Première Partie

Les apports de sciences non-informatiques

Le cadre conceptuel : Terminologie, Principes, Propriétés et Techniques

Everything has now become digitized.
We have created a unimedia, really.

Nicholas Negroponte
in SIGGRAPH'95 advance program

Le cadre conceptuel :
Terminologie, Principes, Propriétés et Techniques

1.1. Introduction

De machine expérimentale qu'il était il y a moins de cinquante ans, l'ordinateur a conquis aujourd'hui un nombre de domaines impressionnant et s'installe dans notre vie quotidienne. Ce mouvement semble s'accélérer constamment et, comme le dit Nicholas Negroponte, tout—au moins toute information—peut être codé sous forme numérique. En particulier, l'image et le son sont maintenant couramment numériques et bénéficient donc des deux possibilités les plus originales de l'information numérisée : la duplication exacte sans dégradation et la transmission par des électrons ou des ondes électromagnétiques. Ces qualités rendent possibles la transmission d'images et de sons par l'intermédiaire de réseaux informatiques, en plus de la transmission de données informatiques pour laquelle ils étaient utilisés jusqu'à présent.

Dans le cadre de notre travail, nous nous sommes intéressés aux systèmes permettant à plusieurs utilisateurs de travailler en commun et de communiquer par des moyens audio et vidéo. Dans ce chapitre, nous commençons par préciser la terminologie et cerner le sujet de notre étude : les systèmes multi-utilisateurs. Puis nous exposons la grille d'analyse qui nous a guidée pour appréhender la conception et la réalisation de ces systèmes, et qui structure aussi ce mémoire. Cette grille d'analyse a trois niveaux d'abstraction : principes, propriétés, techniques.

1.2. Définitions

De la façon la plus élémentaire, on peut définir un système collecticiel comme un système interactif pouvant être utilisé par plusieurs utilisateurs. Mais cette définition manque de précision : la plupart des systèmes informatiques présentent en effet cette caractéristique. Qu'il s'agisse d'un système tel qu'Unix, ou d'un micro-ordinateur relié à un réseau et permettant le partage de ressources (imprimante, par exemple), ces systèmes sont utilisables par plusieurs utilisateurs, simultanément ou successivement, et le système est conçu pour permettre cette utilisation. Pourtant, l'appellation "collecticiel" (ou "groupware") ne recouvre traditionnellement pas ce type de systèmes et suscite plutôt l'évocation de systèmes favorisant l'interaction entre les utilisateurs comme CoLab [Stefik 1988], un des précurseurs. Il nous faut donc définir de façon plus rigoureuse les systèmes qui sont l'objet de notre étude.

Le terme "collecticiel" est traditionnellement utilisé dans la communauté française pour désigner les systèmes interactifs qui permettent le travail coopératif assisté par ordinateur ("computer-supported cooperative work" ou CSCW). Notons que la désignation officielle

est “synergiciel”[1] mais ce terme n’a jamais été utilisé par la communauté française. Pour certains auteurs, les systèmes permettant principalement la communication, tels les mediaspaces dont nous reparlerons au chapitre suivant, ne sont pas couverts par l’appellation collecticiel. Afin d’éviter toute ambiguïté, nous utiliserons le terme plus général de “système multi-utilisateur”.

Pour définir l’espace des systèmes multi-utilisateurs, nous nous appuyons sur le modèle proposé récemment par Ellis et Weiner [Ellis 1994]. Ce modèle définit un système multi-utilisateur comme une combinaison de trois aspects fonctionnels :

- l’aspect *ontologique* : les objets manipulés et les opérations possibles sur ces objets,

- l’aspect *coordination* : l’organisation et les relations entre les activités des participants,

- l’aspect *interface* : la façon dont les participants interagissent avec le système et entre eux.

Avec son modèle ontologique et sa représentation explicite des services de coopération, cette décomposition a le mérite de dégager deux classes de concepts fonctionnels caractérisants des systèmes multi-utilisateurs. Toutefois dans ce modèle, l'interface utilisateur recouvre à la fois l'interaction utilisateur-système et la communication entre utilisateurs. Selon le précepte bien connu de la distinction entre fonctions pures et interface utilisateur, il convient de considérer l'interaction homme-machine comme une conséquence des services de fond dont il est la vitrine perceptible. En conséquence, nous proposons de reprendre les fondements de la décomposition tripartite d’Ellis, mais nous en éliminons l'interface utilisateur. Cette élimination ne signifie pas que l'interface est un composant de seconde classe mais que son rôle doit être distingué de celui du noyau fonctionnel.

Comme le montre la figure 1.1, nous considérons que les services d’un système multi-utilisateur couvrent trois espaces : la production, la coordination et la communication. Ce modèle constitue notre modèle conceptuel de base : le trèfle des systèmes multi-utilisateurs.

[1] Loi du 4 août 1994, Journal Officiel de la République Française.

- L'*espace de production* correspond au modèle ontologique d'Ellis : il désigne les objets qui résultent d'une activité de groupe, par exemple un livre, une œuvre de cinéma, un logiciel, etc. L'espace de production correspond au *modèle conceptuel* tel que nous l'entendons en conception des systèmes mono-utilisateurs. Pour les systèmes multi-utilisateurs, ce modèle décrit les concepts qui motivent l'action de groupe, qui dénotent l'œuvre tangible commune, mais aussi l'espace privé de chaque utilisateur comme dans un système mono-utilisateur.

- L'*espace de coordination* reprend la définition d'Ellis : il s'agit de définir les acteurs (et notamment les individus, les groupes, les rôles, voire des agents logiciels "intelligents"), d'identifier les activités et les tâches (et notamment leurs relations temporelles), de désigner enfin les acteurs responsables des tâches et des activités. Tandis que l'espace de production offre une vue statique du système, l'espace de coordination en définit la dynamique. Sur ce point, nous adhérons au point de vue d'Ellis.

