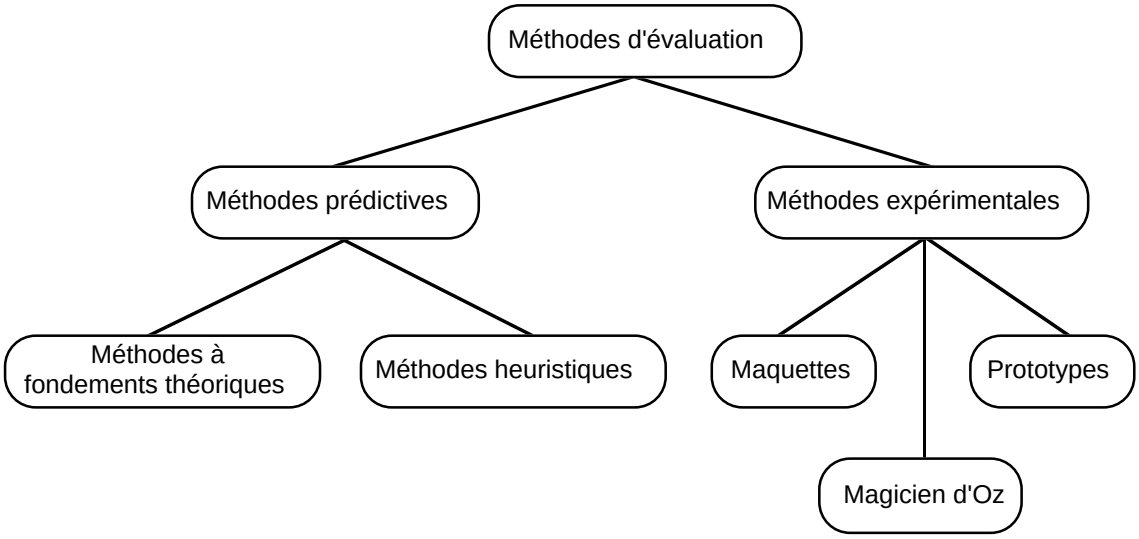

Figure 5.1. Classification des techniques d'évaluation.

L'évaluation expérimentale repose sur l'expérimentation avec un utilisateur final. Les tests de maquettes, prototypes ou de produits finaux, les tests en laboratoire d'utilisabilité, et les expérimentations Magicien d'Oz sont des exemples de techniques d'évaluation expérimentale. L'évaluation expérimentale ne nécessite pas un produit logiciel opérationnel. Les expérimentations à base de maquettes en papier et carton réalisées par IBM pour la messagerie des jeux olympiques de Los Angeles [Gould 1987] sont un exemple célèbre du fait que l'évaluation expérimentale de l'utilisabilité peut être réalisée à coût réduit et tôt dans le développement du système. L'évaluation expérimentale relève de la démarche ergonomique artisanale. Nous présentons plus loin dans ce chapitre l'outil NEIMO, plate-forme Magicien d'Oz et d'observation pour l'évaluation expérimentale.

L'évaluation prédictive est caractérisée par le fait qu'elle ne nécessite pas la présence de l'utilisateur final. Elle ne nécessite pas non plus une réalisation ni même un prototype du système. Elle peut donc intervenir très tôt dans le cycle de développement, dès la phase de spécification. L'évaluation prédictive relève de la démarche sciences appliquées (comme le "cognitive walkthrough" par exemple), de la démarche artisanale (l'utilisation de règles ergonomiques) ou de la démarche ingénierie. Nous illustrons cette dernière approche avec la notation UAN et nos propriétés d'utilisabilité (présentées au chapitre précédent). Un inconvénient des techniques prédictives est lié à la complétude de la théorie sous-jacente ou de l'ensemble des règles heuristiques. La théorie ou les règles sont souvent limitées aux interfaces graphiques traditionnelles. En l'absence d'une extension des supports théoriques, la technique d'évaluation n'est pas applicable à des systèmes plus récents tels

les interfaces multimodales ou les systèmes multi-utilisateurs. Il faut alors recourir à une technique artisanale. Gageons que ces techniques seront étendues tôt ou tard, mais il faut avoir conscience de leurs limitations.

Les techniques d'évaluation ne peuvent pas être envisagées indépendamment du processus de développement. Outre le fait qu'une technique d'évaluation requiert un certain avancement du développement pour être appliquée, et donc présuppose la disponibilité de documents ou d'un artefact pour servir de base à l'évaluation, toutes les techniques d'évaluation ne sont pas adaptées à tous les processus de développement. Nous caractérisons maintenant les processus de développement communément utilisés pour la conception et la réalisation des systèmes multi-utilisateurs.

5.4.2. Classification des processus de développement

Nous avons identifié trois grandes catégories de processus de développement utilisés dans la conception et la réalisation des systèmes multi-utilisateurs. Ces trois catégories sont le cycle génie logiciel, l'approche exploratoire et la conception participative. Nous définissons chacune de ces catégories et proposons une classification de leur usage.

5.4.2.1. Approche génie logiciel

L'approche génie logiciel est maintenant couramment utilisée en informatique. Les modèles sur lesquels elle repose sont typiquement le cycle de vie en cascade, ou les cycles en V ou en spirale. Une telle approche vise à organiser les activités de conception, de réalisation et de test du système de façon structurée. Chacune des étapes est identifiée et donne lieu à la production de documents. Les retours arrière sont possibles tout au long du cycle de vie. Aujourd'hui, les cycles de vie couramment utilisés ne mentionnent pas explicitement l'intégration des tests ergonomiques. Dans la pratique actuelle, ceux-ci ont souvent tendance à être repoussés en fin de cycle, regroupés avec les tests globaux du système. Or, les problèmes mis en évidence par l'évaluation ergonomique requièrent souvent une remise en cause des spécifications du système. Un retour arrière important (des tests du système jusqu'aux spécifications) est extrêmement coûteux. Une étude de Hewlett-Packard évalue qu'un défaut détecté lors des tests du système est cent fois plus coûteux à corriger qu'un défaut détecté lors de la conception (document cité dans [Balbo 1994]). Il est donc indispensable d'intégrer l'évaluation ergonomique le plus tôt possible dans le cycle de développement. N'oublions pas que si beaucoup de grandes entreprises de logiciel ont intégré les pratiques ergonomiques à leur cycle de vie, l'évaluation ergonomique reste encore "la cerise sur le gâteau" dans beaucoup de contextes industriels. Une récente étude révèle que seulement 52% des industriels européens mesurent

l'importance[1] de l'évaluation ergonomique ! Dans la classification de Long et Dowell, le cycle de vie génie logiciel relève de l'approche ingénierie, au moins en ce qui concerne la partie informatique. La structuration rigoureuse du cycle de vie permet aussi d'intégrer les pratiques ergonomiques comme le montre [Balbo 1994].

5.4.2.2. Approche exploratoire

L'approche exploratoire relève suivant les cas de la démarche artisanale ou de la démarche science appliquée. Elle repose sur l'utilisation intensive du prototypage et se caractérise par une suite d'itérations (spécification-)prototypage-évaluation. La granularité des itérations peut varier : le cycle de vie en spirale identifie plusieurs phases de prototypage, mais on observe souvent que le prototypage est utilisé pour tester un ou plusieurs aspects précis du système. On peut ainsi distinguer un "prototypage global" et un "prototypage local", plus concentré sur une caractéristique précise du système (par exemple un détail de l'interface utilisateur). La caractéristique distinctive de l'approche exploratoire est une plus grande implication des utilisateurs : le prototype est destiné à être évalué lors de tests d'utilisabilité et les résultats des tests informent l'itération suivante.

Même si un modèle comme le cycle de vie en spirale propose une structure pour l'approche exploratoire, celle-ci est moins fortement structurée qu'un processus de développement où l'objectif final est bien identifié. Les itérations ne sont pas connues à l'avance et même si l'on peut vouloir limiter leur nombre, leur résultat peut mener à repousser cette limite. En fait, et comme le fait explicitement ressortir le modèle en spirale, chaque itération comporte une phase d'analyse des risques. L'importance de cette phase ne doit pas être négligée : elle nécessite une analyse exhaustive du problème qui fait l'objet du prototypage et constitue à notre sens le "défaut dans la cuirasse" de l'approche exploratoire. Une analyse des risques mal conduite ou incomplète peut entraîner une itération supplémentaire. Notons au bénéfice du modèle en spirale qu'il impose une phase de spécification explicite et qu'il exige de documenter les alternatives lors de la spécification. Cette approche augmente les chances de réaliser un prototype pertinent pour l'évaluation du problème considéré.

5.4.2.3. Conception participative

La conception participative est une approche assez récente qui se distingue par une implication des utilisateurs tout au long du processus de développement. On trouve des exemples d'étude de cas dans la littérature décrivant les mediaspaces, ainsi que dans les présentations de techniques de conception pour les systèmes multi-utilisateurs, notamment celles de l'École Scandinave. Ce processus de développement est

[1] "Mesurer l'importance" ne veut pas dire "appliquer" !

particulièrement adapté aux systèmes multi-utilisateurs. Il prend en compte des aspects sociaux et organisationnels qui ne sont pas abordés par les autres méthodes. Cependant, il relève d'une démarche artisanale. Les études de cas font apparaître un éventail de démarches ad hoc et, si elles permettent d'identifier les grands principes qui sous-tendent cette approche, la méthode n'est pas encore généralisable. Ses résultats sont toutefois très prometteurs et elle a l'avantage de promouvoir la collaboration de disciplines diverses indispensables à la conception des systèmes multi-utilisateurs. Cette multi-disciplinarité de la méthode est sans doute à l'origine de son manque de structuration. Chacune de ces disciplines a en effet ses propres technique, et l'ensemble souffre de l'absence d'un creuset intégrateur permettant de conjuguer leur apports de façon systématique. Des techniques de Design Rationale comme la notation QOC présentée au chapitre 1 peuvent ici apporter une aide.

Suite à cette analyse des processus de développement, nous proposons la classification de la figure 5.2. Les deux axes de cette classification indiquent la structuration du processus de développement et l'implication des utilisateurs dans la conception du système.

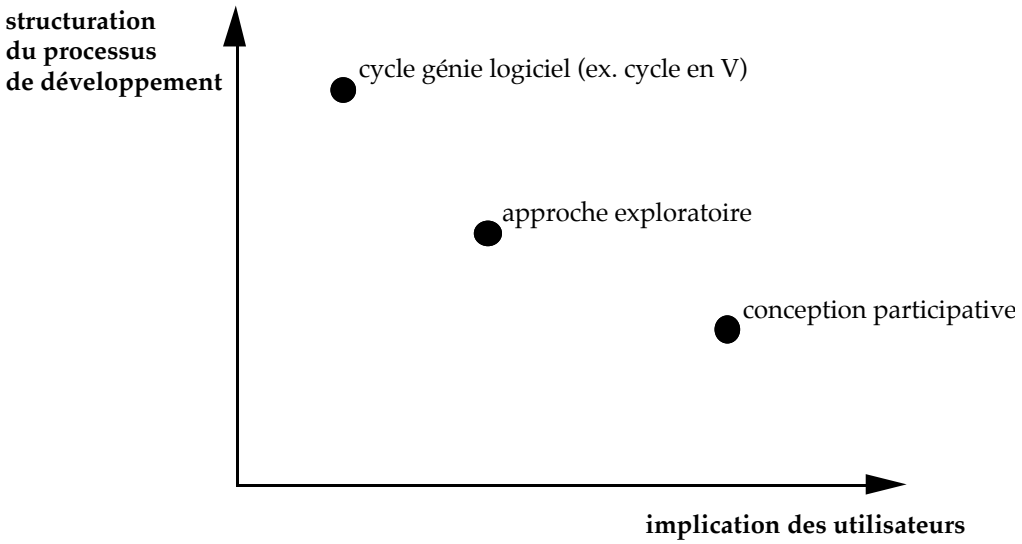

Figure 5.2. Classification des processus de développement reflétant les pratiques actuelles de conception et de réalisation des systèmes multi-utilisateurs.

