HAL open science

Ingénierie de la fluidité interactionnelle : vers des systèmes symbiotiques

Yann Laurillau

To cite this version:

Yann Laurillau. Ingénierie de la fluidité interactionnelle : vers des systèmes symbiotiques. Interface homme-machine [cs.HC]. Université Joseph Fourier, Grenoble, 2014. tel-03286230

HAL Id: tel-03286230

https://hal.science/tel-03286230v1

Submitted on 14 Jul 2021

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

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

UNIVERSITE JOSEPH FOURIER
SCIENCES. TECHNOLOGIE. MEDECINE

GRENOBLE 1

HABILITATION A DIRIGER DES RECHERCHES
DE L'UNIVERSITÉ GRENOBLE 1 (UJF)

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Yann LAURILLAU

Préparée au sein du **Laboratoire d'Informatique de Grenoble**

Ingénierie de la fluidité interactionnelle : vers des systèmes symbiotiques

Thèse soutenue publiquement le **21 novembre 2014**,
devant le jury composé de :

M. James CROWLEY
Professeur à Grenoble INP, Président
Mme. Karin CONINX
Professeur à l'Université de Hasselt (Belgique), Rapporteur
M. Nicholas GRAHAM
Professeur à l'Université Queen's University (Canada), Rapporteur
M. Philippe PALANQUE
Professeur à l'Université Paul Sabatier, Rapporteur
Mme. Gaëlle CALVARY
Professeur à Grenoble INP, Membre
M. Oscar PASTOR
Professeur à l'Université polytechnique de València (Espagne), Membre
Mme. Sylvie PESTY
Professeur à l'Université Pierre-Mendès France, Membre
M. Jean VANDERDONCKT
Professeur à l'Université Catholique de Louvain (Belgique), Membre

Remerciements

Ce travail fût une sorte de cure de rajeunissement, en remontant le temps de mes réalisations, pour se remémorer le temps lointain de la thèse, finalement assez semblable dans ces moments de joie dans la progression, de doute face à l'incertitude, d'inquiétude puis de plaisir par tant de compliments face au jury, et de libération avec le sentiment du devoir accompli. D'ailleurs, dans les remerciements de ma thèse de doctorat, je précisais qu'il s'agissait d'une œuvre finalement collective car sans les nombreux soutiens personnels et professionnels, elle n'aurait pu aboutir : c'est encore vrai cette fois-ci.

Merci à Karin, Sylvie, Jean, Jim, Oscar, Nick, et Philippe qui m'ont fait l'immense honneur d'accepter d'être membre du jury, quitte à venir de très loin. Ce fût un très grand plaisir de vous avoir.

Merci à toi Gaëlle pour tout le soutien apporté durant ce périple, de la rédaction à la soutenance, par ta très grande disponibilité malgré un agenda chargé, pour ton regard critique affûté et constructif qui a très largement contribué à atteindre la qualité nécessaire. Plus largement, merci à l'ensemble de l'équipe IIHM du LIG, notamment ses permanents, par tradition, de se rendre toujours disponible pour écouter et conseiller dans les phases de préparation de l'oral.

Merci à mes collègues de l'IUT pour l'ambiance chaleureuse qui donne envie de venir enseigner. En particulier, merci à Gaëlle d'avoir veillé à ce que je puisse préserver du temps pour la rédaction du mémoire.

Merci à mes proches pour leur soutien moral tout au long de ce parcours.

Table des matières

I. INTRODUCTION : VERS DES SYSTEMES SYMBIOTIQUES 1

1. Une nouvelle ère de l'informatique 1
2. Vers des systèmes symbiotiques 2
3. Mutation des collecticiels 4
4. Fluidité interactionnelle, un requis 5
5. Structure du mémoire 6

II. LIGNES DE RECHERCHE : DIMENSIONS DE LA FLUIDITE INTERACTIONNELLE 7

1. Systèmes symbiotiques 7
2. Architecture d'un système symbiotique 9
   Interface entre physique et numérique :
   espace personnel, espace social 9
   Relations symbiotiques 10
3. Dimensions de la fluidité interactionnelle 11
   3.1. Modalités d'interaction et multimodalité 11
   3.2. Interaction utilisateur et multiutilisateur 12
   3.3. Multiplication des contextes d'usage 13
   3.4. Synthèse 14

III. INGENIERIE DE LA FLUIDITE INTERACTIONNELLE : REQUIS DE FLEXIBILITE 15

1. Du mono-utilisateur au multiutilisateur 16
   1.1. Flexibilité des modes d'interaction 16
   Permettre l'opportunisme 17
   Anticiper l'imprévisible 20
   Indépendance 23

1.2. Architectures logicielles : flexibilité multiniveaux....26
Modèles en couche et à agents....27
Flexibilité longitudinale....29
Flexibilité macro....30
Flexibilité micro....32
Plateforme Clover....33
Flexibilité transversale....35

2. Du monomodal au multimodal....39
2.1. Spécification de l'interaction multimodale....39
Arbre de tâches et multimodalité....39
Spécifier l'interaction gestuelle multiutilisateurs : classes de techniques....43

2.2. Modalités d'interaction gestuelle....46
Interaction gestuelle au touché....47
Interaction muscle-machine....50

3. Du monocontexte au multicontextes....54
3.1. Spécification de l'adaptation....54
Modèle de comportement....54
Modèle de composant....55
Modèle d'architecture....56
Illustration : actions de groupe entrelacées....57

3.2. Hétérogénéité en situation de mobilité....57
Apport de la multimodalité....58
Contenus différenciés : compromis et adaptation....59

3.3. Contexte social : donner du contrôle a l'utilisateur....61
Niveaux d'observabilité : donner du contrôle....62
Visualisation des échanges en contexte social....64
Concept de Perle....69

IV. CONCLUSION ET PERSPECTIVES 71

1. Rappel des contributions....72
Du mono-utilisateur au multiutilisateur....72
Du monomodal au multimodal....73
Du monocontexte au multicontextes....74

2. Perspectives de recherche....75
Fluidité interactionnelle : vers une interaction plus naturelle....75
Flexibilité : vers des environnements de conception centrée humain....76

V. BIBLIOGRAPHIE 77

VI. ANNEXE 87

Introduction

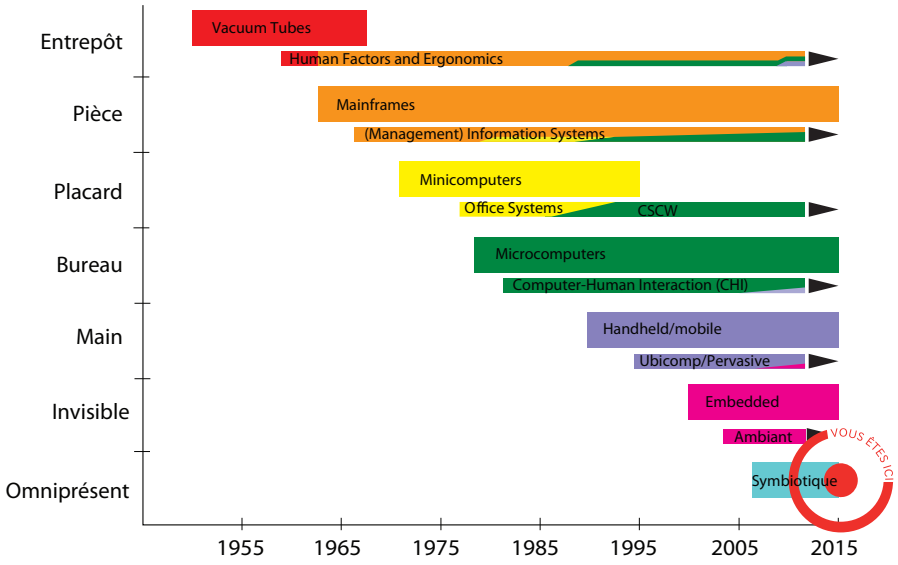
Vers des systèmes symbiotiques, requis de fluidité

« **Many people live and work across many platforms and interact with many people, yet our UI systems architectures support none of this**. »
(Olsen, 2007)

De l'ingénierie des collecticiels, c'est-à-dire des systèmes interactifs multiutilisateurs dédiés à l'activité de groupe, mes travaux de recherche se sont, au fil des ans, progressivement orientés vers l'ingénierie logicielle de systèmes interactifs polyvalents et polymorphes pour tendre finalement vers l'**ingénierie de systèmes interactifs symbiotiques**. Cette évolution reflète les mutations qui se produisent depuis plusieurs décennies dans les domaines de recherche sur l'Interaction Homme-Machine (IHM) et le Travail Collaboratif Assisté par Ordinateur (TCAO).

1. Une nouvelle ère de l'informatique

Figure 1
Hardware Platforms and HCI research fields, adapté de (Grudin, 2012)



Dès 2006 puis en 2012 dans les colonnes du magazine ACM *Interactions*, Jonathan Grudin, de la communauté *Computer-Supported Cooperative Work* (CSCW), met en évidence les mutations de ces deux communautés de recherche en comparant leur évolution avec celle des espèces vivantes (Grudin, 2006 ; Grudin, 2012).

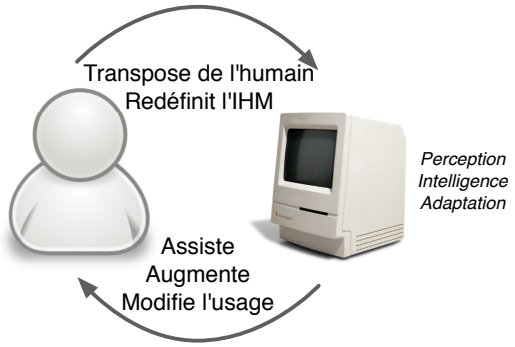
Il montre comment des changements technologiques majeurs sont à l'origine de ces mutations ainsi que de disparitions de certaines espèces comme les serveurs *mainframe* et communautés de recherche afférentes. Grudin illustre par un synoptique (voir Figure 1) l'histoire de ces changements aboutissant à l'apparition de nouvelles communautés, accompagnant une miniaturisation toujours plus poussée des technologies informatiques jusqu'à en devenir invisibles à nos yeux : nous sommes désormais dans l'ère de l'informatique ambiante et il est déjà légitime de s'interroger quand aura lieu la prochaine évolution majeure.

Pour Brangier et coll. (2009), nous sommes au début d'une nouvelle ère, celle des **systèmes informatiques symbiotiques**, prolongeant ainsi l'histoire retracée par J. Grudin (voir Figure 1). La vision proposée par ce dernier est celle de la recherche qu'il convient de compléter par la question de l'usage des technologies de l'information. Notre rapport avec les systèmes informatiques, nos usages, nos pratiques et activités du quotidien ont évolué et se trouvent profondément transformés par l'importance croissante prise par les technologies dans l'espace et dans le temps. Comme le souligne E. Brangier et coll. (2009), nous avons dépassé le stade de la question de l'acceptation dans la relation humain-technologie pour entrer dans le stade du « penser et vivre avec la technologie ». En effet, la relation humain-technologie est désormais marquée par la très grande proximité, de l'ordre de l'intime, dans une « forte dépendance mutuelle » (Brangier et coll., 2009) : l'homme et la machine co-évoluent désormais en symbiose.

2. *Vers des systèmes symbiotiques*

Licklider (1960) a émis l'idée de systèmes informatiques symbiotiques dans un article fondateur, à une époque où l'ordinateur était exploité pour sa fonction de calcul : il imagine un système informatique capable d'assister l'humain dans son processus de réflexion, pour la résolution de problèmes complexes. Il y développe un parallèle avec la biologie : deux organismes **différents** et vivants (les symbiotes) sont en symbiose lorsqu'ils s'associent étroitement et durablement, tirant un avantage réciproque de cette association[1].

Figure 2
Relation symbiotique homme-machine



Prolongeant l'analogie avec les espèces vivantes et en l'appliquant à la relation humain-technologie, l'humain tire un avantage des technologies car elles en constituent son prolongement en l'assistant, en augmentant ses capacités, tout en modifiant ses usages (Brangier et coll. 2009) (voir Figure

[1] Définition tirée du dictionnaire Larousse 2013.

2). De manière réciproque, la technologie devient plus intelligente par la transposition dans l'ordinateur de « fonctionnalités humaines » (« ce qui est de lui-même programmable ») et la redéfinition bien pensée de l'interface selon les nouveaux usages (voir Figure 2). Un tel système serait doté de fonctions d'intelligence, notamment de perception, d'interprétation, de décision et d'action (Suganuma et coll., 2007).

L'informatique nuagique (*Cloud Computing*) et la visualisation pour traiter des problèmes liés, par exemple, au traitement des données massives (*Big Data*) font écho à cette vision. Pour permettre cette symbiose, Licklider souligne également le besoin d'avoir des langages et des dispositifs d'entrée et de sortie adaptés pour aider aux échanges entre l'humain et la machine : il y préfigure déjà l'IHM en termes de modalités d'interaction (visuelle, reconnaissance de la parole, reconnaissance de l'écriture manuscrite), d'interaction multimodale, et de dispositifs d'interaction comme les surfaces interactives.

Depuis Licklider, la relation humain-technologie s'opère désormais dans une multitude de contextes, souvent imprévisibles. Si l'on se réfère de nouveau à l'évolution des espèces selon J. Grudin (voir Figure 1), du fait de la miniaturisation, nous sommes passés d'une technologie informatique monolithique et inamovible à un choix croissant de plateformes informatiques de plus en plus riches, puissantes et disponibles en tout lieu et à tout moment. En conséquence, la relation humain-technologie n'est plus figée à un seul contexte d'usage : l'humain est amené à connaître diverses expériences d'interaction avec des préférences variant suivant les contextes d'usage.

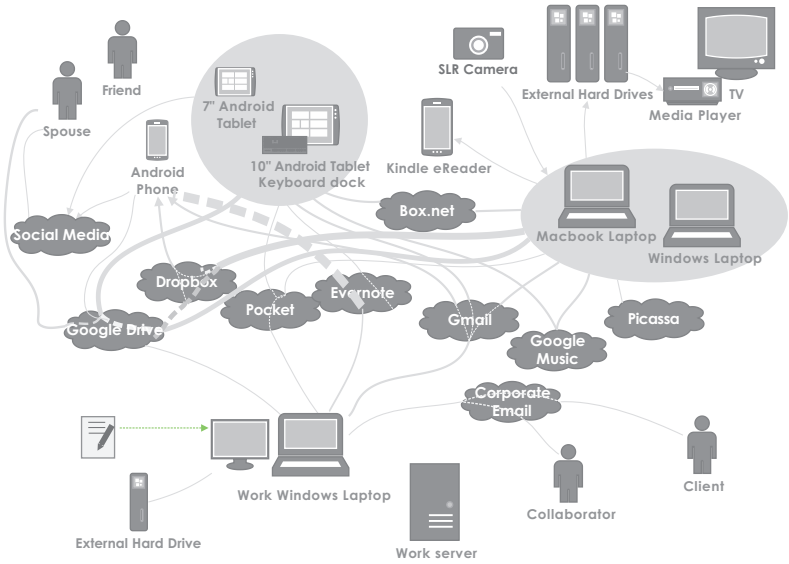


Figure 3
Exemple d'écosystème de dispositifs, de services et d'interactions (tiré de (Santosa et Wigdor, 2013))

En effet, comme le montre Santosa et Wigdor (2013), illustré par la Figure 3, nous disposons d'un large choix de plateformes informatiques allant de l'ordinateur de bureau à la montre intelligente, en passant par le Smartphone, le GPS de la voiture et les tables interactives. Parfois même, nous interagissons avec plusieurs dispositifs simultanément. L'environnement physique et social change également, que nous soyons dans le train entourés d'inconnus, à la maison avec nos proches confortablement installés dans un canapé, au bureau en réunion avec des collègues ou dans la rue en marchant avec des amis, etc. Notre état interne

(cognitif, moral, conatif, etc.) influe également sur la relation avec la technologie que l'on soit dans une situation de stress ou de calme, de grande fatigue ou d'excitation, etc. Comme le souligne Coutaz et Nigay (Coutaz et Nigay, 2012), cette perspective met en évidence que les interactions humain-machine ne peuvent plus être conçues pour un contexte d'usage unique et bien identifié (c'est-à-dire, la station de travail à son bureau) et qu'il convient de concevoir des IHM « distribuées, nomades, multimodales et plastiques ». En particulier, la plasticité d'une IHM « dénote sa capacité d'adaptation au contexte d'usage dans le respect de sa valeur pour l'humain » (Calvary, 2007). Cette capacité d'adaptation à l'humain est justement une des propriétés des systèmes symbiotiques, renforçant la théorie de l'évolution selon J. Grudin (2012) que ces derniers sont une évolution des systèmes interactifs.

3. Mutation des collecticiels

L'évolution vers des systèmes informatiques symbiotiques concerne également les collecticiels. Selon Suganuma et coll. (2007), un système informatique symbiotique est une évolution de l'informatique ubiquitaire couvrant, comme nouvelle dimension, l'aspect social de la relation humain-technologie, traitée par les apports dans le domaine de l'informatique sociale (*social computing*). De tels systèmes doivent être dotés de fonctions capables de reconnaître et de comprendre les activités des utilisateurs dans une société ainsi que de reconnaître et de comprendre les activités de groupe, et de développer des mécanismes d'adaptation conjointement pour l'individu et pour le groupe. J. Grudin (2010) pose le même constat d'une telle mutation et de l'importance croissante de l'aspect social. D'ailleurs, la conférence CSCW, conférence phare de la communauté TCAO, s'intitule maintenant « Computer-Supported Cooperative Work and **Social Computing** ».

A l'origine, l'intérêt pour les collecticiels (*groupware* en anglais), c'est-à-dire des logiciels pour le groupe permettant à plusieurs utilisateurs d'interagir dans un environnement numérique partagé, a émergé au début des années 80 dans une optique prioritairement professionnelle. Il s'agissait avant tout de répondre au besoin d'assister les activités professionnelles (i.e. *Work*) au sein des entreprises à l'aide des technologies informatiques. Le contexte d'usage était alors bien identifié, voire figé, en termes d'activités (c'est-à-dire, le contexte professionnel), d'acteurs et de rôles (par exemple, à travers une structure hiérarchique reflétant l'organisation), de dispositifs d'interaction et d'interfaces utilisateur (par exemple, un ordinateur de bureau avec un client Xwindows), de connectivité (par exemple, une connexion à un serveur mainframe), et de lieu (par exemple, à son bureau, sur son lieu de travail, dans une salle de réunion).

Désormais, complétant le constat initié dans la partie 1, les contextes d'interaction multiutilisateur deviennent imprévisibles pour plusieurs raisons, ce que J. Grudin (2010) résume en revisitant l'acronyme CSCW :

C : « **Computers** are no longer the only digital device of interest. »

S : « Digital technology is no longer confined to a **Support** role; it is integral to many activities. »

C : « The focus was initially on small groups for which **Cooperation** was the norm, but today's world features hacker attacks, spam, privacy concerns, conflict, and competition. »

W : « In 1985 systems capable of supporting groups were mainly affordable in corporate **Work** setting. It's different now. »

En résumé, ce foisonnement de dispositifs informatiques a profondément changé nos usages pour faire de la technologie partie intégrante de notre quotidien. En contrepartie, l'expérience utilisateur s'en trouve fortement fragmentée, ce que Santosa et Wigdor (2013) mettent clairement en évidence, (voir Figure 3), les contextes d'usage devenant de plus en plus imprévisibles et affaire d'opportunités. Plus encore, l'hyperconnectivé qui nous maintient en contact permanent avec autrui conduit à une déstructuration des schémas organisationnels habituels (par exemple, hiérarchique), gommant les frontières entre nos différentes sphères d'activité : individuelle, personnelle et professionnelle. J. Grudin (2010) le formule ainsi : « **barriers between work and non-work activities are even getting fuzzier** ».

Aussi, cette dichotomie entre systèmes interactifs mono-utilisateur et multiutilisateur n'est plus aussi nette (« fuzzier ») qu'auparavant. Les systèmes interactifs tendent à devenir polyvalents et polymorphes, capables d'assurer une interaction utilisateur au caractère imprévisible et de plus en plus mouvante, en un mot : fluide.

4. *Fluidité interactionnelle, un requis*

La fluidité exprime le caractère de ce qui est changeant et, pour un fluide, de ce qui est capable de s'adapter à son contenant. L'opposé est la viscosité et désigne la résistance au changement. Cette dernière (*viscosity*) est régulièrement employée dans le champ de l'interaction homme-machine pour désigner un excès d'actions inutiles induites pour accomplir une tâche donnée (Green, 1990), ainsi que dans le champ du TCAO pour désigner l'effet induit d'une action sur l'activité d'autrui (*distributed viscosity*), se traduisant par une surcharge de travail, inutile dans l'accomplissement de la tâche en cours (Rogers, 1994).

En appliquant cette propriété aux systèmes interactifs symbiotiques, je définis la **fluidité interactionnelle** comme la capacité des systèmes interactifs à mobiliser leurs **polyvalence** et **polymorphisme** pour des adaptations assurant une interaction cohérente en tout contexte d'usage. La fluidité est ainsi une propriété de la plasticité (Calvary, 2007). Elle constitue le thème fédérateur de mes recherches ici synthétisées.

Comment assurer et mettre en œuvre ce requis de fluidité interactionnelle est la question centrale à mes travaux. Précisément, du point de vue de l'ingénierie des systèmes interactifs, il est besoin d'outils permettant aux concepteurs de l'interaction et aux développeurs de l'application de satisfaire ce requis de fluidité interactionnelle. Or, comme le constate Olsen (2007), nos outils et modèles de conception ne sont plus adaptés aux nouvelles pratiques et architectures : « **Many people live and work across many platforms and interact with many people, yet our UI systems architectures support none of this**. » Selon Olsen (2007), la solution réside d'abord par fournir des outils de conception, comme les architectures logicielles, aux bonnes propriétés, dont la flexibilité.

Pour cela, d'un point de vue système, j'aborde la question du requis de **fluidité interactionnelle** en terme de **flexibilité** au travers d'outils de conception.

5. *Structure du mémoire*

Afin de positionner ma démarche de recherche et mes contributions, le chapitre suivant caractérise les systèmes symbiotiques et identifie plusieurs dimensions pour traiter la fluidité interactionnelle, mettant ainsi en évidence des questions de recherche. Le troisième chapitre présente mes contributions pour traiter le requis de fluidité par différentes approches. Précisément, mes travaux relevant du domaine de l'ingénierie des systèmes interactifs, j'exprime la fluidité d'un point de vue système sous l'angle de la flexibilité, par des propositions d'outils de spécification de l'interaction, de conception logicielle et de réalisations logicielles, couvrant les dimensions identifiées dans le second chapitre. Le dernier chapitre ouvre sur de nombreuses pistes de recherche, prolongeant les travaux que j'ai ainsi entrepris.

En annexe, je propose une sélection d'articles cités dans ce document, couvrant une grande partie de mes travaux.

Lignes de recherche

Dimensions de la fluidité interactionnelle

1. Systèmes symbiotiques

J.C.R Licklider (1960) a émis l'idée des systèmes symbiotiques et d'une relation symbiotique entre l'homme et la machine en 1960, prédisant que l'informatique permettra aux individus de réfléchir [à la résolution de problèmes complexes] en interaction avec un ordinateur, de la même façon que l'on cherche à résoudre un problème de concert avec un collègue de compétence complémentaire : « to think in interaction with a computer in the same way that you think with a colleague whose competence supplements your own ». Ainsi, cette vision considère un ordinateur amplifiant l'intelligence de l'humain, en l'assistant dans des tâches complexes, et le suppléant en le déchargeant sur le plan cognitif grâce, notamment, à une interaction « naturelle » et multimodale (reconnaissance et compréhension de l'écriture, reconnaissance et synthèse vocale, etc.).

En effet, pour Brangier et coll. (2009), cette amplification relève en grande partie de la simplification de l'interaction par des « interfaces intelligentes » qui, au delà des questions d'ergonomie, « permettent de suppléer aux cognitions humaines, lorsque l'opérateur se trouve sursollicité par des interfaces dont le traitement cognitif n'est plus possible ». Il convient alors de concevoir des outils, c'est-à-dire des systèmes symbiotiques, « aidant l'opérateur dans sa tâche, parfois à son insu » (Brangier et coll., 2009).

Plusieurs critères sont proposés pour caractériser les systèmes symbiotiques selon une vision centrée utilisateur (Brangier et coll., 2009) :

Amplification de l'intelligence. Il s'agit des moyens disponibles pour augmenter les capacités cognitives de l'utilisateur grâce à des mécanismes permettant de traiter et manipuler de grandes quantités d'information, de raisonner et de prendre des décisions.

Augmentation perceptive. Il s'agit de traiter l'information sensorielle contribuant à une meilleure exploitation de l'information, notamment grâce à la

multiplicité des présentations ou visualisations, adaptées au contexte d'usage (utilisateur et tâche notamment).

Accélérateur opératoire. Il s'agit de rendre plus efficace l'interaction entre l'utilisateur et le système, par simplification de l'interaction voire par la suppression de modes d'interaction. La sollicitation de plusieurs canaux par une interaction multimodale couplée à une adaptation à l'utilisateur contribue à satisfaire ce critère.

Management des connaissances en contexte. Il s'agit de capitaliser, traiter et partager des connaissances constituant un bien commun, en lien avec le contexte social.

Équilibrage émotionnel. Il s'agit d'adapter l'interaction en fonction de l'état émotionnel des utilisateurs, notamment pour limiter les frustrations induites par l'interaction.

Résilience dans la gestion des erreurs. Il s'agit des capacités du système à récupérer, de manière autonome, d'erreurs de fonctionnement et à conserver un comportement stable en cas de perturbations.

Réduction des éléments distracteurs. Il s'agit de focaliser l'attention de l'utilisateur sur sa tâche, par l'usage de filtres intelligents par exemple, et de réduire les sources de distraction (par exemple, des publicités).

Continuité du flux informationnel. Il s'agit d'assurer une équité dans l'accès et la qualité de l'information quelque soient les dispositifs d'interaction et moyens de transmission, grâce à des mécanismes de synchronisation et d'adaptation.

Ces différents critères mettent en évidence que la conception d'un système symbiotique repose sur l'implication de nombreuses disciplines[1] : apprentissage (*machine learning*), informatique perceptuelle (*perceptual computing*), informatique ubiquitaire (*ubiquitous computing*), fouille de données (*data mining*), capteurs intelligents, neuroscience, psychologie, traitement du signal, biomécanique, anthropologie, interaction homme-machine, etc. Par exemple, en systèmes multi-agents, Shiratori et coll. (Shiratori et coll., 2012) identifient un système symbiotique comme un pont entre les mondes physique et numérique, et peuplé d'agents autonomes. Cette approche par agents converge avec la vision développée dans le domaine de l'informatique organique (*organic computing*) (Würtz et coll., 2010) qui transpose les principes de la biologie cellulaire aux systèmes multiagents.

Il existe donc plusieurs facettes du système symbiotique qui peuvent être traitées d'un point de vue purement logiciel. C'est ce type d'approche qui est développée dans le domaine de l'informatique organique et autonomique (Würtz et coll., 2010). Dans mes travaux, je m'intéresse à l'aspect interactionnel de la relation entre les utilisateurs et le système symbiotique pour identifier différentes formes de fluidité interactionnelle : c'est la vision promue par Brangier et Licklider.

[1] Voir l'appel à soumission pour le "International Workshop on Symbiotic Interaction".

2. *Architecture d'un système symbiotique*

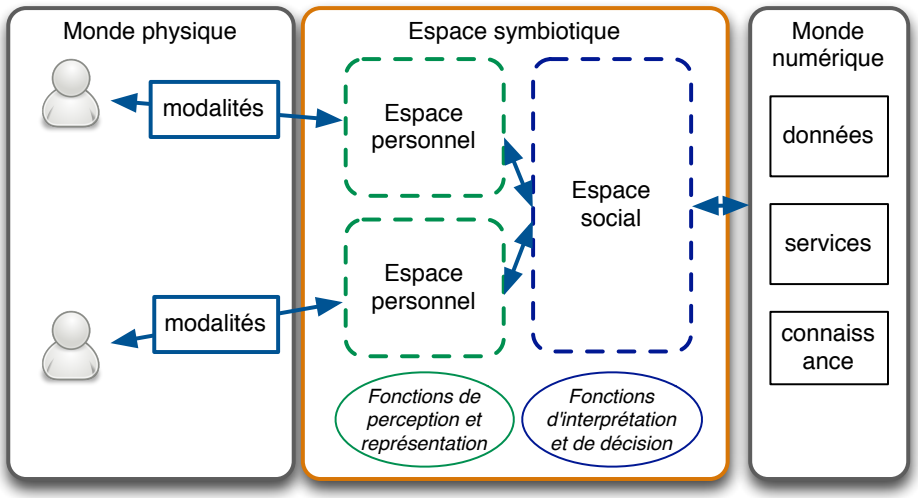


Figure 1
Architecture d'un système symbiotique (tiré et traduit de (Shiratori et coll., 2012))

Pour concevoir un système symbiotique, Shiratori et coll. (2012) proposent une architecture composée de deux couches distinctes (voir le bloc au centre de la Figure 1) formant l'**espace symbiotique** (*symbiotic space*), s'intercalant entre le monde physique (*real space*) dans lequel évoluent les utilisateurs (voir le bloc à gauche de la Figure 1), et le monde numérique (*digital space*) (voir le bloc à droite de la Figure 1). Ce dernier relève du domaine métier (c'est-à-dire du niveau fonctionnel des systèmes informatiques usuels) et héberge des données, des services et des bases de connaissances. L'espace symbiotique est donc une nouvelle forme d'interface utilisateur reposant sur la constitution de relations symbiotiques entre les utilisateurs (monde physique) et le système informatique traditionnel (monde numérique). Selon les auteurs (Shiratori et coll., 2012 ; Suganuma et coll., 2007), sa mise en œuvre implique un système doté de mécanismes de perception, d'interprétation, de décision et d'action, reposant sur des bases de connaissance. Il s'agit notamment de reconnaître l'utilisateur, ses caractéristiques ainsi que l'environnement social et physique. Il s'agit également de lui proposer et lui présenter des informations, connaissances et services du monde numérique, intelligibles et adaptés.

Interface entre physique et numérique : espace personnel, espace social

La première couche (Shiratori et coll., 2012 ; Suganuma et coll., 2007) (voir la partie gauche du bloc central sur la Figure 1) est une agrégation d'*espaces personnels* : chaque espace interagit avec l'utilisateur qui lui est associé, et embarque des classes de fonctionnalités pour percevoir cet utilisateur (système perceptif) et son contexte d'usage, c'est-à-dire les observer et les reconnaître : actions, comportement, environnement, préférences, motivations, activité sociale, etc. Cet espace intègre également des classes de fonctionnalités pour représenter à l'utilisateur des informations et moyens d'action sur l'environnement (système moteur). Ces deux classes de fonctions (perception et représentation) sont les deux classes fondatrices de l'interaction homme-machine : action et réaction. Cette couche a aussi la charge de gérer l'état de l'interaction avec l'utilisateur, de mémoriser ses préférences ainsi que ce qui est perçu de son état interne (par exemple, cognitif, émotionnel). Considérant le système symbiotique dans son ensemble, cette perception n'est pas centrée sur un unique utilisateur mais concerne un ensemble d'utilisateurs : le système symbiotique dépasse le

cadre classique du système interactif mono-utilisateur fondé sur une informatique monolithique : il dispose de capacités pour permettre des interactions multiples pour de multiples utilisateurs, qu'ils interagissent entre eux ou non.

La seconde couche, nommée espace social, embarque des fonctionnalités d'interprétation et de décision en fonction de ce qui est perçu, d'une part, individuellement de chaque utilisateur (informations issues de la première couche, l'espace personnel) et, d'autre part, de la connaissance de l'environnement social des utilisateurs. Ceci fait appel à des fonctions de reconnaissance et d'interprétation des activités, intentions et actions en société, sur la base de modèles d'intelligence sociale et individuelle. Ainsi, cette couche augmente l'information captée par le système perceptif avec des structures sociales, des normes et du sens commun. Par ailleurs, cet espace est situé à la frontière avec le montre numérique : aussi il augmente et donne du sens d'ordre social (par exemple en relation avec un travail de groupe) aux données, services et connaissances du monde numérique (Shiratori et coll., 2012). Selon les auteurs (Shiratori et coll., 2012 ; Suganuma et coll., 2007), cette extraction de connaissances peut reposer, par exemple, sur des techniques de fouille de données (*data mining*) appliquées à de l'informatique sociale (*social computing*).

Finalement, cette vision en deux couches offre une analogie avec le modèle humain puisque la couche en relation avec le monde physique s'inspire du système perceptif et moteur, tandis que la seconde s'inspire du système cognitif et décisionnel, en lien avec le système de mémorisation.

Relations symbiotiques

Dans le modèle développé par Shiratori et coll. (2012), plusieurs formes de relations symbiotiques apparaissent, dressant des ponts entre le monde physique, le monde numérique, et les deux espaces constituant l'espace symbiotique. Ces relations sont vouées à évoluer, c'est-à-dire à apparaître, disparaître et se transformer au cours du temps.

Une première forme de relations émerge des échanges entre les utilisateurs, le monde physique et la couche personnelle. Ce type de relation repose sur un système de perception et de représentation riche, augmentant l'utilisateur en lui facilitant l'interaction avec le monde numérique (tendant vers des interfaces dites « naturelles » (NUI, N*atural User Interface*), c'est-à-dire une interaction directe et intuitive). L'expression de telles relations se caractérise d'abord par un système symbiotique offrant un large spectre de modalités d'interaction. Ceci est rendu possible grâce à une technologie foisonnante débouchant sur une grande diversité de dispositifs d'interaction dotés d'une informatique embarquée puissante bardée de capteurs divers, de plus en plus miniatures et mobiles. De plus, cette technologie est désormais capable de capter l'humain et son environnement (gestes, expressions, position dans l'espace, localisation, état de stress, etc.), permettant un contrôle de plus en plus fin de l'utilisateur (comme l'interaction cerveau-machine ; BCI, *Brain-Computer Interaction*). Aussi, le système symbiotique affine, améliore son système perceptif en accumulant des données sur l'utilisateur et son environnement, aboutissant à une perception plus fine et robuste, tendant vers une plus grande fiabilité. Ces formes de relation relèvent de la propriété d'augmentation perceptive de la relation symbiotique humain-machine d'après Brangier et coll. (2009).

Une seconde forme de relations apparaît au sein même de l'espace symbiotique, relations entre l'espace personnel et l'espace social. Ces relations reposent sur les capacités d'interprétation et de décision du système symbiotique, notamment envers l'environnement social. En effet, les informations perçues et transmises par l'espace personnel à l'espace social sont augmentées avec du sens sur la base de connaissances acquises et de moyens de traitement, et de mise en contexte social par une capacité à reconnaître les activités individuelles et de groupe. Grâce à cette capacité, le système symbiotique est en mesure de gérer des communautés de pratiques et des activités de groupe, de favoriser le partage et les échanges au sein de réseaux sociaux (Brangier et coll., 2009). Ceci est rendu possible grâce à la capacité de l'espace personnel de percevoir les actions des individus et le comportement des groupes (Suganuma et coll., 2009).

La troisième forme de relation symbiotique porte sur les relations entre l'espace social et le monde numérique. Selon les auteurs (Shiratori et coll., 2012), ces relations augmentent le monde numérique par extraction et reconnaissance d'activités d'ordre social (par exemple, appliquer des mécanismes d'analyse de traces ou de fouille de données pour détecter une activité collaborative). Dans la suite, les préoccupations (analyse de traces, fouille de données, etc.) soulevées par ces relations étant éloignées de l'interaction utilisateur, je me concentre sur les deux premières formes de relations symbiotiques.

3. *Dimensions de la fluidité interactionnelle*

Les systèmes symbiotiques et leur conception soulèvent de nombreuses questions relevant de plusieurs domaines de recherche comme l'informatique autonomique (systèmes multi-agents et organiques). Dans mes travaux, j'adopte une démarche centrée utilisateur. Je me focalise sur les questions relatives à l'interaction de l'utilisateur et des utilisateurs avec les systèmes symbiotiques.

Précisément, en m'appuyant sur l'architecture des systèmes symbiotiques telle que décrite dans la partie 2, j'identifie trois dimensions pour caractériser la fluidité interactionnelle :

- Les relations entre utilisateurs et espace personnel ;
- Les relations entre espace personnel et espace social ;
- Les contextes d'usage liés à ces relations.

3.1. MODALITÉS D'INTERACTION ET MULTIMODALITÉ

La première dimension est relative à la relation immédiate entre l'utilisateur et le système. Elle est liée aux modalités d'interaction qui permettent aux utilisateurs d'agir sur le système, c'est-à-dire de communiquer avec le système pour réaliser une action, et de percevoir le résultat de cette action. Pour ce faire, le système doit être capable d'agir sur l'environnement via des dispositifs techniques qui rendent perceptible le changement d'état suite à cette action. Aussi, comme le soulignent Brangier et Hammes-Adelé (2011), le caractère symbiotique de la relation homme-technologie repose sur un sentiment fort que la technologie constitue un prolongement naturel de nous-mêmes, que l'on peut la maîtriser, et qu'elle est simple et efficace. Ces

perceptions contribuent alors à accroître le sentiment d'utilité de la technologie.

Les modalités d'interaction mises à disposition des utilisateurs par le système jouent donc un rôle prépondérant dans l'établissement de la relation symbiotique, ce que mettait en évidence Licklider (1960) dès les années 60. Leur disponibilité, leur robustesse et fiabilité, leur facilité d'usage, ainsi que les mécanismes de perception sous-jacents sont autant de qualités nécessaires à assurer pour concevoir un système symbiotique. En effet, reprenant les constats posés par Brangier et Hammes-Adelé (2011), développer l'impression que la technologie est un prolongement naturel de l'utilisateur et qu'elle confère des sentiments de simplicité et de maîtrise incite à concevoir et mettre en œuvre des modalités d'interaction au plus proche de l'utilisateur. C'est dans cette tendance que s'inscrivent les recherches dans le domaine de l'interaction gestuelle et l'interaction tangible, sur les surfaces interactives, tendant à favoriser la manipulation directe, ancrée dans le monde physique (on parle également d'interfaces utilisateurs naturelles (NUI, *Natural User Interfaces*)). Cette tendance inclut les modalités d'interaction de nouvelle génération captant à la source l'activité physiologique de l'utilisateur comme les mesures de l'activité cérébrale via les interfaces BCI (*Brain-Computer Interfaces* : interfaces reposant sur le traitement de signaux EEG (ElectroEncephaloGraphique)) ou comme les mesures de l'activité musculaire (reposant sur l'analyse des signaux EMG (ElectroMyoGraphique)). Par exemple, le projet MindSee (projet Européen FP7) aborde la question de la relation symbiotique homme-machine sous l'angle de la recherche d'information en s'appuyant sur des modalités de type BCI. Il vise à augmenter les performances de l'utilisateur pour réaliser des requêtes complexes sur des bases de données contenant plus de 50 millions de documents. Autre exemple, le projet Symbitron (projet Européen FP7) aborde la relation symbiotique homme-machine dans un contexte médical sur la question du contrôle d'exosquelettes à destination de personnes souffrant de paralysie.