- L'*espace de communication* offre aux acteurs du système multi-utilisateur la possibilité d'échanger de l'information. Le contenu sémantique de cette information concerne les acteurs communicants. Il est étranger au système qui se contente de servir de messager. Cet aspect, qui constitue l'essence de la *communication homme-homme médiatisée*, n'est pas explicite en tant que tel dans le modèle d'Ellis.

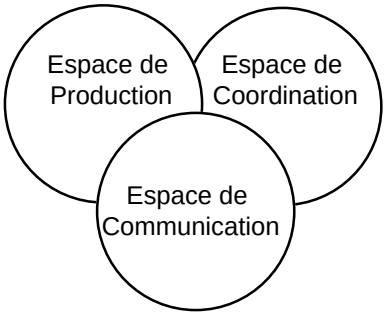

Figure 1.1. Le trèfle des systèmes multi-utilisateurs.

Selon les systèmes multi-utilisateurs, les trois espaces du trèfle fonctionnel n'ont pas la même importance. Par exemple, les systèmes d'édition partagée actuels mettent l'accent sur les services de production, le workflow sur la coordination, et les mediaspaces privilégient la communication. En poussant plus loin l'observation, l'importance relative de chacun de ces trois espaces peut varier au cours de l'interaction. Pour exprimer la variabilité de l'équilibre entre les trois classes de service, nous reprenons le principe du métamodèle Slinky qui, appliqué au modèle d'architecture Arch des systèmes interactifs

[Bass 1992], traduit la variation de l'importance des composants fonctionnels d'un système et leur perméabilité. Cette mouvance peut être statique (l'importance relative des trois composantes est décidée une fois pour toute à la conception) ou bien dynamique (elle varie alors en cours de session). La figure 1.2 montre les possibilités de variation des composants du modèle du trèfle.

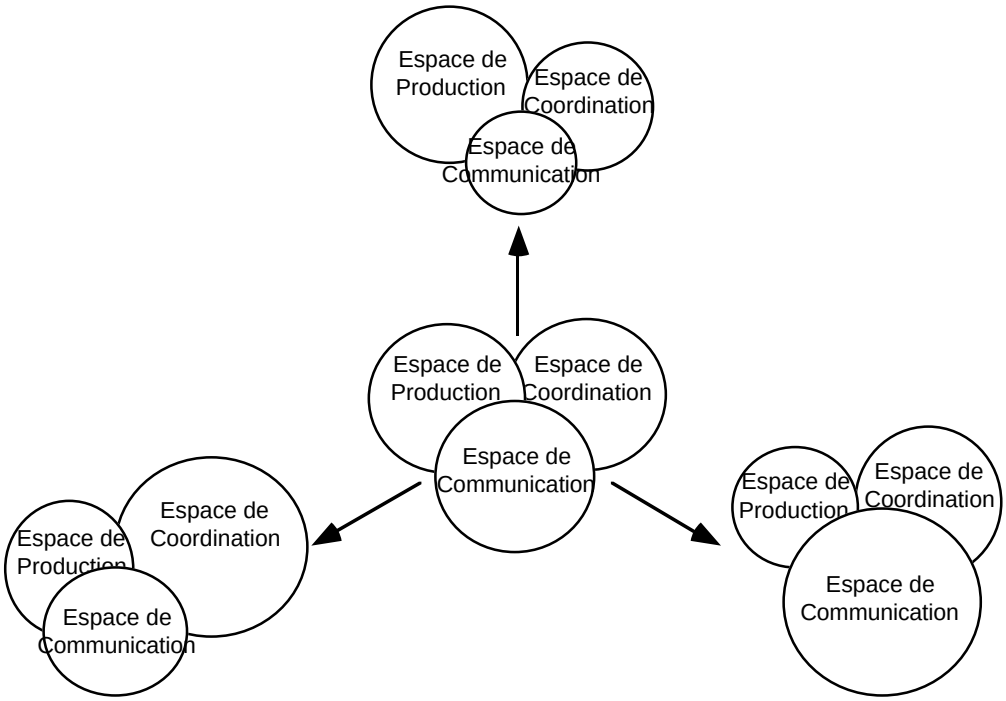

Figure 1.2. Le métamodèle Slinky appliqué au trèfle des systèmes multi-utilisateurs. L'importance de chaque facette peut varier.

L'intérêt du modèle du trèfle est double : il décompose distinctement en trois classes les services que l'on attend d'un système multi-utilisateur et il fournit une définition conceptuelle de ces systèmes. En effet, nous avons trop souvent vu des définitions à partir d'exemples d'applications (par exemple : "les collecticiels sont les systèmes de communication et les éditeurs partagés"). En identifiant clairement les trois aspects information, coordination, communication, ce modèle nous permet d'englober tous les systèmes multi-utilisateurs : à la fois les collecticiels existants et usuellement reconnus comme tels, mais aussi les collecticiels atypiques comme les outils de communication, ainsi que d'éventuelles nouvelles catégories de collecticiels à venir.