Il est important de préciser que cette classification reflète les usages des méthodes présentées, et non leurs caractéristiques intrinsèques. Un cycle génie logiciel devrait impliquer davantage les utilisateurs dans la conception. Rien dans un cycle en V par exemple n'empêche de rajouter des phases d'évaluation ergonomique après chaque étape, dans l'esprit du modèle en étoile[1] de [Hix 1993]. De même, si la conception participative

[1] star life model.

apparaît aujourd'hui comme un processus de développement peu structuré, nous espérons qu'elle évoluera vers une plus grande structuration et permettra d'intégrer les apports des différentes disciplines qu'elle rassemble dans un cycle de vie inspiré de ceux du génie logiciel.

Notre classification des processus de développement fait ressortir que les catégories que nous avons identifiées diffèrent dans la façon dont les utilisateurs finaux sont présents lors du développement. La présence ou non de l'utilisateur final à un certain stade du développement conduira à opter pour une technique d'évaluation prédictive ou expérimentale.

De l'analyse des pratiques ergonomiques et des classifications que nous venons de présenter, nous tirons deux conclusions. Tout d'abord, il nous faut tendre vers la démarche ingénierie et les méthodes d'évaluation prédictive sont pour cela les plus adaptées. Cependant, les approches de conception des systèmes multi-utilisateurs relèvent plus de l'approche exploratoire ou participative que de l'approche génie logiciel. Cette contradiction nous a incité à explorer, en nous appuyant sur nos propriétés, une méthode prédictive flexible et peu coûteuse reposant sur la notation UAN.

5.5. Évaluation prédictive : l'exemple d'UAN

Nous avons dit que les propriétés que nous avons présentées au chapitre 4 peuvent être formalisées dans un but de vérification formelle. Compte tenu des pratiques actuelles de développement des systèmes multi-utilisateurs, la lourdeur d'une telle technique est irréaliste. Nous avons donc exploré le potentiel d'une notation semi-formelle plus flexible et plus simple d'emploi, User Action Notation (UAN). Nous constatons que si cette notation peut bien servir à la vérification de propriétés, son extension à la description des systèmes multi-utilisateurs pose des difficultés.

User Action Notation (UAN) [Hix 1993] est un système de notation semi-formel pour les systèmes interactifs développé originellement par Hartson, Siochi et Hix. UAN est orienté tâche et utilisateur : la notation permet de décrire les tâches, les actions de l'utilisateur et le comportement de l'interface homme-machine. A ce titre, c'est un outil précieux pour la représentation des tâches et pour les spécifications externes. UAN remplace avantageusement les descriptions informelles sous forme de texte et copies d'écran qui constituent aujourd'hui la plupart des spécifications externes des interfaces. Les descriptions textuelles sont en effet souvent ambiguës ou incomplètes, et par là même sujettes à interprétations multiples. Il faut toutefois noter que UAN n'est pas une description formelle comme peut l'être une description en langage Z. Cependant, dans les

cas où un vrai formalisme n'est pas nécessaire (donc hors systèmes critiques), UAN offre une notation plus lisible que la moyenne des formalismes. L'utilisation privilégiée d'UAN est la communication de spécifications, par exemple entre le concepteur et le réalisateur, utilisation qui a d'ailleurs motivé son développement.

Nous présentons d'abord dans ses grandes lignes la notation UAN, puis montrons de quelle façon elle peut être utilisée pour vérifier des propriétés d'utilisabilité. Nous envisageons son extension aux systèmes multi-utilisateurs et terminons par une évaluation de la notation.

5.5.1. Présentation de la notation UAN

UAN structure la description du comportement d'une interface en un ensemble de tableaux. Chaque tableau représente une tâche qu'il est possible d'effectuer avec le système. Ces tâches peuvent aussi bien être des tâches de haut niveau que des sous-tâches ou des tâches feuilles de l'arbre de tâches. Chaque tableau se compose de trois colonnes : les actions de l'utilisateur, la réponse de l'interface et l'état de l'interface. La notation propose un ensemble de symboles figurant des actions utilisateur ou des réponses de l'interface. Par exemple, enfoncer le bouton de la souris est représenté par le symbole **v**[1]. Le changement d'état d'une icône, par exemple son passage en vidéo inverse est représenté par **!**. La notation autorise le libre ajout de variables et de prédicats représentant soit des objets d'interaction soit des états internes de l'interface. La figure 5.3 montre une représentation typique en UAN.

Task: delete file
Precondition: visible(file_icon)

User Action	Interface Feedback	Interface State
~[file_icon]v	other_icon-! file_icon!	selected = file_icon
~[trash_icon]	trash_icon!	
^	trash_icon-! trash_icon!!	
	erase(file_icon)	selected = nil

Figure 5.3. Représentation UAN (simplifiée) de la destruction d'un fichier.

La figure 5.3 est une représentation UAN de la tâche de destruction d'un fichier avec le Finder du Macintosh. La représentation est légèrement simplifiée : le fait que le

[1] Nous prenons ici une légère liberté avec la notation. L'action d'enfoncer le bouton de la souris devrait en fait être notée **Mv**. Nous justifions cet ajustement de la notation au paragraphe 5.5.4.

"fantôme" de l'icône suit le mouvement lorsque l'on déplace l'icône a été omis. En haut à gauche du tableau, on indique la tâche décrite et d'éventuelles préconditions. Ici, la précondition pour que l'on puisse détruire un fichier est que l'icône de ce fichier doit être visible. Les lignes suivantes du tableau représentent la succession dans le temps des actions de l'utilisateur et les réponses de l'interface correspondantes. La première ligne indique que l'utilisateur déplace la souris près de l'icône de façon à pouvoir la saisir. [icon] représente la partie de l'élément d'interaction icon qui permet de le manipuler. Dans le cas d'une icône, il s'agit en général de toute sa surface ; pour une fenêtre, il s'agirait de sa barre de titre. L'utilisateur enfonce ensuite le bouton de la souris. La réponse de l'interface est constituée de deux opérations : toute autre icône sélectionnée auparavant (other_icon) repasse en affichage normal (**-!**) et l'icône du fichier est inversée. La variable d'état de l'interface selected prend la valeur de la nouvelle sélection. Les lignes suivantes décrivent successivement le déplacement de l'icône vers la corbeille et, après relâchement de la souris, le changement d'apparence de l'icône de la corbeille qui se gonfle (**!!**) et la disparition de l'icône du fichier déplacé.

UAN offre également un ensemble complet d'opérateurs permettant d'exprimer les relations entre tâches : séquencement, parallélisme, interruptions, entrelacement, tâches optionnelles, … ainsi qu'un ensemble exhaustif de symboles permettant de prendre en compte le temps et les relations temporelles entre tâches ou actions.

On constate que la notation UAN permet d'exprimer de façon très compacte un comportement d'interface qui est lourd et difficile à décrire en langue naturelle : il suffit de compter le nombre de lignes de texte qui sont nécessaires ci-dessus pour décrire la première ligne d'UAN ! Ce bénéfice jouerait à lui seul en faveur de l'usage d'UAN pour la spécification d'interfaces. Comme nous allons le voir, la notation permet aussi de détecter des problèmes d'utilisabilité.

5.5.2. Utilisation d'UAN pour la vérification de propriétés d'utilisabilité

UAN présente en regard les unes des autres les actions utilisateur et les réponses de l'interface. Cette caractéristique permet de vérifier la propriété d'observabilité présentée au chapitre 4. Nous allons montrer sur un exemple (MATIS) comment vérifier la propriété d'observabilité puis nous donnerons quelques règles qui permettent de vérifier d'autres propriétés à partir de l'examen d'une spécification UAN.

MATIS, qui fonctionne sur NeXT, est un système multimodal de recherche d'information sur les transports aériens [Nigay 1994]. L'application MATIS utilise un système de reconnaissance de la parole et pour pouvoir utiliser ce service doit être lancée depuis

l'environnement Office Manager (OM). Chaque application utilisant les services de reconnaissance de la parole d'OM fournit son propre dictionnaire permettant de guider la reconnaissance. La description UAN en figure 5.4 décrit comment l'utilisateur active une application dans l'environnement OM. Notons que l'analyse UAN de MATIS a été faite après développement. Il s'agit donc d'une évaluation "summative" au sens de [Hix 1993], mais la règle que nous donnons est générale et est aussi valide dans le cas d'une évaluation avant implémentation.

Task: SelOMAppli
Precondition: visible(Appli_Icon)

User Action	Interface Feedback	Interface State
~[Appli_Icon]v ^	Other_Icon! : Other_Icon-! Appli_Icon! *Feedback absent* ✗	SelDict = AppliDict

Figure 5.4. Détection d'un problème d'observabilité à partir de la spécification UAN.

On note dans la colonne Interface State qu'une variable SelDict est mise à jour après que l'utilisateur a cliqué sur une icône pour changer d'application. Ceci indique que le dictionnaire actif est maintenant celui correspondant à la nouvelle application sélectionnée. Or aucun feedback n'est fourni correspondant à la modification de cette variable. Cependant, cette variable est utilisée comme précondition dans la description d'une autre tâche sous la forme SelDict = MATISDict. Comme la modification de cette variable n'a jamais été répercutée dans la colonne Interface Feedback, l'utilisateur n'a aucun moyen de savoir si la précondition qui conditionne l'exécution d'une tâche est vraie ou fausse. La propriété d'observabilité du dictionnaire sélectionné n'est donc pas vérifiée.

A partir de cet exemple, nous pouvons généraliser notre constatation par la règle suivante :

❶ Si une variable d'état de l'interface est utilisée dans la précondition d'une tâche, toute modification de cette variable dans la colonne Interface State doit être répercutée dans la colonne Interface Feedback.

Notons que l'on peut être tenté d'aller plus loin que ce que propose ❶ et dire simplement que toute modification d'une variable d'état de l'interface doit être répercutée dans la colonne Interface Feedback. Mais ce serait en fait outrepasser la propriété d'observabilité. En effet, pour garantir que la variable est pertinente pour la tâche de l'utilisateur, comme

défini par la propriété d'observabilité, la variable doit apparaître comme précondition d'une tâche.

D'autres règles peuvent être établies pour vérifier des propriétés liées à l'observabilité. Nous donnons ici les règles permettant de vérifier les propriétés de non-préemption, dialogue à fils multiples, accessibilité, ainsi que les propriétés CARE.

Non-préemption :

❷ La non-préemption est assurée si toute opération apparaissant dans la colonne Interface Feedback est une réponse à une opération figurant dans la colonne User Action. Dans le cas contraire, l'opération considérée constitue une préemption de la part du système.

Dialogue à fils multiples :

❸ Des opérateurs permettant d'exprimer le parallélisme (**||**) et l'entrelacement de tâches (**<->**) sont inclus dans UAN. Le dialogue à fils multiples est donc particulièrement facile à caractériser : il suffit que les opérateurs de parallélisme ou d'entrelacement soient présents dans la spécification. Remarquons que, conformément à l'énoncé de la propriété, le dialogue à fils multiples peut être exprimé dans UAN à différents niveaux d'abstraction avec les mêmes opérateurs : niveaux tâche, sous-tâche, et action physique.