La qualité des modalités d'interaction et leur diversité contribuent à satisfaire une propriété des systèmes symbiotiques, à savoir être un accélérateur opératoire en simplifiant, en accélérant ou supprimant des modes d'interaction (Brangier et coll., 2009). En particulier, l'interaction multimodale est un moyen pour y parvenir, notamment « en **s'adaptant à divers utilisateurs**, à plusieurs possibilités d'interactions opératoires et en sollicitant plusieurs canaux sensori-moteurs » (Brangier et coll., 2009).

Une autre question émerge relative à l'adaptation à divers utilisateurs dans le cas d'une interaction multimodale multiutilisateur. La multimodalité doit alors être considérée globalement, c'est-à-dire du point de vue de plusieurs utilisateurs établissant une relation symbiotique avec le système. Apparaît alors la question de l'hétérogénéité interactionnelle ainsi que du manque d'équité en matière de ressources interactionnelles entre les différents utilisateurs.

3.2. INTERACTION UTILISATEUR ET MULTIUTILISATEUR

J'identifie deux classes de questions sur les relations symbiotiques pouvant s'établir entre l'espace personnel et l'espace social : la structuration des relations entre ces deux espaces (comment) ; et l'environnement social de ces relations (pourquoi). Précisément, la seconde dimension porte sur la

question de la structuration de relations en lien avec l'espace social (la question du contexte est une question transversale qui se pose également pour les relations entre utilisateurs et espace personnel, et abordée dans la partie suivante précisant la troisième dimension).

Se référant au modèle des systèmes symbiotiques (voir partie 2) (Shiratori et coll., 2012), un tel système rompt la dichotomie qui distingue les systèmes interactifs mono-utilisateur et multiutilisateurs. À l'inverse, ce modèle montre qu'il les juxtapose voire les compose suivant la nature et la structure des relations qui se nouent entre espace personnel et espace social. Notamment, le modèle préconise une structuration pour refléter des normes sociales, organisationnelles, etc.

Toutefois, un tel modèle induit une très grande variété de modes d'interaction « sociale » conciliant interaction mono-utilisateur et multiutilisateur. En effet, la dimension sociale peut être périphérique à une activité mono-utilisateur comme les filtres collaboratifs (personnalisation d'une interaction mono-utilisateur en fonction d'une capitalisation de comportements (par exemple, les systèmes de recommandations des sites de vente en ligne)) (Brangier et coll., 2009), informelle au travers des réseaux sociaux, ou fortement coopérative au travers de collecticiels métiers par exemple. Ce type de modèle nécessite donc un besoin pour des systèmes suffisamment flexibles pour traiter cette juxtaposition et permettre leur mise en œuvre avec divers degrés d'implication des utilisateurs dans une activité « sociale ».

Plus encore, alors que les collecticiels embarquent un modèle social et organisationnel sur lesquels l'interaction multiutilisateur est forgée, le système symbiotique laisse l'opportunité aux utilisateurs de s'organiser. Dès lors le système doit être conçu pour anticiper la constitution opportuniste de relations dans l'espace social, en fonction des choix d'utilisation : « les utilisateurs orienteraient leurs choix d'utilisation si ce dernier [le système symbiotique] favorise les coopérations sociales » (Brangier et coll., 2009).

3.3. MULTIPLICATION DES CONTEXTES D'USAGE

La troisième dimension est transversale aux deux autres car elle concerne l'état de l'ensemble des relations établies pour un utilisateur entre le monde physique, l'espace personnel et l'espace social. Cet état reflète le contexte d'usage de l'interaction entre l'utilisateur avec le système symbiotique et entre les utilisateurs. Il reflète également ce que le système connaît de l'utilisateur (ses préférences, ses motivations, son activité) et ce que le système connaît de l'environnement social dans lequel se déroulent les activités des utilisateurs (les productions d'un groupe, les traces d'activité, etc) (Shiratori et coll., 2012). Ceci est lié à la capacité du système symbiotique à appréhender le contexte d'usage et à s'adapter aux usages : « les actions nécessaires à l'obtention d'un résultat de la technologie peuvent être adaptées en fonction du contexte d'usage et ainsi générer un effet [dé]multipliant l'efficacité des actions de l'utilisateur (accélérateur opératoire) » (Brangier et coll., 2009).

Cependant, comme observé en introduction, contrairement à une informatique monolithique, le contexte d'usage n'est plus unique mais, d'une part, recouvre plusieurs facettes, et, d'autre part, est multiple : multiplication des dispositifs d'interaction aux formes, modalités

d'interaction et ressources variées ; mobilité des utilisateurs ; disponibilité de ressources à la demande (*Big Data*, *Cloud Computing*) ; etc.

Cette multiplicité est également liée à l'importance croissante de la dimension sociale du contexte dans l'interaction utilisateur. Cet aspect soulève plusieurs questions.

D'une part, dans la relation entre les utilisateurs avec l'espace personnel, le système symbiotique doit être conçu pour traiter la diversité des plateformes et dispositifs d'interaction entre utilisateurs interagissant avec le système symbiotique. Dans le cas d'une interaction multiutilisateur, l'hypothèse implicite est une interaction reposant sur une homogénéité des plateformes, ce qui est de moins en moins vérifié. Ce manque d'homogénéité peut constituer un frein à l'activité dans l'environnement social.

D'autre part, dans la relation entre l'espace personnel et l'espace social, le système symbiotique se doit d'adapter l'interaction en fonction de l'utilisateur, de ses préférences, et en relation avec le contexte social. Il y a donc nécessité de différencier les utilisateurs comme fondements de cette adaptation et nécessité d'identifier et spécifier des modes d'interaction sur cette base.

De cette question du contexte et de l'adaptation au contexte d'usage, se pose également la question de l'observabilité de ces changements sur deux plans : cela concerne les changement opérés par le système symbiotique ; cela concerne également l'observabilité des actions des autres utilisateurs conduisant également à un changement de contexte. Dès lors, dans quelle mesure l'utilisateur dispose-t-il de moyens pour observer les conséquences de ses actions et contrôler ce que les autres utilisateurs en perçoivent du fait des relations nouées dans l'espace social ?

3.4. SYNTHÈSE

Sur la base de l'architecture d'un système symbiotique, trois dimensions émergent pour caractériser la fluidité interactionnelle :

- Du monomodal au multimodal ;
- Du mono-utilisateur au multiutilisateur ;
- Du monocontexte au multicontexte.

Le chapitre suivant apporte une réponse unificatrice à l'ingénierie de ces dimensions : la flexibilité logicielle.

Ingénierie de la fluidité interactionnelle

Requis de flexibilité

L'objectif de mes travaux est de proposer des moyens, tels des outils de spécification ou de conception logicielle, afin d'assurer le requis de fluidité interactionnelle pour les systèmes symbiotiques. J'organise mes contributions selon les trois dimensions de la fluidité :

- Du mono-utilisateur au multiutilisateur,
- Du monomodal au multimodal,
- Du monocontexte au multicontexte.

Pour mener ces recherches, j'adopte une méthode multifacette.

D'une part, du point de vue du cycle de vie logicielle, mes travaux abordent à la fois la **spécification de l'interaction** et la **conception logicielle** du système interactif. Aussi, mes contributions portent autant sur des outils conceptuels pour spécifier l'interaction, comme les notations, que les architectures et plateformes logicielles.

D'autre part, mes travaux étudient les différentes fonctions identifiées par le modèle ARCH (Bass et coll., 1992), débouchant sur un éventail de contributions qui s'étend du noyau fonctionnel (par exemple, l'architecture et la plateforme Clover) à l'interaction concrète (par exemple, via l'étude de l'interaction muscle-machine), incluant le dialogue et l'interaction abstraite (par exemple, la notation COMM).

Enfin, mes recherches sont également guidées par le souci de proposer des prolongements conceptuels harmonieux entre concepts et principes relevant de l'interaction mono-utilisateur et multiutilisateur.

1. Du mono-utilisateur au multiutilisateur

Comme le montre le chapitre caractérisant les dimensions de la fluidité interactionnelle, les systèmes interactifs existants tendent vers une plus grande fluidité interactionnelle. Ceux-ci se caractérisent notamment par des systèmes interactifs capables de juxtaposer et de mettre en œuvre conjointement des interactions liées à des activités individuelles et des interactions liées à des activités de groupes. Il en découle un besoin de systèmes interactifs suffisamment flexibles pour mettre en œuvre cette juxtaposition et offrir différents modes d'interaction s'intégrant harmonieusement, qu'ils soient mono-utilisateur ou multiutilisateurs.

Dans mes travaux de recherche, j'aborde cette dimension sous l'angle des outils de spécification de l'interaction et des outils de conception logicielle au travers des architectures logicielles pour les systèmes interactifs. En effet, satisfaire cet aspect de la flexibilité nécessite de prévoir une telle juxtaposition. Aussi, je constate le besoin d'outils capables d'exprimer cette propriété de flexibilité, à la fois par des outils conceptuels, aidant le concepteur de l'interaction dans sa tâche de spécification, et d'outils logiciels comme les modèles d'architecture logicielle et boîtes à outils afférentes.

Dans cette partie, je présente un ensemble de contributions traitant de la flexibilité sur le plan de la conception de l'interaction en proposant des moyens conceptuels pour satisfaire le requis de fluidité interactionnelle. Puis, la seconde sous-partie aborde la question de la conception logicielle de systèmes interactifs flexibles au travers des architectures logicielles.

1.1. FLEXIBILITÉ DES MODES D'INTERACTION

Dans mes recherches, je traite le requis de flexibilité au travers des outils dédiés à la spécification du dialogue, de l'interaction abstraite et concrète, dans une approche à base de tâches et modèles de tâches.

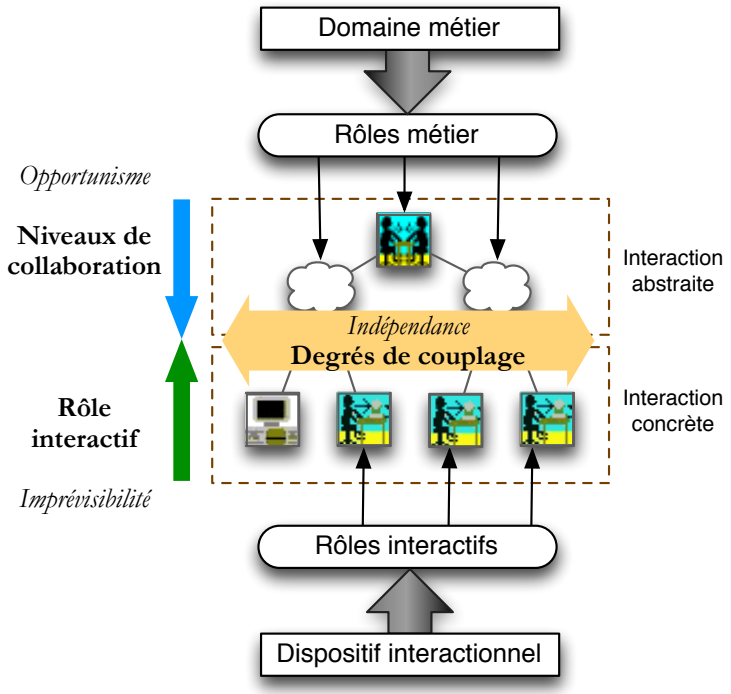
La spécification d'une IHM, qu'elle soit mono-utilisateur ou multiutilisateur, est une démarche bien établie, et de nombreux outils de spécification comme les notations existent afin d'aider le concepteur à spécifier le dialogue, les séquences de l'interaction, les éléments concrets de l'IHM, etc.

Notamment, au-delà de l'analyse de la tâche, les modèles de tâches sont devenus des outils usuels du processus de développement des systèmes interactifs pour spécifier les tâches structurant le dialogue du point de vue du système (van Welie et coll., 1998 ; Calvary et coll., 2003). Comme le montre la Figure 1, un modèle de tâches décrit à la fois l'activité métier dans une version restructurée (suite à une analyse de l'activité ou une étude ethnographique), et traduit et embarque les choix technologiques exprimés en réponse aux besoins idoines (van Welie, 2001 ; Johnson et coll., 1998). Ces derniers sont en grande partie relatifs aux dispositifs interactionnels choisi pour mettre en œuvre l'interaction utilisateur.

J'identifie trois aspects pour traiter ce requis de flexibilité par ces outils (voir Figure 1) :

- **Opportunisme :** considérant une approche descendante du modèle de tâches, guidée par le domaine métier, il s'agit d'offrir une souplesse fonctionnelle dans les rôles métier, c'est-à-dire d'exprimer des **degrés de liberté** en terme de tâche métier (interaction abstraite) pour concilier activité individuelle et activité de groupe, sans nécessairement concourir vers un même objectif.
- **Imprévisibilité :** l'approche inverse (ascendante), guidée par le dispositif technologique interactionnel, vise à offrir une souplesse interactionnelle dans les rôles interactifs, c'est-à-dire à exprimer des degrés de liberté en terme de tâche concrète (interaction concrète) et sur les utilisateurs qui effectivement interagissent avec le système.
- **Indépendance** : transversalement au modèle de tâches, la notion de couplage permet l'expression de degrés de liberté entre tâches. En particulier, mes travaux considèrent le couplage sur le plan de la tâche métier (interaction abstraite) et de la tâche concrète (interaction concrète).

Figure 1
Le modèle de tâches exprime un compromis entre deux classes de requis en tension, notamment via l'expression de rôles : les requis induits par les besoins du domaine métier et les requis induits par le choix de dispositifs interactionnels



Permettre l'opportunisme

À l'origine, les collecticiels ont été conçus pour aider les utilisateurs dans leur activité métier au sein d'une entreprise : « *Cooperative work was the norm* » (Grudin, 2010). Ces outils logiciels accompagnent un processus métier, et suivent une logique organisationnelle reflétant la structure hiérarchique de l'organisation, pouvant s'avérer très contraignant en terme d'usage. Par la suite, Gutwin et Greenberg (1998) posent le constat qu'un collecticiel est voué à l'échec s'il est trop contraignant en ne permettant pas aux utilisateurs et membres d'un groupe d'accomplir à l'écart des tâches individuelles qui, à terme, contribuent à l'effort collectif. Cependant, tâches individuelles et tâches de groupe peuvent nécessiter des requis contradictoires. Aussi, des recommandations sont préconisées (par exemple, proposer des vues radars pour découpler la navigation au sein d'un espace partagé) pour concevoir

des systèmes interactifs multiutilisateurs suffisamment **fluides** pour permettre de balancer entre tâche individuelle, contribuant à l'effort commun, et tâche de groupe. Avec l'apparition de systèmes interactifs multiutilisateurs pour le grand public, notamment avec l'avènement des réseaux sociaux, l'utilisateur n'est plus nécessairement autant impliqué dans l'activité de groupe : il peut viser des objectifs propres.

Ce **degré d'implication** de l'utilisateur dans une activité de groupe apparaît implicitement dans la distinction entre activité coopérative et collaborative, cette dernière offrant un caractère plus **opportuniste** par l'absence de division imposée du travail. Roschelle et Teasley (1995) distinguent ces deux classes d'activités de groupe ainsi :

> ***Cooperative work*** *is accomplished by the division of labour among participants, as an activity where each person is responsible for a portion of the problem solving. [...]* ***Collaboration*** *[is a] mutual engagement of participants in a coordinated effort to solve the problem together.*

Mes travaux sur la navigation sociale (Laurillau, 2002 ; Laurillau et Nigay, 2000 ; Laurillau, 1999) m'ont permis d'affiner cette dichotomie pour aboutir à une échelle de quatre classes d'interaction multiutilisateurs. Celles-ci sont guidées par le **degré d'implication** des utilisateurs dans une activité de groupe, que ce soit pour atteindre un but commun ou individuel :

- *Coopérative.* Cette classe correspond à une structure organisationnelle hiérarchique, avec répartition planifiée des tâches entre acteurs pour atteindre un but commun. Cette classe est conforme à la définition de Roschelle et Teasley. Aussi, l'implication des acteurs est forte et laisse peu de place à l'opportunité.
- *Coordonnée.* Cette classe correspond à des activités de groupe qui ne s'articulent pas sur une structure organisationnelle prédéfinie : les acteurs endossent un rôle qui les place sur un pied d'égalité. Il y a donc un besoin de coordination et d'engagement mutuel. Une plus grande liberté est laissée aux acteurs impliqués dans l'activité de groupe.
- *Opportuniste.* Cette classe correspond à un regroupement d'acteurs aux objectifs différents mais pour un même domaine d'activité. L'activité de groupe est donc totalement opportuniste avec mise en commun ou non d'informations.
- *Conduite.* Cette classe correspond à une activité de groupe dont l'objectif à atteindre est fixé par un acteur mais avec une implication libre des autres acteurs. Dans le cas de la navigation collaborative, cette classe correspond à la notion de visite guidée.

Cette dichotomie se recoupe avec les degrés de couplage identifiés par Tang et coll. (2006) qui, dans leurs travaux sur le couplage de l'interaction autour d'une table interactive, montrent aussi que le couplage est lié au degré d'implication des utilisateurs. En effet, leur échelle, qui compte six degrés, montre que le couplage devient plus faible dès que l'implication des participants se réduit.

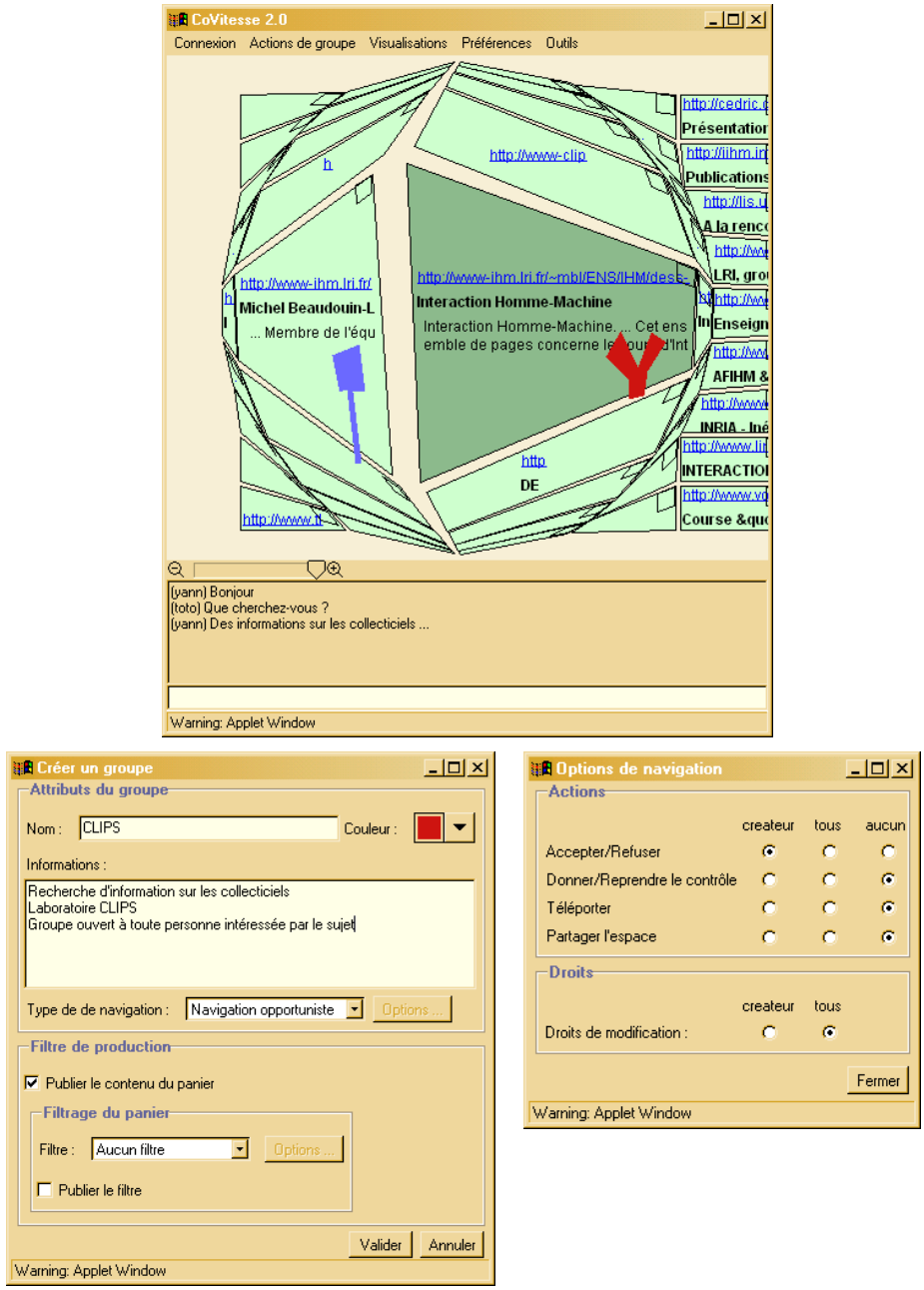
Ces quatre classes m'ont servi de guide pour mettre en œuvre des techniques de navigation sociale sur le web par la réalisation de l'application logicielle CoVitesse (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau et Nigay, 2003; Laurillau, 2002; Laurillau et Nigay, 2000), reposant

sur l'outil Vitesse (Nigay et Vernier, 1998). Cette dernière explore l'usage de techniques de visualisation, comme la vue en œil de poisson, pour faciliter la navigation dans de grands espaces de données. Appliqué à la navigation sociale sur le web, comme le montre la Figure 2 (en haut), chaque tuile verte représente une page web (réponses d'un moteur de recherche à une requête) : des avatars représentent les différents utilisateurs présents dans cet espace (ici, avatars « T » en bleu et « Y » en rouge).

Figure 2
(en haut)
CoVitesse : écran principal ;

(en bas, à gauche)
écran de choix ;

(en bas, à droite)
paramétrage
d'un type de navigation,
c'est-à-dire une classe
d'interaction de groupe



L'application CoVitesse a été conçue et développée pour mettre en œuvre quatre types de navigation collaborative, instanciant les quatre classes d'interaction multiutilisateurs citées précédemment. Comme le montre la Figure 2 (en bas, à gauche), la création d'un groupe permet le choix entre un de ces quatre types : navigation coopérative (structure hiérarchique avec prise de contrôle possible par le responsable du groupe et répartition a priori des tâches à réaliser) ; navigation collaborative (aucune hiérarchie ; structure à plat avec droits identiques entre tous les participants) ;

navigation opportuniste (possibilité de délégation du contrôle de la navigation) ; et visite guidée (contrôle délégué automatiquement au guide mais liberté laissée aux participants pour quitter ou rejoindre le groupe).

Cette classification a servi de guide pour concevoir le modèle de tâches à la base du développement de l'application CoVitesse. Précisément, comme le montre la Figure 2 (en bas, à droite), j'exprime ces quatre types de navigation par des compositions d'actions atomiques (par exemple, accepter un nouveau membre ou téléporter les membres du groupe). De plus, chaque action atomique est associée à un des deux types de rôles ainsi identifiés : le créateur d'un groupe et le membre d'un groupe. Selon le type de navigation, toutes les actions atomiques ne sont pas nécessairement disponibles. Enfin, outre les quatre types de navigation prédéfinis, l'application permet aux utilisateurs de créer un type de navigation collaborative personnalisé sur la base de cet ensemble d'actions atomiques.

Anticiper l'imprévisible

Les travaux menés au cours de la thèse de Frédéric Jourde sur les notations pour la spécification des interactions multiutilisateurs adoptent une approche inverse complémentaire : au travers de la notation COMM (*COllaborative and Multimodal*) (Jourde, 2011 ; Jourde et coll., 2010a ; Jourde et coll., 2010b), s'appuyant sur la notion de rôle métier, la notion de **rôle interactif** est introduite comme moyen d'anticiper l'**imprévisibilité** au sein du modèle de tâches, car induite par le dispositif interactionnel.

La notion de rôle métier[1] est une des notions usuelles employées pour spécifier l'interaction multiutilisateur (par exemple (Van Welie et coll., 1998)). Identifiés en amont par une analyse de la tâche ou une étude ethnographique, le rôle métier reflète la structure du travail du monde réel du point de vue du métier et de la structure organisationnelle du travail (van Welie et coll., 1998). Précisément, l'allocation des tâches aux acteurs repose sur cette notion de **rôle métier**. Ainsi, comme l'illustre la Figure 1, la spécification du modèle de tâches est déduite de l'organisation de l'activité métier, souvent fortement structurée (par un processus métier par exemple), et est guidée par l'ensemble des rôles métiers afférents.

Cependant, mes recherches (Jourde, 2011 ; Jourde et coll., 2008a ; Jourde et coll., 2008b) montrent que les notations existantes offrent peu de flexibilité pour exprimer des tâches multiutilisateurs autres que coopératives du fait de cette imprévisibilité. Pour l'illustrer, considérons une tâche collaborative, nommée Tcollab, sans allocation prédéfinie des rôles métiers aux tâches, consistant à exécuter en séquence une tâche T1 puis une tâche T2, **accomplies par deux rôles métiers distincts**, R1 et R2. L'opportunité est laissée aux acteurs de se coordonner et d'accomplir cette tâche collaborative : l'allocation des tâches aux rôles est réalisée au moment de l'interaction effective des utilisateurs avec le système, en fonction des contraintes induites par le dispositif interactionnel.

[1] L'aspect métier de la notion de rôle est implicite dans la littérature mais fait clairement référence à la position d'un acteur dans une organisation d'un point de vue métier.

Exprimer cette tâche collaborative à l'aide, par exemple, de la notation *Cooperative Concurrent Task Tree* (CCTT) (Paternó, 2000) implique de spécifier toutes les combinaisons possibles d'allocation des tâches aux rôles métiers R1 et R2, à l'aide de deux sous-tâches coopératives, Tcoop.1 et Tcoop.2, formant ainsi la tâche collaborative Tcollab (voir Figure 3 à droite). De plus, comme le montre la Figure 3 (à gauche), il convient également de dupliquer la spécification des deux tâches individuelles T1 et T2 respectivement pour les deux rôles métiers R1 et R2. Le nombre de combinaisons est alors fonction de l'ensemble des rôles métiers, ce qui conduit à une explosion combinatoire du nombre de tâches, et donc à une spécification non compacte et peu flexible.

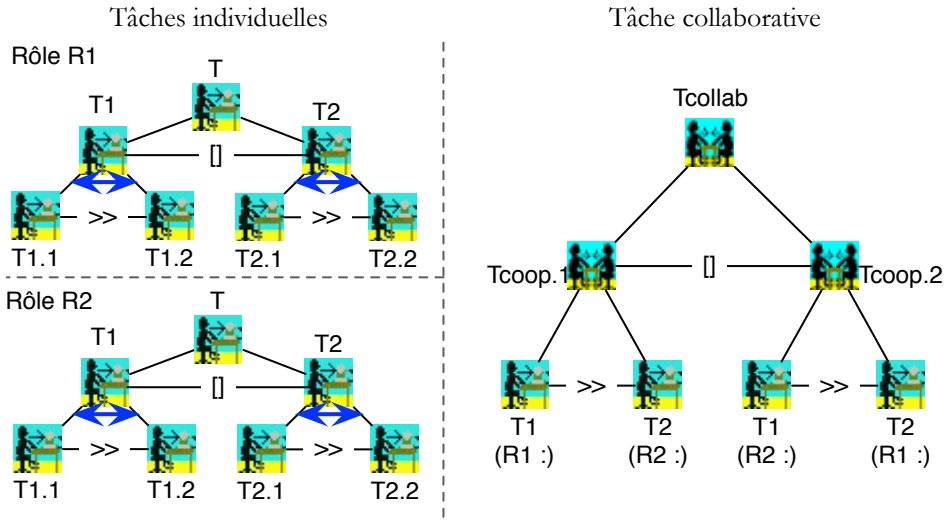


Figure 3
Exemple de tâche collaborative spécifiée à l'aide d'une tâche coopérative avec la notation CCTT :

(à droite) tâches individuelles pour les rôles R1 et R2

(à gauche) tâche coopérative exprimant une tâche collaborative

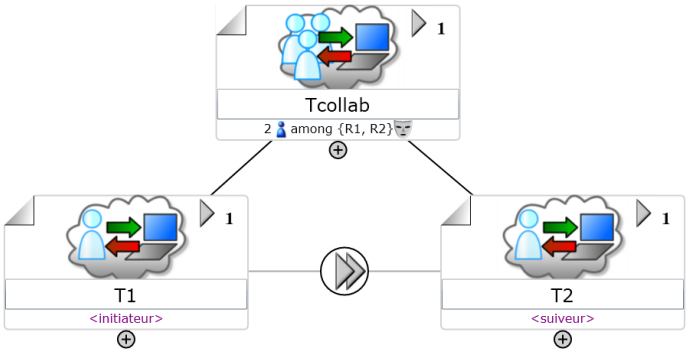
Afin d'apporter plus de flexibilité, la notation COMM introduit la notion de **rôle interactif** pour exprimer l'allocation dynamique des tâches rôles métiers, et des rôles métiers aux acteurs.

Cette notion de rôle interactif désigne un rôle éphémère endossé au moment de l'interaction et est associé à un rôle métier le temps d'accomplir la tâche. Elle répond au caractère dynamique de l'allocation des tâches à des rôles métier. Plus encore, comme je l'aborde dans la partie 2.1, elle permet aussi de préciser l'interaction concrète pour différentes modalités d'interaction. Ainsi, comme l'illustre la Figure 1, alors que le rôle métier exprime la répartition du travail du point de vue de l'activité métier, ce qui correspond à une approche descendante usuelle, le rôle interactif constitue un moyen pour spécifier comment l'activité de groupe est influencée par le contexte d'usage, en particulier en termes de ressources interactionnelles. Autrement dit, ces deux notions de rôles, métier et interactif, explicitent la tension entre les requis d'activité métier et d'interaction concrète.

La notation COMM étend plusieurs concepts de la notation CCTT (*Cooperative Concurrent Task Tree*) (Paternò, 2000). Toutefois, contrairement à la notation CCTT, elle permet la spécification de modèles de tâches uniques pour spécifier conjointement l'interaction mono-utilisateur et multiutilisateurs, combinant tâches individuelles et tâches de groupe. Cette originalité permet de fournir une spécification plus compacte, de spécifier des tâches de nature collaborative, et *in fine* de capter le caractère imprévisible de l'interaction.

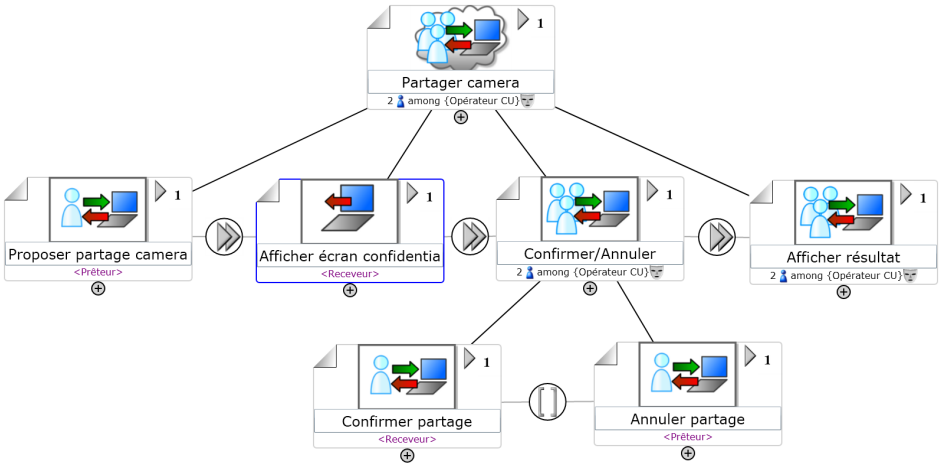
Reprenons l'exemple de la tâche collaborative décrite précédemment en CCTT (voir Figure 3) pour la spécifier à l'aide de la notation COMM : comme le montre la Figure 4, la tâche collaborative Tcollab est associée à deux rôles métiers (R1 et R2), et est composée de deux tâches individuelles accomplies en séquence, T1 et T2. La décoration de la tâche collaborative Tcollab exprime la contrainte que deux rôles métiers distincts, parmi une liste de rôles {R1, R2}, sont nécessaires pour être accomplies. De plus, les deux sous-tâches T1 et T2 étant des tâches individuelles, un seul rôle métier peut y être associé. Aussi, l'**imprévisibilité** de l'allocation des tâches T1 et T2 aux rôles R1 et R2 est spécifiée par la décoration des tâches individuelles par deux **rôles interactifs** distincts, respectivement `<initiateur>` et `<suiveur>`. Ainsi, au moment de l'interaction, un seul des deux rôles métiers R1 et R2 sera `<initiateur>`, et l'autre `<suiveur>`.

Figure 4
Exemple de tâche collaborative reposant sur la notion de rôle interactif, spécifiée avec la notation COMM



Dans le cadre du projet « Facteurs Humains – Partage d'autorité », financé par la DGA (Direction Générale de l'Armement), en partenariat avec des industriels (Bertin technologies, EADS, SAGEM, et PY Automation), la notation COMM a été employée pour spécifier un prototype d'IHM de poste de commande de drones. Ce projet a permis de mettre à l'épreuve la notation COMM et de valider son passage à l'échelle ainsi que son pouvoir expressif (Beaudouin-Lafon, 2004) : plus de 180 tâches ont été spécifiées pour des arbres constitués de branches s'étalant sur 12 niveaux de profondeur. En particulier, la notion de rôle interactif permet l'expression de la délégation de tâches relative au partage d'autorité.

Figure 5
Tâche de délégation de contrôle de charge utile entre opérateurs, exprimée à l'aide de deux rôles interactifs



La Figure 5 illustre un exemple concret tiré de ce projet, spécifié à l'aide de la notation COMM. Le contrôle d'un drone implique plusieurs opérateurs, c'est-à-dire rôles métiers : un opérateur vecteur (VE) pour le contrôle de la trajectoire ; un ou plusieurs opérateurs charge utile (CU) pour le contrôle

des dispositifs embarqués comme une caméra 360°. Le concept de rôle interactif a été utilisé pour spécifier une tâche de partage de charge utile entre deux opérateurs CU. Comme le montre la Figure 5, la tâche collaborative consiste à déléguer le contrôle d'une caméra 360° entre deux opérateurs CU : elle illustre la phase de partage à l'aide des deux rôles interactifs `<Prêteur>`, qui délègue le contrôle, et `<Receveur>`, qui en prend le contrôle.

Outre le concept de rôle interactif, la notation COMM propose un éventail étendu de types de tâches (Jourde et coll., 2010b ; Jourde, 2011) :

- Deux tâches système : calcul (interne : changement d'état) et présentation (par exemple, la tâche « Afficher écran » de la Figure 5) ;
- Trois tâches individuelles : utilisateur (interne : cognitive), action et interaction (qui combine la tâche système présentation avec la tâche utilisateur action), comme la tâche « Annuler partage » à la Figure 5 ;
- Trois tâches de groupe : groupe (interne : décision), action de groupe et interaction de groupe (qui combine la tâche système présentation avec la tâche utilisateur action), comme la tâche « Partager Caméra » à la Figure 5.

La notation préconise aussi un ensemble de règles de composition et de factorisation selon les types de tâches (Jourde, 2011).

Indépendance

Complémentaire à l'opportunisme et l'imprévisibilité, l'**indépendance**, autre aspect de la flexibilité, est transversale à la structure de l'activité (voir Figure 1), c'est-à-dire entre tâches en lien avec l'allocation des tâches aux rôles. La notion de couplage (*work coupling*) est usuelle pour caractériser cette (in)dépendance, exprimant des actions plus ou moins contraintes, liées entre elles (Salvador et coll., 1996 ; Olson et Teasley, 1996) :

> *The dependency of one participant on another refers to the granularity of the working interactions, that is, how much work can one participant do before having to interact with another.*

Cela se traduira, par exemple, par de longues séquences de tâches ou par de nombreuses répétitions pour réaliser une action minimale.

Il est admis que le degré de couplage varie le long d'un continuum dont les extrémités sont faible et fort (Salvador et coll., 1996 ; Olson et Teasley, 1996) :

> *[…], participants can be loosely coupled, where relatively few interactions are required to make relatively significant progress, or tightly coupled, where participants need to interact frequently relative to the amount of work that needs to get done.*

Pinelle et Gutwin (2005) identifient trois critères pour caractériser le couplage : interdépendance, différentiation et intégration. Le critère d'interdépendance exprime le degré d'impact d'une action accomplie par un utilisateur sur un autre utilisateur. Le critère de différentiation exprime une distinction claire et logique entre utilisateurs : la notion de rôle est usuelle pour exprimer une séparation logique du travail entre individus. Le critère d'intégration exprime la fréquence des efforts de synchronisation entre utilisateurs. Aussi, un couplage faible correspond à une différentiation élevée, à une interdépendance et à une intégration faibles. À l'inverse, un

couplage fort correspond à une différentiation faible, à une interdépendance et à une intégration élevées. Aussi, ces trois critères confirment que la notion est guidée par l'aspect métier de l'activité (relevant de l'interaction abstraite) et n'intègre pas la dimension interactionnelle (relevant de l'interaction concrète).

Le postulat que je formule est que le couplage peut être caractérisé par d'autres critères relevant de l'interaction concrète, indépendamment de l'activité métier, induits par le dispositif interactionnel. En effet, pour reprendre le schéma de la Figure 1, l'indépendance s'exprimerait sur au moins deux plans : celui de l'interaction abstraite, expression de l'activité métier, et celui de l'interaction concrète, liée au dispositif interactionnel. De plus, le postulat est formulé que de tels critères peuvent être formalisés au travers de la spécification de l'interaction concrète.