Afin d'éprouver la validité du modèle du trèfle, il est intéressant de le rapprocher de modèles organisationnels de l'utilisation de la technologie. Thompson par exemple distingue trois rôles de la technologie [Thompson 1967] :

- un rôle *médiateur* : la technologie est un creuset qui regroupe les résultats de tâches effectuées par différents utilisateurs, séquentiellement ou simultanément (figure 1.3),

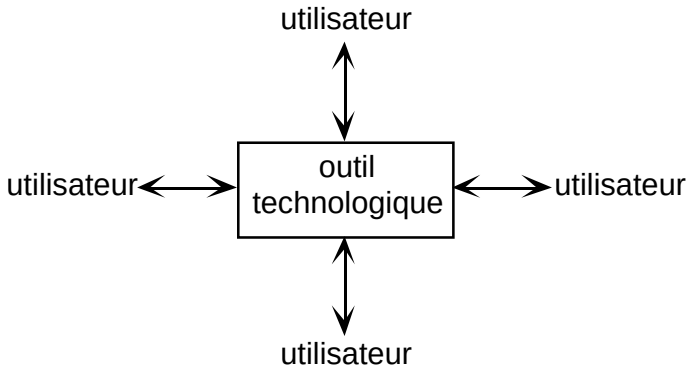

Figure 1.3. L'outil technologique comme médiateur. Différents utilisateurs réalisent des tâches distinctes et l'outil technologique rassemble les différents résultats. Les doubles flèches indiquent que les utilisateurs confient à l'outil le résultat de leurs tâches mais peuvent aussi réutiliser le résultat d'autres tâches pour accomplir leurs tâches.

- un rôle *d'ordonnancement* : les tâches sont interdépendantes séquentiellement dans le temps. Ce modèle est le modèle classique du travail à la chaîne (figure 1.4),

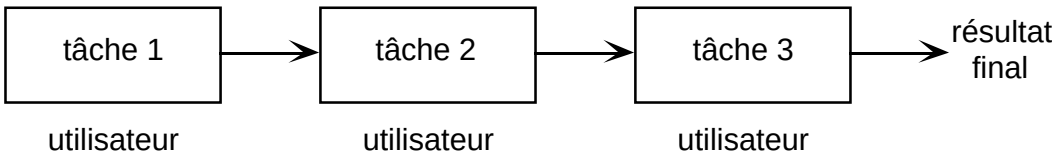

Figure 1.4. Le rôle d'ordonnancement de l'outil technologique. Chaque utilisateur réalise une tâche précise avec le support de l'outil technologique et la tâche (n+1) ne peut être réalisée avant terminaison de la tâche (n).

- enfin la technologie peut être utilisée de façon *intensive* : différentes tâches doivent être réalisées simultanément et peuvent être fortement interdépendantes. Les différents utilisateurs négocient entre eux directement (figure 1.5).

Notons que, même si ce modèle date d'il y a presque trente ans, il nous semble toujours pertinent. La vague récente de "business process reengineering" n'a fait que modifier l'allocation des tâches aux utilisateurs (principalement en leur assignant un plus grand éventail de tâches), mais n'a pas fondamentalement remis en cause les schémas de Thompson.

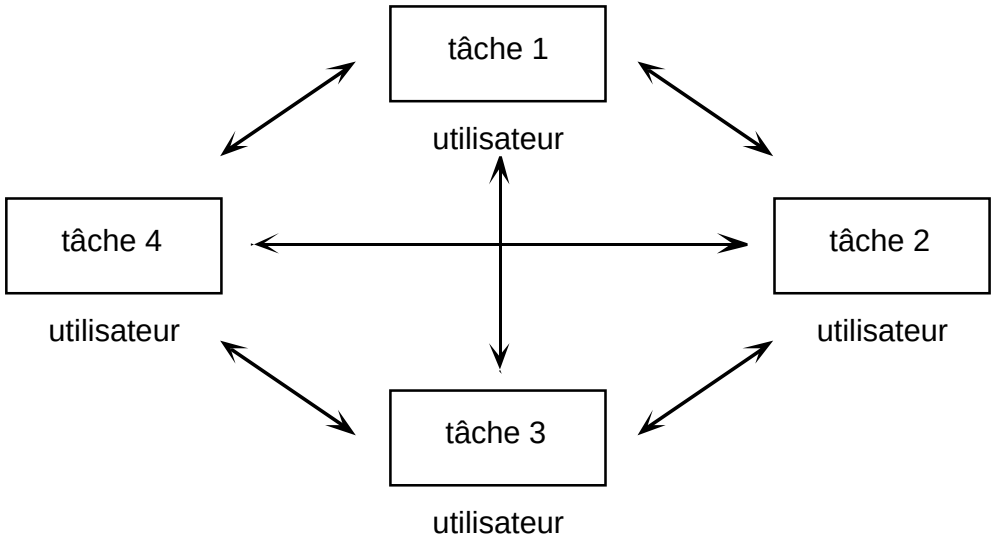

Figure 1.5. Le rôle intensif de la technologie. Chaque utilisateur réalise une tâche qui peut être liée de façon complexe aux tâches des autres utilisateurs. Les flèches indiquent que les utilisateurs doivent négocier entre eux pour clarifier et organiser les dépendances entre les tâches.

Il est instructif de mettre en regard ces trois rôles de la technologie et les trois composantes du trèfle des systèmes multi-utilisateurs. Même si l'on peut en faire une analyse plus subtile, il nous semble que l'on peut rapprocher ces deux modèles de la façon suivante. Le rôle médiateur de la technologie est à rapprocher de l'espace de production du modèle du trèfle. Le but de la technologie est ici un rôle intégrateur, les tâches des utilisateurs concourant à un but commun. Le rôle d'ordonnancement de la technologie privilégie la composante coordination du trèfle. La technologie a alors un rôle d'orchestration des différentes tâches et des différents utilisateurs. Enfin, le rôle intensif de la technologie met en avant la composante communication du trèfle.

Cette analyse rapide d'un modèle organisationnel nous conforte quant à la pertinence du modèle du trèfle pour décrire l'espace fonctionnel des systèmes multi-utilisateurs. Nous terminons ce paragraphe en définissant deux grandes classes de systèmes multi-utilisateurs.