Accessibilité :

❹ Pour vérifier que tous les états de l'interface sont accessibles, il faut d'abord s'assurer qu'il n'y a pas de tâches "orphelines" c'est-à-dire que l'arbre de tâches est un arbre correct. Puis il suffit de vérifier pour chaque tableau UAN que la précondition peut être vraie. Il faut donc vérifier que tous les constituants de toutes les préconditions peuvent être modifiés par l'utilisateur (ou par le système le cas échéant) de sorte que les préconditions puissent être vraies.

Propriétés CARE :

Dans le cadre particulier des interfaces multimodales, la notation UAN doit être étendue pour prendre en compte de nouvelles techniques d'interaction. [Nigay 1994] donne un exemple d'une telle extension pour la reconnaissance de la parole.

❺ La complémentarité, l'assignation, la redondance et l'équivalence des modalités peuvent être déterminées à partir de l'examen de la spécification UAN. L'équivalence est caractérisée par un opérateur de choix (**|**) entre deux suites d'actions physiques chacune exprimée dans une modalité différente.

L'assignation, en revanche, est détectée par l'absence d'un opérateur de choix pour réaliser une tâche.

5.5.3. Extension d'UAN aux systèmes multi-utilisateurs

Nous l'avons noté dans ce chapitre, l'évaluation des systèmes multi-utilisateurs requiert de considérer deux niveaux et donc deux groupes de tâches. Le concepteur peut utiliser UAN pour spécifier les tâches "globales" du système, ou considérer l'interface d'un utilisateur donné et spécifier le comportement de cette interface isolément. Dans un cas comme dans l'autre, UAN révèle des limitations.

Si l'on considère le système de façon globale, la structuration de l'espace des tâches imposée par UAN nécessite de décomposer les tâches globales. Nous avons souligné que cette approche est réductrice, à moins de prendre en compte les tâches de coordination et de communication. [Jambon 1994] va dans ce sens et propose une extension de la notation UAN pour décrire une tâche collaborative faisant intervenir deux participants. Ce travail propose de mettre en regard les actions de chaque participant et les réponses du système ; il introduit également une notation pour indiquer les informations partagées et aborde également les problèmes de coordination (principalement l'attente par un participant d'une réponse de son interlocuteur) et de communication (transmission d'une information d'un utilisateur à un autre). A partir de cette représentation, on peut par exemple garantir qu'une information a été correctement échangée entre les utilisateurs. Mais, comme le remarquent les auteurs, la présentation choisie ne permet de prendre en compte que deux participants ; un plus grand nombre de participants rendrait le tableau UAN illisible. Il est probable qu'un support informatique de la notation permettrait d'envisager d'autres alternatives de représentation. En ce qui concerne l'étude de l'utilité, une spécification UAN prenant en compte la tâche globale et les tâches de coordination et de communication permettrait par exemple d'étudier de façon systématique la coordination, en particulier pour les systèmes de workflow. La complexité de la coordination imposée par le système pourrait être évaluée et une telle étude pourrait aider à détecter des redondances ou des incohérences dans les tâches de coordination. Toutefois, en l'absence d'une représentation de la notation UAN adaptée à plusieurs utilisateurs, et en l'absence d'une notation permettant d'exprimer la coordination et la communication, cette étude ne peut être faite que dans des cas élémentaires.

Si l'on considère la tâche d'un utilisateur donné, les problèmes de représentation sont simplifiés : on se ramène au cas mono-utilisateur. Toutefois, il faut pouvoir exprimer que des événements extérieurs dûs aux autres utilisateurs surviennent. Par sa structure qui fait suivre les actions de l'utilisateur par les réponses du système, il est peu naturel

d'exprimer avec UAN des événements qui ne sont pas une réponse du système à une action de l'utilisateur. Mais la difficulté la plus sérieuse tient au fait que l'on perd la nature multi-utilisateur du système. La spécification du partage d'informations, de l'accès concurrent à des objets d'interaction, des tâches de communication ne sont pas prises en compte par cette approche. Or ce sont précisément ces aspects du comportement du système qui mériteraient de bénéficier d'une description précise comme celle permise par UAN. Pour un éditeur de dessin partagé par exemple, il est intéressant de spécifier précisément le comportement de l'interface lorsque plusieurs utilisateurs sélectionnent le même objet graphique. L'approche considérant un seul utilisateur ne le permet pas.

Dans le cas des systèmes multi-utilisateurs, et quelle que soit l'approche considérée, UAN dans sa forme actuelle se révèle inadapté. Pourtant, les résultats intéressants obtenus avec les systèmes mono-utilisateurs et le potentiel de la notation pour l'étude de l'utilisabilité sont prometteurs. L'adaptation d'UAN à plusieurs utilisateurs reste un problème ouvert, mais certainement une voie de recherche à poursuivre.

5.5.4. Évaluation d'UAN

UAN est d'abord un outil de notation intéressant grâce à sa lisibilité et sa facilité d'apprentissage (au moins de notre point de vue, mais sa diffusion relativement large nous laisse supposer que ce point de vue est partagé). Nous avons cependant jugé utile d'apporter deux modifications à la notation. La première est une simplification : pour exprimer l'appui sur le bouton de la souris, la notation emploie le double symbole **Mv**. Nous l'avons remplacé par **v**, pour alléger la notation puisqu'il n'y a pas ambiguïté dans notre cas (**M** et **v** ne sont jamais utilisés avec une autre signification). Nous avons introduit une deuxième modification moins mineure qui concerne les préconditions : dans UAN, une précondition peut être utilisée à n'importe quel moment dans la colonne User Action. Nous avons souvent trouvé utile de regrouper les préconditions intervenant dans une tâche donnée et de les placer au début du tableau. Cette façon de faire permet aussi de mettre en évidence les variables intervenant dans la précondition. Ainsi elles peuvent être repérées plus facilement pour vérifier la propriété d'observabilité. L'heuristique que nous proposons est donc d'essayer d'isoler les préconditions en début de tableau. D'après notre expérience, si ce n'est pas possible c'est souvent que la tâche en question n'est pas suffisamment décomposée et que l'on peut identifier des sous-tâches dont les préconditions pourront, elles, être isolées en début de leurs tableaux UAN.

D'autres critiques peuvent être faites à UAN. Son caractère semi-formel, qui lui donne l'avantage de la lisibilité, conduit dans certains cas à une sémantique imprécise et à des possibilités d'interprétation multiples. Par exemple la disposition en colonnes amène

parfois à s'interroger sur les relations temporelles entre actions et réponses situées sur une même ligne et dans plusieurs colonnes. UAN souffre aussi de l'absence de certaines facilités d'écriture : il n'y a pas par exemple de mécanismes d'encapsulation comme les macros ou les procédures.

Outre la difficulté d'adaptation d'UAN aux systèmes multi-utilisateurs, nous regrettons aussi que la notation ne permette pas d'exprimer le lien entre l'interface et le noyau fonctionnel. Cette lacune nous a par exemple empêchés de proposer une règle permettant de vérifier la propriété d'annulabilité. En effet, dans ce cas, nous devons pouvoir vérifier qu'après annulation, l'interface et le noyau fonctionnel sont revenus dans leur état antérieur à l'exécution de la commande annulée. Avec UAN, on ne peut vérifier ce fait que pour l'interface ce qui est d'un intérêt limité et ne permet certainement pas de vérifier l'annulabilité. L'ajout d'une quatrième colonne représentant l'état du noyau fonctionnel résoudrait ce problème, mais la lourdeur d'écriture qui en résulterait incite à se demander si les inconvénients ne primeraient pas sur les avantages.

Nous avons montré que l'utilisation d'une technique d'évaluation prédictive reposant sur les propriétés permet de mettre en évidence des défauts d'utilisabilité en raisonnant sur les spécifications. Toutefois, nous avons rencontré des difficultés à étendre cette technique aux systèmes multi-utilisateurs. Nous apportons ainsi confirmation du manque de maturité des techniques prédictives pour l'étude des systèmes multi-utilisateurs. Face à cet état de fait, il est raisonnable de se tourner vers les techniques expérimentales.

5.6. Évaluation expérimentale

L'évaluation expérimentale est le terrain privilégié des ergonomes. Le manque de maturité de beaucoup de techniques d'évaluation prédictive, le rôle d'entraînement de quelques grandes sociétés de logiciel qui se sont équipées de laboratoires d'utilisabilité et l'aspect pratique et tangible des tests d'utilisabilité sont probablement la cause de cet état de fait. On peut se féliciter de cette vogue de l'utilisabilité. Il faut cependant garder à l'esprit les points suivants :

- l'évaluation expérimentale nécessite un prototype ou une maquette du logiciel. Même dans le cas d'une évaluation d'une maquette carton ou papier, des spécifications précises sont nécessaires. Cela signifie que l'évaluation expérimentale intervient plus tard que l'évaluation prédictive dans le cycle de vie. Les problèmes détectés seront donc plus coûteux à corriger.

- L'évaluation expérimentale ne s'improvise pas. La compétence de plusieurs ergonomes est indispensable, à la fois pour la préparation, la conduite et l'exploitation des résultats des tests d'utilisabilité. [Valentin 1993] affirme que la participation de trois ergonomes est nécessaire pour détecter 90% des problèmes d'utilisabilité. Rappelons que l'évaluation expérimentale est une approche artisanale dans la classification de Long et Dowell. La compétence des ergonomes est largement empirique et est donc difficile à transférer à des non-experts.

- Le travail des ergonomes est difficile. Les sessions d'utilisabilité sont longues et nécessitent une attention soutenue. Le dépouillement des sessions d'expérimentation nécessite d'analyser une masse importante de données, sous forme de vidéo, questionnaires, notes d'expérimentation.

L'avantage certain de l'évaluation expérimentale est de fournir des données réelles, obtenues par observation de sujets représentatifs des utilisateurs effectifs. On peut toutefois s'interroger sur la validité de données recueillies dans le contexte d'un laboratoire par opposition à l'observation des utilisateurs dans leur contexte de travail habituel. D'autre part, l'évaluation expérimentale permet d'évaluer, grâce en particulier aux questionnaires, la satisfaction de l'utilisateur. Et nous avons vu que la satisfaction de l'utilisateur contribue à l'utilisabilité d'un système.

Pour les systèmes multi-utilisateurs et les outils de communication homme-homme médiatisée, l'absence de techniques prédictives opérationnelles et prenant en compte explicitement ces systèmes plaide comme nous l'avons vu en faveur de l'approche expérimentale. De plus, la composante humaine et sociale, particulièrement dans les outils de communication, ne peut se satisfaire d'une évaluation uniquement théorique. Cependant, l'impact que ces systèmes peuvent avoir sur l'organisation et le fonctionnement d'un groupe social, ou d'une organisation complexe comme une entreprise, ne peut être raisonnablement estimé lors de sessions d'expérimentation en laboratoire. Des évaluations "sur le terrain" sont alors nécessaires.

Dans cette section, nous présentons d'abord nos observations, "in vivo" ou à partir de discussions ou des comptes rendus, du travail des ergonomes dans quelques laboratoires d'utilisabilité de grandes entreprises informatiques (Hewlett-Packard, CCETT, Lotus, Claris, Apple). A partir de ces observations, nous proposons les éléments d'un cahier des charges pour un ensemble d'outils permettant d'aider les ergonomes dans leurs tâches. Ces éléments ont mené à la réalisation de l'environnement NEIMO présenté dans la section suivante.