Pour cela, s'appuyant sur la théorie de l'action (Norman et Draper, 1996), un dispositif interactionnel a été conçu (Jourde et coll., 2009) sur le principe de variables physiques et psychologiques pour exprimer le couplage sur le plan de l'interaction concrète par des dépendances entre variables.

L'interaction abstraite choisie repose sur une tâche coopérative consistant à faire coopérer deux rôles pour la tâche métier suivante : contrôler un véhicule pour atteindre des cibles disséminées le long d'un parcours à l'aide d'un viseur. Aussi, le couplage entre les rôles et les tâches individuelles (contrôler l'orientation du véhicule et contrôler l'orientation du viseur) est exprimé formellement à l'aide de plusieurs variables physiques et psychologiques. Deux rôles sont identifiés : le conducteur du véhicule et le tireur contrôlant le viseur. À chaque rôle est associée une variable physique : le conducteur contrôle l'orientation du véhicule par rapport au référentiel absolu (angle $\theta_{véh.}$) ; le contrôleur règle l'orientation du viseur dans le référentiel du véhicule (angle $\theta_{\varphi,viseur}$). Si l'on se réfère aux critères de Pinelle et Gutwin (2005), la tâche **métier** impliquerait plutôt un couplage fort étant donné que le critère d'interdépendance et que le critère d'intégration sont élevés : une synchronisation permanente est nécessaire pour ajuster les paramètres selon l'avancée du véhicule et pour réussir la visée. Par la suite, seules les variables relatives au rôle de tireur sont observées, précisément les variables psychologiques et physiques perçues suivantes :

- $\theta_{véhicule}$: angle du véhicule, contrôlé par le conducteur et **perçu** par le tireur ;
- $\theta_{\psi,viseur}$ *:* variable **psychologique**, angle du viseur choisi par l'utilisateur ;
- $\theta_{\varphi,viseur}$ *:* variable **physique** perçue par le tireur, angle du viseur dans le référentiel absolu.

Pour une tâche métier identique, trois interfaces concrètes sont construites pour le rôle métier du tireur selon trois points de vues et différents référentiels. Comme le montre la Figure 6 (en haut), la première vue est une vue de dessus du véhicule (représentée par un rectangle rouge). Le référentiel est relatif au déplacement du véhicule et donne une représentation absolue par rapport à l'écran de l'orientation du véhicule, et de l'orientation du viseur (cône bleu), et la position relative des cibles (cubes jaunes). Le chemin à suivre est représenté par une piste grise.

La seconde vue (voir Figure 6 en bas, à gauche) est une vue en perspective, plaçant le rôle de tireur dans le référentiel du viseur : le viseur apparaît fixe pendant le déplacement. La troisième vue (voir Figure 6 en bas, à droite) est similaire à la précédente et place le rôle de tireur dans le référentiel du véhicule : celui-ci apparaît fixe pendant le déplacement.

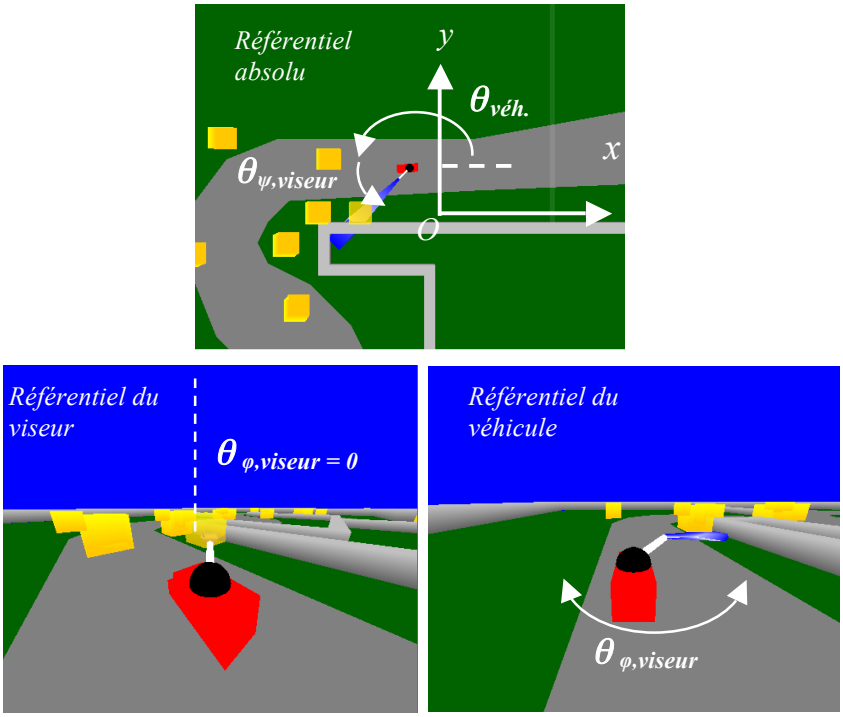
La construction de ces trois vues permet d'expliciter l'indépendance ou la dépendance entre les différentes variables perçues ou psychologiques pour le rôle du tireur, ce que semble confirmer l'expérimentation.

Figure 6
Trois vues :

(en haut) vue de dessus dans le référentiel absolu de la fenêtre;

(en bas, à gauche) vue dans le référentiel du viseur ;

(en bas, à droite) vue dans le référentiel du véhicule



Référentiel	Variables	Angle du viseur	Position d'une cible
Relatif	Couplées, direct	$\theta_{\varphi,viseur} = \theta_{\psi,viseur} + \theta_{véh.}$	$OP_{cible,vue} = OP_{cible,absolu} - OM_{véh.}$
Viseur	Couplées, indirect	$\theta_{\varphi,viseur} = 0°$	$OP_{cible,vue} = R(\theta_{\psi} + \theta_{véh.}).(OP_{cible,absolu} - OM_{véh.})$
Véhicule	Découplées	$\theta_{\varphi,viseur} = \theta_{\psi,viseur}$	$OP_{cible,vue} = R(\theta_{véh.}).(OP_{cible,absolu} - OM_{véh.})$

Table 1
Variables physiques et psychologiques

Définition des autres paramètres
$OP_{cible,vue}$: position d'une cible dans le référentiel de la vue
$OP_{cible,absolu}$: position d'une cible dans le référentiel absolu
$OM_{véh.}$: position du véhicule, contrôlée par le système
$R(\theta)$: matrice de rotation

Les trois vues ainsi construites permettent de définir trois niveaux de dépendance entre ces variables (voir Table 1) :

- **Vue de dessus :** la variable physique relative à l'angle du viseur ($\theta_{\varphi,viseur}$) est explicitement dépendante de l'orientation du véhicule ($\theta_{véhicule}$). Au niveau psychologique, le tireur doit tenir compte des **trois** variables pour ajuster l'angle qu'il souhaite en fonction de cette dépendance. Toutefois, la position des cibles est indépendante du contrôle du conducteur et du tireur. Cette vue est construite pour

induire une impression de couplage **fort**, qui se traduit par la **dépendance explicite** entre les variables relatives à l'orientation du véhicule et de la cible.

- **Vue dans le référentiel du véhicule :** dans ce référentiel, la variable physique relative à l'angle du viseur ($\theta_{\varphi,viseur}$) n'est plus liée à l'orientation du véhicule et il y a une correspondance exacte avec la variable psychologique relative à l'angle souhaité par le tireur ($\theta_{\varphi,viseur} = \theta_{\psi,viseur}$) : il n'y a que **deux** variables à traiter. Le découplage se visualise par une orientation du viseur qui apparaît libre, et l'orientation du véhicule se traduit visuellement par un changement de position des cibles. Cette vue est construite pour induire une impression de couplage **faible**, qui se traduit par l'**indépendance explicite** entre les variables relatives à l'orientation du véhicule et de la cible.
- **Vue dans le référentiel du viseur** : par définition, la variable physique ne varie pas dans le référentiel du viseur pour être constamment égale à 0° : seules **deux** variables sont à gérer. Le contrôle de l'orientation du viseur influe sur la position des cibles. Dès lors, il y a de nouveau une dépendance entre les deux variables physiques puisque la position visuelle des cibles dépend de ces deux paramètres : il y a **dépendance implicite**. Par contre, le rôle tireur se concentre sur un seul objet visuel, la cible, contre deux pour les deux vues précédentes : le viseur et la cible. Cette vue est construite pour induire une impression de couplage à un degré moindre comparé à la vue de dessus.

Ainsi, les résultats de l'expérimentation semblent confirmer l'existence de deux plans caractérisant le couplage, relevant respectivement de l'interaction abstraite et de l'interaction concrète. Les conclusions qui en découlent recoupent celles formulées par Tang et coll. (2006) quant à l'existence d'au moins six degrés de couplage et, en filigrane, l'existence d'une telle dichotomie.

1.2. ARCHITECTURES LOGICIELLES : FLEXIBILITÉ MULTINIVEAUX

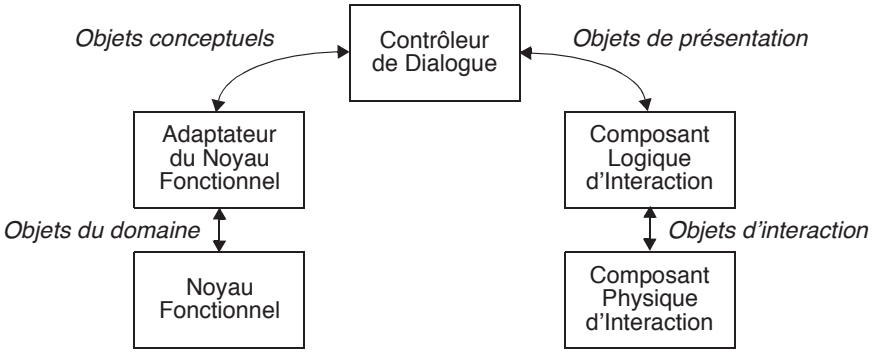
Après la phase de spécification du système interactif, vient la phase de conception logicielle. Le requis de flexibilité doit aussi y être considéré. En particulier, il peut être pris en compte et exprimé au travers d'architectures logicielles.

Précisément, mes travaux étudient comment introduire plus de flexibilité dans des architectures logicielles existantes dédiées à la mise en œuvre de systèmes interactifs, permettant l'intégration conjointe d'interactions individuelles et d'interactions multiutilisateurs. Pour cela, j'identifie deux stratégies pour introduire de nouveaux points de flexibilité dans des architectures logicielles existantes : en **longitudinal** (le long d'un empilement de couches ou composants) à la fois au niveau **macro** et **micro**, et en **transversal** (concomitant à chaque couche).

Modèles en couche et à agents

Dans le domaine de l'IHM, l'architecture ARCH (Bass et coll., 1992) constitue un modèle architectural de référence pour la conception de systèmes interactifs, offrant une décomposition canonique des principaux composants d'un système interactif. Le point clé de cette architecture est d'identifier une séparation claire entre des préoccupations fonctionnelles exprimant la logique de l'application, et l'interface utilisateur.

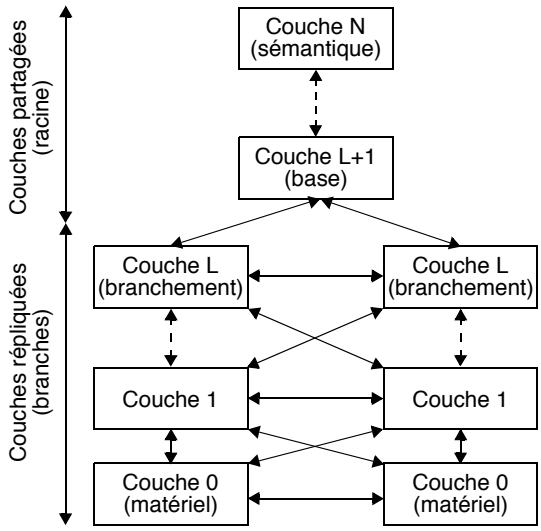
Figure 7
Modèle d'architecture à composants ARCH



Les composants de ce modèle d'architecture sont (voir Figure 7) :

- *Le Noyau Fonctionnel* (NF) qui implante les fonctionnalités et les concepts du domaine indépendamment de leur présentation ;
- *L'Adaptateur du Noyau Fonctionnel* (ANF) jouant le rôle de médiateur entre le NF et le Contrôleur de Dialogue assurant une cohérence entre la présentation et le noyau fonctionnel ;
- *Le Contrôleur de Dialogue* (CD) qui a la charge de gérer le dialogue, c'est-à-dire d'orchestrer l'enchaînement des tâches en fonction des actions de l'utilisateur et de l'état de application ;
- *Le composant d'Interaction Logique* (IL) jouant le rôle de médiateur avec l'interaction physique, rendant ainsi le CD indépendant de la plateforme cible par abstraction ;
- *Le composant d'Interaction Physique* (IP) qui représente les interacteurs concrets pour une plateforme logicielle donnée.

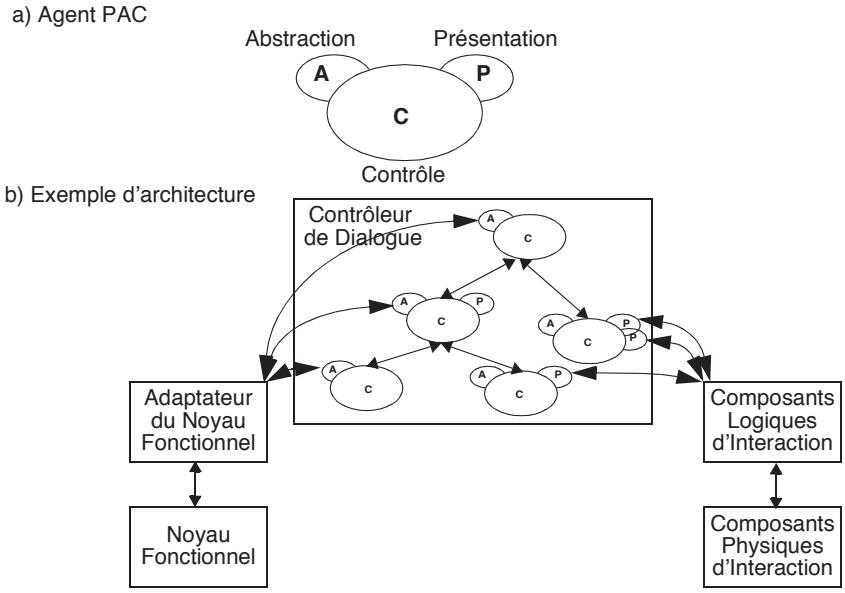
Figure 8
Modèle de Dewan



Le modèle de référence de Dewan (1995) pour la conception logicielle des collecticiels est une forme de généralisation du modèle d'architecture ARCH, décliné pour l'interaction multiutilisateur. Selon une approche similaire, le modèle de référence de Dewan (1995) structure un collecticiel

par un empilement quelconque de couches (ou composants) augmentant ainsi les points de flexibilité. Il distingue deux classes de couches : les abstractions relevant de la logique de l'application, c'est-à-dire relevant du Noyau Fonctionnel (au sommet de la pile) ; les couches les plus basses relevant de l'interaction, nommées interacteur, au plus proche de l'utilisateur (en bas de la pile) et relevant des composant IP et IL de ARCH. Ainsi, tout au long de l'empilement, toute couche placée juste au dessus d'une couche donnée est une abstraction. Les échanges entre couches en **longitudinal** sont assurés par des événements d'interaction. Ce modèle de référence exprime l'interaction multiutilisateur et le caractère distribué d'un collecticiel en considérant que certaines couches d'un même niveau, c'est-à-dire **transversales**, peuvent être répliquées en plusieurs instances (en général les interacteurs), constituant les branches de l'architecture. À l'inverse, certaines couches sont uniques (en général celles relevant du NF) et sont partagées par toutes les instances, constituant ainsi le tronc de l'architecture. Les couches répliquées communiquent par le biais d'événements collaboratifs, constituant ainsi la base pour mettre en œuvre l'interaction multiutilisateur.

Figure 9
(a) Modèle à agent PAC ;
(b) Modèle PAC-Amodeus



Le modèle PAC-Amodeus (Nigay, 1994) est un modèle hybride : il repose sur une extension du modèle ARCH (Bass et coll., 1992) selon une approche multi-agents. Ce modèle reprend les cinq niveaux fonctionnels du modèle ARCH et structure le Contrôleur de Dialogue en une hiérarchie d'agents PAC.

Un agent PAC, comme le montre la Figure 9 (a), est composé de trois facettes :

- Une facette Abstraction (A) qui gère les concepts du domaine et définit la compétence de l'agent indépendamment de la présentation,
- Une facette Présentation (P) qui définit l'interface utilisateur en sortie et en entrée, c'est-à-dire qui présente le système à l'utilisateur et interprète ses entrées au fil de l'interaction,
- Une facette Contrôle (C) qui, d'une part, fait le lien entre les facettes Abstraction et Présentation et, d'autre part, assure et gère les relations avec les autres agents PAC dans la hiérarchie (l'agent père et les agents

fils). Les agents communiquent entre eux uniquement au travers de cette facette Contrôle.

La facette Contrôle se charge donc de mettre en correspondance les phénomènes concrets et abstraits. La propriété de modifiabilité est fortement favorisée par ce modèle d'architecture car il permet l'ajout et le remplacement de facettes à moindre coût grâce à sa structure très modulaire.

Au sein du modèle hybride PAC-Amodeus, les agents PAC, comme le montre la Figure 9 (b), sont organisés selon une hiérarchie qui définit le Contrôleur de Dialogue. A l'opposé du modèle multi-agents PAC où la hiérarchie d'agents constitue l'ensemble du système interactif, dans le modèle PAC-Amodeus, les agents peuplent un seul composant, le Contrôleur de Dialogue. La facette Abstraction de chaque agent communique alors avec le Noyau Fonctionnel via l'Adaptateur du Noyau Fonctionnel et manipule des objets conceptuels. De même, la facette Présentation de chaque agent communique avec le Composant Physique d'Interaction via le Composant Logique d'Interaction.

Flexibilité longitudinale

Revisitant et étendant des modèles d'architecture existants dédiés aux systèmes interactifs, mes travaux sur les architectures logicielles (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau, 2002 ; Laurillau et Paternò, 2004a ; Laurillau et Paternò, 2004b) identifient de nouveaux points de flexibilité, notamment en explicitant le traitement conjoint des actions individuelles et collectives. J'identifie deux stratégies pour traiter la flexibilité **longitudinale** au sein des modèles d'architecture existants : au niveau **macro** par un affinement des composants de l'architecture générique de Dewan sur la base du modèle du trèfle des collecticiels ; au niveau **micro** par extension de l'approche à agents tels PAC à l'ensemble des couches de l'architecture.

Le modèle du trèfle (Salber, 1995), inspiré du modèle conceptuel des collecticiels (Ellis et Wainer, 1994), identifie des classes de fonctionnalités relevant des collecticiels. En effet, il fournit un cadre conceptuel utile pour une analyse fonctionnelle. D'après ce modèle, un collecticiel couvre trois espaces fonctionnels :

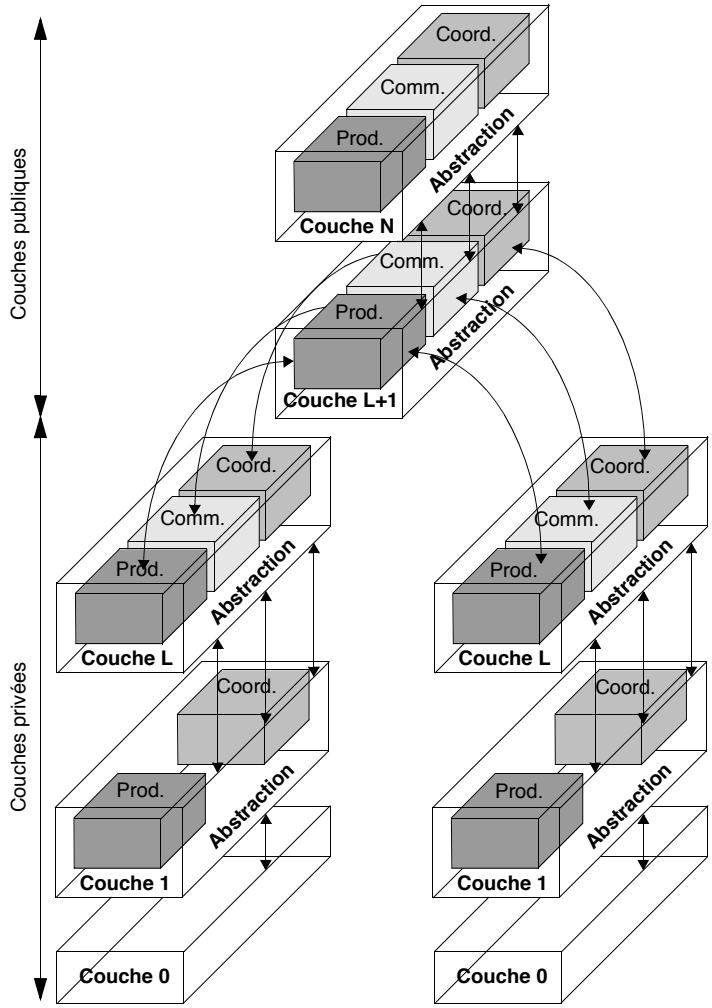
- *Espace de production :* il concerne l'ensemble des fonctionnalités de production d'objets partagés tels des documents communs et la gestion des accès à ces données partagées ;
- *Espace de communication :* il correspond aux fonctionnalités permettant l'échange d'information entre les acteurs du collecticiel. Cet échange est de la communication homme-homme médiatisée, couvrant différents modes : audio, vidéo, gestuel, etc.
- *Espace de coordination :* il concerne les fonctionnalités liées à l'organisation des tâches au sein d'un groupe et à leur allocation aux acteurs d'une activité de groupe en fonction de leurs rôles métiers. Ces fonctionnalités ont pour but de coordonner les acteurs afin d'atteindre des buts communs. Cette coordination peut s'exprimer en terme de planification des tâches de type *workflow* par exemple.

Flexibilité macro

Le modèle d'architecture Clover (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau, 2002) introduit **de nouveaux points de flexibilité au niveau macro** en appliquant au modèle d'architecture de Dewan (1995) les cinq niveaux du modèle de référence ARCH (Bass et coll., 1992), mais selon une approche différente, et les trois classes du trèfle fonctionnel des collecticiels.

Dans la suite, je raisonne en terme de composant structurant l'architecture au lieu d'employer le terme de couche ou niveau, un composant pouvant lui-même être un assemblage de composants : cette approche vise à préserver les propriétés véhiculées par les différents modèles qu'il intègre.

Figure 10
Modèle d'architecture Clover



Les caractéristiques clés de l'architecture Clover sont (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau, 2002) :

Un noyau fonctionnel public et privé. Le modèle Clover distingue le NF selon deux composants, le composant du Trèfle Fonctionnel privé, et le composant du Trèfle Fonctionnel public (voir Figure 10). Ce dernier gère l'ensemble des objets du domaine communs à l'ensemble des utilisateurs, en fonction des restrictions imposées par l'application, et offre un ensemble de services ou de fonctionnalités qui permettent de les manipuler. A l'opposé, le composant du Trèfle Fonctionnel privé gère l'ensemble des objets du domaine à caractère privé, c'est-à-dire propres à un seul utilisateur. L'ensemble des objets privés et publics constitue pour un utilisateur donné

ce que Dewan (1999) nomme "l'état d'interaction" du système. Cette subdivision, caractéristique des collecticiels (Philipps, 1999), explicite au sein de l'architecture les fonctionnalités relevant de la sphère commune et de la sphère personnelle et **constitue ainsi une première brique pour permettre la fluidité interactionnelle**. De plus, cette subdivision permet d'introduire de la variabilité quant à l'importance relative de ces deux composants, traduisant un potentiel de variabilité et donc de fluidité interactionnelle : un composant Trèfle Fonctionnel privé peut être inexistant, comme la mise en œuvre du mécanisme de collaboration transparente (une application mono-utilisateur est augmentée pour être totalement mulitutilisateurs) ; à l'inverse, un composant Trèfle Fonctionnel public peut être réduit à sa portion congrue, assurant un minimum sur le plan communicationnel (notre approche diffère du modèle de Dewan qui autorise une architecture totalement répliquée, c'est-à-dire une architecture sans couche publique ; cette divergence s'explique par le fait que le modèle de Dewan mêle les aspects à la fois conceptuels et implémentationnels).

Des classes de composants selon le trèfle fonctionnel. Chaque composant implantant une couche est une encapsulation de regroupements de fonctionnalités relevant respectivement de l'une des trois classes de fonctionnalités identifiées par le modèle du trèfle : production, communication, et coordination. Ces regroupements sont vus comme un assemblage de trois composants logiciels offrant des services relevant de l'un des trois aspects couverts par le modèle du trèfle. Ainsi, un tel assemblage agit comme une glue fonctionnelle : il assure la communication entre les trois sous-composants et les autres composants dans l'architecture. Ainsi, cette abstraction masque aux autres composants cette structure interne, organisant les fonctionnalités selon les trois facettes du modèle du trèfle. Le composant peut donc être empilé avec des niveaux qui n'ont aucune connaissance de l'existence de ces trois sous-composants, c'est-à-dire qui ne différencient pas les fonctionnalités relevant de la production, de la communication ou de la coordination. Cette approche est aussi valide avec les niveaux qui n'ont aucune connaissance de l'aspect collaboratif de l'application. Le modèle d'architecture Clover considère qu'un composant implantant une couche offre un ensemble de services, autres que ceux fournis par les trois sous-composants, incluant les services systèmes ainsi que les fonctionnalités relevant de l'action individuelle. En effet, ces derniers ne relèvent d'aucun des trois sous-composants. Cette structure est cohérente avec le modèle PAC* (Calvary et coll., 1997) qui organise les trois facettes d'un agent PAC (Présentation, Abstraction, Contrôle) selon les classes du trèfle fonctionnel. Dans la suite, je développe plus en détail les liens du modèle d'architecture Clover avec les modèles PAC et PAC* pour préciser des points de flexibilité au niveau **micro**.

Trois classes d'événements. Au sein du métamodèle d'architecture de Dewan, la communication entre les différentes couches est rendue possible grâce à deux types d'événements : les événements d'interaction échangés le long des branches et de la racine ; et les événements de collaboration échangés entre couches de branches distinctes. Notre modèle respecte les mêmes règles que le modèle de Dewan. Cependant, les événements d'interaction et de collaboration sont différenciés selon trois catégories : production, communication et coordination. Cet affinement permet une communication

directe entre sous-composants dédiés. Notre modèle dispose néanmoins d'un événement générique qui assure la communication avec les autres couches qui ne sont pas des Trèfles Fonctionnels.

Filtrage et observabilité. Le modèle d'architecture Clover considère que chaque composant met en œuvre un mécanisme de filtrage. En effet les ressources, comme les objets du domaine gérés par les composants relevant du NF, peuvent être filtrés selon une politique de publication comme définie dans la partie 3.3. Par exemple, il est possible de définir un degré de publication du Trèfle Fonctionnel : ce degré peut varier du "totalement accessible" à "inaccessible" en passant par "partiellement accessible". Pour le Trèfle Fonctionnel privé, ce degré doit pouvoir être changé dynamiquement. De même, il est possible de définir un degré d'observabilité sur l'interaction utilisateur, avec une granularité plus ou moins fine, par exemple de l'ordre du mouvement de la souris comme mis en œuvre avec le télépointeur, selon le composant dans la hiérarchie.

Flexibilité micro

Dans une approche similaire au modèle d'architecture hybride PAC-Amodeus qui combine les modèles ARCH et PAC, notre modèle d'architecture Clover combine la structure en couche héritée du modèle de Dewan avec le modèle PAC*, pour structurer l'ensemble des couches, visant ainsi une articulation harmonieuse entre les différents modèles.

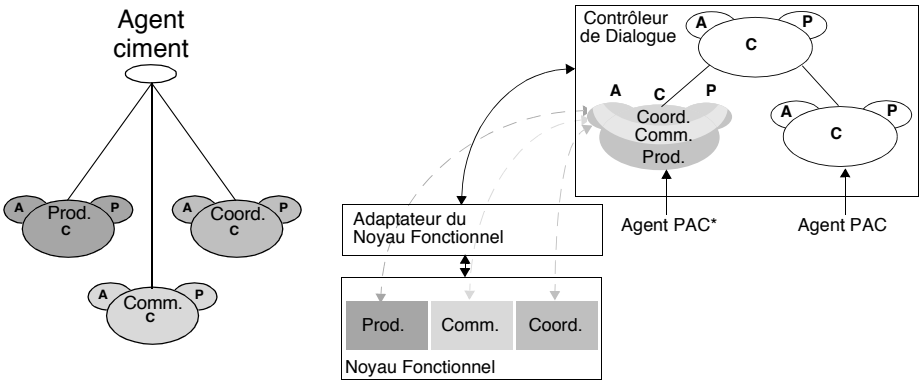


Figure 11
(à gauche)
Agent PAC* centralisé ;

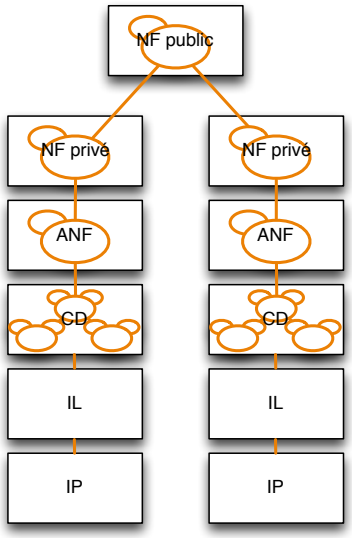
(à droite)
liaison entre le NF
et un agent PAC*

Le modèle PAC* (Calvary et coll., 1997) est la version collaborative du modèle d'architecture hybride PAC-Amodeus. Dans le modèle PAC*, le Contrôleur de Dialogue est composé d'une hiérarchie d'agents PAC et PAC* qui échangent des événements avec le Noyau Fonctionnel via l'Adaptateur du Noyau Fonctionnel. Dans ce modèle, un agent PAC* résulte d'un découpage fonctionnel d'un agent PAC affiné selon les trois espaces fonctionnels du modèle du trèfle : production, communication et coordination. Aussi, les trois facettes d'un agent PAC sont décomposées en trois parties dédiées chacune à une dimension du modèle du trèfle. Un agent PAC* peut exister sous trois formes différentes : la forme centralisée, la forme répartie et la forme hybride. Par exemple, dans la forme centralisée, un agent PAC* (voir Figure 11 à gauche) est composé de trois agents dédiés respectivement à la production, à la communication et à la coordination, et reliés à un agent ciment. Ce dernier assure la communication entre les trois agents et le reste du monde, c'est-à-dire les autres agents de la hiérarchie. Dans ce modèle, un agent PAC* est comparable à un composant public et un agent PAC est comparable à un composant privé. Enfin, la communication entre agents s'inspire du modèle de Dewan.

Toutefois, alors que les modèles PAC-Amodeus et PAC* introduisent un point de flexibilité au niveau micro cantonné au composant Contrôleur de Dialogue, notre modèle préconise une structuration de l'ensemble des composants par une approche à agents PAC et PAC*, offrant ainsi **des points de flexibilité micro à tout niveau de l'architecture**.

D'abord, notre modèle d'architecture Clover permet aux trois facettes Abstraction d'un agent PAC* de communiquer avec le sous-composant correspondant du Trèfle Fonctionnel privé, comme le montre la Figure 11 (à droite). Cette communication n'est pas directe puisque les événements transitent via l'Adaptateur du Noyau Fonctionnel (qui n'est pas nécessairement structuré selon les trois facettes du modèle du trèfle), puis via l'interface logicielle du Trèfle Fonctionnel privé. De plus, comme le montre la (voir Figure 11 à gauche), un agent PAC* dans sa forme hybride est constitué d'un agent ciment. Ce dernier est chargé de gérer la communication entre les trois sous-agents et le monde extérieur, c'est-à-dire les autres agents de la hiérarchie. L'abstraction du Trèfle Fonctionnel de notre architecture Clover joue un rôle similaire étant donné qu'elle gère la communication entre les différentes couches (le monde extérieur) et les trois sous-composants dédiés à la production, communication et coordination.

Figure 12
Illustration de la généralisation de l'approche à agents à toutes les couches du modèle Clover



La généralisation de l'approche à agents à tout composant de l'architecture se traduit en une unique structure hiérarchique reliant l'ensemble des composants (voir Figure 12). Cette stratégie offre plusieurs avantages. Notamment, elle offre un modèle de composant unifié assurant l'interopérabilité entre les couches et facilitant la réutilisation grâce une interchangeabilité plus aisée. Cette facilité est rendue possible grâce à un modèle de communication uniforme entre agents, tel que préconisé par le modèle PAC. C'est également ce modèle qui est mis en œuvre pour assurer la communication entre couches répliquées dans le modèle de Dewan. Comme je l'expose dans la suite, c'est sur cette approche qu'il est possible de proposer des points de **flexibilité longitudinale**.

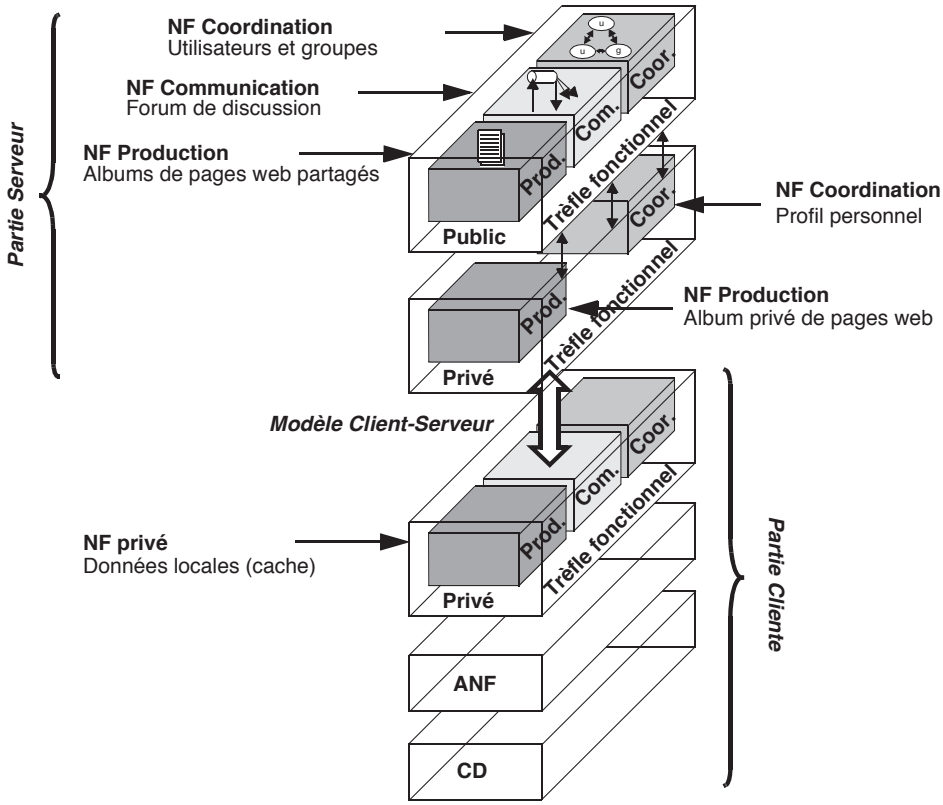
Plateforme Clover

La plateforme logicielle Clover (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau, 2002) a été développée. Elle instancie le modèle d'architecture Clover. Elle est à la base de l'application CoVitesse (voir

partie 1.1), illustrant ainsi le potentiel en **flexibilité longitudinale** à la fois aux niveaux **macro** et **micro**, et validant l'approche par la mise en œuvre d'une preuve de concept logicielle.

La plateforme Clover propose à la fois une interface de programmation (API, *Application Programming Interface*) et un environnement d'exécution reposant sur un modèle client-serveur. Ce dernier permet l'hébergement à l'exécution d'instances de composants relevant du Trèfle fonctionnel public et privé, conformément au modèle Clover. Précisément, la plateforme permet l'hébergement à l'exécution de sous-composants dédiés à la production, à la coordination et à la communication, conformément au modèle du trèfle fonctionnel des collecticiels. L'interface de programmation de la plateforme Clover offre un cadre logiciel (*software framework*) pour programmer des composants selon un modèle à agents, reposant sur un modèle de communication par messages, avec différentiation de classes d'agents selon le modèle du trèfle fonctionnel.

Figure 13
Architecture de CoVitesse selon le modèle Clover

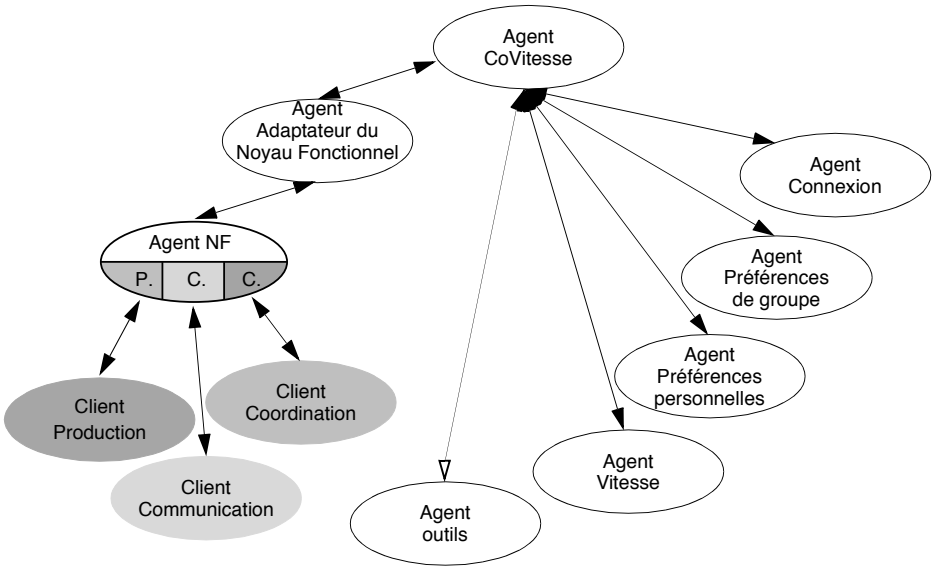


Dans la suite, je m'appuie sur l'application CoVitesse pour illustrer la mise en œuvre des deux stratégies (macro et micro) ainsi que sur la plateforme Clover. L'implantation du composant relevant du Trèfle fonctionnel est décrite à gros grain (se référer à (Laurillau, 2002) pour plus de détails).