Une distinction usuelle est faite entre systèmes multi-utilisateurs synchrones et asynchrones. La classification espace-temps proposée par Ellis [Ellis 1991] nous permet de clarifier cette distinction. La classification espace-temps repose sur deux axes : la distance temporelle entre les utilisateurs du système, et leur distance spatiale (figure 1.6).

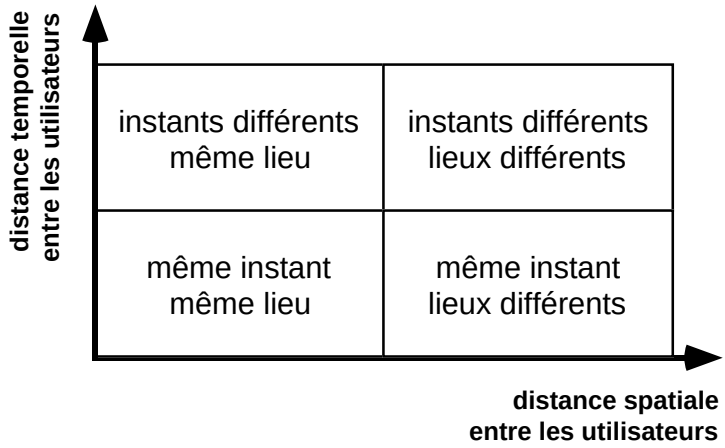

Figure 1.6. La classification espace-temps.

Notons que cette classification peut être affinée. Par exemple, il est plus réaliste de considérer les axes espace et temps comme continus [Salber 1994]. On peut alors considérer un ordre de grandeur pour la distance spatiale, par exemple distance métrique et distance kilométrique. Pour tenir compte des systèmes informatiques mobiles, [Grudin 1994] propose de distinguer “lieux différents mais imprévisibles” et “lieux différents mais prévisibles”. Il convient, à notre sens, d'insister sur le caractère imprévisible de la coopération et, par conséquent de considérer les cas "mêmes lieux mais imprévisibles" et “mêmes lieux mais prévisibles”.

La classification de la figure 1.6 nous permet déjà de distinguer le caractère synchrone et asynchrone des systèmes multi-utilisateurs. Les deux quadrants du haut de la classification, ceux pour lesquels les utilisateurs interagissent à des instants différents représentent les *systèmes asynchrones*. Les deux quadrants du bas, pour lesquels les utilisateurs interagissent au même instant (“même instant” du point de vue des utilisateurs et non du système), représentent les *systèmes synchrones*[1]. Notons qu’il est tentant de rapprocher la classification espace-temps du modèle du trèfle. Potentiellement, chacune des composantes du trèfle peut être considérée dans le cadre de la classification espace-temps. Dans [Salber 1995], nous proposons un modèle intégrant le modèle du trèfle des systèmes multi-utilisateurs et le classique modèle espace-temps.

1.3. La conception des systèmes multi-utilisateurs

La spécification, la réalisation et l’évaluation d’un système interactif sont réputées difficiles. Les systèmes interactifs multi-utilisateurs font apparaître de nouvelles

[1] Le terme “collecticiel temps réel” est également employé pour désigner les systèmes multi-utilisateurs synchrones.

difficultés. Une bonne part des problèmes rencontrés sont d'ordre technique ; mais en amont de la réalisation du système, la définition et la conception du système ne sont pas des tâches simples. La diversité des disciplines qui apportent leurs concours à la conception, chacune ayant une culture, un vocabulaire et un champ d'action différents, est pour beaucoup dans cette complexité. Les approches étudiant la conception des systèmes, telle l'approche Design Rationale peuvent apporter une aide à l'intégration de ces différentes disciplines. Nous avons aussi trouvé utile de structurer notre réflexion suivant une grille d'analyse à trois niveaux que nous présentons ensuite.

1.3.1. Intégrer l'approche multidisciplinaire

Il est maintenant reconnu que le développement des systèmes interactifs requiert la participation de plusieurs disciplines : psychologues, ergonomes, informaticiens apportent tous des compétences et des savoir-faire indispensables. Les systèmes multi-utilisateurs s'adressant à un groupe d'utilisateurs, des disciplines supplémentaires doivent intervenir, telles les sciences sociales. C'est là une des difficultés majeures du développement de ces systèmes. L'expérience montre que la coopération de plusieurs disciplines, chacune ayant des préoccupations, une culture, un vocabulaire différents, est un exercice complexe. Il est nécessaire de disposer d'un canevas intégrateur permettant de prendre en compte les apports de chacune des disciplines intervenant dans le processus de conception. Le Design Rationale se propose justement de jouer ce rôle d'intégration.

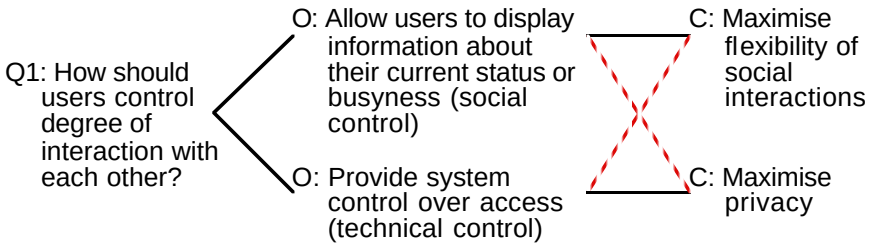

Figure 1.7. Exemple de mise en œuvre de la méthode QOC pour la conception d'un système mediaspace. La question posée Q1 (comment les utilisateurs peuvent-ils contrôler leur degré d'interaction ?) a suscité deux options qui sont examinées en fonction de deux critères. D'après [Duke 1995].