5.6.1. L'activité des ergonomes dans l'évaluation expérimentale

Dans l'évaluation expérimentale d'un système interactif, l'activité des ergonomes peut être découpée en trois phases principales : la préparation de l'expérimentation, la conduite de l'expérimentation et l'analyse des résultats obtenus.

5.6.1.1. La préparation de l'expérimentation

Pour mener correctement une expérimentation, les ergonomes doivent connaître parfaitement le système à évaluer. Leur première tâche est donc de se familiariser avec le système dans son ensemble (y compris la documentation) et le domaine applicatif. Ils identifient les tâches caractéristiques qui peuvent être réalisées avec le système et découvrent des problèmes d'utilisabilité potentiels. Notons qu'en règle générale, les tâches caractéristiques ont été déterminées lors de l'analyse de tâche. Sauf dans des cas flagrants, ces problèmes potentiels devront être infirmés ou confirmés par l'expérimentation.

En règle générale, même les problèmes d'utilisabilité flagrants sont confirmés par l'expérimentation. En effet, voir un utilisateur buter sur une difficulté ou plusieurs utilisateurs répéter systématiquement la même erreur est souvent le meilleur moyen pour un ergonome de convaincre un concepteur ou un développeur du bien-fondé de son analyse. Nous avons remarqué à plusieurs reprises que les ergonomes jouent un rôle pédagogique certain auprès des concepteurs, mais qu'ils doivent être prêts à argumenter toutes leurs assertions. Face à un concepteur convaincu de ses choix de conception, le savoir empirique d'un ergonome est moins démonstratif que l'observation des utilisateurs !

Les ergonomes identifient aussi les utilisateurs représentatifs et déterminent des scénarios caractéristiques. Les utilisateurs sont souvent choisis pour leur connaissance du domaine applicatif et leur niveau d'expertise, suivant la population à laquelle le système est destiné. La mise au point des scénarios repose sur le savoir-faire des ergonomes. Un scénario doit être représentatif des tâches et doit être aussi neutre que possible vis-à-vis du problème d'utilisabilité que l'on souhaite étudier. Le scénario ne doit pas laisser deviner aux sujets le point qui intéresse les ergonomes. Le sujet, se sachant dans une situation de test—même si ce n'est pas ses capacités personnelles mais le système qui est testé—aura parfois tendance à vouloir deviner le but du test. Un ergonome de Lotus insiste ainsi sur la nécessaire cohérence des dessins dans la conception d'une maquette papier : "les dessins doivent être cohérents ; si vous avez par exemple un beau dessin en couleur et que le reste est brouillon, l'utilisateur pensera—ah c'est donc ça qu'ils veulent tester !".

Nous retrouverons cette importance de la cohérence dans la conception et la conduite des expérimentations Magicien d'Oz.

Enfin l'environnement représentatif est déterminé. Si la plupart des laboratoires d'utilisabilité mettent les sujets dans une classique situation d'environnement de bureau, certaines applications grand public requièrent un environnement plus personnel. Les tests des décodeurs de télévision VisioPass par le CCETT par exemple se déroulaient dans une salle recréant un salon d'appartement.

5.6.1.2. La conduite de l'expérimentation

Suivant la complexité du système, le nombre de scénarios élaborés, et aussi la facilité à trouver des sujets adéquats, entre trois et une dizaine de sessions d'expérimentation auront lieu. Souvent, la grande majorité des problèmes d'utilisabilité est détectée dès les deux ou trois premières sessions. La durée d'une session d'expérimentation peut varier, suivant le système et les scénarios, d'une demi-heure à une demi-journée. La technique de verbalisation (*thinking aloud*) est quasiment systématiquement utilisée. Les ergonomes préfèrent faire travailler les sujets par couple : ils sont ainsi conduits à parler entre eux, ce qui rend leurs actions plus compréhensibles aux ergonomes. Sans cet artifice, les ergonomes ont souvent du mal, même en observant avec attention l'écran et le sujet, à deviner les actions et les intentions des sujets. Notons que dans le cas multi-utilisateur et en particulier pour la communication homme-homme médiatisée, la technique du "thinking aloud" est compromise. La communication entre utilisateurs interfère avec le "thinking aloud". Dans le meilleur des cas, la surcharge cognitive du sujet est certaine. Ces constatations ont renforcé notre conviction de la nécessité d'une capture informatique des actions du sujet.

L'équipement et la disposition des laboratoires d'utilisabilité varient, mais on repère certaines constantes. Les ergonomes et les observateurs sont dissimulés derrière une glace sans tain et observent les sujets à travers la glace. L'utilisation de caméras est systématique, avec en moyenne trois caméras : une vue d'ensemble de la pièce, une vue du ou des sujets de face, et une vue de l'écran. Parfois, une caméra supplémentaire fixée au plafond donne une vue d'ensemble du bureau afin de surveiller l'usage de la documentation. Une régie vidéo permet de choisir la vue affichée dans la cabine des expérimentateurs et permet aussi de mélanger les vues des différentes caméras. La vue d'ensemble semble en général assez peu utilisée. La vue affichée dans la cabine est aussi enregistrée sur magnétoscope. La bande vidéo est ensuite archivée pour utilisation lors de l'analyse des résultats. Notons que des expérimentateurs entraînés (chez Hewlett-Packard) ont très peu recours à la bande vidéo. La majeure partie de leur analyse se fait "au vol" et à partir de leurs notes prises en cours de session. La bande vidéo est revue

lorsque les notes sont ambiguës ou imprécises. L'utilité de l'enregistrement dans ce cas semble plus être une sécurité, ou une preuve pour justifier un problème d'utilisabilité auprès d'un concepteur. Cet usage réduit de la vidéo (à la fois lors de la session où l'observation directe est privilégiée, et pour l'analyse) a motivé notre choix de ne pas incorporer immédiatement l'enregistrement vidéo dans NEIMO.

Suivant les laboratoires, le nombre de personnes assistant à une session d'expérimentation est très variable. Chez Hewlett-Packard par exemple, une expérimentation a typiquement lieu en présence de deux ergonomes et du rédacteur de la documentation du système. Ce dernier vérifie l'utilisabilité de la documentation papier et en examine la pertinence. Chez Lotus en revanche, la cabine d'observation est spécialement conçue pour accueillir une quinzaine d'invités : concepteurs, développeurs, autres observateurs, … Il leur est demandé de noter toutes les remarques qui leur viennent à l'esprit lors de la session. Les instructions pour les observateurs précisent explicitement que toutes ces notes sont anonymes. Cette importance accordée aux notes prises au vol dans toute session d'expérimentation nous a conduit a introduire des possibilités d'annotation dans l'environnement d'observation NEIMO.

Lors d'une session, les notes prises par les ergonomes ne sont pas uniquement des annotations. Les expérimentateurs remplissent également un formulaire au fur et à mesure de l'avancement de la session. Les données recueillies sont quantitatives (temps d'exécution de chaque tâche du scénario, nombre d'erreurs) et qualitatives (stratégie utilisée en cas de choix, hésitations, difficultés). Les ergonomes observent et notent également les écarts entre activité (tâche effective) et tâche (prévue). Pour les expérimentations concernant des systèmes multi-utilisateurs, les expérimentateurs rapportent compter le nombre de fois où les sujets disent : "qui a fait ça ?", "qui a dit ça ?", "où es-tu ?". Ces observations permettent de vérifier les propriétés de rétroaction de groupe et d'awareness que nous avons définies au chapitre précédent.

Un soin particulier est pris pour l'accueil et le confort des sujets. Les sujets sont explicitement prévenus de leur participation à un test et du but du test (évaluer la facilité d'utilisation d'un logiciel). Ils savent également qu'ils seront observés et filmés. Chez Lotus, on notifie aux sujets que les enregistrements pourront être utilisés, mais de façon anonyme. Pendant le déroulement des scénarios, les expérimentateurs s'interdisent toute communication avec le sujet. En particulier, ils doivent résister à la tentation naturelle de guider un sujet qui s'enferre dans une suite d'erreurs. Ce n'est qu'exceptionnellement et après hésitation que l'expérimentateur se résout à aider un sujet. Typiquement, le problème rencontré par le sujet a déjà été détecté lors d'une expérimentation précédente et laisser le sujet répéter les mêmes erreurs n'apporterait pas d'informations

supplémentaires. Si le sujet demande de l'aide à l'expérimentateur, celui-ci est tenu de fournir une réponse évasive qui ne puisse pas réellement renseigner le sujet. Une fois les scénarios prévus exécutés, les sujets sont interrogés par les expérimentateurs. Cette interview a plusieurs buts : dissiper d'éventuelles incertitudes des ergonomes sur l'interprétation de ce qu'ils ont observé, obtenir des informations qualitatives et des appréciations sur le système de la part des sujets, évaluer leur satisfaction.

5.6.1.3. L'analyse des résultats de l'expérimentation

Sauf pour les ergonomes experts de Hewlett-Packard auxquels nous avons fait allusion plus haut, la phase d'analyse des résultats de l'expérimentation a posteriori est celle qui demande le plus de travail. Les notes prises pendant les sessions, les vidéos et les questionnaires sont dépouillés. Ils servent à analyser les stratégies (en comparant l'activité des sujets à la tâche prévue) et à étudier les erreurs et problèmes rencontrés par les sujets. Les données quantitatives recueillies pendant les différentes sessions sont pondérées, les données qualitatives et les questionnaires sont rapprochés et résumés.

Ces tâches nécessitent la manipulation et l'analyse d'importants volumes de données : les vidéos en particulier nécessitent une visualisation longue et souvent fastidieuse pour traquer les moments "intéressants". Il faut localiser les événements significatifs, les caractériser et éventuellement les quantifier, puis en abstraire un défaut d'utilisabilité. Cette recherche doit être faite sur tous les enregistrements, et mène à établir des recoupements entre les différentes sessions. On devine ici, étant donné l'absence d'outils de recherche élaborés de séquences vidéo ou de systèmes d'indexation performants, les contraintes de manipulation et de recherche manuelle auxquelles sont soumis les ergonomes. Il existe toutefois des outils permettant l'indexation et la recherche d'index sur des enregistrements vidéo. Le système MUSiC [MacLeod 1993] en est un exemple caractéristique ; mais la nécessité d'indexer manuellement les séquences vidéo limite l'intérêt de tels outils.

Après analyse et synthèse des données recueillies, les ergonomes sont en mesure d'établir un rapport détaillé sur les problèmes d'utilisabilité rencontrés, et de suggérer des améliorations. Ces suggestions peuvent porter sur l'interface de dialogue, l'adéquation des fonctions aux tâches étudiées, la documentation, l'aide en ligne, la formation et l'organisation du travail.

5.6.2. Les outils d'aide à l'expérimentation

L'exposé de nos observations des activités des ergonomes montre leur diversité et leur complexité, comme le dépouillement des données recueillies. Bien que les ergonomes

côtoient des outils informatiques, leur travail n'a pas de support informatisé. On peut s'étonner par exemple que chez Lotus les notes sur Post-It soient l'outil principal utilisé lors d'un test d'utilisabilité. Même si les ergonomes utilisent des outils informatiques classiques (traitement de texte, outils d'analyse statistique), il n'existe pas d'outil dédié pour assister toutes les tâches des ergonomes. Cependant, on trouve des outils permettant d'aider certaines tâches, comme la capture des actions de l'utilisateur, ou l'annotation des vidéos. Nous dressons d'abord un bref état de l'art de ces outils, puis, constatant leurs insuffisances et leur manque d'intégration, nous proposons une approche intégrée du support informatisé pour l'évaluation expérimentale. Nous présentons aussi la technique expérimentale dite du Magicien d'Oz et ses liens avec les autres techniques expérimentales.