Comme le montre la Figure 13, au niveau **macro**, l'application est structurée en plusieurs composants relevant du modèle ARCH et, au niveau du NF, les composants sont constitués de sous-composants traitant l'une des trois classes couvertes par le trèfle fonctionnel : production, communication et coordination. Précisément, la production relève de la gestion d'albums collectant des résultats accumulés suite à une navigation sur le web, sachant qu'une partie est publique car partagée.

La communication met en œuvre un outil de discussion en ligne instantanée (*chat*) : il est à noter que l'instance centralisée du composant du Trèfle fonctionnel privé n'implante pas ce sous-composant mais permet l'échange de message relevant de la communication entre les couches qui l'englobe. Les sous-composants relatifs à la coordination permettent, d'une part, la gestion des préférences personnelles, comme le profil et les informations d'identification, et la gestion de différents types de navigation collaborative (voir partie 1.1). De plus, l'implantation de la plateforme Clover fournit des mécanismes génériques pour réaliser du filtrage, conformément aux niveaux d'observabilité publiée (voir Partie 3.3). Précisément, les mécanismes de filtrage tels que décrits pour l'application CoVitesse (voir partie 1.1) sont développés selon le modèle d'implantation prescrit par l'interface de programmation de la plateforme Clover.

Figure 14
Architecture de CoVitesse selon le modèle PAC



Au niveau **micro**, comme le montre la Figure 14, toutes les couches sont implantées par des agents et, pour les couches relevant du noyau fonctionnel, ces agents sont triples, en cohérence avec le modèle du trèfle fonctionnel et le modèle PAC*. J'ai démontré que la flexibilité induite par l'usage d'agents au niveau micro a permis d'obtenir une plateforme à potentiel de généricité pour permettre l'exécution conjointe d'une autre application collaborative, un tableau blanc partagé (pour plus de détails, se référer à (Laurillau, 2002)).

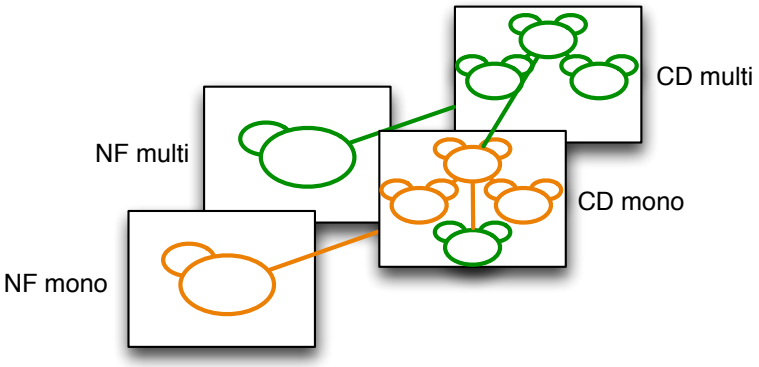
Flexibilité transversale

J'identifie une approche complémentaire pour introduire des points de flexibilité dans les architectures logicielles, transversalement aux couches de l'architecture. Précisément, j'identifie des moyens pour l'**augmentation transversale** d'une architecture logicielle initialement mono-utilisateur afin de permettre un support conjoint de l'interaction mono-utilisateur et multiutilisateur, offrant ainsi une autre façon de mettre en œuvre la fluidité interactionnelle. La stratégie présentée dans cette partie repose également sur l'approche à agents généralisée à l'ensemble des composants constituant le modèle ARCH, tel que décrit dans la partie précédente. Dans la suite, je m'appuie sur l'application CoCicero pour illustrer cette seconde approche.

Figure 15
Écran principal de l'application CoCicero

CoCicero (Laurillau et Paternò, 2004a ; Laurillau et Paternò, 2004b) est une application interactive de visite ludo-éducative et collaborative de musées, déployée au musée du marbre à Carrare (Italie), développée pour des PDA (PocketPC) (voir Figure 15). Elle repose sur l'application mono-utilisateur Cicero qui accompagne l'utilisateur dans sa visite, offrant accès à du contenu multimédia en fonction des œuvres ou artéfacts vus par le visiteur. L'application CoCicero repose sur un redéveloppement complet de l'application Cicero (Ciavarella et Paternò, 2003). Elle étend cette dernière par l'ajout de nouveaux éléments dans l'IHM, en particulier des énigmes individuelles et collectives, ainsi que des éléments d'interface permettant d'organiser, de gérer la visite collaborative, et de communiquer entre participants.

Figure 16
Illustration de l'augmentation transversale du modèle ARCH+PAC pour le noyau fonctionnel (NF) et le contrôleur de dialogue (CD)



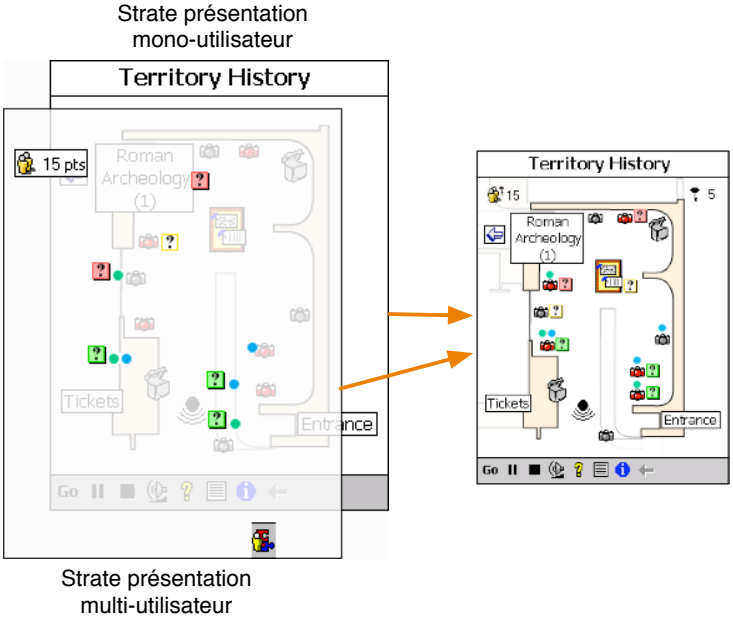
En considérant le Contrôleur de Dialogue comme point central d'une architecture basée sur le modèle PAC-Amodeus mono-utilisateur, le principe de **l'augmentation transversale** consiste à concevoir un Contrôleur de Dialogue pour un système interactif multiutilisateurs dont un sous-ensemble de la hiérarchie d'agents est la hiérarchie d'agents peuplant le Contrôleur de Dialogue du système mono-utilisateur. De nouveaux agents et sous-ensembles d'agents apparaissent dans l'arborescence ainsi obtenue, en différents points de l'architecture :

- Comme sous-ensembles frères du Contrôleur de Dialogue dans sa version mono-utilisateur ;
- Comme nouveaux agents s'intégrant, à l'exécution, dans la hiérarchie du Contrôleur de Dialogue, si ce mécanisme est prévu.

En effet, comme l'illustre la Figure 16, la hiérarchie d'agents constituant le Contrôleur de Dialogue de l'application Cicero constitue un des descendants du Contrôleur de Dialogue de l'application CoCicero. Aussi, les nouveaux agents ajoutés par l'application CoCicero permettent de gérer le dialogue pour les énigmes collectives et les énigmes individuelles ainsi que le paramétrage de la visite collaborative.

Cette augmentation implique une extension des autres composants, notamment ceux relevant du noyau fonctionnel et de l'interaction logique, faisant ainsi coexister des composants relevant de l'interaction mono-utilisateur et de l'interaction multiutilisateur. En effet, m'appuyant sur une approche à agents généralisée à l'ensemble des composants du modèle ARCH, cela conduit à la coexistence de deux branches concomitantes dont le point central est le Contrôleur de Dialogue de l'application CoCicero. Comme le montre la Figure 16, deux agents relevant du Noyau Fonctionnel coexistent sans interférer puisque le Noyau Fonctionnel ajouté assure notamment la communication entre les différents PDA et la propagation des messages relatifs à l'interaction avec les énigmes.

Figure 17
Interface concrète : composition de l'interface concrète de la version mono-utilisateur avec l'interface concrète de la version multi-utilisateurs



Cette extension s'applique également à l'interaction logique et physique. Ceci a été rendu possible par le développement d'un moteur générique pour le rendu de l'IHM fonctionnant par strates (*layers*). Comme le montre la Figure 17, les composants graphiques constituant l'IHM des deux applications sont placés à l'écran dans différentes strates. Ainsi, les nouveaux composants d'interaction ajoutés par l'application CoCicero sont positionnés dans une strate différente. Ceux-ci sont traités par l'application mono-utilisateur sans connaissance de la couche collaborative. En effet, selon l'élément d'interface avec lequel l'utilisateur interagit, des messages interactifs sont propagés le long de la hiérarchie d'agents. Aussi, les messages propres à l'application CoCicero remontent à la racine car inconnus de l'application mono-utilisateur. L'augmentation de l'IHM et l'ajout des composants dans différentes couches se fait sur la base d'un modèle de description de l'IHM (fichier XML) : une description de l'IHM de la version mono-utilisateur et une description disjointe de l'IHM collaborative augmentant la version mono-utilisateur.

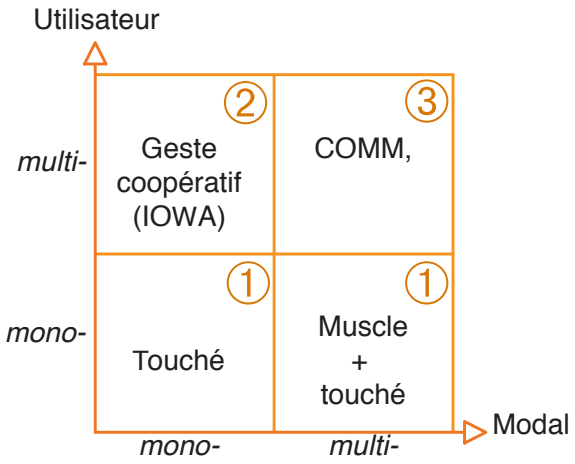
Cette méthode se distingue des approches du type « collaboration transparente », comme le permet l'environnement Flexible JAMM (Begole et coll., 1999), qui transforme une application mono-utilisateur en une application multiutilisateur avec une IHM identique. En effet, ce type d'approche consiste simplement à propager les événements d'interaction (par exemple, un mouvement de la souris) à toutes les instances grâce à une modification des boîtes à outils graphiques. Au contraire, dans notre approche, l'IHM est augmentée de nouveaux éléments d'interfaces relatifs à l'interaction multiutilisateur, prévus dès la conception pour s'intégrer harmonieusement avec l'IHM dans sa version mono-utilisateur.

2. Du monomodal au multimodal

La seconde dimension que j'explore pour satisfaire le requis de flexibilité porte sur la relation entre l'utilisateur et le système interactif en termes de modalités d'interaction. J'aborde cette dimension sous deux angles : par la spécification de modalités ; et par leur mise en œuvre.

Le besoin de flexibilité dans les outils tel que je l'ai traité du point de vue de l'interaction multiutilisateur s'applique également à l'interaction multimodale, plus encore lorsque nous l'abordons conjointement avec l'interaction multiutilisateur. En effet, comme le montre le schéma de la Figure 18, mes travaux traitent cette question selon deux approches, en croisant interaction mono/multiutilisateur et interaction mono/multimodale. Notamment, à gros grain j'aborde la spécification conjointe de l'interaction multimodale et multiutilisateur (voir Figure 18 (3)). À grain plus fin, j'aborde la fluidité interactionnelle en termes de spécification en me focalisant sur l'interaction gestuelle au touché (voir Figure 18 (2)) dans un contexte d'interaction mixte : mono-utilisateur et multiutilisateur.

Figure 18
Croisements entre interaction multiutilisateurs et interaction multimodale



Mes travaux sur les modalités d'interaction s'intéressent également à la mise en œuvre de deux modalités d'interaction gestuelle (voir Figure 18 (1)) : au touché sur surface interactive et par interaction musculaire. En effet, considérer la flexibilité sur le plan de la spécification de l'interaction multimodale au travers des outils ne suffit pas pour satisfaire le requis de fluidité interactionnelle : il convient également de mettre en œuvre des modalités d'interaction fiables et robustes.

2.1. SPÉCIFICATION DE L'INTERACTION MULTIMODALE

Outre l'interaction multiutilisateur, la notation COMM (Jourde, 2011 ; Jourde et coll., 2010a) permet la spécification de l'interaction concrète et, en particulier, de l'interaction multimodale. Elle couvre donc à la fois la spécification de l'interaction abstraite, qui souvent relève de l'interaction de groupe, et la spécification de l'interaction concrète par l'introduction du type de tâche modale.

Arbre de tâches et multimodalité

Ce nouveau type de tâche sert à définir une modalité d'interaction comme étant un couple `<dispositif, langage d'interaction>` (Nigay, 1994). Une tâche modale est donc une tâche comprenant deux nouvelles

décorations, relatives au dispositif et au langage d'interaction. Dans le traitement de l'interaction multimodale, conjointement au traitement de l'interaction multiutilisateur, la notation introduit d'autres leviers pour introduire de la flexibilité dans l'expression de l'interaction multimodale et multiutilisateur.

Comme le montre la Figure 19, avec la notation COMM, la représentation visuelle d'une tâche modale rend explicites le dispositif et l'élément du langage d'interaction (action atomique d'interaction) pour accomplir la tâche modale. Dans cet exemple, la reconnaissance de la parole est spécifiée par une tâche modale, une modalité en entrée, dont l'action atomique du langage est de formuler oralement l'ordre « bouger » et le dispositif d'interaction en entrée est un microphone.

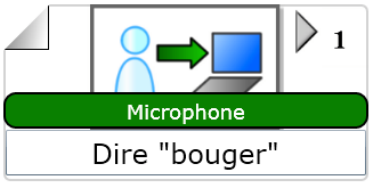


Figure 19
Exemple de tâche modale relative à une modalité en entrée reposant sur de la reconnaissance vocale

Certaines notations emploient une notation externe pour spécifier l'interaction concrète comme la notation GTA (spécification de l'interaction abstraite multiutilisateur) (Van Welie, 2001) qui repose sur la notation NUAN (Venema, 1999). À l'inverse, la notation COMM propose une représentation unique pour spécifier conjointement et harmonieusement l'interaction abstraite et concrète. Sur ce point, notre approche offre des similarités avec la notation Dynamo-Aid (Clerckx et coll., 2005) qui introduit la notion de modalité associée aux feuilles d'un arbre de tâches.

Cette notion de tâche modale constitue une base pour exprimer l'interaction multimodale : les tâches élémentaires (feuilles de l'arbre) peuvent être précisées comme étant une composition de tâches modales pour spécifier l'interaction concrète. L'exemple représenté par la Figure 20 est un cas concret tiré du projet FH-PA (voir partie 1.1) qui spécifie le transfert du contrôle de charge utile d'un drone entre deux opérateurs, et illustre deux tâches multimodales (« Select payload » et « Send request to récipient »), chacune composée de deux tâches modales.

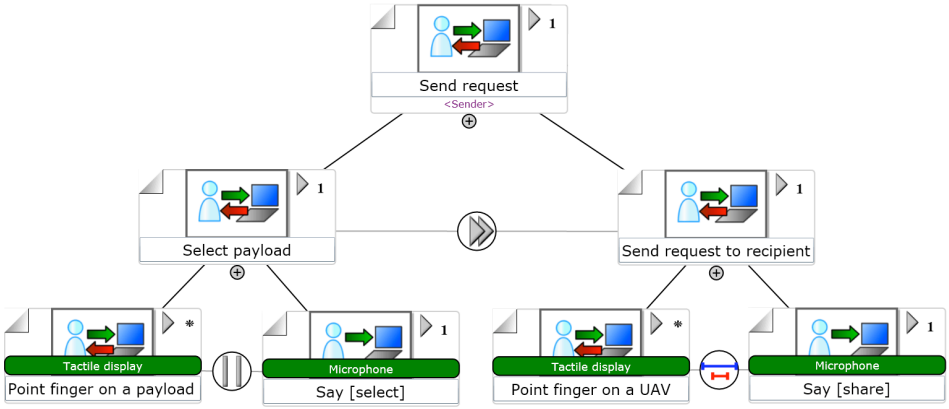


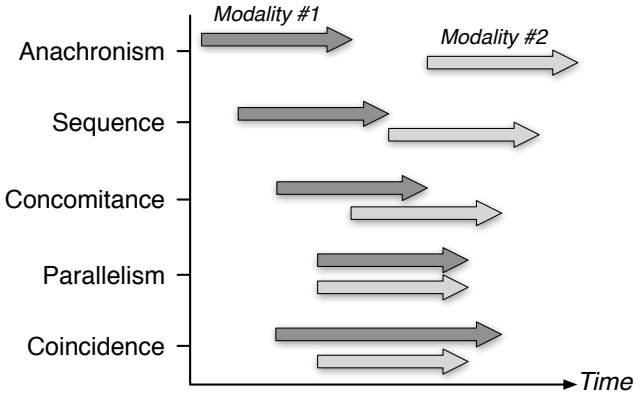
Figure 20
Tâche multimodale de partage de contrôle de charge utile d'un drone

Les propriétés CARE (Complémentarité, Assignation, Redondance, Équivalence) (Coutaz, 1995) sont une première brique pour exprimer la multimodalité par une composition de tâches modales. L'**assignation** exprime l'absence de choix : une tâche élémentaire est simplement une

tâche modale. Il n' y a pas de contrainte temporelle. L'**équivalence** exprime une situation de choix entre plusieurs modalités : une tâche élémentaire est alors une composition de tâches modales reliées par un opérateur d'alternative, avec ou sans contrainte temporelle. La **complémentarité** exprime l'usage conjoint de plusieurs modalités (c'est-à-dire qu'aucune modalité ne peut être utilisée individuellement pour accomplir la tâche) dans une fenêtre temporelle : une tâche élémentaire est alors une composition d'au moins deux tâches modales, reliées par un opérateur temporel. Par exemple, la Figure 20 illustre deux tâches multimodales (tâches « Select payload » et « Send request to recipient ») composées chacune de deux tâches modales complémentaires : l'opérateur de parallélisme (||) exprime que leur exécution est concomitante dans le temps. La **redondance** exprime l'usage de modalités pour exprimer la même information soit en parallèle soit en séquence, dans une fenêtre temporelle : une tâche élémentaire est alors la composition d'au moins deux tâches modales reliées par un opérateur temporel.

En outre, la notation COMM propose un éventail étendu des opérateurs temporels LOTOS présents dans la notation CTT, par des opérateurs précisant des agencements temporels à grain plus fin sur la base des relations de Allen (Allen, 1983) appliquées à la multimodalité (Vernier et Nigay, 2001). En effet, lorsque plusieurs modalités sont accomplies au même moment, il devient important de préciser quelles sont les contraintes temporelles régissant l'ordre dans lequel doivent être réalisées les tâches modales. De telles contraintes sont nécessaires pour mettre en œuvre un véritable mécanisme de fusion afin de détecter quand et comment la fusion des actions doit être réalisée. Ainsi, reprenant l'exemple de la Figure 20, afin de réaliser la tâche « Select payload », l'utilisateur doit poser son doigt sur l'écran tactile tout en formulant oralement l'ordre « select ». L'opérateur de parallélisme reliant ces deux modalités exprime donc une contrainte temporelle entre celle-ci, à savoir une exécution concurrente. Seulement, cette contrainte temporelle n'est pas assez précise pour exprimer une exécution concomitante.

Figure 21
Combinaisons temporelles entre modalités



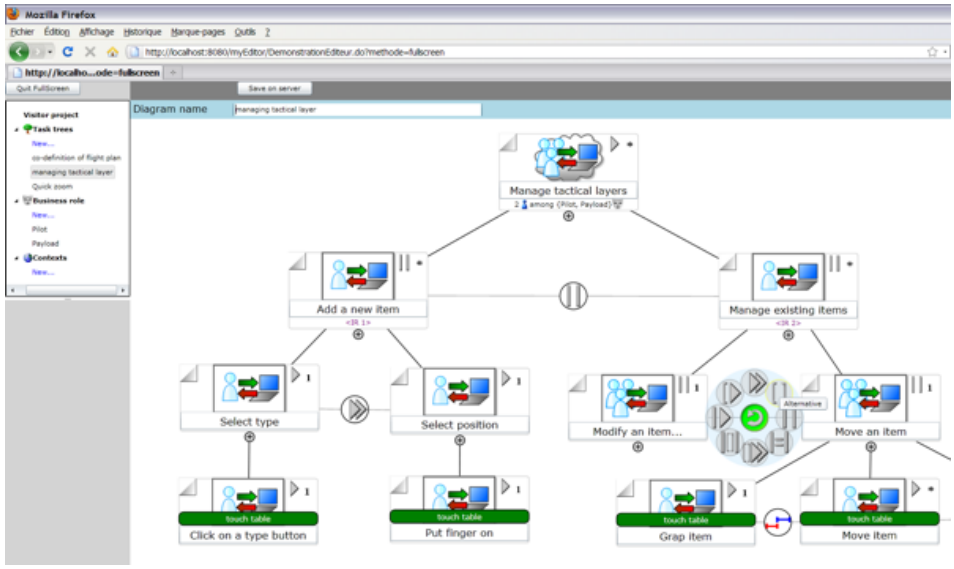
Aussi, l'introduction de nouveaux opérateurs temporels permet de gagner en précision. Comme le schématise la Figure 21, cinq opérateurs temporels supplémentaires sont introduits (Vernier et Nigay, 2001) : anachronisme (pause entre l'exécution de deux modalités), séquence (enchaînement temporel de deux modalités), concomitance (chevauchement partiel dans l'exécution de deux modalités), parallélisme, et coïncidence (une modalité est exécutée dans le contexte d'exécution d'une autre). Ainsi, reprenant

l'exemple illustré par la Figure 20, la seconde tâche d'interaction concrète « Send request to recipient » est plus précise que la tâche « Select payload » : les deux tâches modales « Point Finger on UAV » et « Say [share] » sont reliées par l'opérateur de concomitance. Cela signifie que l'exécution de l'une des deux modalités démarre avant l'autre et que son exécution se terminera avant. L'usage de tels opérateurs constitue donc une seconde brique pour spécifier l'interaction multimodale.

Toutefois, la notation COMM conserve les opérateurs LOTOS tirés de CTT et préconise leur usage pour exprimer des relations temporelles entre tâches non modales, à plus haut niveau d'abstraction. De plus, cette approche assure une continuité entre la spécification des tâches relevant de l'interaction abstraite et de l'interaction concrète. En effet, les cinq opérateurs temporels additionnels apportent une spécification plus fine par affinement des deux opérateurs temporels de base. Ainsi, l'opérateur exprimant le parallélisme (| |) est affiné en trois opérateurs exprimant : la concomitance, la coïncidence, et le parallélisme. De même, l'opérateur de séquence est affiné en deux opérateurs : l'anachronisme et la séquence.

Afin de faciliter la conception, la notation COMM est outillée par l'éditeur e-COMM[2] (Jourde, 2011 ; Jourde et coll., 2010a ; Jourde et coll., 2010b). Il est développé sous la forme d'une application en ligne. Celui-ci repose sur la technologie SilverLight de Microsoft et est programmé en C#. L'interaction a été pensée pour autant que possible favoriser la manipulation directe et limiter la multiplication des fenêtres pop-up. La principale fenêtre pop-up disponible a été pensée pour limiter le nombre de champs du formulaire. Aussi, l'espace d'affichage disponible est principalement un espace d'édition dédié à la construction d'arbres. Les différents espaces de travail sont accessibles par une barre verticale située sur le côté gauche de l'interface. A cela s'ajoute une barre de commandes réduite à quelques boutons dans la partie supérieure de l'interface. Lors de la création d'un nouvel arbre, l'éditeur s'ouvre sur une seule tâche.

Figure 22
Ecran principal de l'éditeur e-COMM



L'ajout de tâches filles est réalisable en cliquant sur le symbole (+) (voir Figure 22) situé en dessous de chaque tâche. Pour attribuer un nom à une

[2] e-comm.imag.fr

tâche, il suffit de cliquer dans la zone inférieure du rectangle associé et de réaliser une saisie au clavier. Si une tâche se révèle être mal placée, il suffit de la déplacer (technique glisser/déposer). L'éditeur fait alors apparaître tous les points d'ancrage possibles à la manière d'un aimant. Pour la suppression, il suffit de sélectionner une tâche et d'activer la suppression au clavier. Bien sûr, l'éditeur offre la sélection et le déplacement d'une grappe de tâches, le zoom avant et arrière, et le déplacement de la zone de travail. Dès que l'on ajoute une nouvelle tâche fille, celle-ci est reliée par un opérateur inconnu.

Figure 23
Palette d'opérateurs :

(à gauche) opérateurs LOTOS ;

(à droite) opérateurs de Allen

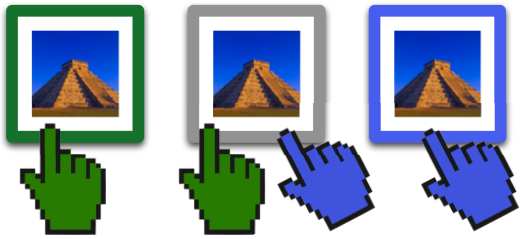
Pour choisir un des opérateurs CTT, hérités de LOTOS, il suffit de cliquer sur l'opérateur. Une palette circulaire apparaît (voir Figure 23 à gauche). Un second clic au centre de la palette fait apparaître la palette circulaire d'opérateurs temporels basés sur les relations de Allen (voir Figure 23 à droite).

Comme mentionné dans la partie 1.1, l'éditeur e-COMM a été utilisé pour produire tous les modèles de tâches du projet PEA FH-PA. Il est également utilisé depuis 2011 en enseignement dans le cadre du Master 2 informatique de l'université Joseph Fourier dans le cours sur les collecticiels.

Spécifier l'interaction gestuelle multiutilisateurs : classes de techniques

Comme je l'évoque en introduction, dans un contexte d'interaction multiutilisateur, la gestuelle a une place importante dans le processus de collaboration (Wang et Blevis, 2004) autant dans des situations de face à face qu'à distance car elle est un vecteur essentiel de la communication (Van den Hoven et Mazalek, 2011) : elle constitue un support à l'auditeur pour mieux comprendre l'intervenant ; elle permet à l'intervenant de verbaliser des concepts tout en réduisant la charge cognitive ; elle aide à conceptualiser et à réaliser des tâches de résolution de problèmes comme la conception d'artefacts (Visser et Maher, 2011). Elle constitue ainsi une forme de communication alternative et néanmoins complémentaire à la communication intentionnelle et conséquentielle Gutwin et Greenberg (2000), relative à l'expression du langage du corps (*body language*) ainsi qu'à la désignation (*deictic references*).

Figure 24
Illustration d'un geste coopératif : transfert de propriété



Les surfaces interactives recouvrent une famille de dispositifs d'interaction reposant justement sur l'interaction gestuelle. Plus encore, elles favorisent naturellement la coopération, la communication et l'interaction multiutilisateurs dans des situations de face à face (Wang et Blevis, 2004 ;

Scott et coll., 2003). Leur émergence est désormais l'objet de nombreuses recherches et de nombreux développements d'applications collaboratives. En particulier, ce type de dispositif interactif permet le développement de techniques d'interaction gestuelles coopératives (*coopérative gestures*, Morris et coll. (2006)) dont leur accomplissement nécessite la coordination et l'implication de plusieurs utilisateurs.

Les gestes coopératifs reposent sur une modalité gestuelle et sont des interactions interprétées par le système comme étant une combinaison d'actions réalisées individuellement par différents utilisateurs mais associés à une seule commande. Ce type d'interaction permet d'obtenir, par exemple, un consensus pour réaliser une action irréversible (par exemple, effacer tous les objets numériques présents à l'écran) ou pour réaliser un vote. Par exemple, la Figure 24 illustre un type de geste coopératif : le transfert de droit ou de propriété. Sur une surface interactive, l'utilisateur *vert* manipule une image de sa bibliothèque personnelle : il est le seul à pouvoir interagir avec et à réaliser des modifications. L'utilisateur *vert* souhaite en transférer la propriété à l'utilisateur *bleu* : après avoir activé le mode pour le transfert de droits, les utilisateurs *vert* et *bleu* se synchronisent en touchant simultanément l'image concernée. Dès que l'utilisateur *vert* relâche son doigt, l'image devient la propriété de l'utilisateur *bleu*. Ainsi, ce type de geste permet d'améliorer l'implication des participants dans la collaboration, de limiter les erreurs et les incompréhensions entre individus, et d'avoir une meilleure perception de l'activité de groupe (*group awareness*). Toutefois, ce type de technique d'interaction nécessite un dispositif interactionnel capable (1) de traiter des interactions multiples en entrée avec un même dispositif et (2) de **différentier les utilisateurs**, c'est-à-dire de leur attribuer une identité.

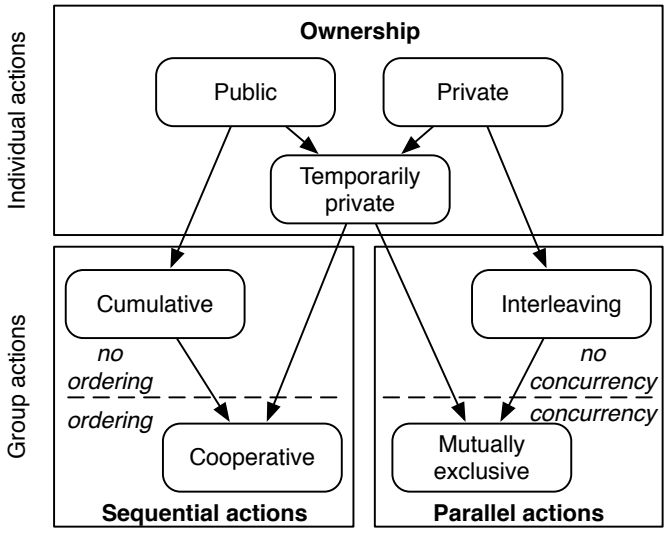
La capacité du système à identifier l'utilisateur permet également de traiter l'interaction individuelle dans un contexte multiutilisateurs. Par exemple, l'environnement idLenses (Schmidt et coll., 2010) exploite l'identité de l'utilisateur pour préserver son espace d'interaction, et donc sa vie privée, en appliquant des filtres. Il permet également de personnaliser son environnement comme en remplissant automatiquement des formulaires sur la base de son identité.

Dans le prolongement de mes travaux sur la modalité gestuelle et de l'interaction basée sur le touché, j'aborde la spécification de techniques d'interaction mono et multiutilisateurs dans le cas où le système dispose de cette information essentielle (Laurillau, 2013) : **l'identité de l'utilisateur**. Précisément, prolongeant les travaux de Morris et coll. (2006), j'identifie des classes d'interaction gestuelle multiutilisateur et mets en évidence comment ces classes s'articulent avec l'interaction gestuelle mono-utilisateur lorsque l'identité de l'utilisateur est accessible. Il convient de noter que cette cartographie ne se limite pas à l'interaction au touché et inclut d'autres modalités (par exemple par l'usage de plusieurs souris (Druin et coll., 1997)) traduisant le geste et permettant d'identifier les utilisateurs.

J'identifie sept classes d'interaction gestuelle (voir Figure 25) combinant à la fois interaction mono-utilisateur et interaction multiutilisateurs, sur la base de l'identité des utilisateurs (Laurillau, 2013). Ces classes se regroupent en trois catégories relatives à des :

Actions individuelles. Ces actions dépendent de la notion de propriété d'un artéfact. Un artéfact **public** signifie que tout utilisateur peut interagir simultanément avec et ne peut devenir la propriété d'un des utilisateurs. Ce type d'artéfact permet de peupler une IHM avec des composants graphiques d'utilité publique dont l'action résultante est propre à chaque utilisateur et ne participe pas à l'action commune. Un artéfact **privé** est au contraire la propriété exclusive d'un utilisateur et aucun autre ne peut interagir avec. L'effet d'une action est restreint à son utilisateur. Ce type de composant permet donc de gérer la question de la vie privée. Un artéfact **temporairement privé** est un artéfact public permettant son usage exclusif pendant la durée de l'interaction. En d'autres mots, pendant ce temps d'interaction, les actions des autres utilisateurs sur cet artéfact sont ignorées sur la base de l'identité de l'utilisateur associée aux actions.

Figure 25
Classes d'interaction gestuelle multiutilisateurs

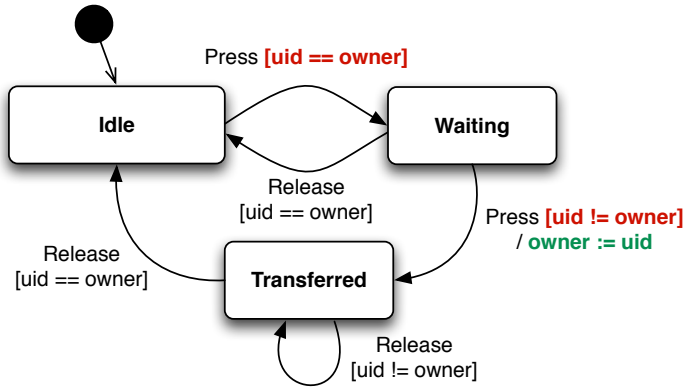


Actions de groupe séquentielles. Ces classes d'actions identifient des actions nécessitant une action conjointe d'un groupe d'utilisateurs. Ces actions sont réalisées en séquence. Il peut s'agir d'une action de groupe **cumulative**, comme par exemple un vote pour obtenir un consensus : ce qui compte c'est le nombre ; l'ordre de la séquence n'a aucune importance. Il peut également s'agir d'une action de groupe **coopérative**, ce qui correspond à la notion de geste coopératif (Morris et coll., 2006) : les utilisateurs doivent se synchroniser pour réaliser une séquence d'actions dans un ordre bien défini. Le scénario de transfert de propriété (voir Figure 24 et Figure 26) relève de cette classe d'action de groupe.

Actions de groupe parallèles. Ces classes identifient des actions de groupe permettant un entrelacement d'actions individuelles. Deux classes sont identifiées : les actions **entrelacées** non concurrentes et les actions concurrentes réalisées en **exclusion mutuelle**. L'exécution d'actions parallèles non concurrentes permet le développement de composants d'utilité publique capables de gérer différents modes d'interaction simultanément et ainsi prévenir les actions des utilisateurs d'actions potentiellement destructrices. Par exemple, ce type de classe permet d'envisager une application de dessin collaborative capable de gérer simultanément différents modes d'édition, autorisant des utilisateurs à être dans un mode d'écriture pendant que d'autres sont dans un mode pour

effacer de l'écrit : les utilisateurs peuvent écrire et effacer localement sans concurrence. Enfin, des actions peuvent être réalisées en exclusion mutuelle pour gérer un accès concurrent à un artéfact commun et placer en attente l'utilisateur n'ayant pas eu l'exclusivité. C'est ce type de mécanisme qui est mis en œuvre au sein de la bibliothèque logicielle Diamondspin (*Waiter*) (Shen et coll., 2004).

Figure 26
Exemple de modélisation du geste coopératif de transfert de propriété à l'aide d'une machine à états.



Pour établir cette cartographie identifiant trois classes d'interactions gestuelles, j'ai étudié des dispositifs interactionnels et boîtes à outils logicielles existants permettant la mise en œuvre de telles techniques ainsi que des applications logicielles développées avec ces dispositifs et outils logiciels. Pour analyser et comparer ces techniques sur la base de leur mise en œuvre logicielle, j'ai modélisé leur comportement à l'aide de machines à états. Reprenant le scénario de geste coopératif de transfert de propriété, illustré par la Figure 24, la Figure 26 représente la machine à états associée à cette technique d'interaction : l'évolution du comportement en fonction de l'identité de l'utilisateur se traduit par des transitions dont une fraction des conditions est exprimée en fonction de cette identité. Par exemple, la transition entre l'état « *idle* » et « *waiting* » indique que seul le propriétaire de l'artéfact peut le sélectionner et le faire changer d'état : la condition `[uid==owner]` repose sur l'identité des utilisateurs pour réaliser le changement d'état.

De plus, pour compléter l'analyse de l'existant, j'ai implanté huit techniques d'interaction sensibles à l'identité de l'utilisateur tirées de la littérature, notamment de (Ryall et coll., 2006 ; Morris et coll., 2006). Afin de permettre leur comparaison, ces techniques ont également été modélisées par des machines à états.

Ces classes m'ont permis d'identifier des patrons de conception et constituent donc un guide pour spécifier et concevoir des techniques d'interaction gestuelle flexible, combinant à la fois interaction mono-utilisateur et multiutilisateurs.

2.2. MODALITÉS D'INTERACTION GESTUELLE

Dans notre quotidien, la gestuelle est un élément important de la communication car elle contribue à une meilleure compréhension entre interlocuteurs, réduisant ainsi la charge cognitive (Van den Hoven et Mazalek, 2011). Plus encore, la gestuelle joue un rôle important pour des activités collaboratives comme la résolution de problèmes ou la conception coopérative (Visser et Maher, 2011). En effet, comme l'ont mis en évidence Gutwin et Greenberg (2000), la gestuelle est un mécanisme élémentaire de

la collaboration comme la désignation (*deictic*) et la communication conséquentielle (*consequential communication*). En Interaction Homme-Machine, l'interaction gestuelle est également employée pour faciliter la communication avec un système interactif car « son pouvoir expressif permet une interaction moins contraignante et plus intuitive qu'avec les IHM traditionnelles » (Van den Hoven et Mazalek, 2011).

Mes travaux m'ont donc conduit à explorer l'interaction gestuelle, à la fois sur le plan de l'interaction entre utilisateurs et sur le plan de l'interaction avec le système interactif. J'ai étudié deux modalités d'interaction, en particulier la robustesse et fiabilité de leur mise en œuvre : l'interaction muscle-machine et l'interaction au touché, deux formes d'interaction gestuelle.

Interaction gestuelle au touché

Complémentaire à mes travaux sur l'interaction muscle-machine, un autre pan de mes travaux porte sur la mise en œuvre de modalités d'interaction basées sur la reconnaissance de gestes de la main et du touché. Ces travaux incluent la mise à disposition de briques logicielles réutilisables pour le prototypage rapide de tels systèmes interactifs. Proposer un système interactif réellement utilisable avec ce type de modalité nécessite des mécanismes sous-jacents robustes. J'ai abordé plusieurs aspects : la reconnaissance fiable et robuste du geste de la main et du touché ; la différenciation entre mains d'un même utilisateur ou entre mains de plusieurs utilisateurs, pour le geste ou le touché.