L'approche "Design Rationale" du processus de conception consiste à documenter, à l'aide de diverses techniques, toutes les décisions prises lors de la conception. Cette documentation participe à la traçabilité en génie logiciel. Nous avons choisi de détailler la méthode QOC, représentative de cette approche, mais on peut citer d'autres méthodes comme Design Space Development (DSD) [Bernsen 1994]. La méthode QOC (Questions, Options, Criteria) [MacLean 1991] est une représentante de ces techniques de Design Rationale. Elle consiste à explorer le plus systématiquement possible l'espace solution

correspondant à un problème donné. Pour chaque question qui apparaît lors de la conception d'un système, les options possibles constituant l'espace solution sont explorées et documentées. En regard sont indiqués les critères que satisfont ou non chacune des options. La figure 1.7 montre un exemple de mise en œuvre de la méthode QOC pour la conception du système mediaspace ECOM. Deux options sont envisagées pour la question Q1 et deux critères sont envisagés. Les traits pleins entre options et critères indiquent que le critère est satisfait par l'option considérée. Un trait pointillé indique que le critère s'oppose à l'option choisie.

Trois critiques peuvent être faites à QOC : la première porte sur l'utilisabilité pratique de la méthode, les deux autres portent sur le fait que, si QOC documente bien les choix de conception, la méthode n'offre pas de guide pour déterminer les critères et leur pertinence.

L'utilisabilité pratique de la méthode QOC peut être mise en question. L'espace solution, même pour un choix de conception précis, peut être d'une taille imposante. Il n'est pas rare de trouver cinq à dix options et autant de critères pour une question donnée. Plus il y a d'intervenants, plus il risque d'y avoir d'options et de critères. Dans une approche multidisciplinaire, chaque discipline fournit son propre jeu d'options et de critères correspondants. Plus il y a de disciplines impliquées, plus l'espace de conception est vaste. En l'absence de support informatique, l'espace de conception devient vite ingérable avec la méthode QOC [Bellotti 1994]. Dans [Nigay 1994], nous proposons des mécanismes d'encapsulation permettant d'améliorer la lisibilité de grands espaces de conception décrits avec la méthode QOC.

Une critique plus sérieuse que l'on peut faire à la méthode QOC est qu'elle n'inclut pas de guide permettant de déterminer les critères pertinents. Lorsque nous avons utilisé QOC pour décrire un espace de conception logicielle, nous avons choisi comme critères des propriétés issues du génie logiciel. Nous pensons que cette approche est généralisable pour d'autres espaces de conception. Au paragraphe suivant, nous proposons une grille d'analyse qui situe les propriétés et nous présenterons la nature des propriétés au chapitre 4.

Notre troisième critique met en évidence une contradiction de la méthode QOC. QOC ne permet pas de trancher le choix d'une option dans le cas où deux options satisfont des critères différents et non contradictoires. La figure 1.7 en donne un exemple simple : les deux options considérées satisfont chacune un critère différent. Comment le concepteur peut-il trancher ? Il est possible de donner plus ou moins d'importance aux différents critères, par exemple en les pondérant. Mais dans ce cas les décisions ayant mené à

l'attribution de tel poids à tel critère ne sont pas explicitement documentées dans le diagramme QOC. A la rigueur, ces décisions peuvent être consignées implicitement dans le cahier des charges ou le plan d'assurance qualité. Ce choix des poids, essentiel puisqu'il permet de prendre une décision dans des cas difficiles (puisque "tangents"), n'est pas capturé par le Design Rationale. Ce point est en contradiction avec la motivation même du Design Rationale. Même si les critères sont exprimés sous forme de propriétés, favoriser une propriété plutôt qu'une autre est influencé par des choix de conception d'un plus haut niveau. Ces choix de haut niveau sont les principes de notre grille d'analyse.

1.3.2. Une approche à plusieurs niveaux : principes, propriétés et techniques

Pour essayer de structurer notre vision du processus de développement des systèmes multi-utilisateurs, nous proposons une grille d'analyse à trois niveaux. Cette proposition vise à prendre en compte les apports des différentes disciplines qui participent à la conception des systèmes multi-utilisateurs et à intégrer leurs apports sous forme d'une "lingua franca" de propriétés. Les propriétés ainsi déterminées peuvent ensuite être utilisées avec une technique d'intégration telle que QOC, comme nous l'avons évoqué plus haut. Nous avons identifié trois niveaux échelonnés dans l'ordre chronologique du processus de conception : les principes, les propriétés et les techniques. La figure 1.8 résume notre approche.

Précisons tout de suite pour fixer les idées que l'on peut tracer un parallèle entre nos trois niveaux de principes, propriétés et techniques et un espace référentiel proposé pour le génie logiciel par McCall [McCall 1977]. McCall identifie aussi trois niveaux dans un processus de développement logiciel mettant en jeu un client, un concepteur et un réalisateur : les facteurs traduisent les attentes du client telles qu'exprimées dans le cahier des charges. Les critères s'adressent au concepteur : ils traduisent les facteurs dans les termes du concepteur. Enfin les solutions s'adressent au réalisateur et traduisent les facteurs en termes de solutions techniques. On retrouve dans notre proposition ces niveaux d'affinement successifs, chacun s'adressant à différents acteurs du cycle de développement.