Des outils peuvent apporter une aide aux ergonomes pour faciliter les tâches d'observation, d'analyse en fournissant des données précises comme les outils de capture ou en facilitant la manipulation des enregistrements vidéo. Notons que la plupart de ces outils sont encore utilisés dans un contexte de recherche et que très peu sont disponibles hors de ce cadre.

Les outils de capture des actions de l'utilisateur sont potentiellement extrêmement utiles pour les ergonomes. Les systèmes existant capturent les événements système dans un fichier à des fins d'analyse ou pour rejouer la session. Le système de [Hammontree 1992] effectue une capture au niveau des éléments d'interaction (widgets) : il enregistre les événements clavier ou souris et les date. Mais cette limitation au niveau widget ne permet qu'une couverture partielle des domaines applicatifs : la manipulation directe du type MacDraw par exemple n'est pas prise en compte. Il est possible de filtrer et de n'enregistrer que certains types d'événements. Ce système est couplé à un enregistreur vidéo et les événements capturés sont synchronisés avec l'enregistrement vidéo. Le système permet aussi d'enregistrer des annotations verbales qui sont elles aussi synchronisées. Il existe d'autres systèmes effectuant des captures d'événements système mais celui de Hammontree est l'un des plus complets. Mais cette approche souffre de limitations.

La capture d'événements d'un bas niveau d'abstraction comme les clics souris est de peu d'utilité directe aux ergonomes. Il faut ensuite à partir de ces événements de bas niveau retrouver les commandes exécutées. Le lien avec l'enregistrement vidéo facilite cette opération. La capture informatique permet d'envisager un traitement automatique comme l'approche EMA proposée par [Balbo 1994]. Rejouer une session peut aussi soulever des difficultés : il faut envoyer les événements au système en respectant exactement les dates des événements. D'autre part, le contexte de départ doit être fixé précisément (position

des fenêtres à l'écran par exemple). Si l'on envoie l'événement "relâcher la souris" un tout petit peu trop tard, la position de la fenêtre que l'on était en train de déplacer n'est plus la même que dans la session originale, le clic suivant n'aura pas un contexte correct et risque de perdre sa signification. Ces problèmes de précision et de contexte typiques de la capture à bas niveau d'abstraction, sont suffisamment sérieux pour être un réel obstacle à l'utilisation de cette technique. Le filtrage proposé dans l'outil de Hammontree est intéressant : il permet de limiter la quantité d'informations capturées. Mais il a l'inconvénient d'être fixé de façon statique et ne peut être changé en cours de session. Or au cours d'une session, au fur et à mesure de l'exécution des différentes tâches par le sujet, les événements intéressants varient.

Les outils d'indexation vidéo comme l'outil d'Hammontree ou MUSiC facilitent la manipulation des enregistrements vidéo. Mais comme nous l'avons vu auparavant, l'indexation doit être faite manuellement, ce qui limite leur intérêt. Dans l'outil d'Hammontree toutefois, la synchronisation avec le fichier de capture et les annotations permet de retrouver rapidement les passages utiles au travail d'analyse.

Le manque d'outils réellement adaptés à l'expérimentation nous a conduit à développer le projet NEIMO, une plate-forme d'observation et de capture du comportement de l'utilisateur. NEIMO étant aussi une plate-forme pour des expérimentations Magicien d'Oz, nous présentons cette technique d'évaluation.

5.6.3. La technique du Magicien d'Oz

Une expérimentation Magicien d'Oz consiste à faire simuler par un "compère" humain les services non implémentés d'un logiciel, utilisé par un sujet qui ignore la présence du compère. En général le compère observe le sujet grâce à des moyens informatiques. Lorsque le sujet déclenche l'exécution d'une commande non implémentée, le résultat de la commande est généré par le compère et transmis au sujet. Lors de l'expérimentation, les échanges informatiques entre le sujet et le compère sont enregistrés pour analyse ultérieure. Jusqu'à présent, les systèmes Magicien d'Oz ont généralement été utilisés pour simuler la reconnaissance du langage naturel, parlé ou écrit, pour des services d'informations téléphonées [Richards 1984], l'interrogation de bases de données [Dahlbäck 1988] ou de systèmes experts [Diaper 1989]. Pour l'interrogation de services téléphoniques, le sujet appelle un service supposé automatique. Pour parfaire l'illusion, la voix du compère qui lui répond est déformée par un filtre électronique qui donne l'impression d'une voix artificielle. Hors des systèmes de reconnaissance de langage naturel, nous n'avons trouvé que deux systèmes visant l'étude d'autres types d'interfaces qui prennent en compte la manipulation directe. [Dahlbäck 1989] décrit un système

Magicien d'Oz développé pour l'étude d'applications graphiques. Mais les auteurs reconnaissent avoir rencontré des difficultés dans la mise en œuvre et l'utilisation du système. Le dispositif duplique la fenêtre de travail du sujet sur l'écran du compère et toutes les événements de bas niveau générés par les actions du sujet sont répercutés vers la machine du compère. Les problèmes de synchronisation évoqués plus haut pour rejouer des événements de bas niveau semblent ici avoir été la cause des problèmes rencontrés.

Le système Turvy [Maulsby 1993] est un agent "intelligent", en fait une simulation d'agent que l'on peut programmer par démonstration. Le système permet d'utiliser la parole et la manipulation directe pour programmer l'agent simulé par le compère. Ce système s'est révélé d'une grande aide pour le prototypage rapide de l'agent logiciel Turvy. Enfin [Mignot 1993] décrit l'utilisation d'un dispositif Magicien d'Oz pour l'étude de l'interaction multimodale. Le système est utilisé pour simuler entièrement un système de dessin utilisant la voix et la manipulation directe.

Cet état de l'art des systèmes Magicien d'Oz met en évidence l'utilisation privilégiée de ce type de systèmes : la simulation pour l'étude de nouvelles techniques d'interaction. Deux contextes sont cependant à distinguer : dans de nombreux cas on étudie une technique d'interaction en la simulant dans le but d'étudier la technique elle-même et son usage, et non une interface déterminée. En reconnaissance de la parole par exemple, cette technique permet de constituer des corpus et de rendre plus robuste un système de reconnaissance existant. Dans d'autres cas, c'est une interface précise qui est étudiée et, en l'absence de la technologie adéquate ou du composant logiciel nécessaire, certains services sont simulés. Cette dernière approche montre que la technique du Magicien d'Oz peut être un auxiliaire précieux pour le prototypage. Cependant les expériences Magicien d'Oz présentent des difficultés de mise en œuvre.

La simulation du langage naturel, donc d'une seule technique d'interaction se heurte à des difficultés lorsqu'un seul compère assure la simulation [Polity 1990]. En particulier, les temps de réponse deviennent vite rédhibitoires pour le sujet qui utilise les services simulés. En effet, les tâches demandées au compère, même si elles semblent simples, exigent un travail cognitif important. Les demandes de l'utilisateur sont difficilement prévisibles et surtout le compère doit se conformer à certaines règles pour parvenir à simuler de façon réaliste les réponses d'une machine : les réponses doivent être cohérentes au regard de leur contenu, leur style et leur régularité. Deux demandes identiques du sujet doivent obtenir la même réponse. Le temps de réponse doit être conforme aux attentes du sujet : si le compère est trop lent, le sujet risque d'éviter l'usage du service simulé. Pour garantir le respect de ces contraintes de cohérence, plusieurs

approches ont été essayées. Tout d'abord, les tâches du compère sont précisément définies ainsi que leurs limites d'action. Par exemple une requête non prévue doit provoquer un message "Je ne comprends pas". Les compères sont préparés à leurs tâches et suivent un entraînement intensif : on ne s'improvise pas compère ! Le compère peut aussi disposer d'un support informatique : dans [Mignot 1993], le compère peut déclencher l'émission de messages parlés préenregistrés ; dans [Dahlbäck 1989], le compère combine des éléments de réponse à l'aide de menus. Enfin, une seule expérimentation à notre connaissance utilise plus d'un compère afin de tenter d'alléger la tâche de simulation. Dans [Mignot 1993], deux compères coopèrent pour la simulation : l'un des compères a la charge de la communication avec le sujet, et l'autre compère génère effectivement le résultat des requêtes sous le contrôle du premier. Dans cette expérimentation, un troisième compère est parfois intervenu, chargé de vérifier la cohérence des réponses fournies au sujet et de coordonner l'action des deux autres. Nous avons poussé plus loin ce découpage des tâches dans NEIMO pour tenir compte des exigences de l'interaction multimodale.

5.7. Réalisation : NEIMO

NEIMO (Nouvelle Evaluation des Interfaces par Magicien d'Oz) est un environnement pour le test d'utilisabilité des logiciels interactifs. Il représente une première étape vers un laboratoire numérique d'utilisabilité. Il fournit des outils pour l'expérimentation et pour l'analyse des données recueillies. Nous exposons ces deux facettes de l'environnement NEIMO, puis nous présentons brièvement une des expérimentations que nous avons réalisées dans cet environnement. Cette expérimentation concerne un système de communication homme-homme médiatisée. Enfin, nous tirons des leçons de cette réalisation et proposons un agenda de recherches pour faire évoluer cet outil.

Le cœur de l'environnement NEIMO est une plate-forme Magicien d'Oz pour l'étude de l'interaction multimodale. Comme nous l'avons indiqué au paragraphe 5.6.3 un dispositif Magicien d'Oz demande aux compères un lourd travail de simulation. C'est pourquoi NEIMO a été conçu dès l'origine pour permettre à plusieurs compères de collaborer pour simuler les services manquants du logiciel testé. De plus, NEIMO est aussi une plate-forme d'observation. Des observateurs prennent part à la session et disposent d'outils d'annotation. La figure 5.5 montre une configuration typique de NEIMO.

Le sujet observé a à sa disposition un ensemble de modalités d'interaction. Ici, par exemple, la parole est utilisée et la reconnaissance de la parole est simulée par un compère. Deux observateurs participent aussi à la session. Le travail de simulation est coopératif : les compères se répartissent la tâche de simulation et peuvent, par exemple,

simuler chacun un service différent. Les observateurs et le sujet sont aussi des participants actifs avec des rôles particuliers. NEIMO est donc un système multi-utilisateur. Nous décrivons l'environnement dans cette perspective au paragraphe 5.7.4. Nous détaillons en particulier les services que nous avons développés pour tenir compte de l'interaction multi-utilisateur. Une description de l'architecture logicielle du système NEIMO est présentée au chapitre 7.