Suivant les dispositifs techniques, différentes stratégies et mécanismes sont à mettre en œuvre pour assurer une reconnaissance fiable et robuste ainsi que pour distinguer les mains interagissant avec le système. Deux approches ont été explorées : une basée sur la vision par ordinateur et une basée sur une table sensible au touché, la table DiamondTouch (Dietz et Leigh, 2001).

Figure 27
(à gauche)
Système MagicBoard, élément du projet
(à droite)
Facilitator Room

La première approche (Bérard et coll., 2001) repose sur l'exploitation de techniques de vision par ordinateur. Le dispositif technique[3] repose sur l'usage d'une caméra filmant les gestes dans une zone planaire rectangulaire blanche, comme le représente la Figure 27 (à gauche), dans l'espace de couleur (depuis, l'usage de caméras captant dans l'espace infrarouge, couplées à une caméra captant la profondeur, comme c'est le cas de la technologie Kinect, ont largement amélioré ces solutions). Le principe de vision par ordinateur consiste à analyser la séquence d'images filmée par la caméra et d'en extraire l'information pertinente pour déterminer les mains et la position des doigts. Le processus (Bérard, 1999) repose sur du traitement d'image pour filtrer (segmentation), relier les pixels significatifs appartenant à un même groupe (par exemple la main ; par analyse de

[3] Vidéo, http://iihm.imag.fr/demos/magicboard/movies/MagicTableDivX.avi

connexité), éliminer les formes ou groupes de couleur non candidats (différents de la couleur de peau par exemple ; par analyse statistique), de relier les mouvements de groupements de pixels entre images successives afin d'en extraire une gestuelle (filtre de Kalman).

Aussi, comme le montrent la Figure 27 (à gauche) et la vidéo, l'usage de jetons de couleur caractéristique a été retenu afin d'obtenir un suivi robuste et peu sensible aux variations des conditions environnementales, à savoir les variations des conditions d'illumination ambiante (naturelle ou artificielle). Cette technique a été développée pour mettre en œuvre le système MagicBoard (Bérard et coll., 2001), un tableau blanc augmenté permettant l'interaction avec du contenu physique (écrits sur le tableau blanc) et avec du contenu numérique (encre numérique à haute résolution), captée à partir de l'écrit physique) via des jetons de couleurs. Le système MagicBoard est un des éléments du projet Facilitator Room[4] (voir Figure 27 à droite) développé dans l'équipe VisMod (Alex Pentland) du MIT MediaLab. Comme le montre la Figure 27, ce système permet l'interaction multiutilisateur en situation de face à face grâce à un suivi simultané de plusieurs jetons en mouvement sur la surface. La robustesse du mécanisme permet de différencier les différents jetons et donc les différentes mains des utilisateurs en présence. Pour cela, différentes techniques d'interaction devenues désormais usuelles comme le *pinch* (rapprochement de deux doigts d'une main) ont été développées pour permettre l'interaction avec le contenu numérique.

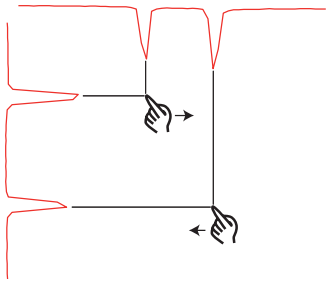


Figure 28
(à gauche) DiamontTouch ;

(à droite) Exemple de signaux produits par le contact d'un doigt sur la table

Dans le prolongement de ces travaux explorant la vision par ordinateur pour mettre en œuvre l'interaction gestuelle au touché, j'ai également abordé la question de la robustesse et de la fiabilité de la reconnaissance du **multitouche** pour des dispositifs capables de reconnaître le touché. Parmi les technologies existantes, je me suis intéressé à la table interactive DiamondTouch, une des premières tables interactives véritablement accessible et unique par sa capacité à différentier les utilisateurs (voir Figure 28 (à gauche)), facilitant le développement de systèmes interactifs multiutilisateurs. Il existe de nombreuses techniques et technologies pour construire des surfaces interactives sensibles au touché.

Certaines technologies reposent sur la vision par ordinateur via des caméras filmant dans l'espace infrarouge (FTIR : *frustrated total internal reflection*), d'autres analysent le contact électrique du doigt sur la surface selon deux méthodes dites « résistives » et « capacitives ». C'est cette dernière qui est mise en œuvre dans la table DiamondTouch. Comme le montre la Figure 28 (à droite), deux signaux sont captés par deux réseaux d'antennes qui

[4] Facilitator Room (MIT MediaLab), http://vismod.media.mit.edu/vismod/demos/facilitator-room/

constituent la table, l'un en abscisse, l'ordre en ordonnée, et se traduisent par des pics dont les sommets correspondent aux coordonnées du point de contact sur la table. Ces pics sont différentiés du bruit ambiant parasitant les signaux par seuillage.

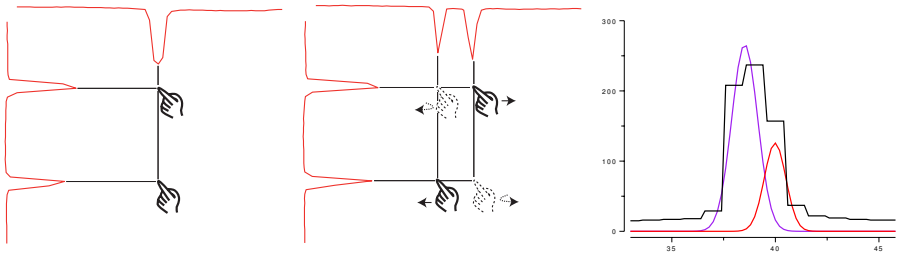


Figure 29
(a) Deux doigts sur une même ligne ;
(b) Ambiguïté pour associer les gestes ;
(c) Différenciation par signaux gaussiens

Toutefois, comme c'est le cas pour des technologies similaires (par exemple, le tableau DViT), une limite technique de cette table est qu'elle n'est pas prévue pour le multitouche. En effet, comme le montre la Figure 29 (a), lorsqu'un utilisateur positionne deux doigts alignés verticalement ou horizontalement, d'une même main ou non, le signal capté montre un seul pic traduisant l'alignement : il y a ambiguïté. Ensuite, lorsqu'un des deux doigts se déplace, le système est alors incapable de faire la bonne association (voir Figure 29 (b)).

Afin de permettre le multitouche, la solution proposée repose sur la modélisation des signaux captés par des signaux gaussiens et d'appliquer un algorithme temporel de suivi des pics, permettant ainsi de lever l'ambiguïté dans la majorité des situations (Bérard et Laurillau, 2009).

$$S(i) = \sum_{j=0}^{k-1} M_j(i)$$

$$M_j(i) = s_j * N_{\mu_j,\sigma_j}(i)$$

L'approche consiste à modéliser le signal acquis comme étant la résultante $S(i)$ d'une combinaison de signaux gaussiens $M_j(i)$ (*gaussian mixture*), notamment le bruit de fond. Ainsi, le signal émis par la table est analysé pour déterminer combien de signaux gaussiens sont combinés et les paramètres correspondants. Les paramètres des fonctions gaussiennes $N_{\mu_i,\sigma_j}(i)$ sont évalués en appliquant l'algorithme *Expectation Maximization*. Outre la modélisation du signal par des fonctions gaussiennes, la méthode considère un algorithme de suivi des pics pour relier dans le temps les captures et fiabiliser le suivi de gestes. Aussi, comme le montre la Figure 29 (c), la combinaison de ces deux algorithmes permet de différencier deux touchés (courbe violette et courbe rouge) en déplacement lorsqu'un seul pic est détecté (courbe noire en escalier).

Ce travail sur la mise en œuvre de modalités d'interaction, de leur fiabilité et robustesse est encore plus indispensable pour ensuite élaborer des techniques d'interaction multimodale et multiutilisateurs. Ce travail a fait l'objet d'une capitalisation au sein de la bibliothèque logicielle GIL[5] dédiée au prototypage rapide de systèmes interactifs. Outre les surfaces interactives, cette bibliothèque logicielle multiplateformes offre de

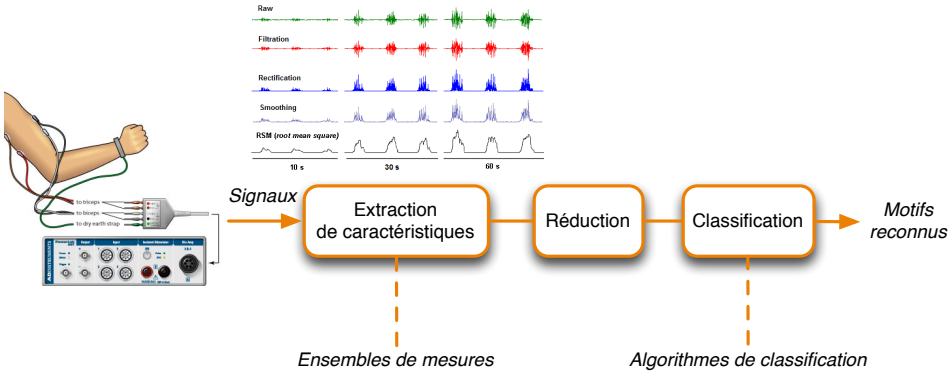
[5] gil.imag.fr

nombreux composants pour construire des systèmes interactifs post-WIMP, notamment de réalité augmentée.

Interaction muscle-machine

Complémentaires aux travaux sur l'interaction au touché, mes travaux sur l'interaction muscle-machine visent à proposer une interaction au plus proche de l'utilisateur en captant à la source ses actions et à exploiter finement sa gestuelle, comme le mouvement ou la différenciation des doigts, pour concevoir une interaction utilisateur riche. Précisément, mes travaux portent sur la mise en œuvre d'une modalité d'interaction liée à l'activité musculaire de l'avant-bras, incluant le poignet, les actions de la main dans son ensemble et des doigts individuellement.

Figure 30
Chaîne de traitement des signaux EMG



Pour détecter l'activité musculaire, je m'appuie sur une technique visant à capter les signaux ÉlectroMyoGraphiques en surface (sEMG) : il s'agit d'une mesure de différence de potentiel entre deux points sur la surface de l'épiderme, points localisés à proximité du muscle dont on veut mesurer l'activité (voir Figure 31 en haut : positionnement de capteurs sur trois différents muscles de l'avant-bras). Bien que principalement employée dans le domaine médical pour le diagnostic de patient et le contrôle de prothèses, cette technique est étudiée depuis quelques années pour proposer de nouvelles modalités d'interaction innovantes. En effet, Saponas et coll. (2008) ont démontré la faisabilité de ce type d'Interfaces Homme-Machine par l'analyse des signaux EMG produits par les muscles de l'avant-bras et la classification de gestes avec une précision de 95%. Notamment, le bracelet Myo[6] fait depuis peu l'objet d'une commercialisation pour un usage grand public. Toutefois, mon ambition est de proposer ce type de modalité d'interaction pour mettre en œuvre des techniques d'interaction gestuelle tangible qui combinent gestes et manipulation d'objets physiques (*grasping gestures*).

Cependant, la mise en œuvre d'une telle modalité d'interaction est complexe car cela implique de disposer d'un outillage capable de capter et d'interpréter les signaux EMG. Cette interprétation doit être suffisamment robuste aux variations d'utilisation (variation de la position des capteurs pendant un mouvement ou entre deux séances, variation de l'intensité musculaire pour un même geste, fatigue musculaire, etc.) mais aussi robuste aux différentes constitutions musculaires des individus. Cela conditionne fortement l'utilisabilité d'une telle modalité et donc du système interactif reposant sur son usage. Aussi, il est nécessaire de considérer la chaîne de traitement dans son ensemble (voir Figure 30), depuis la capture des signaux

[6] www.thalmic.com/en/myo/

jusqu'au traitement d'événements par le système interactif. Cette approche conjointe permet d'aiguiller le choix d'une méthode de traitement des signaux EMG par des questions d'utilisabilité.

Mes travaux[7] couvrent à la fois des recherches sur le traitement de ces signaux EMG, du point de vue de la biomécanique, et des recherches en IHM sur la question de l'utilisabilité de telles modalités d'interaction. En particulier, le travail accompli apporte des résultats pour améliorer les phases d'extraction des caractéristiques et de classification sur plusieurs plans : robustesse aux variations dans le temps, précision de la classification, choix optimal de mesures pour allier précision et temps de calcul réduit, et contournement de la phase d'apprentissage du système de classification (*training*).

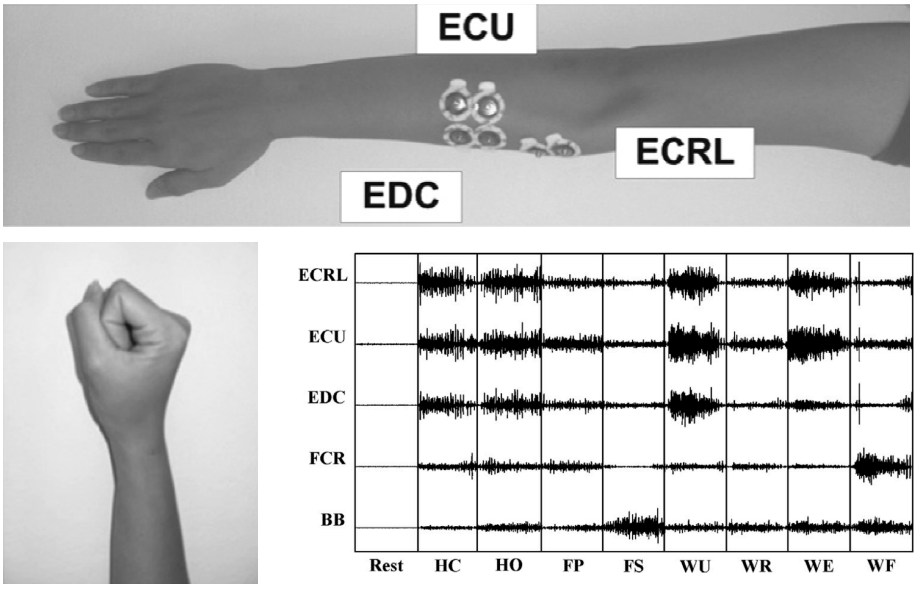


Figure 31
(en haut)
Position de capteurs sur 3 muscles ;

(en bas, à gauche)
Exemple de geste de : fermeture de la main (HC);

(en bas, à droite)
Exemple de signaux EMG captés (muscle/geste) : la section BB-HC montre le signal reçu indiquant l'activité musculaire du biceps (BB) lors de la fermeture de la main (HC)

La majorité des systèmes de reconnaissance de geste existants basés sur l'EMG requiert des phases de calibration et d'apprentissage pour compenser les différences morphologiques entre individus. Ceci limite leur utilisabilité car, pour être réellement utilisables, ces systèmes devraient être en mesure de s'adapter à différents types de muscles dont la taille varie en fonction de la morphologie des individus ainsi que d'autres caractéristiques musculaires, comme la typologie musculaire, le degré d'entrainement ou de fatigabilité. Afin de limiter la phase d'apprentissage pour calibrer le système de reconnaissance de geste et de s'affranchir des différences morphologiques entre individus, ces travaux ont étudié l'usage de techniques anthropométriques (Phinyomark et coll., 2014 ; Phinyomark et coll., 2013a), c'est-à-dire basées sur des mesures de caractéristiques physiques, comme le poids de l'individu ou le volume musculaire, pour calibrer semi-automatiquement, voire automatiquement, le système de reconnaissance. Il est envisageable de considérer des équipements portés capables de faire automatiquement ces mesures à l'image du bracelet capable de mesurer le diamètre du poignet (Cannan and Hu, 2011; Saponas

[7] Ce travail présenté a été mené en collaboration avec l'équipe SAIGA du laboratoire GIPSA-lab, avec F. Quaine, spécialiste de biomécanique et de l'EMG, principalement dans le cadre du projet TIGRE (Tangible Interaction based on Gesture REcognition). Ces travaux ont pu être réalisés grâce à la présence d'A. Phinyomark, post-doctorant intervenant pour le projet TIGRE.

et coll., 2009) ou les e-textiles (par exemple, des recherches sont menées pour intégrer des matrices d'EMG dans les vêtements (Taelman et coll., 2007)).

Une première étude consiste donc à identifier des variables anthropologiques ayant une forte corrélation avec les signaux EMG permettant de configurer un moteur de classification et de s'affranchir de la phase d'apprentissage.

Pour cela, plusieurs variables anthropologiques (poids, taille, indice de masse corporelle, circonférence du biceps, distance entre le coude et l'extrémité des doigts) ont été considérées pour déterminer des corrélations fortes avec les signaux EMG générés par les muscles de l'avant-bras (par exemple le biceps). Différentes classes usuelles de mouvements de la main ont été examinées : extension du poignet, flexion du poignet, ouverture et fermeture de la main (voir Figure 31 en bas, à gauche), etc. Pour analyser les signaux EMG (voir Figure 31 en bas, à droite) : exemples de signaux captés pour différents muscles en fonction du geste exercé), plusieurs mesures ont été prises en compte comme (voir Figure 32) l'amplitude moyenne absolue du signal (MAV, *Mean Absolute Value*) sur un signal échantillonné à 1024Hz pour une fenêtre de temps de 2 secondes (L est donc égal à 2048), ou une mesure de complexité (WL, W*aveform Length*) qui indique si le signal varie grandement ou non pour un échantillon.

Figure 32
Deux mesures de signaux EMG : MAV et WL

$$\mathrm{MAV} = \frac{1}{L}\sum_{i=1}^{L} |x_i| \qquad \mathrm{WL} = \sum_{i=1}^{L-1} |x_{i+1} - x_i|$$

Une campagne de mesure a été menée avec 20 sujets (10 hommes, 10 femmes), étalée sur 4 jours. Chaque jour, les sujets doivent répéter 15 fois 8 mouvements (ordre aléatoire) tenus pendant 2 secondes. L'analyse des mesures montre une forte corrélation (plus de 70%) entre la largeur des épaules et les signaux EMG produits pour deux muscles de l'avant-bras (biceps et extenseur ulnaire du carpe (extenseur et adducteur de la main)), traités avec les mesures MAV et WL. De plus, cette analyse montre la faible incidence sur la corrélation des variations de la condition musculaire sur plusieurs jours ainsi que des différences de positionnement des capteurs.

L'autre aspect de mes travaux (Phinyomark et coll., 2013b ; Phinyomark et coll., 2013c ; Phinyomark et coll., 2013d)concerne l'optimisation des phases d'extraction de caractéristiques et de classification (voir Figure 30) pour une meilleure robustesse aux variations, alliée à un coût réduit du temps de traitement. Ce dernier point est fortement lié aux choix du nombre de mesures et de leur complexité pour analyser les signaux EMG.

Il existe trois classes de mesures :

1. *Domaine temporel.* Les mesures sont effectuées sur le signal brut dans le domaine temporel et sont basées sur l'amplitude du signal ;
2. *Domaine fréquentiel.* Elles reposent sur l'analyse du signal dans le domaine fréquentiel et sont basées sur le spectre du signal : ce type de mesure nécessite une phase de transformation à l'aide d'une transformée de Fourier ;
3. *Domaine temporel-fréquentiel.* Elles reposent sur l'analyse du signal dans le domaine temporel et fréquentiel, permettant ainsi de localiser les

fréquences dans le temps : ce type de mesure nécessite une phase de transformation à l'aide d'ondelettes.

La première classe de mesures pour l'analyse de signaux EMG est privilégiée. En effet, différents travaux montrent que (1) les mesures dans le domaine fréquentiel ne sont pas adaptées pour couvrir tout type de signaux, notamment les signaux stationnaires et non stationnaires (moyenne et variance non stables dans le temps) ; (2) et les mesures dans le domaine temporel sont tout aussi précises que celles effectuées dans le domaine temporel-fréquentiel tout en étant plus efficaces de par l'absence de transformation et de la faible complexité des mesures. De fait, ce choix de mesures convient parfaitement à des équipements embarqués à faible capacité de calcul. De plus, les mesures relevant de la première classe ne nécessitent pas (ou peu) de réduction lors de la seconde phase, c'est-à-dire de réduire le nombre de caractéristiques identifiées par les mesures.

Pour gagner en performance et robustesse au long de cette chaîne, des combinaisons de mesures sont comparées, parmi une cinquantaine tirée de la littérature comme la mesure MAV ou WL. À ces combinaisons de mesures s'ajoute la recherche d'un algorithme de classification le plus efficace pour un ensemble donné de mesures. Huit algorithmes de classification tirés de la littérature, comme celui du plus proche voisin (kNN, *k-Nearest Neighbor*), ont été étudiés.

Cette étude s'appuie sur un corpus de mesures de signaux EMG, menées pendant 21 jours, pour des activités de la vie courante. Une combinaison de 4 mesures, principalement une mesure d'entropie d'un échantillon de signal (SampEn, mesure de similarité entre deux séries temporelles), couplée à un algorithme de classification d'analyse discriminante linéraire (LDA, *Linear Discriminant Analysis* ; algorithme de classification qui détermine l'appartenance d'un individu à une classe, et qui a la propriété d'être un algorithme simple, efficace et ayant besoin de peu d'apprentissage), permet d'atteindre une précision de 97,75% avec un temps de traitement temps réel et une performance stable.

Ces résultats constituent une base solide pour concevoir des modalités d'interaction basées sur l'EMG. En particulier, l'interaction gestuelle tangible est visée avec des gestes de prise d'objets (*grasping gestures*) pour des objets de forme et taille différentes. Une première classification entre technique d'interaction et forme d'objet a été établie[8] (Tofan, 2013). Le travail se poursuit actuellement pour détecter au niveau musculaire des signatures selon la forme et l'objet, d'identifier si la taille a une influence sur l'interaction et de compléter la classification.

[8] Ce travail a été dans le cadre du projet de recherche de Master 2 international Mosig (UJF) d'A. Tofan, en relation avec le projet TIGRE.

3. Du monocontexte au multicontextes

La troisième dimension couverte par mes travaux porte sur le contexte d'usage de l'interaction et sur la capacité du système interactif à s'adapter au changement de contexte, une autre caractéristique de la flexibilité. De nos jours, comme je le souligne en introduction, nous interagissons avec une toujours plus grande variété de dispositifs d'interaction qui ne se cantonnent désormais plus à l'ordinateur de bureau inamovible avec un utilisateur immobile, ni au triplet souris-clavier-écran comme périphériques d'interaction : nous interagissons désormais dans des contextes d'usage toujours plus variés et de moins en moins prévisibles. Se posent alors plusieurs questions, notamment comment anticiper l'imprévisibilité à la conception et doter le système interactif de moyens pour lui permettre de s'adapter au contexte d'usage. Du point de vue de l'utilisateur, se pose également la question de l'observabilité des contextes et du contrôle du changement, notamment face à un environnement social en perpétuelle évolution qui fait de plus en plus partie intégrante des systèmes.

Mes travaux abordent ces questions relatives au contexte d'usage et d'adaptation en prolongeant d'abord mes recherches sur les outils de spécification et la flexibilité au travers des modèles IOWA. Ensuite, j'aborde ces questions sous deux autres angles, en termes, d'une part, de moyens pour appréhender des interactions multiutilisateurs entre utilisateurs mobiles, pour des contextes d'usage hétérogènes et, d'autre part, de représentation du contexte social de l'interaction et des moyens donnés à l'utilisateur pour contrôler ses échanges. Donner du contrôle à l'utilisateur contribue à satisfaire le requis de fluidité interactionnelle.

3.1. SPÉCIFICATION DE L'ADAPTATION

Aborder des contextes d'usage multiples nécessite des systèmes interactifs adaptables. Le modèle IOWAState (Laurillau, 2013) constitue une proposition pour spécifier et implanter des éléments d'interfaces capables d'adapter leur comportement en fonction de l'identité d'un utilisateur ou en fonction d'un groupe d'utilisateurs. Il repose sur l'assemblage de trois familles de modèles usuels. Ce modèle a notamment été utilisé pour implanter huit techniques d'interaction pour illustrer la classification présentée dans la partie 2.2 qui identifie des classes de techniques d'interaction pour l'interaction gestuelle tangible sensible à l'identité des utilisateurs. Aussi, sur la base de cette classification, je propose plusieurs patrons de conception de modèles de machines à états hiérarchiques selon le modèle IOWAState pour concevoir des interfaces dotées de capacité d'adaptation à l'identité de l'utilisateur.

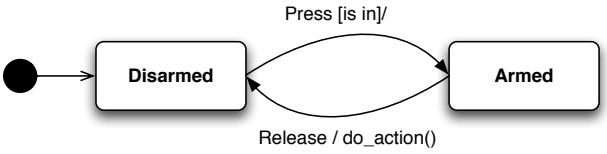


Figure 33
Exemple de machine à états hiérarchique modélisant le comportement d'un bouton

Modèle de comportement

Pour modéliser le comportement, je m'appuie sur une modalisation à base de machines à états hiérarchiques (*Hierarchical State Machine*), une évolution des machines à états. Ce type de modèle se révèle adapté (Newman, 1968) pour spécifier le comportement d'une interface utilisateur. Différents

travaux ont démontré la faisabilité et le bénéfice d'une telle approche (par exemple : Wellner, 1989 ; Blanch et Beaudouin-Lafon, 2006 ; Myers et coll., 2008), notamment sur la qualité du code produit, limitant ainsi l'effet « spaghetti » (Myers et coll., 2008).

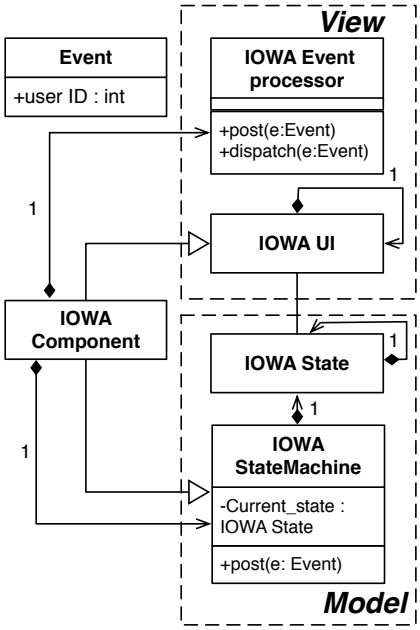
Une machine à états est une combinaison d'états et de transitions reliant les états (voir Figure 33). Conformément au langage UML pour le diagramme état-transition, les transitions sont décorées par des conditions selon la syntaxe :

```
déclencheur [garde] / effet.
```

Le déclencheur identifie un événement déclenchant la transition d'un état vers un autre. La garde correspond à un ensemble de conditions qui, si elles sont satisfaites, conditionnent le déclenchement de la transition. Outre le changement d'état, l'effet spécifie la réalisation d'une action, conséquence du déclenchement de la transition. De plus, une machine à états hiérarchiques considère qu'un état peut lui-même être décrit par une machine à états. Les points d'entrée et de sortie de l'état correspondent alors à l'état initial et à un des états finaux de la machine à états interne décrivant son comportement.

Le principal avantage des machines à états hiérarchiques est qu'elles permettent de plus facilement contrôler le risque d'explosion combinatoire. D'autres approches plus formelles mais plus complexes à aborder sont également proposées comme les réseaux de Pétri (Navarre et coll., 2009). Toutefois, les principes et propriétés que j'ai identifiés pour spécifier le comportement d'éléments d'interfaces adaptables à l'identité de l'utilisateur à l'aide de machines à états hiérarchiques sont transposables à d'autres formalismes comme les réseaux de Pétri.

Figure 34
Modèle de composant IOWA



Modèle de composant

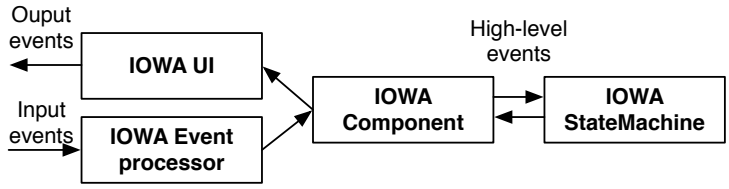
Le modèle de composant IOWA représente une structure possible pour implanter des éléments d'interfaces sensibles à l'identité de l'utilisateur. Comme le montre la Figure 34, le modèle de composant IOWA compose plusieurs aspects, s'inspirant explicitement du modèle MVC (*Model-View-*

Controller) puisqu'il identifie clairement des éléments relatifs au modèle (*Model*) et à la présentation (*View*).

Toutefois, le modèle de composant IOWA se distingue du modèle MVC sur plusieurs points :

- *Modèle = machine à états hiérarchique sensible à l'identité.* Le modèle de composant IOWA préconise une implantation du comportement du composant par une machine à états. Cette machine à états repose sur un mécanisme d'envoi d'événements pour déclencher des transitions entre deux états (via l'appel de la méthode `post(e : Event)`, voir Figure 34). Toutefois, la particularité est que (1) un événement véhicule une référence sur l'identité de l'utilisateur (classe `Event`, attribut `userID`) à l'origine de celui-ci et que (2) les conditions décorant les transitions peuvent exprimer des contraintes quant à l'identité de l'utilisateur pour être accomplies. C'est ce mécanisme qui me permet de réaliser la différentiation entre utilisateurs et de disposer d'éléments d'interfaces capables d'adapter leur comportement en fonction de cette identité.
- *Vue = fusion entrée et sortie.* L'autre particularité est de considérer l'aspect vue comme implantant conjointement la partie présentation en sortie (classe `IOWAUI`) et le traitement des entrées (classe `IOWAEventProcessor`). Cette approche repose sur un principe identique au modèle MVP (*Model-View-Presenter*), notamment mis en œuvre au sein des bibliothèques logicielles Java Swing ou Microsoft .NET. De plus, dans une approche similaire à celle développée par la bibliothèque hsmTk (Blanch et Beaudouin-Lafon, 2006), la partie vue d'un composant IOWA est une collection d'objets de type `IOWAUI` : chacun de ces objets est associé à un état décrivant le comportement du composant. Cette approche permet ainsi, outre un comportement différencié en fonction de l'identité de l'utilisateur, de proposer une présentation adaptée à chaque utilisateur.
- *Exécution parallèle de machines à états.* Afin de permettre l'entrelacement des actions des différents utilisateurs ainsi que de considérer des actions concurrentes, le modèle de composant IOWA peut embarquer plusieurs machines à états associées, une par utilisateur. Dès lors, lorsqu'un événement est reçu par le composant (via l'appel de la méthode `post(e : Event)`, voir Figure 34), celui-ci est redirigé vers la machine à états associée à l'utilisateur en fonction de l'attribut `userID`. Cette approche permet notamment de mettre en œuvre des éléments d'interfaces capables de gérer plusieurs modes d'interaction simultanément.

Figure 35
Architecture d'un composant IOWA



Modèle d'architecture

Comme le montre le Figure 35, le modèle d'architecture de composant IOWA précise l'articulation entre les constituants du composant IOWA et la chaîne de traitement des événements échangés entre ces constituants. Etant donné qu'un composant IOWA s'inspire du modèle MVP, le modèle

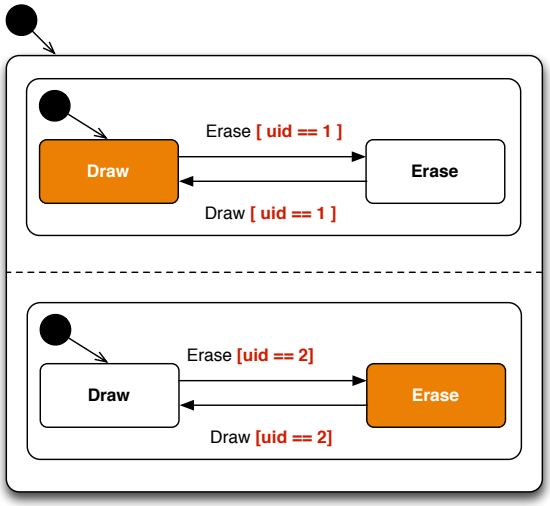
d'architecture sous-jacent met en évidence une articulation similaire des constituants en trois couches, l'élément central étant le composant IOWA assurant la liaison entre le modèle (classe `IOWAStateMachine`) et la présentation (classe `IOWAUI` et `IOWAEventProcessor`). En particulier, le composant IOWA génère des événements de plus haut niveau, échangés avec la machine à états à partir des événements de bas niveau, reçus de la vue, et correspondant aux actions des utilisateurs. De plus, les événements de haut niveau reçus par la machine à états sont également redistribués aux multiples machines à états pouvant être embarquées dans une même machine.

Le modèle de composant IOWA a fait l'objet d'une mise en œuvre logicielle dans le langage Python, basée sur la bibliothèque logicielle SWIT[9], fournissant une base pour l'implantation de machines à états hiérarchiques. Cette implantation a été employée pour développer huit composants d'interfaces conformes au modèle IOWA, illustrant les différentes classes de techniques d'interaction sensibles à l'identité de l'utilisateur (voir partie 2.1).

Illustration : actions de groupe entrelacées

La Figure 36 représente un exemple de mise en œuvre d'une technique d'interaction permettant une gestion entrelacée de modes d'interactions multiples, un par utilisateur (actions de groupe parallèles entrelacées selon la classification présentée en partie 2.1). Il s'agit d'un exemple simplifié d'un mini éditeur de dessin autorisant deux modes : le mode de dessin ou le mode pour effacer. Par exemple, lorsque l'utilisateur No 2 choisi le mode pour effacer (état « Erase » représenté par une forme remplie), seules ses actions effacent une partie du dessin. Aussi, les actions de l'utilisateur No 1 seront toujours des actions de dessin s'il n'a pas changé de mode.

Figure 36
Modèle de comportement pour des actions de groupe entrelacées sans ordonnancement



3.2. HÉTÉROGÉNÉITÉ EN SITUATION DE MOBILITÉ

Le nombre croissant de dispositifs d'interaction ne garantit plus aucune équité en ressources : l'hypothèse d'environnements homogènes pour concevoir une interface multiutilisateurs est caduque (Chalmers et Galani, 2004). Dans le même temps, c'est le signe d'une plus grande flexibilité et d'un moyen pour permettre la fluidité interactionnelle. Toutefois, les utilisateurs montrent qu'ils sont aptes à s'adapter à cette hétérogénéité et que de petites différences peuvent se révéler bénéfiques pour dynamiser une

[9] Développée par R. Blanch.

activité de groupe (Chalmers et Galani, 2004). Lorsque les différences sont trop importantes, elles deviennent alors nuisibles à la dynamique de groupe.

Apport de la multimodalité

L'interaction multimodale peut constituer un moyen pour compenser le manque d'équité interactionnelle, notamment l'usage du son pour véhiculer des informations essentielles sur le déroulement de l'activité de groupe : la conscience de groupe (*group awareness*).

Notamment, mes travaux sur l'interaction multimodale par la sonification d'interfaces pour dispositifs mobiles (voir partie 2.1) abordent également la navigation collaborative sur le web pour des contextes d'interactions multiutilisateurs hétérogènes : des utilisateurs interagissant à plusieurs soit avec l'application CoVitesse dans sa version ordinateur de bureau (Figure 37 à gauche), soit avec la version PDA de l'application avec, par conséquent, une zone d'affichage réduite (Figure 37 à droite).

Mes travaux abordent la mise en œuvre logicielle d'interactions multiutilisateurs et multimodales avec le développement de l'application PalmVitesse, une version PDA (*Personal Digital Assistant*) de l'application CoVitesse (Laurillau, 2002) (voir partie 1.1), qui exploite la modalité sonore et des techniques de visualisation pour mettre en œuvre une **interaction multimodale en sortie (visuelle et sonore)**. Pour cela, je me suis appuyé sur des techniques de **sonification** de l'interface utilisateur de l'application CoVitesse[10] pour véhiculer des informations relatives à l'activité de groupe et ainsi entretenir la conscience de groupe.

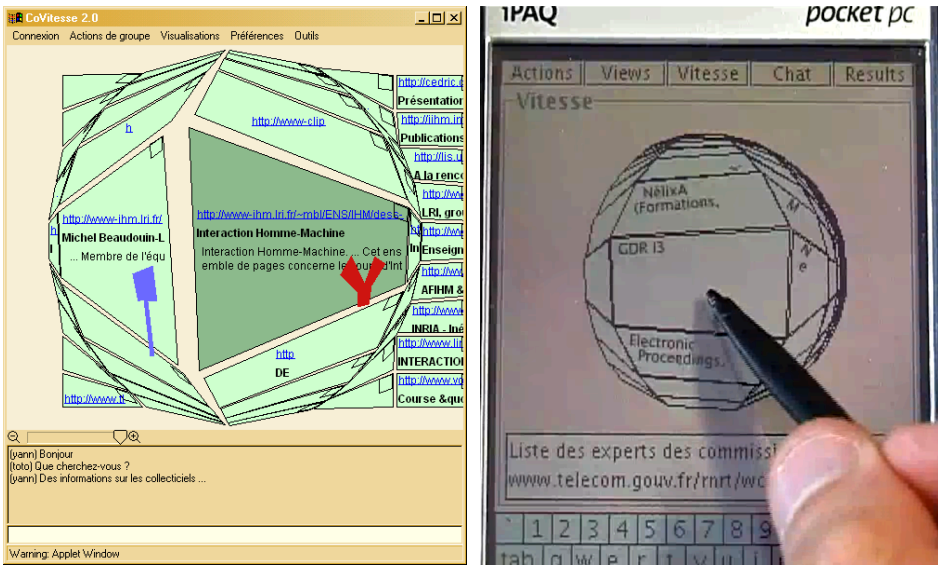


Figure 37 (à gauche) interface de CoVitesse version ordinateur de bureau ; (à droite) interface de CoVitesse version PDA

En effet, les travaux de Buxton et coll. (1994) sur l'usage du son au sein d'interfaces visuelles, qu'il soit réaliste ou synthétique, ont montré qu'il pouvait grandement améliorer l'utilisabilité, que ce soit pour une interaction mono-utilisateur ou multiutilisateurs. En particulier, le son est un bon support pour entretenir la conscience de groupe pour des interactions multiutilisateurs (Gutwin et coll., 2011) et donc véhiculer des informations périphériques sur les activités. Aussi, la sonification de l'interface de l'application CoVitesse a été réalisée pour deux aspects de l'activité multiutilisateurs :

[10] Ce travail a été réalisé en 1999 au sein de l'équipe GIST avec **S. Brewster**, spécialiste de la sonification des interfaces, de l'université de Glasgow.