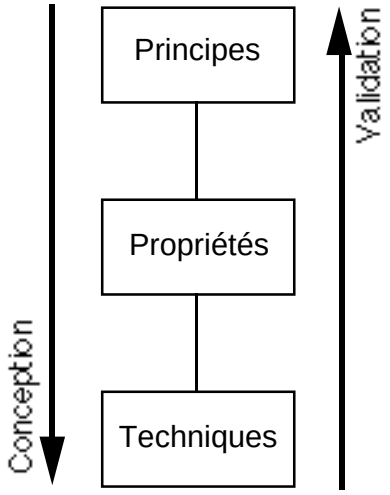

Figure 1.8. Les trois niveaux des principes, propriétés, et techniques. Le processus d'affinement descendant est effectué lors de la conception. Il doit être suivi d'un processus "remontant" de validation.

1.3.2.1. Principes

Nous utilisons ici le mot *principe* dans son sens "cause première"[1]. Un principe est une contrainte permettant de guider les choix de conception et qui est établi très tôt dans le processus de conception. Ce peut aussi être une philosophie qui préside à la conception. Un principe décrit un aspect qualitatif du système à concevoir. Un principe est en général indépendant de l'artefact informatique ; les indications exprimées par un principe sont d'un haut niveau et ne préjugent pas de la réalisation : elles doivent s'appliquer au système informatique mais un principe pourrait être tout autant valide dans le cas d'une réalisation non informatique. Ce dernier point est utile pour intégrer les apports des disciplines non informatiques. En effet, les enseignements que ces disciplines peuvent apporter sont souvent exprimés indépendamment de toute réalisation informatique.

En fait, les principes ne sont pas spécifiques aux systèmes multi-utilisateurs et on peut déjà les identifier dans la conception des systèmes interactifs mono-utilisateurs. Dans une approche génie logiciel, les principes seront exprimés, parfois implicitement, dans le cahier des charges. Ils peuvent aussi être modélisés, par exemple sous forme d'un modèle d'utilisateur établi au début de la conception, ou être intégrés dans le modèle de tâche. Pour les systèmes multi-utilisateurs cependant, de nouveaux types de principes doivent être pris en compte. Ils peuvent être exprimés sous forme d'un modèle du groupe utilisateur du système, et d'un modèle social ou organisationnel. Pour concrétiser cette notion de principe, nous donnons maintenant quelques exemples de principes sociaux ou organisationnels. Notons que ces principes sont présentés hors de leur contexte et qu'ils

[1] Principe : Cause première active, primitive et originelle. (*Le Robert*, dictionnaire de la langue française)

ne sont pas des principes universels. Ils ne sont valides que dans le cadre de la conception d'un système donné.

- "La vie privée des individus doit être protégée."
  Ce principe est issu des réflexions sur les aspects éthiques de l'usage des outils informatiques. Il est particulièrement important pour les systèmes de communication homme-homme médiatisée. Nous justifions ce principe au chapitre 2, en nous appuyant sur les travaux des sciences sociales. Toutefois, la conception d'un système particulier pourrait choisir d'ignorer ce principe.

- "Les utilisateurs sont souvent interrompus dans leur travail."
  Ce principe a été identifié par [Rouncefield 1994], lors d'une étude ethnographique. Les auteurs en déduisent implicitement des propriétés que devrait respecter un système informatique adapté à ces utilisateurs. Par exemple, les écrans ne doivent pas s'éteindre automatiquement après quelques minutes d'inactivité de l'utilisateur, ni surtout perdre des données en cours de saisie en s'éteignant. Ici, une propriété telle que la stabilité de l'écran serait pertinente.

- "L'automatisation totale du travail n'est pas souhaitable."
  Nous avons rencontré ce principe dans la description du travail des contrôleurs aériens. Ce principe a une influence certaine sur les choix de conception du système : on visera à fournir une information adéquate aux contrôleurs aériens, mais jamais le système ne se substituera à eux, en particulier pour prendre des décisions. La propriété de non-préemption présentée au chapitre 4 devrait être appliquée à la conception d'un tel système.

- "Les réunions doivent être courtes et efficaces."
  Ce principe pourrait guider le développement d'un système de réunion informatisée. Ce principe est organisationnel et n'est pas lié à un artefact technologique. On peut citer à titre d'anecdote la façon dont Hubert Beuve-Méry, fondateur et directeur du quotidien *Le Monde*, appliquait ce principe : les réunions de rédaction se déroulaient autour d'une table sans chaises, tous les participants restant debout pendant la réunion. De fait, les réunions de rédaction étaient toujours courtes ! Si l'on voulait appliquer ce même principe dans le cadre d'un outil de réunion informatisé, on pourrait par exemple envisager une limitation du temps de parole de chaque participant. La propriété qui étudie le rythme de l'interaction serait pertinente dans ce cas.

- "Laisser les utilisateurs résoudre leurs différends."
  Ce principe est proposé par Mike Godwin pour les "communautés virtuelles" [Godwin 1994]. Il montre encore une fois qu'un principe a un domaine d'application précis. Ce principe ne serait sans doute pas acceptable dans beaucoup d'entreprises où un conflit est généralement arbitré par un supérieur hiérarchique. Pour le cas des systèmes adaptés aux communautés virtuelles, il plaide contre les "modérateurs", au moins pour l'arbitrage des différends. Il est à rapprocher du principe suivant, plus général.

- "Préférer les solutions sociales, et seulement si tout échoue, recourir aux solutions techniques."
  Ce principe a été proposé par Amy Bruckman, suite à son expérience avec la communauté virtuelle LambdaMOO [Bruckman 1994]. Il recommande de privilégier les solutions négociées entre utilisateurs, et de ne recourir à des solutions techniques contraignantes que lorsque tous les autres recours ont été épuisés. La propriété contrôle technique/contrôle social serait alors applicable.