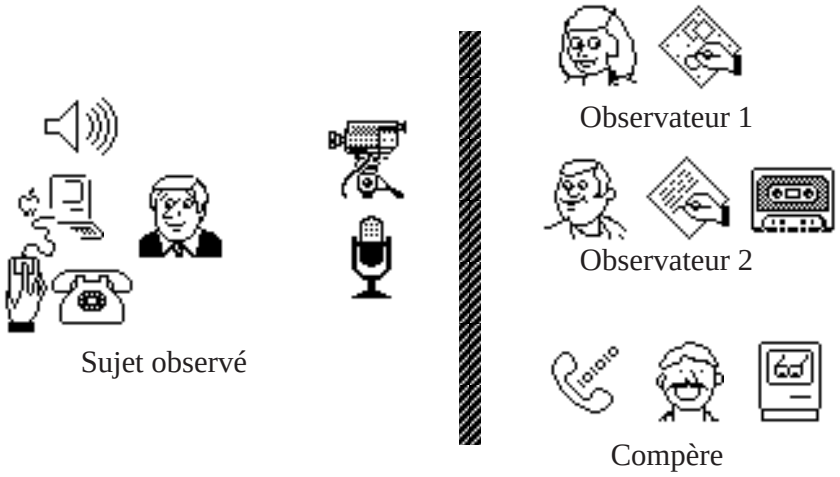

Figure 5.5. Exemple de configuration de NEIMO

5.7.1. NEIMO : outils pour l'expérimentation

Pour assister l'expérimentation, NEIMO offre une plate-forme Magicien d'Oz qui permet la simulation de services manquants dans le logiciel testé, l'observation et la capture informatique du comportement de l'utilisateur. NEIMO fournit aussi des outils aux ergonomes pour la conduite d'une session d'expérimentation.

5.7.1.1. Adaptation d'un logiciel à l'environnement de test

Pour pouvoir utiliser l'environnement de test d'utilisabilité, une application doit être adaptée à NEIMO. Plus précisément, elle doit être interfacée avec NEIMO via une interface de programmation définie. Cette interface permet au développeur de spécifier les actions de l'utilisateur qui doivent être transmises aux compères et/ou capturées pour analyse ultérieure. Par exemple, si le dispositif Magicien d'Oz est utilisé pour simuler la reconnaissance de la parole, le logiciel testé envoie à l'environnement NEIMO les phrases prononcées par le sujet sous forme de sons. Le compère reçoit ces sons et interprète les phrases du sujet et agit sur l'interface utilisée par le sujet pour réaliser les commandes demandées.

L'inconvénient de notre approche est qu'elle nécessite un développement spécifique. Nous avons toutefois essayé de minimiser le coût de ce développement. Une fois identifiées les informations pertinentes à transmettre aux compères ou à capturer, il faut faire un appel à la bibliothèque NEIMO pour chaque information différente. La bibliothèque est réduite pour minimiser son apprentissage (cinq à dix fonctions sont utiles au développeur). Comme le logiciel doit être adapté, il faut disposer de son code source pour pouvoir l'utiliser avec NEIMO. Cette limitation n'est pas gênante pour l'évaluation "formative" utilisée en cours de développement. Elle interdit cependant l'évaluation "summative", pour évaluer par exemple des logiciels commerciaux déjà développés. Nous expliquons au paragraphe 5.7.5 comment nous envisageons de nous affranchir de cette limitation.

Cette nécessaire adaptation a heureusement plus d'avantages que d'inconvénients : en intégrant explicitement dans le code source le lien avec l'environnement de test d'utilisabilité, elle donne plus de souplesse aux concepteurs. Par exemple, ils sont à même de choisir le niveau d'abstraction des actions de l'utilisateur qu'ils souhaitent capturer : du niveau événement jusqu'au au niveau commande. Un clic sur un bouton peut par exemple être enregistré comme { clic en (140, 203) }, comme { appui sur le bouton Raccrocher } ou comme { commande Raccrocher }. Sans un modèle de l'application, il n'est pas possible d'abstraire les commandes à partir des événements de bas niveau.

L'adaptation à NEIMO du logiciel à tester a deux autres avantages qui relèvent aussi d'une plus grande souplesse : la sélectivité et l'extensibilité. Contrairement à une solution automatique capturant les événements de bas niveau ou à une solution fondée sur un modèle de l'application, notre approche permet aux concepteurs de choisir les éléments pertinents à capturer. Ainsi, NEIMO permet une capture sélective et les expérimentateurs peuvent ainsi se concentrer sur un ensemble de problèmes d'utilisabilité précis. Nous avons vu que l'importance en volume des données recueillies lors d'une expérimentation est un des problèmes auxquels sont confrontés les ergonomes. Limiter le volume de données à la source pour éliminer le plus tôt possible le "bruit" de l'information intéressante nous semble une contribution importante de notre solution. Enfin, en laissant au développeur le choix des informations à capturer, cette solution garantit l'extensibilité du système. Ce point est particulièrement important pour le test des interfaces multimodales dans lesquelles les modalités utilisées sont susceptibles d'évoluer.

5.7.1.2. Conduite d'une expérimentation avec NEIMO

Lors d'une expérimentation avec l'environnement NEIMO, quatre types de participants interviennent : sujet, observateur, compère et super-compère.

- Un ou plusieurs *sujets* utilisent le logiciel à tester suivant un scénario mis au point par des ergonomes. Les actions du sujet sont capturées comme indiqué au paragraphe précédent. Même si jusqu'à présent nos expérimentations n'ont mis en jeu qu'un seul sujet, NEIMO permet la participation de plusieurs sujets.

- Des *observateurs* suivent la session. Ils disposent d'une station et d'une interface adaptée grâce à laquelle ils voient sur leur écran une copie de l'écran du sujet. Ils peuvent créer des annotations écrites ou verbales qui seront capturées. Ils surveillent aussi l'exécution du scénario, et pour chacune des tâches identifiées au préalable, ils enregistrent deux indications : leur opinion sur la bonne fin de la tâche, et la satisfaction de l'utilisateur quant à la réalisation de la tâche. Ces informations correspondent à celles inscrites par les ergonomes sur des formulaires dans une session d'utilisabilité. Les temps d'exécution des tâches et des scénarios sont calculés par le système d'après les indications de début et de fin données par les observateurs.

- Les *compères* ont la charge de simuler des services manquants du système, le cas échéant. Chaque compère a une tâche bien identifiée. Par exemple un compère dédié à cette tâche peut simuler un reconnaisseur de parole en écoutant les commandes vocales du sujet et en simulant, via des commandes adaptées, l'effet de ces commandes. Chaque compère dispose d'une station et d'une interface prévue pour sa tâche de simulation. Nous avons été particulièrement attentifs à cette interface. Comme nous l'avons dit plus haut, la tâche d'un compère est cognitivement exigeante. De plus, il doit satisfaire des contraintes de rapidité et de cohérence de ses réactions aux commandes émises par le sujet. L'interface inclut donc un ensemble d'éléments prêts à l'emploi que le compère peut réutiliser. L'objectif ici est double : minimiser les actions du compère dans un but de rapidité (optimisation au niveau "keystroke") et préparer à l'avance tout ce qui peut l'être en fonction du scénario dans un souci de cohérence. Dans le cas multi-utilisateur, un compère peut simuler un autre utilisateur ou un interlocuteur dans le cas de la communication homme-homme médiatisée. Nous en verrons un exemple avec l'expérimentation Supratel au paragraphe 5.7.4.

- Le *super-compère* est un compère qui n'intervient que lorsque plusieurs compères participent à la session et dont le rôle est particulier. Il a la tâche de superviser et d'orchestrer les actions des compères. Il s'occupe aussi de l'administration et de la configuration de la session. Un aspect important de sa tâche est de surveiller la cohérence des réponses des compères. Dans ce but, il n'a pas de support

informatique proprement dit et l'interface qui lui est dédiée est proche de celle d'un observateur.

Lors de la session, des données comportementales correspondant aux actions de l'utilisateur sont capturées à différents niveaux d'abstraction comme nous l'avons vu précédemment. Les annotations des observateurs, et éventuellement les actions des compères sont également enregistrées. Ce fichier historique sert ensuite de support à l'analyse de la session. L'environnement NEIMO comporte un outil d'analyse de la session a posteriori.

5.7.2. NEIMO : outils pour l'analyse

NEIMO offre des outils à l'ergonome pour l'analyse d'une session a posteriori. Ces outils permettent de rejouer la session en visualisant les actions du sujet et les simulations réalisées par les compères, de retrouver les annotations des observateurs et de les analyser en liaison avec un diagramme représentant les tâches. La figure 5.6 montre des copies d'écran de ces différents outils dans leur application à l'analyse d'une session Supratel [Lischetti 1994]. L'expérimentation Supratel est présentée au paragraphe suivant.

Les outils permettent de visualiser les différentes informations capturées durant la session :

- les actions du sujet et les interventions des compères sont visualisées sur une réplique de l'interface. L'interface utilise la métaphore du magnétoscope et permet de se déplacer dans l'enregistrement.

- Les scénarios sont présentés sous une forme synthétique. Pour chaque scénario, les annotations des observateurs sont visibles. Par exemple, la tête souriante et le pouce levé visibles en bas de la figure 5.6 indiquent que le scénario est considéré comme réussi à la fois du point de vue de l'observateur et du point de vue du sujet. Les annotations indiquent aussi les difficultés rencontrées par les utilisateurs et indiquent pour chacune son nom, le type d'erreur, la tâche concernée et la gravité de l'incident. Pour chaque scénario, les tâches réalisées sont présentées sous forme d'un diagramme de Gantt. Il permet de voir les instants de début et de fin de chaque tâche. Les annotations sont matérialisées sur l'échelle de temps par les icônes en forme de point d'exclamation.