L'activité de communication. Une stratégie possible pour adapter l'interface utilisateur de la version ordinateur de bureau pour un PDA consiste à diviser en espaces de travail distincts accessibles via un mécanisme d'onglets ou de menu par exemple. Alors que la version ordinateur de bureau (voir Figure 37 à gauche) présente conjointement l'espace de travail partagé (zone centrale reposant sur l'usage de techniques de visualisation type œil de poisson) et la zone de discussion en ligne (partie inférieure), la version pour PDA dissocie les deux zones en écrans distincts, accessibles via des menus (voir Figure 37 à droite). Aussi, alors que l'utilisateur de la version ordinateur de bureau peut visualiser de manière périphérique soit l'activité textuelle liée au défilement du texte, soit l'activité de navigation par le déplacement des avatars, cette perception périphérique conjointe est perdue dans la version pour PDA. Pour traduire l'activité de communication, l'application est conçue pour produire des sons véhiculant l'intensité de cette activité. Deux paramètres sont considérés pour générer des sons basés sur une banque MIDI d'instruments (en l'occurrence une corde de basse) : la vitesse de production de messages textuels et la longueur de chaque message. La durée de la note est calculée par rapport à la longueur et la vitesse est calquée sur la vitesse de réception des messages.

L'activité de navigation. La modalité sonore est également employée pour traduire l'activité liée à la navigation des utilisateurs dans l'espace d'information. Cette modalité sonore permet de transcrire l'activité lorsque l'utilisateur est dans une fenêtre autre, comme la zone de discussion. Le son produit (*sweep pad*) dépend de plusieurs paramètres : la distance absolue par rapport au focus, le déplacement en abscisse, et le déplacement en ordonnée. Précisément, le volume sonore est calculé en fonction de la distance : plus l'avatar est proche du focus, plus le volume est élevé ; inversement, plus l'avatar est éloigné, plus le volume est faible. Toutefois, un volume minimal est assuré pour traduire l'activité. Pour traduire un déplacement en abscisse, la spatialisation du son est exploitée : à gauche, au centre ou à droite. Pour traduire un déplacement en ordonnée, le son varie selon la hauteur de la note : vers les aigus pour un déplacement vers le haut ; vers les graves pour un déplacement vers le bas.

De plus, l'avantage du son est qu'il n'occupe pas d'espace visuel et constitue donc une modalité complémentaire adaptée pour des dispositifs d'interaction avec des surfaces d'affichage réduites. Elle permet aussi de pallier la faible résolution de l'écran lorsque l'utilisateur visualise et navigue dans l'espace d'information. En effet, malgré l'usage de techniques de visualisation pour représenter un maximum d'informations à l'écran, lorsque des avatars sont situés en bordure, ils ont une taille de quelques pixels, ce qui se révèle peu utilisable. Aussi, l'application est développée pour produire un son traduisant le déplacement des avatars relativement au focus de l'utilisateur.

Contenus différenciés : compromis et adaptation

Outre les modalités d'interaction, l'hétérogénéité peut se manifester par des contenus différenciés, adaptés à chaque utilisateur, présentés et accessibles dans l'interface. En effet, l'interaction sur support mobile encourage à développer des applications interactives capables d'adapter le contenu en fonction du contexte d'usage comme la localisation de l'utilisateur. Notamment dans le domaine de la visite de musées, l'adaptation à la localisation améliore la flexibilité et l'efficacité de tels guides (Ghiani et coll.,

2009) : ce moyen permet de proposer un contenu en phase avec le déroulement de la visite en rendant accessibles des contenus et informations les plus appropriés, améliorant à la fois la navigation dans les mondes réel (le musée) et numérique (l'interface du guide) : l'expérience utilisateur, dans ces deux mondes, est alors plus grande.

Cependant, la visite de musée est souvent considérée comme une expérience individuelle. Aussi, les guides électroniques sont rarement conçus pour favoriser les activités groupe. Au contraire, la visite est un acte intrinsèquement social (Falk et Dierking, 2000) : notre façon d'apprécier une exposition ou une œuvre est fortement influencée par le contexte social (Bowers et coll., 2001 ; Leinhardt et Crowley, 2003), c'est-à-dire le comportement de chacun. Dans le même temps, ce type de guide se doit de préserver l'expérience utilisateur individuelle. Il est alors nécessaire de trouver le bon compromis face au risque d'une trop grande hétérogénéité.

Le système CoCicero (voir partie 1.2) (Laurillau et Paternò, 2004a ; Laurillau et Paternò, 2004b) repose sur le système Cicero (Ciavarella et Paternò, 2003), une application mono-utilisateur qui vise à offrir des contenus numériques personnalisés en fonction de la localisation du visiteur, et des informations complémentaires à propos des œuvres présentées dans le musée. Trois types de tâches sont considérés :

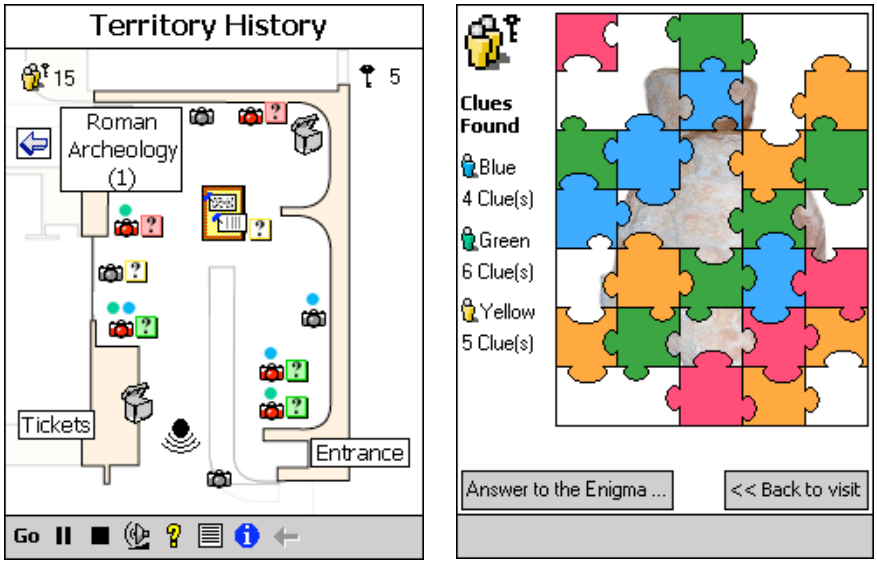


Figure 38
CoCicero :
(à gauche)
écran principal ;

(à droite)
énigme partagée

- S'orienter dans le musée : trois niveaux d'information sont disponibles : une carte du musée, une carte des sections du musée et un plan de la pièce en cours de visite, affichant une série d'icônes représentant les œuvres présentes dans la pièce, comme le montre la Figure 38 (à gauche). En sélectionnant une icône, le système affiche une photo de l'œuvre accompagnée d'informations factuelles et d'un commentaire audio. L'apparition de la photo a pour objectif de permettre à l'utilisateur de repérer l'œuvre dans la pièce et d'associer les commentaires avec la bonne œuvre. La présence de détecteurs infrarouges dans chaque pièce du musée permet au système de mettre à jour automatiquement le plan de la pièce en cours de visite et d'activer les commentaires audio associés.
- Contrôler l'interface en agissant, par exemple, sur les paramètres relatifs aux commentaires audio.

- Accéder aux informations générales sur le musée selon trois niveaux de granularité (musée, section et pièce).

A tout moment, le système est capable d'indiquer à l'utilisateur où il se trouve. Toutefois, le visiteur peut modifier le contenu et choisir d'afficher le plan d'une autre pièce.

Pour faire profiter les visiteurs d'une expérience utilisateur complète, le guide CoCicero permet à la fois de mener une visite en groupe et individuellement. Un compromis a donc été recherché pour offrir des contenus différenciés (hétérogénéité) tout en assurant une cohérence pour les activités de groupe. Cet aspect est en partie traité dans la mise en œuvre d'une architecture conciliant les deux aspects (voir partie 1.2) : individuel et collectif. Je l'ai également traité au travers de la réalisation de l'IHM et des fonctionnalités et interactions offertes aux visiteurs.

Pour ce second aspect, outre les informations sur le musée, le guide repose sur l'exploitation de jeux éducatifs individuels et collaboratifs. En effet, le jeu constitue une approche intéressante pour promouvoir les interactions et le partage entre les visiteurs (Hindmarsh et coll., 2002). De plus, une telle approche permet de préserver le caractère individuel de la visite, qui est, comme l'ont remarqué les auteurs du projet Sotto Voce (Grinter et coll., 2002), un aspect important.

Aussi, dans ce guide, deux types de jeux sont proposés : le premier met en œuvre une collaboration explicite entre les compagnons tandis que le second, reposant sur une activité individuelle, participe indirectement à l'activité de groupe. Le premier type de jeu est la résolution d'énigme commune (voir Figure 38 à droite), présentée sous forme de chasse au trésor : les visiteurs ont pour objectif de récolter une série d'indices permettant de découvrir une œuvre mystère. Cette énigme se caractérise par un quizz et un puzzle représentant l'œuvre à découvrir. Aussi, ce jeu nécessite que les visiteurs partagent et communiquent sur ce qu'ils ont vu. Le second type de jeu est une série de jeux, tels que le quizz, qui permettent à chacun de récolter un indice dès que la solution a été trouvée. Cet indice, qui se matérialise sous la forme d'une bribe d'information textuelle ou d'un morceau de puzzle, doit permettre au groupe de résoudre ultérieurement l'énigme commune. Ce type de jeu permet à chaque utilisateur de réguler sa visite et de ne pas être soumis au mettre rythme que ses compagnons.

A l'issue de la visite, et afin d'encourager les visiteurs à essayer la version collaborative du système, le noms des participants s'affichent sur le tableau des scores sachant que le score sera d'autant plus élevé que la collaboration aura été importante.

3.3. CONTEXTE SOCIAL : DONNER DU CONTRÔLE A L'UTILISATEUR

Avec l'avènement des réseaux sociaux, le contexte d'usage s'élargit du fait de la part grandissante de la dimension sociale de l'interaction utilisateur. En effet, nous sommes désormais en permanence connectés avec autrui (via facebook, twitter, instagram, et bien d'autres) pour la moindre tâche à accomplir avec le système interactif. Le corolaire est que nous laissons des traces de notre activité, volontaire ou non. Comme le souligne J. Grudin (2010), la question de la vie privée devient cruciale. Se pose donc la question du contrôle laissé à l'utilisateur pour contrôler et visualiser les traces de son activité. Mes travaux abordent ces questions au travers du principe

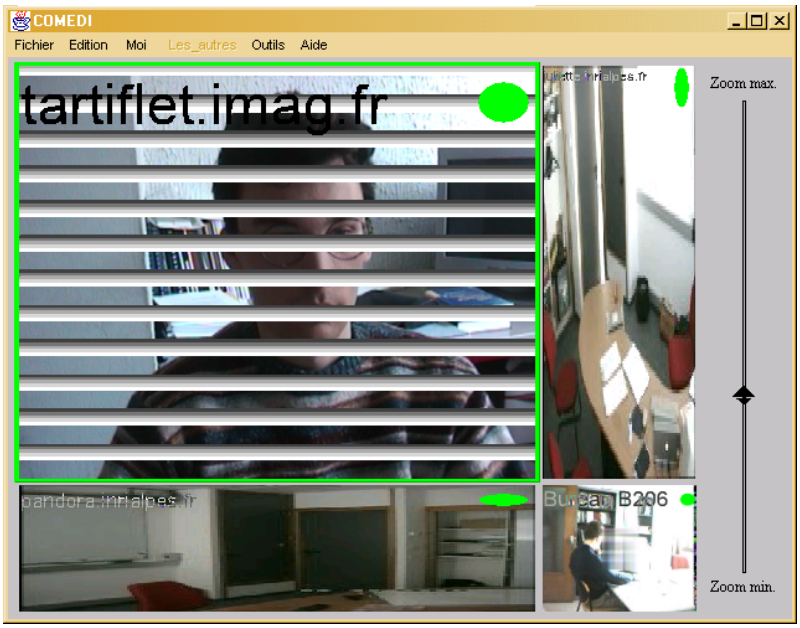
d'observabilité publiée pour traiter du contrôle et au travers de techniques de visualisation pour représenter les échanges en contexte social.

Niveaux d'observabilité : donner du contrôle

L'observabilité (Dix et coll., 2004 ; Gram et Cockton, 1996) est l'un des principes forts de l'utilisabilité d'un système interactif : elle caractérise un système offrant la possibilité à un utilisateur d'évaluer l'état interne du système à l'aide de représentations au travers de son IHM. Dix et coll. (2004) déclinent ce principe d'observabilité pour les systèmes interactifs multiutilisateurs en termes d'observabilité transversale (*feedthrough*) : il caractérise un système interactif multiutilisateurs capable de rendre observable à chacun les actions réalisées par autrui. Ce principe ne se limite pas aux actions. Il s'applique également à la publication d'informations privées, propres à chaque utilisateur (Coutaz et coll., 1999 ; Salber, 1995) : l'**observabilité publiée**.

Ce principe d'observabilité publiée est intimement lié à celui du respect de la vie privée (*privacy*) et sous-entend que l'utilisateur dispose de moyens pour contrôler cette publication, notamment par l'usage de **mécanismes de filtrage** (Coutaz et coll., 1999). Ce dernier à la base de l'application MediaSpace COMEDI (Coutaz et coll., 1999) et mes travaux y apportent une contribution par la participation à son développement logiciel.

Figure 39
MediaSpace COMEDI

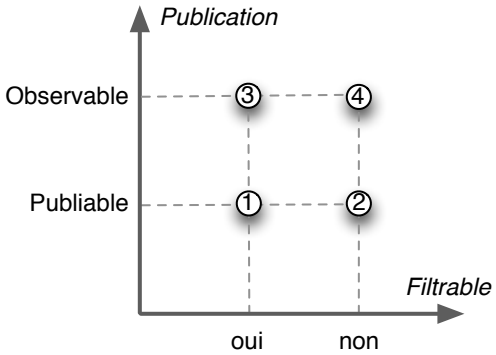


Un MediaSpace (par exemple, Dourish et Bly, 1992) est un collecticiel ayant pour but d'entretenir un lien entre les membres d'un groupe répartis dans différents lieux, à différentes échelles : bureau, étage, bâtiment, etc. Il s'agit également de fournir un moyen pour favoriser des rencontres informelles entre membres d'un groupe pour une plus grande cohésion. L'application COMEDI (Coutaz et coll., 1999) explore cette classe d'application sous l'angle du respect de la vie privée et des moyens possibles à mettre en œuvre pour donner le contrôle à l'utilisateur afin de préserver sa vie privée tout en diffusant suffisamment d'information pour maintenir une conscience de groupe. Comme le montre la Figure 39, l'aspect filtrage a été exploré pour traiter les images vidéos émises en continu de membres filmés en permanence dans leur bureau : métaphore du store vénitien, filtre d'activité

laissant apparaître les contours des objets en mouvement, flou sur une zone, etc.

De ces recherches sur l'usage du filtrage comme moyen de préserver la vie privée, couplées à mes travaux sur la navigation sociale sur le web (Laurillau et Nigay, 2002a; Laurillau et Nigay 2002b; Laurillau et Nigay, 2003; Laurillau, 2002; Laurillau et Nigay, 2000), je propose une extension du principe d'observabilité publiée pour y intégrer le principe de filtrage. Précisément, j'identifie plusieurs formes d'informations observables et de mécanismes de contrôle de publication (Laurillau, 2002) :

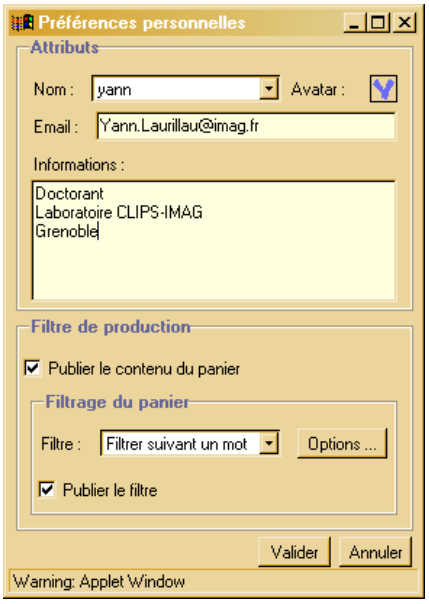
Figure 40
Niveaux d'observabilité publiée



- *Observable :* ce qui est souhaitable que l'utilisateur perçoive ;
- *Publiable :* ce qu'un utilisateur décide de rendre observable ;
- *Filtrable :* la forme de ce qui est observable. Les informations ainsi filtrées ont la visibilité choisie par l'utilisateur.

J'en déduis quatre niveaux d'observabilité (voir Figure 40) (Laurillau 2002) selon deux axes Publication et Filtrage. Ces niveaux constituent un guide pour le concepteur afin de préciser quel niveau d'observabilité est nécessaire pour que l'utilisateur puisse réaliser sa tâche tout en préservant son espace privé et celui des autres.

Figure 41
Covitesse : paramétrage du filtrage



J'ai appliqué ces quatre niveaux à mes travaux sur la navigation sociale sur le web avec le développement de l'application CoVitesse : elle a été conçue pour permettre à chaque utilisateur de paramétrer le degré d'observabilité lié à son activité individuelle ainsi qu'à son activité au sein d'un groupe. Ce

principe est étendu au groupe et, selon son organisation, différentes règles de publication et de filtrage peuvent être appliquées. La Figure 41 (à droite) montre la fenêtre permettant à un utilisateur de paramétrer le filtre à appliquer sur les résultats collectés au cours de la navigation. Le paramétrage permet également de rendre observable la politique de filtrage si l'utilisateur le désire. De plus, l'application CoVitesse repose sur la plateforme Clover et son implantation se conforme au modèle d'architecture Clover (voir partie 1.2).

Selon ces quatre niveaux, le MediaSpace COMEDI correspond au niveau publiable et filtrable.

Ces deux exemples illustrent l'extension du principe d'observabilité pour proposer une IHM fluide permettant à l'utilisateur de mener une activité individuelle conjointement à une activité de groupe en offrant des mécanismes, ici publication et filtrage, proposant un compromis entre ces modes d'interaction. Toutefois, ces approches reposent sur des interfaces utilisateurs préconçues ne disposant pas de mécanismes d'adaptation pour tenir compte, par exemple, de l'hétérogénéité des dispositifs ou d'un changement de contexte.

Visualisation des échanges en contexte social

La question du contrôle par l'utilisateur de l'observabilité publiée, appliquée à la navigation sociale sur le WWW, se prolonge et est approfondie par les travaux sur la proposition d'IHM Perlée (Martins-Baltar et coll., 2012 ; Martins-Baltar et coll., 2013; Martins-Baltar et coll., 2014; D'orazio et coll., 2014) visant, notamment, à révéler et rendre observable à l'aide de techniques de visualisation la dimension sociale de l'activité (individuelle ou en groupe). Pour cela, je m'appuie sur (1) une proposition conceptuelle, la taxonomie des entités à valeur sociale ajoutée (VSA), et sur (2) une technique d'interaction promouvant une visualisation interactive de l'activité dans son contexte social.

En particulier, ces travaux ont été appliqués dans le domaine médical, précisément la gastroentérologie qui est une spécialité reposant grandement sur la production d'images d'endoscopie digestive. Avec l'importance et l'usage croissant des TIC dans le monde de la santé, ce domaine met bien en évidence le caractère critique et crucial de cette question du contrôle et de la traçabilité face à un besoin de partage socioprofessionnel des données médicales (entre praticiens, avec les patients et ou la famille) : il s'agit d'assurer une meilleure qualité de soins, tout en restant en conformité avec le cadre légal régissant le secret médical. En effet, les réseaux socio-médicaux sont un aspect important de l'activité médicale (Eysenbach, 2008). De plus, à partir d'une étude de terrain menée auprès de professionnels de la santé (gastroentérologie) et d'une étude de cas (rétroconception) menée sur une application logicielle de santé, deux requis de premier ordre pour les IHM de santé (potentiellement généralisable) sont identifiés, guidant mon approche :

Promouvoir la dimension sociale des activités. Des visualisations de ces réseaux semblent opportunes et permettent, par exemple, d'observer quels sont les contacts les plus proches, les données médicales partagées avec des confrères, ou de visualiser qui est le médecin référent d'un patient.

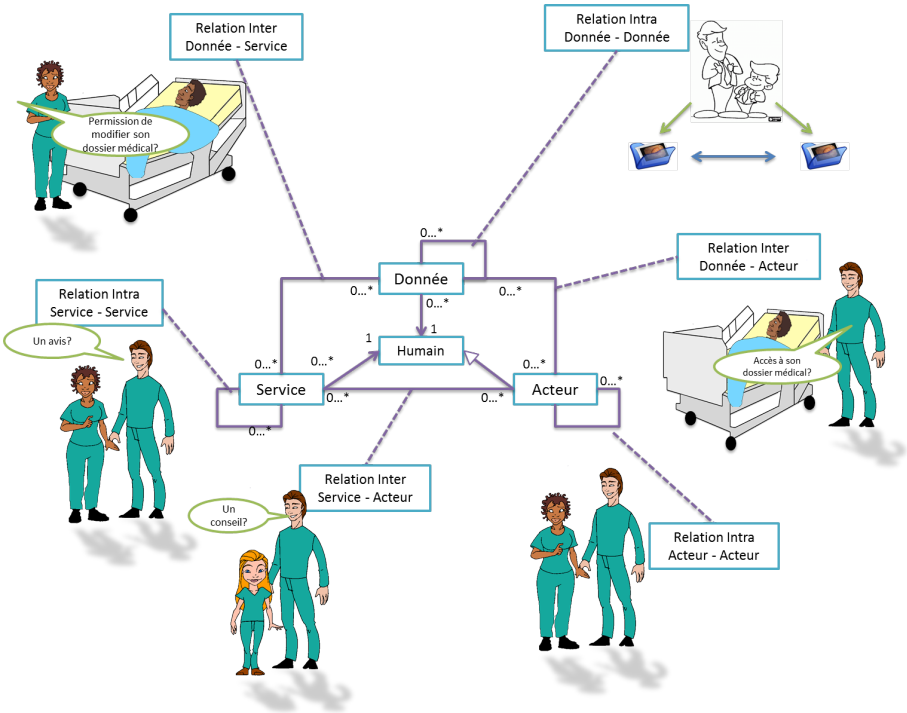
Assurer le partage et la traçabilité. Le besoin de support pour le partage des données médicales et la communication entre professionnels de la santé devient essentiel.

Des entités de type **donnée**, **acteur** et **service** sont considérées comme fondements de cette proposition conceptuelle, en support à la réalisation d'une tâche en contexte. Ces entités font référence aux concepts usuels de tâche, d'objet du domaine et d'agent, tels qu'ils sont traités dans les domaines de l'IHM et du travail collaboratif assisté par ordinateur (TCAO) (Van Welie et coll., 1998). Notamment, le service est abordé comme étant un des moyens disponibles pour accomplir la tâche. La Figure 42 représente une taxonomie de ces entités et des relations entre ces entités sous la forme d'un diagramme de classe UML (chaque entité étant simplement représentée par une classe).

L'entité **donnée** dénote l'information manipulée par un acteur lorsqu'il accomplit une tâche. Dans le contexte médical, une donnée peut être aussi bien un dossier médical qu'une image endoscopique, etc. Des services en ligne, tels qu'un service d'analyse d'image, de reconstruction 3D, ou de la visualisation d'un historique d'examens, sont consommateurs de données.

L'entité **acteur** dénote les individus impliqués dans l'interaction avec le système : ils manipulent des données et réalisent des tâches. Dans le domaine médical, les acteurs peuvent être, par exemple, des patients, des médecins, des secrétaires médicales ou des étudiants en médecine. Des services en ligne peuvent être mis à leur disposition, tels qu'un système de *workflow* en santé ou un service de suivi de dossier médical.

Figure 42
Taxonomie des entités à valeur sociale ajoutée



L'entité **service** dénote le moyen employé par des acteurs pour accomplir une tâche. Sachant qu'une tâche est classiquement décomposée en sous-tâches, cette approche s'applique également à la notion de service : les services sous-tendent la réalisation des sous-tâches. Appliqué au médical, il peut simplement s'agir du suivi du patient. Par exemple, un médecin

gastroentérologue, afin d'accomplir un suivi médical, doit réaliser de nombreuses tâches et faire appel aux services idoines comme pratiquer des examens, produire des images endoscopiques, échanger avec des patients ou des confrères, etc.

Aussi, ces entités pivots sont enrichies pour prendre en compte leur valeur sociale et mettre en évidence les relations entre entités sur la base de cette valeur. On peut également parler de statut social. De ces constats, une taxonomie d'entités à valeur sociale ajoutée (VSA) est proposée, enrichissant les entités pivots (donnée, acteur et service), mettant en évidence leurs relations **intra** et **extra** (voir Figure 42).

Les relations entre les entités pivots constituent une information supplémentaire, aboutissant ainsi aux entités à valeur sociale ajoutée. Les relations **intra** font référence aux relations entre entités de même nature :

Acteur-Acteur : les acteurs sont les utilisateurs du système. Dans notre cas d'étude, plusieurs acteurs ont été identifiés : les patients, les médecins, les secrétaires, et les étudiants en médecine. La notion de proximité sociale entre acteurs (par exemple, la collaboration fréquente entre deux médecins, la relation patient-médecin) peut enrichir les entités dites classiques et être considérée comme des entités VSA.

Donnée-Donnée : ces entités ne sont pas simplement liées entre elles, comme les liens hiérarchiques ou les relations dans une base de données, mais ont une réelle valeur sociale ajoutée. La décoration d'un lien par son type en est un exemple (relation généalogique entre le dossier médical d'un père et celui de son fils pour la prédiction de pathologie héréditaire).

Service-Service : étiqueter les services avec les bonnes pratiques des experts est un exemple d'entité VSA. Cette information supplémentaire permettrait de proposer aux médecins des recommandations. Aussi, des services en ligne de recommandations, comme une aide au diagnostic, reposant sur des systèmes experts peut s'appliquer à ces entités VSA.

Les relations **extra** font référence entre entités de nature différente :

Donnée-Acteur : les acteurs produisent et manipulent les données pour atteindre leurs objectifs. Les données, quant à elles, donnent des informations sur ces différents acteurs. Dans notre cas d'étude, un dossier médical (donnée) est associé au patient (acteur) et bien sûr au médecin (acteur) qui traite ce patient. Ce lien peut également donner lieu à des relations indirectes entre deux acteurs : ces acteurs ne se connaissent pas forcément mais ils partagent de l'information. Des services comme le partage d'information (confidentiel ou anonyme) ou recherchant des corrélations entre dossiers médicaux peuvent s'appliquer à ces entités VSA.

Donnée-Service : un service est socialement lié à une donnée dès lors que le statut social d'une donnée influence l'accomplissement d'une tâche. Par exemple, la mise à jour (tâche) d'un dossier patient (donnée) par un autre médecin que le médecin référent est possible seulement si ce dossier est partagé. À l'inverse, une tâche peut implicitement apporter une valeur sociale à une donnée, comme la production (tâche) d'une image (donnée) qui sera ajoutée au dossier médical disponible par le patient et le médecin, ou le partage (tâche) d'une image (donnée) pour une demande d'expertise. Des services comme un tableau de bord de l'activité médicale (dernier

examen, dernier rendez-vous, dernière mise à jour, etc.) peuvent s'appliquer à ces entités VSA.

Service-Acteur : un acteur peut employer des services différents pour accomplir une même tâche, dans des contextes sociaux différents. Par exemple, dans le domaine médical, la prise de décision peut se faire lors d'une réunion pluridisciplinaire, impliquant différents spécialistes, ou lors d'un examen, impliquant seulement le chirurgien. Des services spécifiques, tels que faire une demande d'avis ou partager une donnée médicale, peuvent s'appliquer à ces entités VSA.

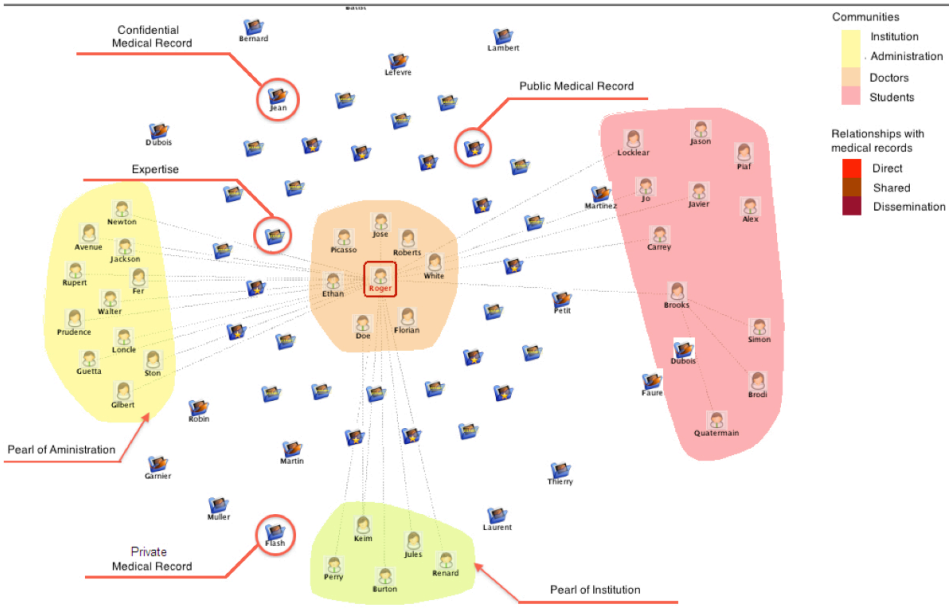


Figure 43
Visualisation des relations VSA Acteur-Donnée et Acteur-Acteur

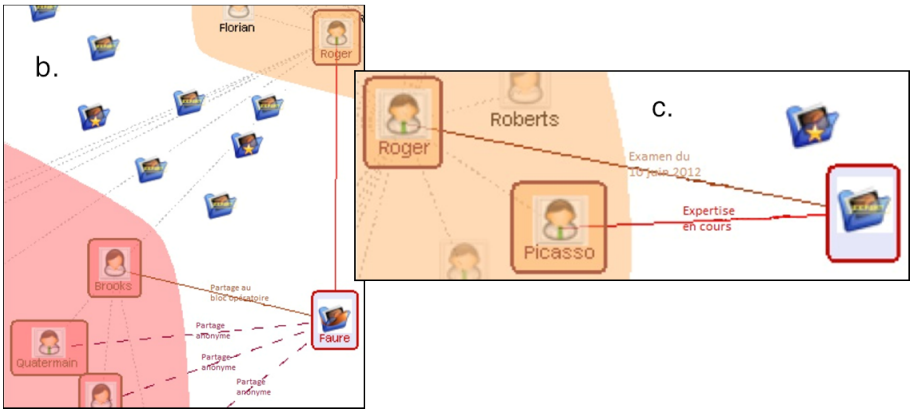


Figure 44
(à gauche)
Zoom sur le partage d'une donnée entre acteurs ;

(à droite)
Zoom sur l'état d'un partage

Sur la base de cette taxonomie, une visualisation du réseau socio-professionnel du médecin est proposée, en lien avec les données médicales (images endoscopiques et dossier patient). Cette visualisation met en évidence les relations VSA **intra** Acteur-Acteur et les relations **extra** Acteur-Données. Les travaux issus de la visualisation d'information, comme les travaux de B. Schneiderman ou J.D. Fekete, constituent une source d'inspiration pour concevoir ces vues. Aussi, la visualisation à la base de l'interface utilisateur ainsi conçue représente un réseau social sous forme de graphe (voir Figure 43), représentation la plus adaptée d'après les travaux en visualisation de Riche et Fekete (2010) : elle fait l'objet d'une preuve de

concept par l'implantation d'un prototype[11] programmé en Java. Précisément, ce prototype est appliqué à la gastroentérologie et cible les praticiens comme classe d'utilisateurs. La Figure 43 illustre cette représentation pour un **médecin fictif**, le Dr. Roger : le graphe représente le réseau socio-professionnel du médecin complété par l'ensemble des dossiers médicaux gérés par celui-ci. Notamment, y sont représentées les relations directes et indirectes entre les différents acteurs (lignes grises), ainsi que les diverses communautés liées au Dr Roger (médecins, étudiants, etc.).

La distance entre les icônes indique la proximité sociale entre les différents acteurs. Par exemple, la communauté des médecins (communauté au centre de la Figure 43) est la communauté qui a la plus grande proximité avec le Dr Roger. Comme le montre la Figure 44 (à droite), les liens entre les dossiers et les acteurs sont décorés par une étiquette pour indiquer le statut de l'activité médicale en cours. Trois catégories sont considérées : (1) tâche la plus récente ou prochaine tâche à accomplir (par exemple, le dernier ou prochain examen) ; (2) nature du partage d'une donnée médicale ; et (3) traçabilité.

Un dossier médical est une collection d'informations médicales de natures différentes : images endoscopiques, comptes rendus médicaux, ordonnances, résultats d'examen biologiques, etc. Le statut social des dossiers médicaux, visible sur l'icône, peut-être : (1) privé – seul le médecin référent et le patient y ont accès ; (2) confidentiel – le dossier privé est partagé avec le médecin traitant par exemple, ou pour une demande d'expertise ; (3) public – le dossier est anonyme et publié pour un usage pédagogique. Il est possible d'y inclure la représentation des liens d'ordre social entre les différents dossiers, comme les liens entre dossiers pour des patients appartenant à une même famille.

La traçabilité est mise en œuvre au travers de la visualisation en indiquant la nature du partage sur un lien, direct ou indirect. Par exemple, le partage d'un dossier médical privé avec un confrère, un étudiant en stage ou l'administration, est direct. Inversement, la dissémination d'un dossier médical rendu anonyme (par exemple, entre deux étudiants) est un partage indirect. Ce type de partage est représenté par des lignes en pointillé. Ce partage s'appuie sur un mécanisme de contrôle qui repose sur la notion de statut social, visualisé ici par une icône représentant le degré de partage (privé, confidentiel ou public). Par exemple, la Figure 44 (à gauche) montre que le dossier de Mr. Faure est partagé directement entre le Dr Roger et l'étudiante Brooks (ligne grise entre ces deux acteurs). Toutefois, ce dossier est partagé indirectement, donc anonyme, entre le Dr. Roger et trois autres étudiants via l'étudiante Brooks : des lignes hachurées violettes sont représentées entre le dossier médical de Mr. Faure et ces trois étudiants (par exemple, Simon). Cela indique que ces étudiants ont un accès partiel à des données médicales non confidentielles car rendues anonymes. Une telle visualisation offre un moyen puissant pour représenter le réseau de diffusion. Ce point est crucial, comme l'ont confirmé les médecins partenaires, dans le processus de soin. Il convient de noter que ces trois aspects (traçabilité, partage et contrôle) sont traités sous l'angle de l'IHM au

[11] Le code source du prototype a fait l'objet d'un dépôt auprès de l'APP (Agence pour la Protection des Programmes) et fait l'objet d'un transfert industriel auprès la société Yansys medical.

travers de ces visualisations, fondés sur l'hypothèse qu'une mise en œuvre soit possible au niveau fonctionnel.

Une évaluation qualitative a été menée avec quatre praticiens et un expert en systèmes d'information de santé. Elle confirme la pertinence du concept et de la proposition d'implantation, tout en pointant les aspects à approfondir (Martins-Baltar et coll., 2012).

Concept de Perle

Ces travaux sur la représentation de la dimension sociale des activités, au travers des entités VSA, se prolongent par la proposition et la formalisation du concept de Perle pour caractériser un contexte d'usage (Martins-Baltar et coll., 2013). Ce concept me permet également d'aborder la question de la dynamique des contextes. Il est sous-jacent à la métaphore d'IHM Perlée (Martins-Baltar et coll., 2012 ; Martins-Baltar et coll., 2014).

En effet, un contexte d'usage peut être caractérisé par un ensemble d'entités pivots (**acteur, donnée, service**) et par les relations entre ces entités en support à la réalisation d'une tâche : services disponibles pour accomplir la tâche (i.e. moyens), données nécessaires et accessibles, et acteurs associés.

Formellement, une Perle est une collection hétérogène d'entités pivots organisées en graphe. C'est un arc d'un hypergraphe (arc à plusieurs sommets) dont les sommets sont des entités pivots : donnée, acteur et service. Ces sommets sont organisés en graphe au sein de cet arc : un tel graphe entre entités de natures différentes permet d'expliciter les relations **extra** entre les entités VSA. Le choix de classes d'entités pivot (par exemple, une classe d'acteur définie par un rôle) ainsi que le choix d'un type de relations entre entités (par exemple, des images médicales relatives à des acteurs appartenant à une même famille) et les métriques (par exemple, une proximité sociale entre acteurs par rapport à une pathologie) définit la structuration en graphe de ces entités au sein de l'arc et précise ainsi le contexte d'usage défini par une Perle. L'hypergraphe représente donc un ensemble de Perles, c'est-à-dire un ensemble de contextes d'usages.

Les entités pivots de même nature (acteur, donnée ou service) sont organisées en graphe selon un type de relation et de métriques. Plusieurs graphes peuvent être produits à partir d'un même ensemble d'entités (i.e. une vue sur les entités) suivant le choix d'une classe d'entités, d'un type de relation et de métriques. Trois types de graphes sont considérés : graphe d'Acteurs, graphe de Données et graphe de Services.

Graphe d'acteurs. Un graphe d'acteur met en évidence les relations sociales entre acteurs : c'est la notion usuelle de réseau social. Les acteurs en sont les sommets. Les relations entre acteurs constituent les arcs et la proximité entre acteurs est déterminée selon le poids des arcs ou de la distance entre acteurs. Un graphe d'acteurs permet ainsi d'expliciter des relations intra entre entités Acteurs, et exprime les quatre caractéristiques suivantes : sens de transmission de l'information (unilatérale ou bilatérale) ; structure du groupe (par exemple hiérarchique) ; proximité sociale ; dynamique de groupe (évolutions des relations).

Graphe de données. Un graphe de données est composé d'entités de type données constituant les sommets. Un graphe de Données permet d'expliciter les relations intra entre entités Données, comme la similarité.

Toutefois, cette formulation n'aborde pas la question complexe de l'agrégation des données et de leur hétérogénéité.

Graphe de services. Un graphe de services est un graphe dont les sommets sont des entités de type service. Un graphe de Services permet d'expliciter les relations intra entre entités Services, et exprime les deux caractéristiques suivantes : dépendance (lien entre deux services) ; disponibilité (utilisateur, environnement et plateforme).