Bien sûr, nous ne pouvons pas prétendre décrire exhaustivement les principes. Chaque contexte de conception et les différentes disciplines impliquées dans le processus de conception feront apparaître des principes différents. Pour illustrer ce point, nous donnons deux exemples, le premier tiré d'un contexte particulier, le second concernant un point d'interface précis.

Le premier exemple montre comment un contexte de développement particulier peut générer un principe en contradiction absolue avec un principe de la conception centrée utilisateur. Un des principes de la conception centrée utilisateur est de viser à ce qu'une interface soit adéquate à la tâche de l'utilisateur. Or il est un domaine notable où ce principe ne doit *pas* être respecté : les jeux. En effet si l'on admet que, dans un jeu, la tâche de l'utilisateur est de gagner, un bon jeu sera justement celui où gagner est plus difficile. Dans cet objectif, l'interface d'un jeu viole certaines propriétés de façon flagrante. On pense aux jeux d'aventure comme Myst [Myst 1993] dont l'interface contient de nombreux éléments cachés et où la structure même de la tâche n'est ni explicite ni prévisible (les indices sont à rechercher dans un certain ordre, lui-même à découvrir). Les propriétés d'affordance, d'observabilité, d'honnêteté[1], pour ne citer que celles-là, sont délibérément contredites.

[1] Ces propriétés sont définies au chapitre 4.

Un autre exemple nous est fourni par un problème courant et important pour certains systèmes de communication homme-homme médiatisée : la conception de l'interface d'entrée d'un mot de passe par l'utilisateur. Un principe applicable ici est celui du respect de la confidentialité des utilisateurs. Des principes de sécurité peuvent être aussi considérés. Ces principes conduisent en général à ne pas respecter la propriété d'observabilité : le mot de passe ne s'affiche pas. Notons des différences d'approche cependant : sous Unix, taper un mot de passe ne provoque absolument aucun écho. En revanche, taper un mot de passe pour se connecter à un serveur AppleShare renvoie un caractère "•" à la place de chaque caractère tapé. Dans ce deuxième cas, une partie de l'état interne est observable : le nombre de caractères tapés. On peut se demander dans quelle mesure la satisfaction partielle de la propriété d'observabilité contredit des principes de sécurité (par exemple, on peut savoir que mon mot de passe AppleShare a 8 caractères). Un autre conflit entre la sécurité et l'observabilité peut apparaître : lorsque l'utilisateur doit entrer son nom et son mot de passe, en cas d'erreur, un système Unix affichera un message "login error". Pour satisfaire un principe de sécurité, le système n'affichera pas un message d'erreur informatif. En particulier il ne précisera pas si le nom est inconnu ou si le mot de passe est incorrect. Notons qu'AppleShare affiche deux messages d'erreur distincts. Mais en règle générale, ce choix d'interface contredit les propriétés de réparation d'erreur, d'informativité des messages et d'observabilité pour satisfaire des principes de sécurité.

A notre connaissance, il n'y a pas de méthode a priori pour vérifier que le système construit satisfait les principes énoncés au début de la conception. L'évaluation de la conformité du système aux principes souhaités ne peut se faire que par observation de l'utilisation du système et des transformations qu'il peut induire dans l'organisation considérée. Il est probable qu'une évaluation multidisciplinaire sera nécessaire. Force est de constater que ce domaine de recherches est encore peu abordé.

En résumé, les principes sont des contraintes de haut niveau, en général qualitatives et indépendantes de toute réalisation informatique, élaborées très tôt dans le processus de conception. Les principes peuvent trouver leur place dans le cahier des charges du système à réaliser, mais certains principes de conception, en particulier les principes organisationnels ou sociaux pour les systèmes multi-utilisateurs sont souvent implicites dans le cahier des charges. Une fois clairement exprimés, les principes peuvent être affinés en propriétés.

1.3.2.2. Propriétés

Une *propriété* d'un système est une caractéristique du système directement vérifiable. A l'aide de métriques appropriées, une propriété est quantifiable. Au contraire des principes,

les propriétés caractérisent le système informatique. Elles peuvent donc être utilisées comme critères de la notation QOC. Comme les principes, les propriétés sont élaborées à partir de l'apport des différentes disciplines intervenant dans la conception du système. Pour un développement donné, un ensemble de propriétés souhaitables est identifié. Le choix de ces propriétés et leur importance relative sont guidés par les principes énoncés au début de la conception.

Les propriétés pertinentes pour le développement des systèmes interactifs et en particulier des systèmes multi-utilisateurs sont exposées plus en détail au chapitre 4. Comme les propriétés concernent directement le système informatique et sont quantifiables, elles sont vérifiables par analyse du système construit. L'évaluation de la conformité du système aux propriétés énoncées est donc possible. Cette évaluation détermine l'utilisabilité du système. Dans la deuxième partie de ce mémoire, nous définirons des propriétés et montrerons de quelle façon elles participent à l'évaluation du système.

1.3.2.3. Techniques

Les *techniques* sont mises en œuvre pour réaliser le système une fois qu'il est spécifié. Il s'agit principalement de modèles et d'outils informatiques, et leur choix est guidé par les propriétés souhaitées du système. En effet, si l'on utilise une méthode telle que QOC, l'examen des propriétés souhaitées (c'est-à-dire des critères) va permettre de faire un choix dans l'espace des solutions et souvent induire un choix de techniques et d'outils. Par exemple, pour satisfaire la propriété de portabilité, on voudra utiliser une boîte à outils virtuelle ou un environnement de développement multi-plate-forme.