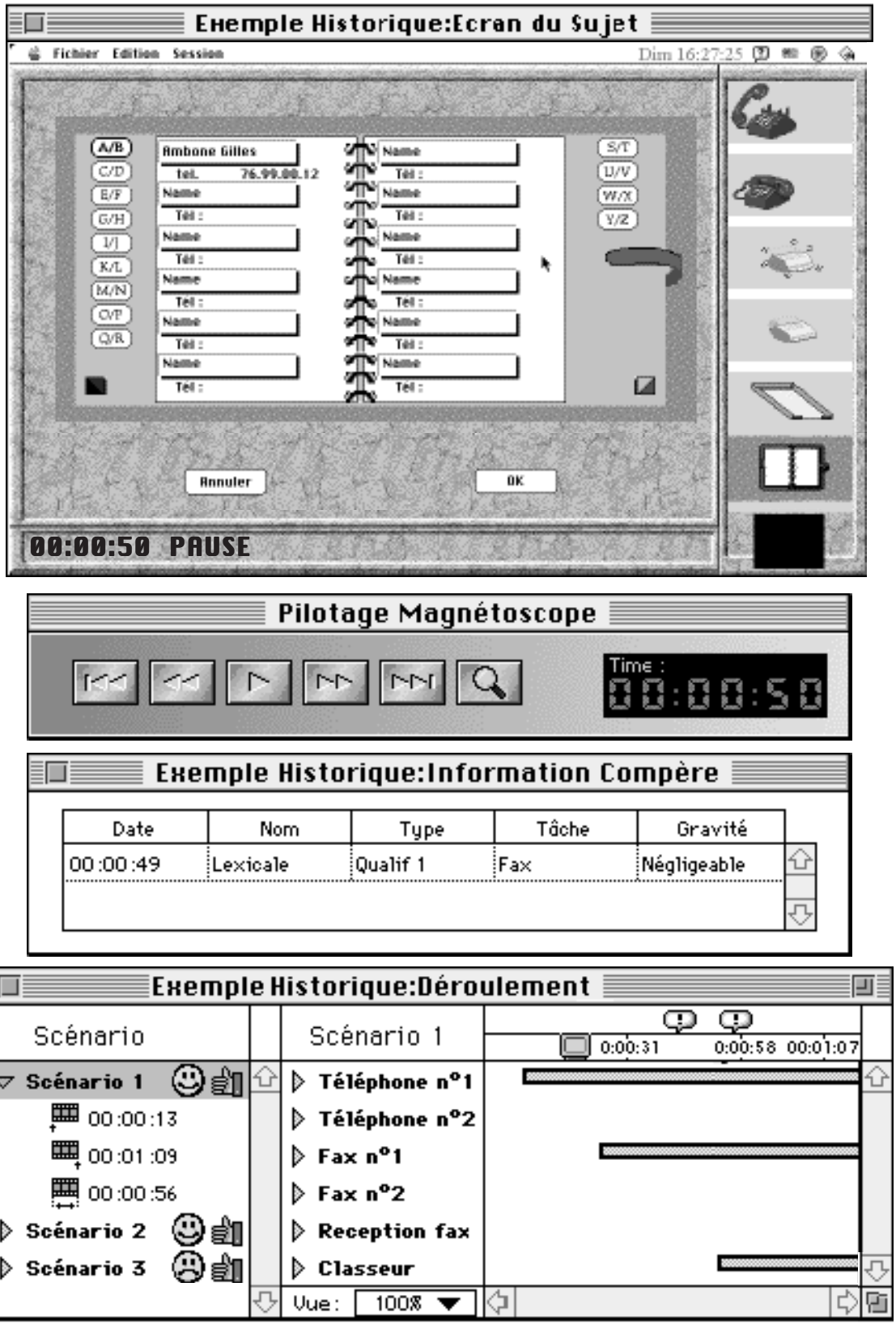

Figure 5.6. Copies d'écran des outils d'analyse de NEIMO. En haut, la visualisation des actions du sujet sur une réplique de l'interface étudiée. En dessous, l'interface de contrôle de la visualisation de la session et la liste des annotations des observateurs. En bas, un récapitulatif des scénarios et des tâches sous forme d'un diagramme de Gantt.

Avec ces outils, les ergonomes peuvent continuer leur analyse commencée pendant la session. Ils peuvent notamment compléter leurs annotations et revoir les passages qui ont posé des difficultés à l'utilisateur. Plusieurs efforts ont été faits pour permettre une analyse sélective des résultats. En effet, même si les mécanismes de NEIMO permettent déjà une capture sélective, la quantité de données capturées au cours d'une session peut être importante. Deux approches complémentaires sont utilisées :

- un mécanisme de vues qui permet de filtrer les informations de l'historique. Les vues agissent comme des filtres suivants des critères définis par l'ergonome : type des actions de l'utilisateur (ces types sont déterminés lors de la conception de l'application étudiée), fenêtre temporelle, utilisateur concerné dans le cas d'une application multi-utilisateur, tâches qui ont donné lieu à des annotations, etc. Différents filtres peuvent être combinés.

- Une présentation hiérarchique des scénarios et des tâches et une présentation synthétique des relations entre les tâches de l'utilisateur sous forme de diagramme de Gantt. Cette présentation permet de voir rapidement les relations temporelles entre tâches pour le cas de dialogue à fils multiples (par exemple entrelacement ou parallélisme).

Enfin, les outils d'analyse de NEIMO peuvent exporter des données numériques comme les dates de début et de fin de chaque tâche vers un tableur comme Excel. Il est ensuite possible d'effectuer des comparaisons entre plusieurs expérimentations avec la même interface.

A la différence des outils existants qui permettent de rejouer une bande vidéo et de l'annoter, cet outil d'analyse présente l'intérêt de visualiser une session d'expérimentation à plus haut niveau d'abstraction : en identifiant les scénarios et les tâches et en les liant aux annotations prises pendant la session, il fournit une aide supplémentaire au travail d'analyse. Nous verrons cependant au paragraphe 5.7.5 que ces outils ne représentent qu'un premier pas vers l'analyse automatique.

5.7.3. L'expérimentation Supratel

Supratel est un prototype de terminal de télécommunication multiservices développé par le CCETT [Charon 1993]. Son interface a été adaptée à l'environnement NEIMO pour l'étude de son utilisabilité. La figure 5.7 montre une copie d'écran de Supratel.

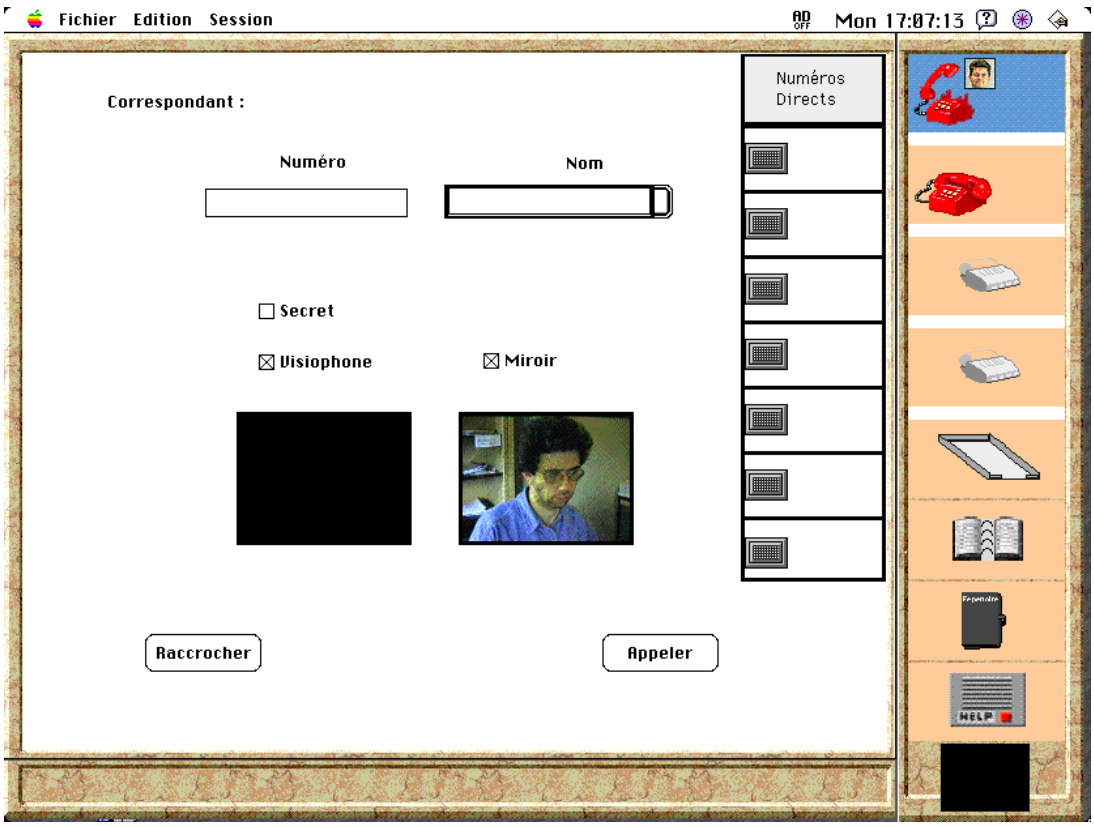

Figure 5.7. Une copie d'écran de l'interface du système Supratel. Le visiophone est en service.

Supratel offre un ensemble de services de communication homme-homme médiatisée et des services plus classiques mono-utilisateur. Ces différents services sont accessibles depuis la rangée d'icônes située à droite de l'écran sur la figure 5.7. On note par exemple des possibilités de communication téléphonique et visiophonique ainsi que fax (le système est destiné à être utilisé avec le réseau RNIS[1]), et des outils classiques comme l'agenda ou le répertoire.

Une des particularités de cette application est le grand éventail de tâches qui s'offrent à l'utilisateur et qui peuvent être utilisées simultanément. L'utilisateur est quasiment systématiquement dans un cas de dialogue à fils multiples. Pour l'expérimentation NEIMO, un ou deux observateurs suivent le comportement du sujet, et un compère simule les correspondants du sujet. Il peut par exemple entrer en communication visiophonique avec le sujet ou simuler la réception d'un fax sur la station du sujet. Les scénarios sont constitués d'un ensemble de tâches simples mais qui exploitent toutes les possibilités de l'application. Par exemple, le sujet doit vérifier sa disponibilité dans

[1] Réseau Numérique à Intégration de Services, traduction de *ISDN*. Réseau téléphonique numérique à débit moyen (64 Kb/s).

l'agenda, appeler un hôtel pour effectuer une réservation et envoyer simultanément un fax de confirmation. Pendant ces opérations, un appel extérieur survient et l'utilisateur doit vérifier un numéro dans son répertoire.

Lors d'une session d'expérimentation, le comportement de l'utilisateur et des compères est capturé à haut niveau d'abstraction, en général au niveau des commandes effectuées. L'analyse de la session permet d'étudier la satisfaction de certaines propriétés et la pertinence vis-à-vis des tâches des propriétés que vérifie l'interface. L'analyse nous permet d'étudier par exemple la vérification des propriétés CARE appliquées aux systèmes de communication, ou l'utilisation de la propriété vidéo miroir, vérifiée par Supratel. Notons que Supratel ne satisfait pas la propriété de réversibilité vidéo. Ce choix est délibéré : les tâches typiques réalisées avec Supratel dans les scénarios sont des tâches de communication "pures" ; le système ne permet pas de tâches de production multi-utilisateurs. L'observabilité publiée peut aussi être étudiée : Supratel comporte une fonction "secret" qui permet de suspendre momentanément la transmission de l'image de l'utilisateur. Enfin, la propriété de contrôle du rythme de l'interaction peut aussi être examinée : il est par exemple possible de mettre les appels en attente.

L'expérimentation Supratel a permis d'appliquer NEIMO à l'étude de la communication homme-homme médiatisée. Cette expérience a permis de vérifier des propriétés sur l'interface de Supratel et aussi la façon dont les sujets tirent parti de ces propriétés. Notons que certaines propriétés n'ont pu être satisfaites, principalement à cause de limitations techniques. A cause de l'environnement réseau utilisé pour les expériences, la régularité des flots par exemple n'est pas satisfaite.

5.7.4. NEIMO comme système multi-utilisateur

NEIMO constitue un système multi-utilisateur original : il comporte une variété de rôles et est à la fois synchrone et asynchrone. Nous avons vu qu'une session d'expérimentation implique plusieurs rôles : sujets, observateurs, compères et super-compère. Il faut y rajouter le rôle d'analyste rempli par l'ergonome qui effectue l'analyse a posteriori. Pendant une session, NEIMO est un système multi-utilisateur synchrone qui permet diverses tâches : la tâche globale est une tâche de production. Il s'agit de réaliser une session d'expérimentation dont le résultat est un fichier de capture. Mais NEIMO fait aussi intervenir des tâches de communication et de coordination entre le sujet et les compères et les observateurs. Si l'on considère l'aspect analyse a posteriori, NEIMO est aussi un système multi-utilisateur asynchrone : à partir du fichier élaboré pendant la session, l'ergonome analyste complète les observations et en tire une synthèse.

5.7.5. Leçons et perspectives

NEIMO nous a permis de réfléchir à la construction des systèmes multi-utilisateurs. Nous présentons au chapitre 7 l'architecture logicielle du système. Par sa complexité et la diversité des rôles qu'il prend en compte, il nous a aussi montré la difficulté de l'analyse des tâches multi-utilisateurs. La notation UAN présentée dans ce chapitre a été utilisée, par exemple pour la réalisation des interfaces du sujet et des observateurs et compère de Supratel. Toutefois, nous avons considérée dans cette analyse les rôles indépendamment les uns des autres. En effet, il n'y a collaboration qu'avec le sujet. Le système ne permet pas de collaboration entre les observateurs par exemple. Et même la collaboration avec le sujet est en général limitée à l'observation, sauf dans le cas du compère. L'absence d'outils d'analyse de tâches permettant de mieux prendre en compte l'interaction multi-utilisateurs a certainement été un frein au développement de possibilités de collaboration plus complètes dans NEIMO.