La représentation des données médicales dans le contexte du réseau socio-professionnel du médecin (voir partie précédente) est une illustration de la mise en œuvre du concept de Perle et d'IHM Perlée pour les entités données et acteurs : les structures sous-jacentes sont des ensembles de données (les dossiers médicaux) et d'acteurs (communautés) dont la visualisation met en évidence les relations (relations au sein de l'hypergraphe). Dans ces travaux, plusieurs classes de Perles et des représentations conceptuelles sont proposées (ces points sont détaillés dans Martins-Baltar et coll., 2013).

Ce travail fait actuellement l'objet de prolongements avec l'étude de l'évolution du contexte d'usage dans le temps.

Conclusion et perspectives

Que seront les systèmes interactifs de demain ?

Les systèmes interactifs font clairement partie intégrante de nos activités quotidiennes : nous avons dépassé le stade de l'acceptation pour entrer dans une relation symbiotique avec la technologie (Brangier et coll., 2009), comme le pressentait Licklider (1960). Prolongements de nous-mêmes, ils augmentent nos capacités cognitives, physiques, sensori-motrices, etc., gagnent en intelligence et en connaissance sur nous-mêmes par leurs capacités perceptives. Ces systèmes symbiotiques se caractérisent par leur **polyvalence** avec une couverture fonctionnelle en perpétuelle expansion et par leur **polymorphisme**, les rendant capables d'adaptation à toute forme et dispositif d'interaction et d'environnements. **Le risque est une expérience utilisateur fragmentée** (Santosa et Wigdor, 2013).

La **fluidité interactionnelle** est le sujet fédérateur de mes travaux : une propriété des systèmes symbiotiques que je définis comme la capacité à s'adapter en tout contexte d'usage tout en préservant une interaction cohérente, continue, unifiée. Dans le second chapitre, au regard de la structure des systèmes symbiotiques (Shiratori et coll., 2012), je défends que la fluidité interactionnelle se caractérise selon trois dimensions relatives aux modalités d'interaction, à l'interaction mono-utilisateur et multiutilisateur, et à la multiplicité des contextes d'usage. Je contribue à la mise en œuvre de ce requis de fluidité interactionnelle : c'est mon sujet de recherche.

Les travaux synthétisés dans ce document s'intéressent à l'ingénierie logicielle de la fluidité que je traite du point du vue système en termes de **flexibilité logicielle**. En effet, comme le souligne Olsen (2007), les outils et modèles logiciels existants, comme les architectures logicielles, sont insuffisants. Les concepteurs ont besoin d'outils adaptés qui apportent de la flexibilité pour concevoir les systèmes interactifs de demain : les systèmes symbiotiques.

1. Rappel des contributions

Du mono-utilisateur au multiutilisateur

La première partie de mes contributions porte sur la dimension relative aux interactions mono-utilisateur et multiutilisateurs : cette partie montre qu'exprimer de la flexibilité au sein d'un système interactif s'anticipe dès la conception et nécessite des outils ayant ce pouvoir d'expression. En particulier, je m'intéresse à la **polyvalence** du système interactif dans sa capacité à couvrir conjointement des interactions mono-utilisateur et multiutilisateurs au travers de deux classes d'outils : les outils de spécification de l'interaction que sont les notations à base de modèles de tâches ; et les outils de conception logicielle que sont les modèles d'architecture logicielle. J'adopte une démarche consistant à revisiter et étendre les outils existants, notamment pour intégrer des aspects liés à l'interaction multiutilisateurs, afin d'assurer une continuité conceptuelle et d'offrir un cadre unifié cohérent.

Cette partie nous enseigne que le pouvoir expressif des outils est crucial. En effet, la phase de spécification est rendue plus complexe car les situations d'interaction ne sont plus uniques (par exemple, au bureau) mais multiples : le concepteur ne peut plus envisager toutes les situations possibles et doit anticiper l'imprévisible. En contrepartie, le système gagnera en flexibilité. Plus encore, l'interaction multiutilisateurs ne se déroule plus dans un cadre strict, figé par une organisation, mais est plus une affaire d'opportunités avec une dépendance plus ou moins forte entre les différents acteurs.

Aussi, j'identifie plusieurs leviers de flexibilité dès la phase de spécification et formule plusieurs propositions :

- En identifiant des classes de tâches de groupe couvrant un continuum entre activités strictement organisées vers plus d'opportunisme, ce que j'illustre par quatre classes de navigation collaborative sur le web ;
- En anticipant l'imprévisibilité de la répartition des rôles métier dans la spécification des tâches par le biais de la notation COMM, grâce notamment au concept de rôle interactif ;
- En exprimant plus finement le couplage (dépendance entre acteurs) autant du point de vue de la tâche métier que de l'interaction concrète.

Je montre également qu'il est possible d'introduire plus de flexibilité dans la conception logicielle du système interactif au travers des modèles d'architecture. En effet, en revisitant des modèles usuels (ARCH, Dewan, PAC et PAC*), je propose plusieurs stratégies pour introduire de nouveaux points de flexibilité dans ces architectures et ainsi considérer un système plus polyvalent, toujours sur le plan des interactions mono-utilisateur et multiutilisateurs :

- Au niveau macro, par un découpage des couches logicielles selon les trois classes de fonctionnalités couvertes par le modèle fonctionnel du trèfle des collecticiels (production, communication et coordination) ;
- Au niveau micro, par une généralisation de l'approche à agents à l'ensemble d'une architecture en couche ;
- Par augmentation longitudinale d'une architecture en couche structurant un système mono-utilisateur avec des composants relevant de l'interaction multiutilisateur, avec des stratégies de recomposition du contrôleur de dialogue et des composants relevant de l'interaction.

Du monomodal au multimodal

La seconde partie de mes contributions traite de la mise en œuvre d'interactions multimodales afin de satisfaire le requis de fluidité interactionnelle dans sa dimension relative aux relations symbiotiques entre les utilisateurs et le système. En effet, Brangier et coll. (2009) identifient la multimodalité comme un des moyens pour permettre au système symbiotique d'agir comme accélérateur opératoire.

Le besoin de flexibilité dans les outils tel que je l'ai traité du point de vue de l'interaction multiutilisateurs s'applique également à l'interaction multimodale, plus encore lorsque je l'aborde conjointement avec l'interaction multiutilisateur. Ainsi, je m'intéresse au caractère **polymorphe** de tels systèmes. En effet, je considère l'interaction multimodale à la fois pour des systèmes mono-utilisateur et pour des systèmes multi-utilisateurs à plusieurs échelles : depuis leur spécification (gros grain) jusqu'à la mise en œuvre individuelle des modalités d'interaction (grain fin).

Notamment, je montre dans cette partie, au travers de la notation COMM, comment raisonner tant au niveau de l'interaction abstraite qu'au niveau de l'interaction concrète tout en garantissant une représentation unifiée et une continuité conceptuelle. En effet, avec COMM, la spécification de l'interaction multimodale repose sur :

- Le concept de tâche modale permettant de spécifier une action atomique, relative à un langage d'interaction, couplée à un dispositif d'interaction ;
- Le concept de rôle interactif ;
- La spécification d'un modèle unique combinant la spécification de l'interaction abstraite et de l'interaction concrète. Ces caractéristiques de la notation permettent ainsi l'expression conjointe d'interactions multiutilisateurs et multimodales.

Outre la possibilité de raisonner à différents niveaux d'abstraction, la notation propose deux niveaux de granularité pour exprimer l'interaction multimodale grâce à une extension du jeu des opérateurs temporels sur la base des relations de Allen (1983). Cette approche permet, d'une part, de composer des modalités d'interaction selon les propriétés CARE (Complémentarité, Assignation, Redondance, Equivalence) (Nigay, 1994) et, d'autre part, contribue à renforcer le caractère unificateur de la notation et sa continuité conceptuelle.

À l'échelle d'une modalité d'interaction, appliquée à l'interaction au touché, la démarche entreprise avec la notation COMM (recherche de cadre unificateur et continuité conceptuelle) se prolonge au travers d'une taxonomie identifiant des classes d'interaction combinant interaction mono-utilisateur et interaction multiutilisateurs sur la base de l'identité des utilisateurs. Cette taxonomie constitue un guide pour introduire de la flexibilité dans la spécification de ce type de modalités d'interaction grâce à leur capacité à s'adapter à l'identité de l'utilisateur.

Cette partie nous enseigne également que considérer la flexibilité sur le plan de la spécification de l'interaction multimodale et multiutilisateurs au travers d'outils idoines ne suffit pas, et que satisfaire le requis de fluidité interactionnelle, outre proposer une grande diversité de modalités d'interaction, repose aussi sur la mise en œuvre de modalités d'interaction **fiables et robustes**. Précisément, une partie de mes travaux portent sur

cette question dans la mise en œuvre de deux types de modalités d'interaction gestuelle : au touché et musculaire basée sur l'analyse de signaux ElectroMyoGraphiques (EMG). Pour des modalités d'interaction gestuelle au touché, la technique proposée permet de distinguer de façon plus robuste le multitouche en modélisant avec des signaux captés par une table interactive. Pour le second type, deux approches complémentaires sont proposées rendant plus utilisable ce type de modalité grâce à :

- Une plus grande précision dans la reconnaissance du geste ;
- Une méthode pour réduire voire éliminer les mécanismes d'apprentissage sous-jacents au processus de reconnaissance de geste.

Du monocontexte au multicontextes

La dernière partie de mes travaux aborde la fluidité interactionnelle dans sa dimension relative au contexte d'usage. Les contextes d'usage se multiplient à l'envie du fait de la variété des dispositifs d'interaction et de leur succès auprès des utilisateurs, de la multiplication des services en ligne, et de l'accès ubiquitaire qui en est fait par les usagers : le requis de fluidité en est d'autant plus exacerbé. Aussi, les interfaces plastiques constituent une réponse pour accompagner le changement de contexte par adaptation : les travaux développés dans cette dernière partie s'inscrivent dans cette filiation.

Dans un premier temps, j'aborde cette question de l'adaptation de l'interaction, toujours sur le plan des outils au travers du modèle IOWAState, à mi-chemin entre la spécification de l'interaction et la conception logicielle. Le modèle IOWAState préconise un modèle de composant d'interface et une démarche pour spécifier le comportement d'un tel composant pour adapter l'interaction en fonction de l'identité de l'utilisateur. En fonction du contexte de l'interaction, le comportement de l'interface, en entrée et en sortie, s'adapte en fonction des utilisateurs interagissant simultanément avec le système.

Je contribue aussi à la question du contexte par des principes et mécanismes qui permettent aux utilisateurs d'appréhender la multiplicité des contextes dans un cadre social, c'est-à-dire avec interaction directe (formelle) ou non (informelle) avec les autres utilisateurs. D'abord, je m'intéresse au risque d'hétérogénéité interactionnelle dans des situations d'interaction multiutilisateurs. En effet, l'équité de ressources n'est plus forcément garantie. Aussi, pour appréhender cette éventualité d'hétérogénéité (dispositifs interactionnels différents, contenus différenciés), je montre par la réalisation concrète de deux applications que la multimodalité et un compromis entre contenus différenciés et contenus partagés permettent de minimiser l'impact de ce manque d'équité sur la qualité de l'interaction multiutilisateurs.

Ensuite, je m'intéresse à la question du contrôle accordé à l'utilisateur pour maîtriser son image dans différents contextes. Pour cela, je propose des principes pour caractériser les niveaux de contrôle d'observabilité, c'est-à-dire de ce que chacun laisse entrevoir de lui-même aux autres. J'étends cette question du contrôle par la proposition d'un patron de conception, reposant sur la taxonomie d'entités VSA (Valeur Sociale Ajoutée) et d'une technique d'interaction promouvant une visualisation interactive de l'activité dans son contexte social pour la spécification d'IHM représentant le contexte social de l'interaction. En particulier, ce patron est appliqué en gastroentérologie.

Au travers des entités VSA, ce travail se prolonge par la proposition et la formalisation du concept de Perle pour caractériser un contexte d'usage.

2. Perspectives de recherche

En perspectives de mes travaux, je vais approfondir l'étude des systèmes symbiotiques au travers des propriétés de fluidité interactionnelle et de flexibilité logicielle, prolongeant ainsi les travaux initiés ces dernières années. Sur la première (fluidité), je vise des recherches vers une interaction toujours plus « naturelle », au plus proche de l'utilisateur. Pour la seconde (flexibilité), je souhaite « mettre » plus d'humain dans les systèmes, par la conception de systèmes interactifs dotés de capacités de conception centrée humain.

Fluidité interactionnelle : vers une interaction plus naturelle

L'avenir est aux systèmes symbiotiques, dans une relation devenant fusionnelle entre l'homme et la machine. En effet, reprenant les constats posés par Brangier et Hammes-Adelé (2011), développer l'impression que la technologie est un prolongement naturel de l'utilisateur et qu'elle confère des sentiments de simplicité et de maîtrise incite à concevoir et mettre en œuvre des modalités d'interaction au plus proche de l'utilisateur. C'est dans cette tendance que s'inscrivent les recherches dans le domaine des interfaces utilisateurs naturelles (NUI, *Natural User Interfaces*). Cette tendance[1] inclut également les modalités d'interaction en entrée de nouvelle génération captant à la source l'activité physiologique de l'utilisateur comme les mesures de l'activité cérébrale via les interfaces BCI (*Brain-Computer Interfaces* : interfaces reposant sur le traitement de signaux EEG (ElectroEncephaloGraphique)).

Dans cette optique, un premier pan de mes perspectives concerne la détection de l'intention de l'utilisateur, permettant d'aborder la question de l'adaptation de l'interaction. Ces perspectives sont un prolongement de mes travaux sur l'interaction gestuelle, basés sur la reconnaissance de gestes à partir de signaux EMG (ElectroMyoGraphique) produits par l'activité musculaire des muscles de l'avant-bras. Cette ligne de recherche me permettra aussi d'approfondir mes travaux sur l'adaptation au contexte d'usage. En effet, l'analyse de l'activité humaine, à savoir l'activité musculaire de l'avant-bras, est une des sources possibles d'information pour construire un système interactif proactif, doté de capacités pour s'adapter aux besoins de l'utilisateur. Plus encore, les signaux EMG peuvent être exploités pour prendre en compte les changements sensori-moteurs (comme la fatigue musculaire) afin d'adapter l'interaction.

En outre, cette perspective relative à l'interaction gestuelle ouvre également des pistes pour aborder l'interaction multiutilisateurs sous un autre angle. En effet, des interactions reposant sur l'interaction gestuelle tangible (geste avec manipulation d'objet) peuvent faciliter la collaboration et la communication, en exploitant les capacités humaines de communication non-verbale, un aspect clef pour l'échange des idées entre concepteurs (Kim et Maher, 2008). Aussi, la détection de l'intention constitue une piste

[1] Par exemple, voir le projet MindSee (FP7) qui étudie les systèmes symbiotiques reposant sur les interfaces BCI (Brain-Computer Interfaces).

prometteuse pour contribuer et renforcer la communication sur un plan non verbal.

Prolongeant la coopération initiée avec le laboratoire GIPSA-lab (équipe SAIGA, F. Quaine), j'ai commencé des recherches abordant ces perspectives au travers du projet pluridisciplinaire et exploratoire TAPIOCA (Labex Persyval). Mes premiers travaux étudient la conception de techniques d'interaction basées sur la reconnaissance de gestes de prise et manipulation d'objets, avec des objets aux formes (comme un cube, un cylindre, ou une pyramide) et tailles variées.

Flexibilité : vers des environnements de conception centrée humain

La promesse des systèmes symbiotiques, c'est-à-dire des systèmes interactifs hautement dynamiques capables de s'adapter à de nouvelles pratiques et de proposer spontanément de nouvelles fonctionnalités (on parle d'exaptation), rend la tâche du concepteur encore plus ardue. En effet, nous interagissons avec une variété croissante de dispositifs d'interaction qui ne se cantonnent plus à l'ordinateur de bureau inamovible face à un utilisateur mobile, ni au triplet souris-clavier-écran comme périphériques d'interaction : nous interagissons désormais dans des contextes d'usage toujours plus variés et de moins en moins prévisibles. Aussi, certains choix de conception ne peuvent plus être faits par le concepteur lors des phases du cycle de vie logicielle du système. Dès lors, il faut considérer la délégation de certains choix de conception soit à l'utilisateur final (*end-user programming*), soit au système.

Aussi, un second pan de mes perspectives concerne la conception de systèmes interactifs dotés de capacités de conception, prolongeant ainsi mes travaux initiés sur les outils de spécification de l'interaction et les outils de conception logicielle. Se posent alors plusieurs questions, notamment comment anticiper l'imprévisibilité à la conception et doter le système interactif de moyens pour lui permettre, au-delà de s'adapter au contexte d'usage, de s'augmenter en réalisant lui-même une conception centrée humain.

L'intégration de ces deux lignes de recherche tend vers une symbiose totale de l'écosystème physique-numérique-humain, jusqu'à la dynamique des rôles d'utilisateur et de concepteur.

Bibliographie

Les publications pour lesquelles je suis auteur sont marquées en gras.

(Allen, 1983) Allen, J.F. Maintaining Knowledge About Temporal Intervals. Commun. ACM 26, 11. ACM Press, 1983, 832–843.

(Bass et coll., 1992) Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., Seacord, R., Sheppard, S. and Szczur, M. A Metamodel for the Runtime Architecture of an Interactive System. ACM Special Interest Group Computer-Human Interface bulletin (SIGCHI), 24(1). ACM Press, 1992, 32-37.

(Beaudouin-Lafon, 2004) Beaudouin-Lafon, M. Designing interaction, not interfaces. Conference Advanced Visual Interfaces (AVI'04). ACM Press, 2004, 15–22.

(Bederson et coll., 2004) Bederson, B.B., Clamage, A., Czerwinski, M.P., and Robertson, G.G. DateLens: A Fisheye Calendar Interface for PDAs. ACM Trans. Comput.-Hum. Interact. 11, 1 (2004), 90–119.

(Bederson et coll., 1996) Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics. Journal of Visual Languages & Computing 7, 1 (1996), 3–32.

(Begole et coll., 1999) Begole, J., Rosson, M.B., and Shaffer, C.A. Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-sharing Systems. ACM Trans. Comput.-Hum. Interact. 6, 2. ACM Press, 1999, 95–132.

(Bérard et coll., 2001) **Bérard, F., Pentland, S., and Laurillau, Y. Augmentation d'un tableau blanc par des techniques de Vision par Ordinateur. Vidéo présentée à la journée AFIHM de la conférence ASTI, (2001).**

(Bérard, 1999) Bérard, F. Vision par ordinateur pour l'interaction homme-machine fortement couplée. PhD, UJF, 1999, 218 pages.

(Bérard et Laurillau, 2009) **Bérard, F. and Laurillau, Y. Single User Multitouch on the DiamondTouch: From 2 x 1D to 2D. International Conference on Interactive Tabletops and Surfaces (ITS'09). ACM Press, 2009, 1–8.**

(Blanch et Beaudouin-Lafon, 2006) Blanch, R. and Beaudouin-Lafon, M. Programming rich interactions using the hierarchical state machine toolkit. Proceedings of the working conference on Advanced visual interfaces, (2006), 51–58.

(Brangier et coll., 2009) Brangier, E., Dufresne, A., and Hammes-Adelé, S. Approche symbiotique de la relation humain-technologie : perspectives pour l'ergonomie informatique. Le travail humain Vol. 72, 4. PUF, 2010, 333–353.

(Brangier et Hammes-Adelé, 2011) Brangier, É. and Hammes-Adelé, S. Beyond the Technology Acceptance Model: Elements to Validate the Human-Technology Symbiosis Model. In M.M. Robertson, ed., Ergonomics and Health Aspects of Work with Computers. Springer Berlin Heidelberg, 2011, 13–21.

(Bowers et coll., 2007) Bowers, J., Bannon, L., Fraser, M., et al. From the Disappearing Computer to Living Exhibitions: Shaping Interactivity in Museum Settings. In N. Streitz, A. Kameas and I. Mavrommati, eds., The Disappearing Computer. Springer Berlin Heidelberg, 2007, 30–49.

(Buxton et coll., 1994) Buxton, W., Gaver, W.W., and Bly, S. Auditory Interfaces: The Use of Non-Speech Audio at the Interface. 1994.

(Calvary, 2007) Calvary, G. Plasticité des Interfaces Homme-Machine. Thèse d'Habilitation à Diriger des Recherches préparée au Laboratoire d'Informatique de Grenoble (LIG), Université Joseph Fourier. 2007, 97 pages.

(Calvary et coll., 2003) Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying Reference Framework for multi-target user interfaces. Interacting with Computers 15, 3. Elsevier, 2003, 289–308.

(Calvary et coll., 1997) Calvary, G., Coutaz, J., and Nigay, L. From Single-user Architectural Design to PAC*: A Generic Software Architecture Model for CSCW. Conference on Human Factors in Computing Systems (CHI'97). ACM Press, 1997, 242–249.

(Cannan et Hu, 2011) Cannan, J.A.R. and Hu, H. Automatic Circumference Measurement for Aiding in the Estimation of Maximum Voluntary Contraction (MVC) in EMG Systems. In S. Jeschke, H. Liu and D. Schilberg, eds., Intelligent Robotics and Applications. Springer Berlin Heidelberg, 2011, 202–211.

(Carlis et Konstan, 1998) Carlis, J.V. and Konstan, J.A. Interactive Visualization of Serial Periodic Data. Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, ACM (1998), 29–38.

(Chalmers et Galani , 2004) Chalmers, M. and Galani, A. Seamful interweaving: heterogeneity in the theory and design of interactive systems. Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS'04). ACM Press, 2004, 243–252.

(Ciavarella et Paternò, 2003) Ciavarella, C. and Paternò, F. Design Criteria for Location-Aware, Indoor, PDA Applications. Conference Human-Computer Interaction with Mobile Devices and Services (MobileHCI'03). Springer Berlin Heidelberg, 2003, 131–144.

(Clerckx et coll., 2005) Clerckx, T., Luyten, K., and Coninx, K. DynaMo-AID: A Design Process and a Runtime Architecture for Dynamic Model-Based User Interface Development. In R. Bastide, P. Palanque and J. Roth, eds., Engineering Human Computer Interaction and Interactive Systems. Springer Berlin Heidelberg, 2005, 77–95.

(Cockburn et coll., 2009) Cockburn, A., Karlson, A., and Bederson, B.B. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. ACM Comput. Surv. 41, 1 (2009), 2:1–2:31.

(Coutaz et Nigay, 2012) Coutaz, J. and Nigay, L. Multimodalité et plasticité des interfaces homme-machine en informatique ambiante : concepts et espaces de conception. In F.S., J-M. Ogier P. Marquis, ed., Information Interaction Intelligence le point sur le i3. Editions Cépaduès, 2012, 179–214.

(Coutaz et coll., 1999) Coutaz, J., Bérard, F., Carraux, E., and Crowley, J. Early experience with the mediaspace CoMedi. Conference Engineering for Human-Computer Interaction (EHCI'99). Springer, 1999, 57–72.

(Coutaz et coll., 1995) Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., and Young, R.M. Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. INTERACT. 1995, 115–120.

(D'orazio et coll., 2014) Bartoli, A., Baetz, A., Beorchia, S., Calvary, G., Chabane, Y., Chadebecq, F., Collins, T., Laurillau, Y., Martins-Baltar, L., Mohamad, B., Ponchon, T., Rey, C., Tilmant, C., and Torti, S. Multimodal and Multimedia Image Analysis and Collaborative Networking for Digestive Endoscopy. Innovation and Research in BioMedical Engineering (IRBM) 35, 2. Elsevier, 2014, 88–93.

(Dewan, 1995) Dewan, P. Multiuser architectures. Conference EHCI'95. Chapman & Hall, 1995, 247–270.

(Dietz et Leigh, 2001) Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. ACM symposium on User interface software and technology (UIST'01). ACM Press, 2001, 219–226.

(Dix et coll., 2004) Dix, A., Finlay, J., Abowd, G.D., and Beale, R. Human computer interaction. Pearson Prentice Hall, Harlow ; Munich [u.a.], 2004, 834 pages.

(Dourish et Bly, 1992) Dourish, P. and Bly, S. Portholes: Supporting Awareness in a Distributed Work Group. Conference on Human Factors in Computing Systems (CHI'92). ACM Press, 1992, 541–547.

(Dragicevic et Huot, 2002) Dragicevic, P. and Huot, S. SpiraClock: A Continuous and Non-intrusive Display for Upcoming Events. CHI '02 Extended Abstracts on Human Factors in Computing Systems, ACM (2002), 604–605.

(Druin et coll., 1997) Druin, A., Stewart, J., Proft, D., Bederson, B., and Hollan, J. KidPad: a design collaboration between children, technologists, and educators. Conference on Human factors in computing systems (CHI'97). ACM Press, 1997, 463–470.

(Ellis et Wainer, 1994) Ellis, C. and Wainer, J. A Conceptual Model of Groupware. Conference on Computer Supported Cooperative Work (CSCW'94). ACM Press, 1994, 79–88.

(Eysenbach, 2008) Eysenbach, G. Medicine 2.0: social networking, collaboration, participation, apomediation, and openness. Journal of medical Internet research 10, 3 (2008), e22.

(Falk et Dierking, 2000) Falk, J.H., and Dierking, L.D., Learning From Museums, *Visitor Experiences and the Making of Meaning*, 2000, 288 p., Rowman & Littlefield.

(Ghiani et coll., 2009) Ghiani, G., Paternò, F., Santoro, C., and Spano, L.D. UbiCicero: A location-aware, multi-device museum guide. Interacting with Computers 21, 4 (2009), 288–303.

(Gram et Cockton, 1996) Gram, C. and Cockton, G. Design Principles for Interactive Software. Springer, 1996, 272 pages.

(Green, 1990) Green, T.R.G. The cognitive dimension of viscosity: A sticky problem for HCI. Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction, North-Holland Publishing Co. 1990, 79–86.

(Grinter et coll., 2002) Grinter, R., Aoki, P., Hurst, A., Szymanski, M., Thornton, J., and Woodruff, A. Revisiting the Visit: Understanding How Technology Can Shape the Museum Visit. Conference ACM CSCW'02. ACM Press, 2002, 146-155.

(Grudin, 2006) Grudin, J. A missing generation: office automation/information systems and human-computer interaction. Interactions 13, 3. ACM Press, 2006, 58–61.

(Grudin, 2012) Grudin, J. Punctuated equilibrium and technology change. Interactions 19, 5. ACM Press, 2012, 62–66.

(Grudin, 2010) Grudin, J. CSCW: time passed, tempest, and time past. Interactions 17, 4. ACM Press, 2010, 38–40.

(Gutwin et coll., 2011) Gutwin, C., Schneider, O., Xiao, R., and Brewster, S. Chalk sounds: the effects of dynamic synthesized audio on workspace awareness in distributed groupware. ACM conference on Computer supported cooperative work (CSCW'11). ACM Press, 2011, 85–94.

(Gutwin et Greenberg, 2000) Gutwin, C. and Greenberg, S. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). IEEE, 2000, 98–103.

(Gutwin et Greenberg, 1998) Gutwin, C. and Greenberg, S. Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. Computer Supported Cooperative Work (CSCW'98). ACM Press, 1998, 207–216.

(Hascoët, 1998) Hascoët, M. Analytical Versus Empirical Evaluation of Spatial Displays. CHI 98 Cconference Summary on Human Factors in Computing Systems, ACM (1998), 257–258.

(Hewagamage et coll., 1999) Hewagamage, K.P., Hirakawa, M., and Ichikawa, T. Interactive visualization of spatiotemporal patterns using spirals on a geographical map. 1999 IEEE Symposium on Visual Languages, 1999. Proceedings, (1999), 296–303.

(Hindmarsh et coll., 2002) Hindmarsh, J., Heath, C., vom Lehn, D., and Cleverly, J. Creating Assemblies: Aboard the Ghost Ship. ACM Conference on Computer Supported Cooperative Work (CSCW'02). ACM Press, 2002, 156–165.

(Huot et Lecolinet, 2006) Huot, S. and Lecolinet, E. SpiraList: A Compact Visualization Technique for One-handed Interaction with Large Lists on Mobile Devices. Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles, ACM (2006), 445–448.

(Johnson et coll., 1988) Johnson, P., Johnson, H., Waddington, R. et Shouls, A. Task-Related Knowledge Structures: Analysis, Modeling and Application. People and Computers IV. BCS HCI, 1988, pp. 35–62.

(Jourde, 2011) Jourde, F. Collecticiel et Multimodalité : spécification de l'interaction, la notation COMM et l'éditeur e-COMM. PhD, 2011, 301 pages.

(Jourde et coll., 2010a) Jourde, F., Laurillau, Y., and Nigay, L. COMM Notation for Specifying Collaborative and Multimodal Interactive Systems. ACM Symposium on Engineering Interactive Computing Systems (EICS'10). ACM Press, 2010, 125–134.

(Jourde et coll., 2010b) Jourde, F., Laurillau, Y., and Nigay, L. e-COMM, Un éditeur Pour Spécifier L'Interaction Multimodale Et Multiutilisateur. International Conference of the Association Francophone D'Interaction Homme-Machine (IHM'10). ACM Press, (2010), 225–228.

(Jourde et coll., 2009) Jourde, F., Laurillau, Y., and Nigay, L. Collecticiels: Neuf Degrés De Couplage. Conference on Association Francophone D'Interaction Homme-Machine (IHM'09). ACM Press, 2009, 265–272.

(Jourde et coll., 2008a) Jourde, F., Laurillau, Y., Moran, A., and Nigay, L. Towards Specifying Multimodal Collaborative User Interfaces: A Comparison of Collaboration Notations. In T.C.N. Graham and P. Palanque, eds., Interactive Systems. Design, Specification, and Verification (DS-VIS'08). Springer Berlin Heidelberg, 2008, 281–286.

(Jourde et coll., 2008b) Jourde, F., Laurillau, Y., Nigay, L., and Moran, A. Conception de Systèmes Collaboratifs Multimodaux: Analyse Comparative De Notations. International Conference of the Association Francophone D'Interaction Homme-Machine (IHM'08). ACM Press, 2008, 121–128.

(Kim et Maher, 2008) Kim, M.J. and Maher, M.L. The Impact of Tangible User Interfaces on Designers' Spatial Cognition. Human–Computer Interaction 23, 2. 2008, 101–137.

(Laurillau, 2013) Laurillau, Y. IOWAState: Implementation Models and Design Patterns for Identity-aware User Interfaces Based on State Machines. ACM Symposium on Engineering Interactive Computing Systems (EICS'13). ACM Press, 2013, 59–68.

(Laurillau et Paternò, 2004a) Laurillau, Y. and Paternò, F. Supporting museum co-visits using mobile devices. Conférence *Internationale Mobile Human-Computer Interaction (MobileHCI'04)*. Springer, 2004, LNCS 3160, 451-455.

(Laurillau et Paternò, 2004b) Laurillau, Y. and Paternò, F. CoCicero: Un Système Interactif pour la Visite Collaborative de Musée sur Support Mobile. Conférence *Francophone Interaction Homme-Machine (IHM' 04)*. ACM Press, 2004, 101-108.

(Laurillau et Nigay, 2003) Laurillau, Y. et Nigay, Laurence. CoVitesse : a Groupware Interface for Collaborative Navigation on the WWW. Conférence *Internationale HCI International (HCII'03)*, Theory and Practice (Part II), Vol. 2. Lawrence Erlbaum, 2003, 954-958.

(Laurillau et Nigay, 2002a) Laurillau, Y. and Nigay, L. Clover Architecture for Groupware. Conférence *Computer Supported Cooperative Work (CSCW'02)*. ACM Press, 2002, 236-245.

(Laurillau et Nigay, 2002b) Laurillau, Y. and Nigay, L. Le modèle d'Architecture Clover pour les Collecticiels. Conférence *Francophone Interaction Homme-Machine (IHM'02)*. ACM Press, 2002, 113-120.

(Laurillau, 2002) **Laurillau, Y. Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe: le modèle et la plate-forme Clover. PhD, 2002, 232 pages.**

(Laurillau et Nigay, 2000) **Laurillau, Y. and Nigay, L. Modèle de Navigation Collaborative Synchrone pour l'Exploration des Grands Espaces d'Information. Conférence *Francophone Interaction Homme-Machine (IHM' 00)*. CRT ILS&ESTIA, 2000, 121-128.**

(Laurillau, 1999) **Laurillau, Y. Synchronous Collaborative Navigation Techniques on the WWW. Conférence *ACM Conference on Human Factors and Computing Systems (CHI'99)*. ACM Press, 1999, 308-309.**

(Leinhardt et Crowley, 2003) Leinhardt, G. and Crowley, K. Objects of learning, objects of talk: Changing minds in museums. Perspectives on object-centered learning in museums. Lawrence Erlbaum, 2003, 301–324.

(Licklider, 1960) Licklider, J.C.R. Man-Computer Symbiosis. IRE Transactions on Human Factors in Electronics HFE-1. 1960, 4–11.

(Lim, 2004) Lim, Y.-K. Multiple Aspect Based Task Analysis (MABTA) for User Requirements Gathering in Highly-contextualized Interactive System Design. Conference on Task Models and Diagrams (TAMODIA'04). ACM Press, 2004, 7–15.

(Mackinlay et coll., 1994) Mackinlay, J.D., Robertson, G.G., and DeLine, R. Developing Calendar Visualizers for the Information Visualizer. Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, ACM (1994), 109–118.

(Mackinlay et coll., 1991) Mackinlay, J.D., Robertson, G.G., and Card, S.K. The perspective wall: Detail and context smoothly integrated. Proceedings of the SIGCHI conference on Human factors in computing systems, (1991), 173–176.

(Martins-Baltar et coll., 2014) **Martins-Baltar, L., Laurillau, Y., and Calvary, G. Pearly UIs for Cloud Computing: First Experience in Health Care IT. Cloud Computing and Digital Media: Fundamentals, Techniques, and Applications. CRC Press, Taylor & Francis Group, 2013.**

(Martins-Baltar et coll., 2013) **Martins-Baltar, L., Laurillau, Y., and Calvary, G. Débridons l'interaction homme-machine pour une meilleure qualité des soins. Ingénierie des systèmes d'information 18, 6. Lavoisier, 2014, 113–139.**

(Martins-Baltar et coll., 2012) **Martins-Baltar, L., Laurillau, Y., and Calvary, G. IHM perlées pour le Cloud Computing : premiers retours d'expérience en informatique médicale. Conférence francophone sur l'Interaction Homme-Machine. ACM Press, 2012, 100–107.**

(Morris et coll., 2006) Morris, M.R., Huang, A., Paepcke, A., and Winograd, T. Cooperative gestures: multi-user gestural interactions for co-located groupware. Conference on Human Factors in computing systems (CHI'06). ACM Press, 2006, 1201–1210.

(Myers et coll., 2008) Myers, B., Park, S.Y., Nakano, Y., Mueller, G., and Ko, A. How designers design and program interactive behaviors. Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on, (2008), 177–184.

(Navarre et coll., 2009) Navarre, D., Palanque, P., Ladry, J.-F., and Barboni, E. ICOs: A Model-Based User Interface Description Technique dedicated to Interactive Systems Addressing Usability, Reliability and Scalability. ACM Transactions on Computer-Human Interaction 16, 4 (2009), 1–56.

(Newman, 1968) Newman, W.M. A system for interactive graphical programming. Proceedings of the April 30–May 2, 1968, spring joint computer conference, (1968), 47–54.

(Nigay, 1994) Nigay, L. Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales. PhD, University Joseph Fourier, 1994, 363 pages.

(Nigay et Vernier, 1998) Nigay, L. and Vernier, F. Design Method of Interaction Techniques for Large Information Spaces. Conference on Advanced Visual Interfaces. ACM Press, 1998, 37–46.

(Norman et Draper, 1986) Norman, D.A. and Draper, S.W. User Centered System Design; New Perspectives on Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986.

(Olsen, 2007) Olsen Jr, D.R. Evaluating user interface systems research. Proceedings of the 20th annual ACM symposium on User interface software and technology. ACM Press, 2007, 251–258.

(Olson et Teasley, 1996) Olson, J.S. and Teasley, S. Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation. Conference on Computer Supported Cooperative Work (CSCW'96). ACM Press, 1996, 419–427.

(Paternó, 2000) Paternó, F. Model-Based Design and Evaluation of Interactive Applications. Springer London, 2000, 208 pages.

(Phillips, 1999) Phillips, G. Architectures for Synchronous Groupware. Rapport Technique, Queen's University, Canada, 1999, 1999-425, 53 pages.

(Phinyomark et coll., 2014) Phinyomark, A., Quaine, F., and Laurillau, Y. The Relationship between Anthropometric Variables and Features of Electromyography Signal for Human–Computer Interface. Applications, Challenges, and Advancements in Electromyography Signal Processing. IGI Global.

(Phinyomark et coll., 2013a) Phinyomark, A., Quaine, F., Charbonnier, S., Serviere, C., Tarpin-Bernard, F., and Laurillau, Y. A feasibility study on the use of anthropometric variables to make muscle–computer interface more practical. Engineering Applications of Artificial Intelligence 26, 7. Elsevier, 2013, 1681–1688.

(Phinyomark et coll., 2013b) Phinyomark, A., Quaine, F., Charbonnier, S., Serviere, C., Tarpin-Bernard, F., and Laurillau, Y. EMG feature evaluation for improving myoelectric pattern recognition robustness. Expert Systems with Applications 40, 12. Elsevier, 2013, 4832–4840.

(Phinyomark et coll., 2013c) Phinyomark, A., Quaine, F., Laurillau, Y., Thongpanja, S., Limsakul, C., and Phukpattaranont, P. EMG Amplitude Estimators Based on Probability Distribution for Muscle-Computer Interface. Fluctuation and Noise Letters 12, 03. WolrdScientific, 2013, 1350016.

(Phinyomark, 2013d) Phinyomark, A., Thongpanja, S., Quaine, F., Laurillau, Y., Limsakul, C., and Phukpattaranont, P. Optimal EMG amplitude detectors for

muscle-computer interface. 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2013, 1–6.

(Pinelle et Gutwin, 2005) Pinelle, D. and Gutwin, C. A Groupware Design Framework for Loosely Coupled Workgroups. In H. Gellersen, K. Schmidt, M. Beaudouin-Lafon and W. Mackay, eds., Conference European Conference on Computer-Supported Cooperative Work (ECSCW'05). Springer Netherlands, 2005, 65–82.

(Riche et Fekete, 2010) Riche N.H., and Fekete J.D. Novel Visualizations and Interactions for Social Networks. Handbook of Social Network Technologies and Applications. Springer, 2010, 611-636.