Nous montrons dans les deux chapitres suivants comment les disciplines non informatiques peuvent générer des principes. Nous en avons déduit des propriétés, présentées avec d'autres propriétés usuelles au chapitre 4. Le chapitre 5 étudie comment la satisfaction des propriétés peut être évaluée. Enfin, les techniques, sous forme de techniques d'évaluation, de modèles d'architecture et d'outils, font l'objet des chapitres 5, 6, 7 et 8 de ce mémoire.

1.4. Synthèse

Pour répondre à la complexité du développement des systèmes multi-utilisateurs, due principalement à la multiplicité des participants impliqués dans le processus de conception, nous avons identifié trois niveaux d'affinement dans le processus de développement. Ces trois niveaux mettent en évidence des principes, des propriétés et des techniques que le système construit devra utiliser ou satisfaire. Le passage d'un niveau à un autre correspond à un processus de réification pour finalement aboutir à la

spécification du système et à l'ensemble des outils nécessaires à son développement effectif. En ce qui concerne le processus de validation, qui permet de vérifier la conformité du système construit aux niveaux des propriétés et des principes, nous devons constater les limites actuelles. A chaque niveau correspond un ensemble de concepts et un espace de référence. Notre structuration en trois niveaux n'est pas une méthode de conception, mais une méthode d'analyse. Elle ne contraint pas l'ordonnancement et est donc adaptée au comportement souvent opportuniste des concepteurs : la conception n'est jamais exclusivement "top-down" ou "bottom-up". Le processus descendant de conception fait pour le moment l'objet de l'essentiel des recherches, et il faudra sans doute attendre que celui-ci soit plus élucidé avant de voir apparaître des méthodes de validation. En particulier, l'évaluation multidisciplinaire du système vis-à-vis des principes énoncés lors de la conception du système nous semble une voie de recherche à explorer.

Références

[Bass 1992] L. Bass, R. Little, R. Pellegrino, S. Reed, R. Seacord, S. Sheppard et M. R. Szczur. *The UIMS Tool Developers' Workshop: A Metamodel for the Runtime Architecture of an Interactive System,* in *SIGCHI Bulletin*, 24(1), janvier 1992. pp. 32-37.

[Bellotti 1994] V. Bellotti et A. MacLean. *Integrating and Communicating Design Perspectives with QOC Design Rationale*, Amodeus Project, Working Paper, ID/WP29, 1994.

[Bernsen 1994] N. O. Bernsen et J. Ramsay. *An executive summary of the DSD framework illustrated by two worked exemplars*, Amodeus Project, 1994.

[Bruckman 1994] A. Bruckman. *Approaches to Managing Deviant Behaviour in Virtual Communities (Panel)*, CHI 95, ACM Conference on Human Factors in Computing Systems, Boston, Massachussets, USA, 1994. pp. 183-184.

[Duke 1995] D. J. Duke, (editor), A. Aboulafia, A. E. Blandford, S. J. Buckingham-Shum, J. Darzentas, J. May, L. Nigay, J. Ramsay, D. Salber et S. Verjans. *Integration Techniques for Multi-Disciplinary HCI Modelling: A Survey*, Projet européen ESPRIT BRA 7040 AMODEUS 2, RP3 Working Paper (en cours de soumission), ID/WP, 1995.

[Ellis 1994] C. Ellis et J. Wainer. *A Conceptual Model of Groupware*, CSCW 94, ACM Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, USA, 1994. pp. 79-88.

[Ellis 1991] C. A. Ellis, S. J. Gibbs et G. L. Rein. *Groupware: some issues and experiences,* in *Communications of the ACM*, 34(1), janvier 1991. pp. 38-58.

[Godwin 1994] M. Godwin. *How to make virtual communities work,* in *Wired*, 2(6), juin 1994. pp. 92.

[Grudin 1994] J. Grudin. *CSCW: History and Focus,* in *IEEE Computer*, 27(5), mai 1994. pp. 19-26.

[MacLean 1991] A. MacLean, R. M. Young, V. Bellotti et T. Moran. *Questions, Options and Criteria: Elements of Design Space Analysis,* in *Human-Computer Interaction*, 6(3 & 4), 1991. pp. 201-250.

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

[Myst 1993] *Myst 1.0.* Logiciel pour Macintosh (CD-ROM). 1993.

[Nigay 1994] L. Nigay, J. Coutaz et D. Salber. *Initial draft preliminary on-the-fly PAC analysis for the ECOM Interface*, Projet européen ESPRIT BRA 7040 AMODEUS 2, Integration Report, SM/IR 4, 1994.

[Rouncefield 1994] M. Rouncefield, J. A. Hughes, T. Rodden et S. Viller. *Working with "Constant Interruption": CSCW and the Small Office*, CSCW 94, ACM Conference on Computer-Supported Cooperative Work, Chapel Hill, North Carolina, USA, 1994. pp. 275-286.

[Salber 1994] D. Salber et J. Coutaz. *Taxinomie des mécanismes de connexion pour la communication homme-machine-homme*, IHM'94, Sixièmes Journées sur l'Ingénierie des Interfaces Homme-Machine, Lille, France, 1994. pp. 183-189.

[Salber 1995] D. Salber, J. Coutaz, D. Decouchant et M. Riveill. *De l'observabilité et de l'honnêteté : le cas du contrôle d'accès dans la Communication Homme-Homme Médiatisée*, soumis à IHM'95, Conférence sur l'Ingénierie des Interfaces Homme-Machine, Toulouse, France, 1995.

[Stefik 1988] M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning et L. Suchman. *Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings,* in *Computer-Supported Cooperative Work: A Book of Readings.* I. Greif, (ed.) Morgan Kaufman, San Mateo, California, 1988. pp. 335-366.

[Thompson 1967] J. D. Thompson. *Organizations in Action,* McGraw-Hill, New York, New York, USA, 1967.