Deux aspects de NEIMO ouvrent des perspectives intéressantes : l'adaptation de l'interface à étudier à la plate-forme d'expérimentation, et l'analyse a posteriori des résultats d'une session.

Nous avons vu au paragraphe 5.7.1.1 que l'adaptation de l'interface à étudier à l'environnement NEIMO nécessitait un développement spécifique. Cette contrainte empêche notamment l'étude de logiciels dont les sources ne sont pas accessibles. Pour surmonter cette limitation, une voie de recherche à poursuivre est la définition d'une interface standard permettant la connexion d'un logiciel à un environnement d'étude de l'utilisabilité. Comme les circuits numériques qui disposent de points de test normalisés, on peut envisager qu'un logiciel comporte des points de test de l'utilisabilité. Un logiciel pourrait être ainsi connecté à n'importe quelle plate-forme d'étude de l'utilisabilité conforme à ce standard.

En ce qui concerne l'analyse, deux aspects méritent une étude approfondie : l'automatisation partielle ou totale et la visualisation. [Balbo 1994] distingue trois niveaux d'automatisation de l'analyse de l'activité d'évaluation : l'automatisation de la capture des actions de l'utilisateur, de la détection des problèmes d'utilisabilité, et de la correction ou de l'explication de ces problèmes. Dans ce cadre d'analyse, NEIMO offre la capture automatique des actions de l'utilisateur lors de la session d'expérimentation. Nous nous dirigeons vers l'analyse automatique avec l'utilisation de techniques comme la détection MRP (*Multiple Repeating Patterns*) de répétition de schémas d'actions dans le comportement de l'utilisateur. Mais comme le note Balbo, il n'existe pas à l'heure actuelle de solution satisfaisante pour la correction ou l'explication automatique. La technique

EMA proposée par Balbo est un premier pas dans cette direction. Guidée par l'utilisation des propriétés, elle pourrait ouvrir des perspectives prometteuses. Pour la visualisation synthétique de l'ensemble des tâches d'un scénario réalisé pendant une session, nous avons utilisés les diagrammes de Gantt. Cependant nous nous heurtons ici à un problème classique de la visualisation d'un grand espace d'informations : le compromis entre la vue globale contextuelle et la vue détaillée. Même si notre interface autorise la parcourabilité de l'ensemble du diagramme (avec un facteur de zoom, visible en bas de la figure 5.6), cette solution impose à l'analyste de naviguer entre vue globale et vue détaillée. L'intérêt de techniques de visualisation plus synthétiques, comme le "perspective wall" [Mackinlay 1991], bien adapté à la visualisation d'informations linéaires, reste à explorer dans le cadre de NEIMO.

5.8. Synthèse

En conclusion, force nous est de constater l'insuffisance des techniques d'évaluation ergonomique pour les systèmes multi-utilisateurs. Nous confirmons ainsi les conclusions du workshop "Design and Evaluation of Groupware" qui s'est tenu lors de la conférence CHI'95. En ce qui concerne les techniques d'évaluation prédictive, nous avons vu qu'en l'absence de théories prenant en compte les systèmes multi-utilisateurs, elles ne sont pas directement utilisables. Nous avons proposé d'utiliser la notation UAN, alliée à nos propriétés pour la détection de problèmes d'utilisabilité. Mais là encore, l'inadaptation de la notation au cas multi-utilisateur, en particulier pour la prise en compte des tâches de communication et de coordination, est un obstacle. Face à ce constat, nous nous sommes tournées vers des techniques relevant de l'approche artisanale, en particulier l'évaluation expérimentale. Notre observation de l'activité des ergonomes nous a conduit à la réalisation de l'outil NEIMO, un premier pas vers un laboratoire numérique d'utilisabilité. NEIMO est une plate-forme d'observation du comportement de l'utilisateur et de simulation par la technique du Magicien d'Oz. Cet environnement intègre aussi un outil d'analyse des résultats d'une session d'observation. Cette approche expérimentale, guidée par les propriétés du chapitre précédent, a été appliquée à l'étude d'un outil de communication homme-homme médiatisée. Toutefois, l'étude de l'utilisabilité d'un système multi-utilisateur plus complexe se heurte à des difficultés encore non résolues.

Références

[Balbo 1994] S. Balbo. *Evaluation ergonomique des interfaces utilisateur: un pas vers l'automatisation.* Thèse de doctorat, Université Joseph Fourier, Grenoble I, 1994.

[Boersma 1994] P. Boersma. *Experimental Research into Usability and Organisational Impact of Workflow Software.* Master's Thesis, Université de Twente, Pays-Bas, 1994.

[Brothers 1990] L. Brothers, V. Sembugamoorthy et M. Muller. *ICICLE: Groupware for Code Inspection*, CSCW'90, ACM Conference on Computer Supported Cooperative Work, Los Angeles, California, USA, 1990. pp. 169-181.

[Charon 1993] J.-P. Charon, L. Tézier, M. Carli et S. Liska. *Le projet Supratel : étude de l'utilisabilité d'un terminal de communication multiservices*, DESS Génie Informatique, Laboratoire de Génie Informatique, Université Joseph Fourier Grenoble 1, Rapport de stage, 1993.

[Dahlbäck 1988] N. Dahlbäck et A. Jönsson. *Talking to a computer is not like talking to your best friend*, SCAI-88, Scandinavian Conference on Artificial Intelligence, 1988. pp. 53-68.

[Dahlbäck 1989] N. Dahlbäck et A. Jönsson. *Empirical studies of discourse representations for natural language interfaces*, Fourth conference of the european chapter of the ACL, 1989. pp. 291-298.

[Diaper 1989] D. Diaper. *The Wizard's Apprentice: A Program to Help Analyse Natural Language Dialogues*, 5th Conference of the British Computer Society HCI SIG, 1989.

[Dix 1993] A. Dix, J. Finlay, G. Abowd et R. Beale. *Human-Computer Interaction*, Prentice Hall, New York, New York, 1993.

[Dowell 1989] J. Dowell et J. Long. *Towards a conception for an engineering discipline of human factors*, in *Ergonomics*, 32(11), pp. 1513-1535.

[Gould 1987] J. D. Gould, S. J. Boies, S. Levy, J. T. Richards et J. Schoonard. *The 1984 Olympic message system: a test of behavioral principles of system design*, in *Communications of the ACM*, 30(9),

[Grudin 1989] J. Grudin. *Why groupware applications fail: problems in design and evaluation*, in *Office: Technology and People*. Elsevier, 1989. pp. 245.

[Hammontree 1992] M. L. Hammontree, J. J. Hendrickson et B. W. Hensley. *Integrated data capture and analysis tools for research and testing on graphical user interfaces*, CHI'92, ACM Conference on Human Factors in Computing Systems, Monterey, California, USA, 1992. pp. 431-432.

[Hix 1993] D. Hix et H. R. Hartson. *Developing User Interfaces, Ensuring Usability Through Product & Process*, John Wiley & Sons, New York, New York, 1993.

[Jambon 1994] F. Jambon et L. Karsenty. *Formalisation des interfaces et travail coopératif : quelles conséquences ?*, IHM 94, Sixièmes Journées sur l'Ingénierie des Interfaces Homme-Machine, Lille, France, 1994. pp. 163-168.

[Lewis 1991] C. Lewis, P. Polson, C. Wharton et J. Rieman. *Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces*, CHI'91, ACM Conference on Human Factors in Computing Systems, New Orleans, Louisiana, USA, 1991. pp. 235-242.

[Lischetti 1994] N. Lischetti. *SupraAnalyse : un outil d'aide à l'analyse de l'utilisabilité. Application à un terminal de télécommunication multiservices*. Mémoire CNAM, Laboratoire de Génie Informatique, Université Joseph Fourier Grenoble 1, 1994.

[Long 1989] J. Long et J. Dowell. *Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering*, Fifth Conference of the British Computer Society HCI SIG, 1989.

[Mackinlay 1991] J. D. Mackinlay, G. G. Robertson et S. K. Card. *The Perspective Wall: Detail and Context Smoothly Integrated*, CHI'91, ACM Conference on Human Factors in Computing Systems, New Orleans, Louisiana, USA, 1991. pp. 173-179.

[MacLeod 1993] M. MacLeod. *DRUM, Diagnostic Recorder for Usability Measurement*, NPL, DITC HCI Group, Teddington, UK, 1993.

[Maulsby 1993] D. Maulsby, S. Greenberg et R. Mander. *Prototyping an Intelligent Agent through Wizard of Oz*, InterCHI'93, ACM/IFIP Conference on Human Factors in Computing Systems, Amsterdam, Pays-Bas, 1993. pp. 277-284.

[Mignot 1993] C. Mignot, C. Valot et N. Carbonell. *An Experimental Study of Future 'Natural' Multimodal Human-Computer Interaction*, InterCHI'93, ACM/IFIP Conference on Human Factors in Computing Systems, Amsterdam, Pays-Bas, 1993. pp. 67-68.

[Nielsen 1994] J. Nielsen. *Usability Engineering*, Academic Press, Boston, Massachussets, USA, 1994.

[Nigay 1994] L. Nigay. *Conception et réalisation des systèmes interactifs: Application aux Interfaces Multimodales.* Thèse de doctorat, Université Joseph Fourier, Grenoble I, 1994.

[Polity 1990] Y. Polity, J.-M. Francony, R. Palermiti, P. Falzon et S. Kazma. *Recueil de dialogues homme-machine en langue naturelle écrite*, Criss, Les cahiers du Criss n°17, 1990.

[Richards 1984] M. Richards et K. Underwood. *How Should People and Computers Speak to Each Other*, Interact'84, IFIP Conference on Human-Computer Interaction, 1984. pp. 268-273.

[Scapin 1990] D. L. Scapin. *Des critères ergonomiques pour l'évaluation et la conception d'interfaces utilisateur*, XXVIè Congrès de la SELF, Montréal, Canada, 1990.

[Senach 1990] B. Senach. *Evaluation ergonomique des interfaces homme-machine : une revue de la littérature*, INRIA, Programme 8, Communication Homme-Machine, n° 1180, 1990.

[Shackel 1991] B. Shackel et S. Richardson. *Human Factors for Informatics Usability*, Cambridge University Press, 1991.

[Smith 1986] S. L. Smith et J. N. Mosier. *A design evaluation checklist for user-system interface software*, The MITRE Corporation, Bedford, Massachussets, USA, #MTR-9480 EDS_TR_84-358, 1986.

[Tognazzini 1991] B. Tognazzini. *On High-Altitude Computing*, in *Apple Directions*, 1991.

[Valentin 1993] A. Valentin, G. Vallery et R. Lucongsang. *L'évaluation ergonomique des logiciels, une démarche itérative de conception*, ANACT, 1993.

[Watabe 1990] K. Watabe, S. Sakata, K. Maeno, H. Fukuoka et T. Ohmori. *Distributed Multiparty Desktop Conferencing System: MERMAID*, CSCW'90, ACM Conference on Computer Supported Cooperative Work, Los Angeles, California, USA, 1990. pp. 27-38.