(Rogers, 1994) Rogers, Y. Exploring Obstacles: Integrating CSCW in Evolving Organisations. Conférence CSCW. ACM Press, 1994, 67–77.

(Roschelle et Teasley, 1995) Roschelle, J. and Teasley, S.D. The Construction of Shared Knowledge in Collaborative Problem Solving. Computer Supported Collaborative Learning. Springer Berlin Heidelberg, 1995, 69–97.

(Ryall et coll., 2006) Ryall, K., Esenther, A., Forlines, C., et al. Identity-differentiating widgets for multiuser interactive surfaces. Computer Graphics and Applications, 26, 5. IEEE, 2006, 56–64.

(Salber, 1995) Salber, D. De l'interaction individuelle aux systèmes multi-utilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée. PhD, 1995.

(Salvador et coll., 1996) Salvador, T., Scholtz, J., and Larson, J. The Denver model for groupware design. ACM SIGCHI Bulletin 28, 1. ACM Press, 1996, 52–58.

(Saponas et coll., 2009) Saponas, T.S., Tan, D.S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J.A. Enabling always-available input with muscle-computer interfaces. symposium on User interface software and technology (UIST'09). ACM Press, 2009, 167–176.

(Saponas et coll., 2008) Saponas, T.S., Tan, D.S., Morris, D., and Balakrishnan, R. Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. Conference on Human Factors in Computing Systems (CHI'08). ACM Press, 2008, 515–524.

(Santosa et Wigdor, 2013) Santosa, S. and Wigdor, D. A field study of multi-device workflows in distributed workspaces. ACM Press, 2013, 63–72.

(Scott et coll., 2003) Scott, S.D., Grant, K.D., and Mandryk, R.L. System guidelines for co-located, collaborative work on a tabletop display. Conference ECSCW. Elsevier, 2003, 159–178.

(Schmidt et coll., 2010) Schmidt, D., Chong, M.K., and Gellersen, H. IdLenses: Dynamic personal areas on shared surfaces. ACM International Conference on Interactive Tabletops and Surfaces (ITS'10). ACM Press, 2010, 131–134.

(Shen et coll., 2004) Shen, C., Vernier, F.D., Forlines, C., and Ringel, M. DiamondSpin: an extensible toolkit for around-the-table interaction. ACM Conference on Human factors in computing systems (CHI'04). ACM Press, 2004, 167–174.

(Shiratori et coll., 2012) Shiratori, N., Sugawara, K., Manabe, Y., Fujita, S., and Chakraborty, B. Symbiotic Computing Based Approach Towards Reducing User's Burden Due to Information Explosion. Journal of Information Processing 20, 1. IEEE, 2012, 37–44.

(Suganuma et coll., 2007) Suganuma, T., Sugawara, K., and Shiratori, N. Symbiotic Computing: Concept, Architecture and Its Applications. In J. Indulska, J. Ma, L.T. Yang, T. Ungerer and J. Cao, eds., Ubiquitous Intelligence and Computing. Springer Berlin Heidelberg, 2007, 1034–1045.

(Tang et coll., 2006) Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S. Collaborative coupling over tabletop displays. Conference on Human Factors in computing systems (CHI'06). ACM Press, 2006, 1181–1190.

(Taelman et coll., 2007) Taelman, J., Adriaensen, T., van der Horst, C., Linz, T., and Spaepen, A. Textile integrated contactless EMG sensing for stress analysis. International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. IEEE, 2007, 3966–3969.

(Tofan, 2013) Tofan, V.-A. Grasping Gestures. Mémoire de Master 2 Recherche, UJF, 2013, 38 pages.

(Van den Hoven et Mazalek, 2011) Van den Hoven, E. and Mazalek, A. Grasping gestures: Gesturing with physical artifacts. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 25, 03. Cambridge, 2011, 255–271.

(van Welie et coll., 1998) Welie, M. van, Veer, G.C. van der, and Eliëns, A. An Ontology for Task World Models. In D.P. Markopoulos and P.D.P. Johnson, eds., Design, Specification and Verification of Interactive Systems (DS-VIS'98). Springer Vienna, 1998, 57–70.

(van Welie, 2001) Van Welie, M. Task-Based User Interface Design. 2001, PhD Thesis, 217 pages.

(Venema, 1999) Venema, D.C. The N-UAN; a New User Action Notation. Masther thesis. Amsterdam, Vrije Universiteit. Department Computer Science, 1999.

(Vernier et Nigay, 2001) Vernier, F. and Nigay, L. A framework for the combination and characterization of output modalities. In Interactive Systems Design, Specification, and Verification. Springer, 2001, 35–50.

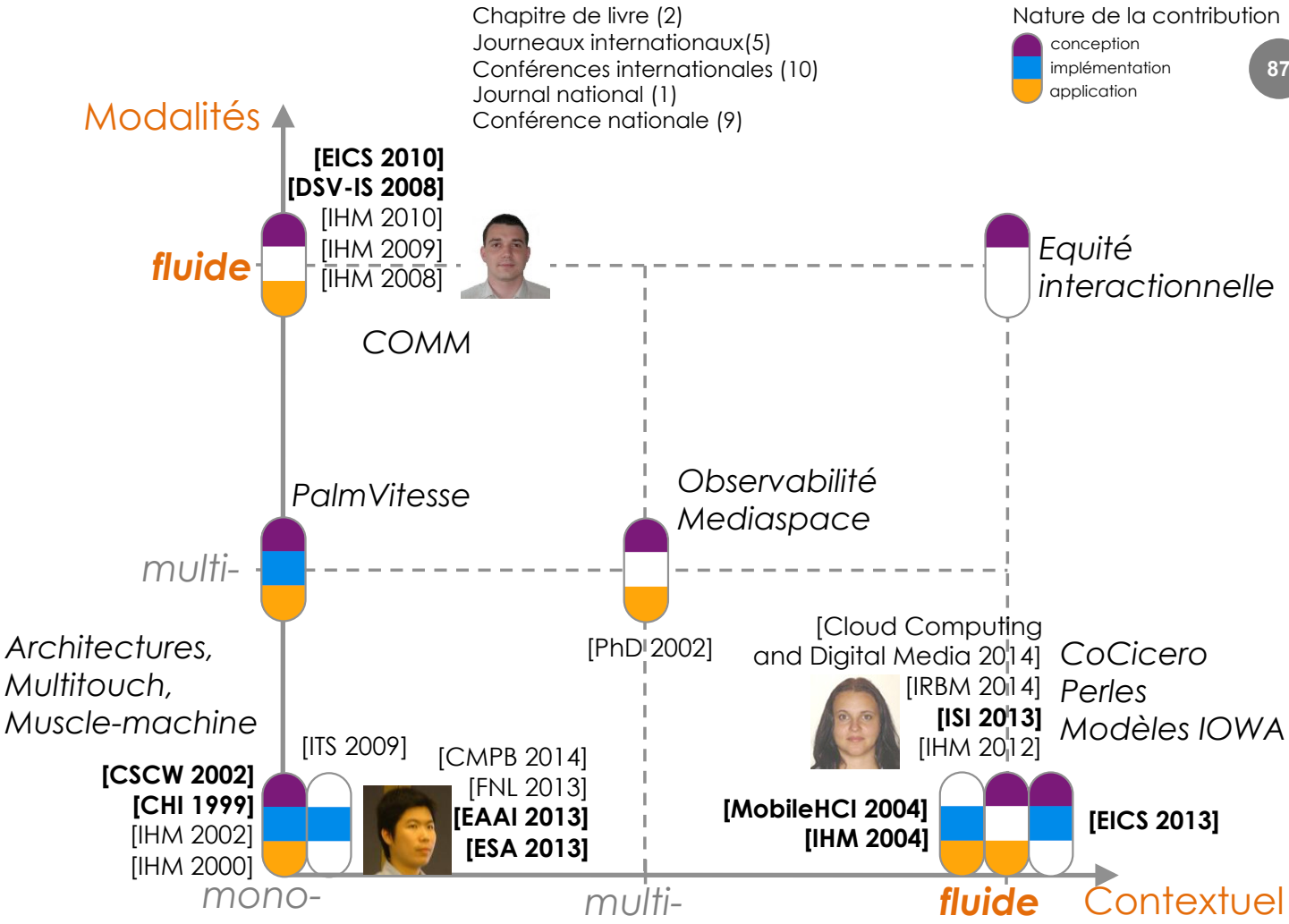
(Visser et Maher, 2011) Visser, W. and Maher, M.L. The role of gesture in designing. AI EDAM-Artificial Intelligence Engineering Design Analysis and Manufacturing 25, 3. 2011, 213.

(Wang et Blevis, 2004) Wang, H. and Blevis, E. Concepts that support collocated collaborative work inspired by the specific context of industrial designers. ACM conference on Computer supported cooperative work (CSCW'04). ACM Press, 2004, 546–549.

(Würtz et coll., 2010) Würtz, R.P., Bellman, K.L., Schmeck, H., and Igel, C. Editorial: Special Issue on Organic Computing. ACM Trans. Auton. Adapt. Syst. 5, 3. ACM Press, 2010, 9:1–9:3.

Annexe

Cette annexe contient un recueil d'articles couvrant une partie des travaux présentés dans ce document. Le diagramme ci-dessous illustre mes travaux aux travers de mes publications organisées selon les trois dimensions présentées dans le manuscrit (sociale, modale, contextuelle), avec la dimension sociale comme dimension de première classe. Les publications jointes à ce mémoire sont celles marquées en gras.



IOWAState: Models and Design Patterns for Identity-Aware User Interfaces Based on State Machines

Yann Laurillau
University of Grenoble, UPMF, CNRS, LIG Laboratory
110 av. de la Chimie, Domaine Universitaire, BP 53, 38041 Grenoble cedex 9, FRANCE
{first name}.{last name}@imag.fr

ABSTRACT
The emergence of interactive surfaces and technologies able to differentiate users allows the design and development of Identity-Aware (IA) interfaces, a new and richer set of user interfaces (UIs). Such user interfaces are able to adapt their behavior depending on who is interacting. However, existing implementations, mostly as software toolkits, are still ad-hoc and mostly based on existing GUI toolkits which are not designed to support user differentiation. The problem is that the development of IA interfaces is more complex than the development of traditional UIs and still requires extra programming efforts. To address these issues, we present a set of implementation models, named IOWAState models, to specify the behavior as state machines, the architecture and the components of IA interfaces. In addition, based on our IOWAState models and a classification of IA user interfaces, we detail a set of design patterns to implement the behavior of IA user interfaces.

Author Keywords
Identity-aware user interfaces; Interactive surfaces; Software design patterns; Architecture model; State machine model.

ACM Classification Keywords
H.5.2. User Interfaces: Graphical User Interfaces, Interaction styles, Prototyping, User-Centered design.
H.5.3. Group and Organization Interfaces: Web-based interaction.
D.2.2. Design Tools and Techniques: User interfaces.

INTRODUCTION
Research on multi-touch interactive surfaces, in particular interactive tabletops, is now well established in the fields of Human-Computer Interaction (HCI) and of Computer-Supported Cooperative Work (CSCW). The directness of interaction and the multiuser capabilities of tabletops may explain the growing interest for these systems. Currently, several technological solutions are available [12,32] including commercial ones [9,17]. Among these technologies, few are able to differentiate users touching the surface [9,16,24,32].

In conjunction with the growing number of technological solutions allowing user identification and differentiation (e.g., [1,16]), work is done on the development of identity-aware (IA) user interfaces, taking advantage of user differentiation and showing the capabilities and benefits of such UIs (e.g. [26,27]). For instance, SIDES [25] is an IA multi-user tabletop-based interactive system designed to develop effective social skills. It shows that such category of technology is helpful for a therapeutic purpose considering teenagers with Asperger's syndrome. In particular, IA widgets requiring synchronous actions were key in its success.

As Identity-Aware User Interfaces (IAUIs) are more complex than traditional and single-user interfaces, their development is still challenging. We identify several issues:

Lack of implementation models and guidelines: developing IAUIs requires extra programming efforts due to the lack of models and of capitalization of best practices (e.g. guidelines, design patterns). We observed that existing IA applications are mostly developed from scratch and, similarly to the development of multi-touch gesture-based interactive systems, developers must deal with low-level events.

User interfaces' behavioral model split across the code: traditional UI toolkits (e.g. Java's Swing), including UI toolkits that support user differentiation (e.g. DiamondSpin [28] toolkit is based on Java's Swing), massively rely on the well-known callback-based programming model: developers have to write a bunch of callbacks to handle each input event for each UI component. Thus, they must maintain the state of the UI component across these callbacks which usually leads to produce "spaghetti" of code [21].

Dealing with concurrent inputs and differentiated outputs: although a traditional UI receives and deals with events generated by the same user, an IAUI has to manage input events generated by different users due to simultaneous actions, sometimes concurrent. Furthermore, such an UI must maintain a much more complex state model in order to produce consistent and customized outputs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
EICS'13, June 24–27, 2013, London, United Kingdom.
Copyright © ACM 978-1-4503-2138-9/13/06...$15.00.

At implementation level, although most of the work done focuses on technical issues to allow user differentiation such as dedicated software toolkits, we investigate the building of software models that would help and drive the development of IAUI components. In particular, we investigate the use of state machines as a means to address the two last issues.

This paper is structured as follows. First, we introduce an example to illustrate IAUIs. Then, we present the IOWAState models, our first contribution: a set of models to specify the behavior, the main components and the architecture of IAUI components. Based on our IOWAState models, we detail our second contribution, seven design patterns to implement the behavior of IAUI components, and our methodology. We conclude with a discussion and perspectives.

ILLUSTRATIVE EXAMPLE

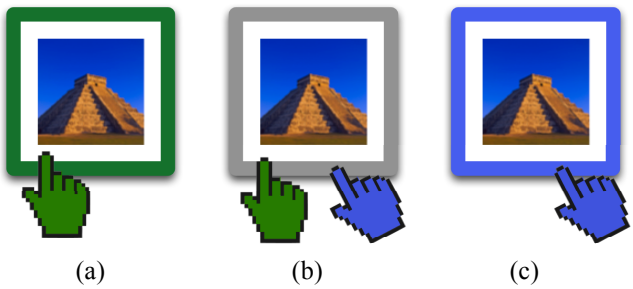


Figure 1: Cooperative gesture to transfer ownership [18].

Let us consider the following scenario: two users, Green and Blue, are interacting simultaneously on a user-differentiating multitouch surface, manipulating digital artifacts (widgets, images, shapes, etc). Some are public while others are private. Thanks to user differentiation, supporting privacy, private artifacts are accessible by their owner only. However, user Green wants to give an image he/she owns to user Blue. Thanks to user differentiation, the users Green and Blue just have to accomplish a cooperative gesture [18] to transfer ownership. As shown in Figure 1, having first activated ownership transfer mode, (a) user Green touches the image he/she wants to relinquish; (b) user Blue touches Green's image to indicate that he/she will be the next owner; (c) ownership is granted to user Blue when user Green releases his/her finger from the surface. This example is used further in the part about design patterns.

BACKGROUND AND RELATED WORK

As underlined in introduction, multi-touch technologies, especially interactive surfaces, are intensively studied and are now well known in our research communities. Therefore, in this paper we concentrate on IA User Interfaces and on development tools supporting user differentiation.

Identity-aware user interfaces

In the 90's, researchers started to investigate the development of groupware using a single and shared display: Single Display Groupware systems (SDG) are ancestors of actual research on interactive surfaces such as tabletops: co-located users were able to interact simultaneously using multiple input devices [29]. Therefore, assigning an input device per user allows user identification and thus the development of identity-aware applications. The most basic example is multi-pointers on a shared display: each user owns a pointer and is allowed to manipulate simultaneously the shared UI elements displayed on the screen. In particular, MMM [4], Pebbles [22], and Kidpad [10] are usually considered in the literature as the very first systems implementing and illustrating the concept of SDG. These systems are the first to take advantage of user identification to develop identity-aware interfaces.

Proxy-Sketch [1] is another example of identity-aware interface dedicated to the creation of GUI prototypes. User identification is used to associate owners to content. It also supports casual observers (i.e. not logged in) that prevent from accidental changes.

Idlenses [27] is an identity-aware interaction technique that revisits magic lenses to provide a moveable personal area. Once identified, users benefit of personal tools that support access control to restricted and personal data, personalized actions such as automatic filling of web forms with personal data, a private clipboard, etc.

Tse et al. [30] have investigated multi-user and multimodal identity-aware interactive systems for gaming, based on DiamondTouch [9]. The underlying mechanism for multimodal fusion uses user identification to link speech with gesture.

To capitalize the work done in this area, Ryall et al. [26] propose the conceptual iDwidgets framework. The authors define identity-aware widgets (i.e. called iDwidgets for identity-differentiated widgets) as an extension of "*the widget concept by including identity as an input parameter, which lets us customize interactions in a variety of ways*". For instance, an identity-aware paintbrush tool will adapt its color or stroke size according to the user.

Toolkits supporting user differentiation

In order to facilitate the development of identity-aware interfaces and widgets, several toolkits have been designed and developed to support user differentiation.

The very first toolkits used peripherals as a means to differentiate users. The implicit user differentiation mechanism was "*one input device, one user; one user, one input device*". For instance, Multiple Input Devices (MID) [13] is a software library built on top of Java. In order to support multiple mice, MID revisits the underlying Java event mechanism. Therefore, it allows developers to implement identity-aware interfaces based on the mouse ID. Such a piece of information is implemented as an extra attribute of event objects.

SDGToolkit [31] is an extension of MID as it supports multiple keyboards. At UI level, the toolkit provides mechanisms to support orientation in tabletop setups. This toolkit is built on top of the .NET framework and is written in C#. Similarly to MID, events generated by input devices are associated to devices based on a device ID. It allows the use of standard widgets provided by the .NET framework to develop identity-aware interfaces as well to develop its own identity-aware widgets from scratch. This toolkit gave rise to IdenTTop [24], adding support for any multi-touch devices and support for a Polemus motion tracker. In addition, IdenTTop proposes a development framework for identity-aware applications based on a set of software components.

For touch surfaces, especially DiamondTouch [9], DiamondSpin [28] is the most well-known toolkit. It is built on top of Java and extends Java's Swing GUI toolkit to support widget orientation. User identification is achieved using a similar mechanism as SDGToolkit: events generated by touches are associated to users by the way of a specific attribute: a user ID. In particular, the toolkit provides identity-aware frames (DSFrame component) allowing users to customize the appearance: a frame can be rotated, zoomed or resized. Similarly to SDGToolkit, it allows developers to reuse standard Java's Swing components in a DSFrame. Compared to DiamondSpin, the GIL Library (gil.imag.fr) is another toolkit based on DiamondTouch but built on top of Tcl/Tk

While the java-based T3 toolkit focuses on high-resolution tabletop interfaces using wireless pens as devices for user identification [32], TouchID [16] goes beyond user identification as it investigates user-, hand-, and handpart-aware tabletops. Similarly to SDGToolkit and IdenTTop, TouchID is build on top of the .NET framework and based on the Microsoft Surface touch table [17].

IOWASTATE MODELS

As our model is intended for the design and the implementation of Identity-Aware UIs (IAUI), the IOWAState model encompasses three modeling primitives:

- A behavior model based on standard state machine models to describe the behavior of an IAUI. As detailed further, we used this modeling primitive to identify recurrent behavior patterns. In particular, we highlight how user differentiation is achieved in terms of state machine.
- A component model that identifies the main components of an IAUI and their relationships. In particular, this model highlights how we handle multiple state machines in order to allow parallel or concurrent user actions on an IAUI.
- An architecture model to describe the structure of an IAUI component. It illustrates how low-level events are processed to produce high-level events and are propagated to sub-components.

In the following, as the IOWA component model is based on the Model-View-Controller (MVC) design pattern, we will refer to it.

IOWA Behavior model

We chose to model and implement the Model part using hierarchical state machines (HSM), a derivative of finite state machines (FSM). Since Newman's work [23], user interfaces are often specified using state machines [15,21,33]. In addition, several works have demonstrated the feasibility and the benefits implementing state machine-based UIs [1,5,14].

As state machines are well suited to specify mode-driven interactions, we allow the Model to encompass several state machines, one per user, and support their parallel execution. Indeed, collaborative settings such as tabletops enable the interleaving modal actions.

In addition, using state machines facilitated the comparison of identity-aware widget's implementations and helped us to identify classes of identity-aware widgets based on their implementation model.

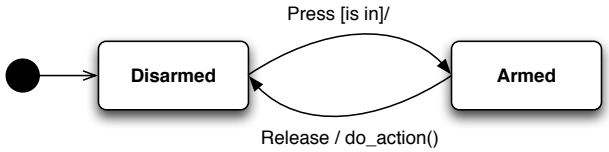


Figure 2: Example of state model of a button.

A state machine is a combination of states and transitions connecting states. Using UML statecharts, transitions are labeled according to the following syntax: *trigger* [*guard*] / *effect*. *Trigger* is an event name, *guard* is a set of conditions and *effect* is an action executed when the transition is triggered. Figure 2 shows a classic state model of a button constituted of two states: disarmed: the button is raised; armed: the button is pushed. Such a state model responds to the press and release events. For instance, if the active state is "Armed" while a release event occurs, the *do_action()* is fired and the button goes in the "Disarmed" state.

Finally, the main advantage of Hierarchical State Machines is to facilitate the control of the state explosion problem as it allows the refinement of states as finite state machines. Indeed, specifying a state model using HSMs is a top-down approach like problem solving: an overall state model is decomposed into FSMs as problems are decomposed into smaller problems. For instance, HSMs are part of UML to specify state machines.

IOWA Component model

The IOWA component model slightly differs from the MVC design pattern as an IOWA Component inherits from an IOWA StateMachine (i.e. Model) and an IOWA UI (i.e. View). The main advantage is to present a component that looks externally as a whole, hiding the model and view

parts, while preserving modularity and loose coupling between the View and the Model.

In order to support the design of IAUIs, user differentiation is first achieved at the Model level. As shown in Figure 3, the Model is an instance of an IOWA StateMachine that describes the behavior of an IAUI, as explained in the previous part, with an IA state machine. Such a state machine is hierarchical as each state (i.e. IOWA State) may be described as a hierarchy of states. Transitions between states are triggered by events sent through a *post()* operation. Events are propagated in the state hierarchy. As events carry the identity of the user (i.e. user ID) who performs the associated action, this mechanism allows the design of IA state machines.

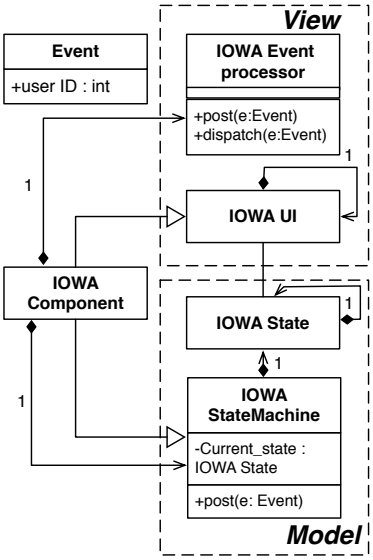


Figure 3: IOWAState's component model.

In order to support the interleaving of different user's actions and concurrent actions, although an IOWA Component is already statemachine, an IOWA Component may handle a set of IOWA StateMachines, one per user. Indeed, each event received by an IOWA Component and processed by the *post()* operation is dispatched to the state machine associated with the user ID that produces such an event.

An IOWA State component is responsible for handling high-level events supplemented with a user ID and achieving user-differentiation. Indeed, depending on the event type and the user ID, an IOWA State component verifies conditions on transitions associated to it: if a condition is verified, this component indicates to the related IOWA StateMachines component what the new state is.

As part of the View, an IOWA UI produces an output representation to the user. It defines the look and feel of an IAUI. In this model, similarly to HsmTk [5], an IOWA UI is a composition of IOWA UIs, one per state. For input events, an IOWA UI is associated with an IOWA Event Processor that receives low-level events and produces high-level events sent to the IOWA Component through a *post()* operation. Such an IOWA Event Processor may be seen as a pipeline of event filters.

IOWA Architecture model

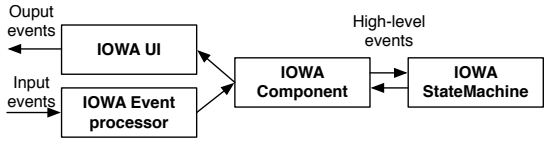


Figure 4: IOWAState's architecture model.

As shown in Figure 4, the IOWA architecture model is layered according to the MVC design pattern. As explained previously, an IOWA UI and an IOWA Event Processor constitutes the View while an IOWA StateMachine constitutes the Model. They are assembled to constitute an event processing chain that processes user' input events and generates an output representation. As an IOWA Component may be a composition of sub-IOWA Component, in addition to the dispatch of events to the state machine, the IOWA Event Processor dispatches events to the sub-components. Furthermore, the state machine may generate events that are also dispatches to the sub-components.

IMPLEMENTING IOWASTATE MODELS

The IOWAState Models, in particular the IOWA behavior model, may be directly specified with an object-oriented programming language that allows a one-to-one correspondence between the IOWAState Models and the implementation. We chose such an approach because, as underlined in introduction, IAUIs are more complex to design and to implement than traditional single-user UIs. The implementation step is usually complex as existing toolkits that support user differentiation mostly rely on usual WIMP toolkits (e.g. Java's Swing). To address this issue, in particular about the implementation of state machines, existing works advocate a developer-centric approach claiming a tight integration of models with dynamic programming languages [2,5,11]. Indeed, a state machine leads to produce code easier to read and to maintain. In addition, it supports a better reusability and extensibility as we may easily add, remove or modify states and transitions thanks to the inheritance mechanism supported by object oriented programming languages.

In order to demonstrate the validity of our IOWAState models, without giving implementation details, we implemented eight very different IAUI components. Although existing implementations focus on customization of appearance [16,24,28,30,32] (e.g. orientation to a particular user), we focus on component's behavior in terms of internal/external functionality and group input [26]. Precisely, in order to cover the largest range of IAUI component classes as identified by Ryall et al. [26], the components we implemented are taken and adapted from [18,19,20,26].

For instance, one of the eight components we implemented is a multi-user slider having a differentiated behavior, performing the same action (i.e. selecting a value) whoever the user is. However, it behaves with different styles depending on the user's identity. For instance, one user may slide the cursor from tick to tick and select a value on a discrete scale, another user would slide the cursor continuously.

Another example is a cumulative voting component allowing different users clicking on a same button to perform an action. Achieving the action requires a minimum number of users performing the interaction.

The height IAUI components we implemented are developed in Python, to be used with a Diamondtouch device [9]. In order to be independent from any GUI toolkit and their associated programming paradigm, we used basic graphic primitives to draw the components (i.e. OpenGL rendering engine). In order to support identity-awareness, we rely on the user-differentiation mechanism provided by the Diamondtouch device [9], able to differentiate up to four users. The low-level events sent by the device are supplemented with a user ID represented as an integer value in a range of 0-3. It allowed us to implement an event loop that sends high-level events supplemented with a user ID to the user interface and thus to our IAUI components.

DESIGN PATTERNS

Methodology

In order to identify recurring design patterns for IAUIs, we defined and followed a twofold method. The first part of this method consists in analyzing and in reverse-engineering the source code of existing identity-aware widgets to detect recurring implementation patterns. The second part of this method consists in developing identity-aware widgets using state machines to model and implement widget's behavior. We chose to reuse and adapt existing identity-aware widgets that are the ones described in the previous section. Obviously, these developments are on our IOWA state models.

Code-based analysis of existing IA widgets

Concomitantly with the development of the eight widgets detailed in the previous section, we analyzed the code of a set of existing prototypes that includes IA widgets. We focused on prototypes developed with toolkits allowing user identification: SDG [31], DiamondSpin [28], T3 [32], TouchID [16] and GIL [3]. We did not consider the IdenTTop toolkit [24] because the code is not publicly available. Although several IA widgets and the related source code are available online, we also requested additional examples from the authors of the DiamondSpin and GIL toolkits.

We analyzed seven IA widgets taken from SDG, DiamondSpin, and GIL. We found no relevant widget for the T3 and TouchID toolkits. The source code was reversed engineered to identify implementation patterns of identity-aware widgets. First, we carefully examined the code as follows: (1) identification of callbacks or related methods managing user input events supplemented with a user ID; (2) identification of attributes used to store the component state; (3) identification of control structures that use the user ID to update the attributes related to the component state. Then, we modeled IA widgets using state machine representations. In order to verify our models, we compared the models at runtime. In order to classify state machines and to derive patterns, based both on our developments and on the analysis of existing components, we focused on similarities and differences in terms of states (e.g. associated states) and transitions (i.e. conditions).

The IA widgets and IA interaction techniques we analyzed are:

- From SDG toolkit, a multi-user button (SDGButton) allowing two interaction modes: (1) restricted interaction to the first user pushing the button (one-user-at-a-time); (2) cumulative effect; a multi-user check button (SDGCheckButton) that paints parts of its border with the color related to the users that checked it; a multi-user slider with multiple cursors (SDGTrackBar), one per user.
- From DiamondSpin toolkit: an identity-aware and moveable menubar (DSMenuBar); a multi-user chess board [8] (RealTimeChess); a RingMenu.
- From GIL toolkit: a cooperative design application to assemble shapes in order to design a building.

Design patterns

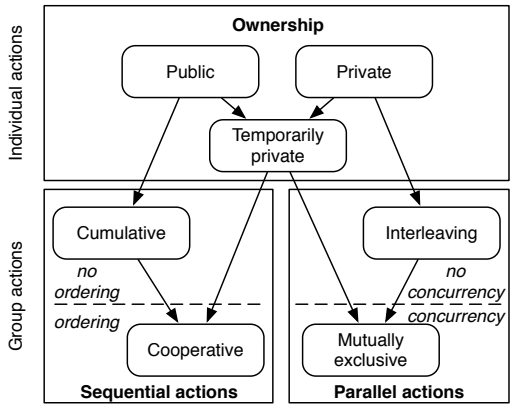


Figure 5: Design pattern graph.

As shown in Figure 5, our method leads us to identify three categories of patterns related to:

- *Individual actions*: these patterns deal with ownership, i.e. how a UI component is owned by one or multiple users. We identify three kinds of ownership: (a) public UI components that are free and not owned; (b) private UI components that are owned by one or multiple users and that can exclusively be used by the owners; (c)

temporarily UI components that are free UI components owned for a limited amount of time.

- *Group actions to achieve a sequence of actions*: these patterns identify UI components that require multiple users to achieve a group action: (a) cumulative UI components that take into account the number of users whatever the sequence of action is; (b) cooperative UI components that imply a well-defined and ordered sequence of actions.
- *Group actions allowing parallel execution of actions*: we identify two situations: (a) the interleaving of actions with no concurrency; (b) mutually exclusive UI components to deal with concurrency.

In the following, we detail each design pattern using Borchers' pattern language [6]. In addition, illustrations of state machines are given using UML statecharts.

Public IAUI

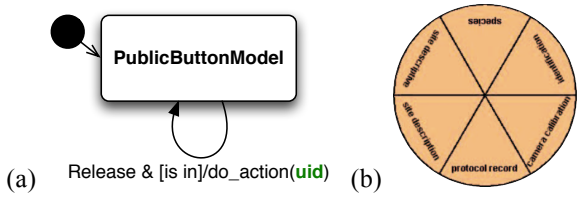


Figure 6: (a) SM model for Public IAUI; (b) TeamTag centralized control [19].

Context: in order to achieve an individual task, different users simultaneously interact with a same UI element (e.g. a button) of the shared workspace to issue a command that acts on an artifact associated with her/him.

Problem: First, traditional widgets are single-user and do not support simultaneous actions. Secondly, the display may offer a limited amount of space: replicated UI elements would clutter the interacting space and would waste pixels. Thirdly, simultaneous but opposite actions on a same UI element would produce an inconsistent visual representation or have no effects: for instance, a user is pressing his/her finger on a button that should look armed while another user releases his/her finger on the same button that should look disarmed.

Solution: a single instance of an identity-unaware state machine composed of a single state would support simultaneous actions: transitions are labeled without uid-based conditions. Thus, user differentiation is achieved by an external function triggered when an action is performed on the UI (i.e. associated to the triggered state transition such as the function `do_action(uid)` shown in Figure 6 (a)). Such a function takes the user id associated to a user event as an argument: different actions are executed according to the user id.

To support presentation consistency for simultaneous actions, a unique output representation is coupled with the state machine because the state machine is composed of a single state.

Examples: TeamTag's IA controls [19] (Figure 6 (b)).

Private IAUI

Context: an interactive surface is partitioned into shared and private territories, allowing users to interact with private artifacts located in their private territory and to perform individual tasks.

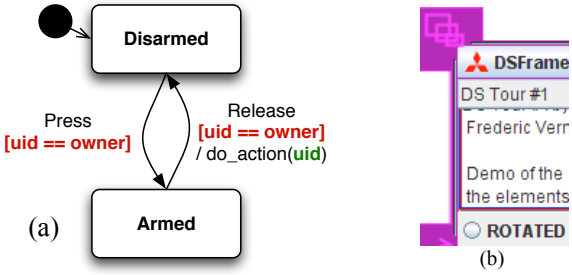


Figure 7: (a) SM model for Private IAUI; (b) Swing widgets in a DSFrame [28].

Problem: an interactive surface is naturally a public shared resource as everything is visible and potentially free, including private territories. Tacit social rules are the most common mechanism that preserves private territories.

Solution: an IAUI exclusively associated to an owner, based on his/her user id, prevents other users to interact with such private UI elements. All transitions of the state machine associated with the private IAUI must be labeled with uid-based conditions: when an event is received, a transition is triggered if the user ID carried by the event matches the owner ID (e.g. condition `[uid == owner]` as shown in red in Figure 7 (a)). We may consider that an owner is associated to such an IAUI element at instantiation time.

Examples: Swing widget in a DSFrame [28] (Figure 7 (b)), IdLenses [27].

Temporarily Private IAUI

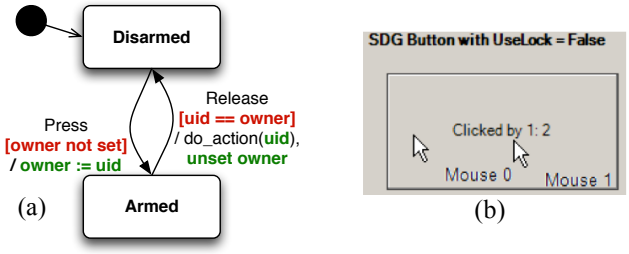


Figure 8: (a) SM model for Temporarily Private IAUI; (b) Single-user lock SDGButton [31].

Context: different users simultaneously access to a shared and free UI element such as a widget or an artifact (e.g. digital photo).

Problem: although some UI elements are public and freely available, some UI elements may only support interactions for one user at a time.

Solution: an IAUI element temporarily owned by the current user interacting with the IAUI: ownership is granted

to the very first user that interacts with the IAUI element; ownership is released when the user action is completed. To support such mechanism, the state machine associated to a temporarily private IAUI element should be designed based on two categories of transitions: transitions labeled (a) without and (b) with uid-based conditions. The first category allows any user to take ownership on a free IAUI element (e.g. condition `[owner not set]` as show in red in Figure 8 (a)): when this kind of transition is triggered, the current user is then marked as the current owner of the IAUI element he/she is manipulating (e.g. effect `owner := uid` as shown in green in Figure 8 (a)). Therefore, the IAUI element is considered as private. Similarly to Private IAUI, the remaining transitions are related to the second category (e.g. condition `[uid == owner]` as shown in red in Figure 8(a)). However, when triggered, at least one transition of the second category must release ownership (e.g. effect `unset owner` as show in green in Figure 8 (a)).

Examples: DSMenuBar [28], Single-user lock SDGButton [31] (Figure 8 (b)).

References: PUBLIC IAUI, PRIVATE IAUI.

Cumulative IAUI

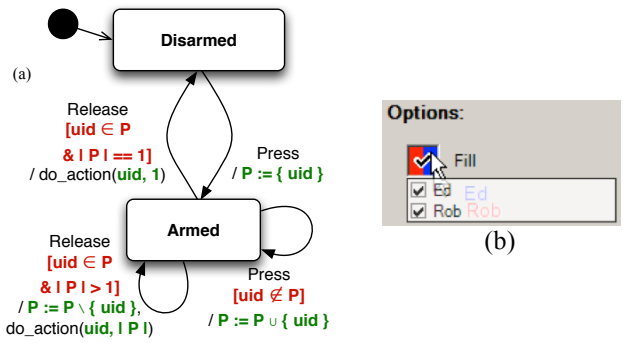


Figure 9: (a) SM model for Cumulative IAUI; (b) Cumulative SDGCheckButton [31].

Context: different users are interacting with the same UI element to perform a group and synchronized action.

Problem: the UI element must consider how many users (i.e. critical mass) are interacting to achieve a group action (e.g. majority). Furthermore, this UI element must remember who is interacting to take into account each user only once: for instance, a user touching a button with two different fingers must be counted as a single touch.

Solution: an IAUI element that maintains a list of users already interacting with it. This list is updated when transitions of the associated state machine are triggered. Three categories of conditions are observed:

- Conditions verifying if a user is not already in the list to avoid duplicate entries (e.g. condition `[uid ∉ P]` as shown in Figure 9 (a)). Consequently, for transitions that verify such a condition, the associated action consists in adding the new interacting user to the list (e.g. condition `[P := P ∪ {uid}]` as shown in Figure 9 (a)).
- Conditions verifying if a user is already on the list (e.g. condition `[uid ∈ P]` as shown in Figure 9 (a)) when the user interaction is completed. Consequently, for transitions that verify such a condition, the associated action consists in removing the associated user from the list (e.g. condition `[P := P \ {uid}]` as shown in Figure 9 (a)).
- Conditions verifying if no more users are interacting with the IAUI element to maintain state consistency (e.g. condition `[|P| > 1]` where `|P|` denotes the cardinality of set `P` as shown in Figure 9 (a)). Such a condition can be seen as threshold to reach in order to select a state transition in case of alternatives.

Although a Public IAUI element responds to individual actions, a Cumulative IAUI element responds to group actions. Similarly, there is no owner associated with it.

Examples: SDGButton [31] (Figure 9 (b)), Voting button [20], SIDES [25], SDGTrackBar [31].

References: PUBLIC IAUI.

Cooperative IAUI

Context: different users are interacting with the same UI element to perform a synchronized group action, involving a limited number of users. Achieving the group action requires to execute actions in a certain order (i.e. ordered sequence of actions). Depending on the number of users or depending on who is interacting, the UI element behaves in different ways (modes).

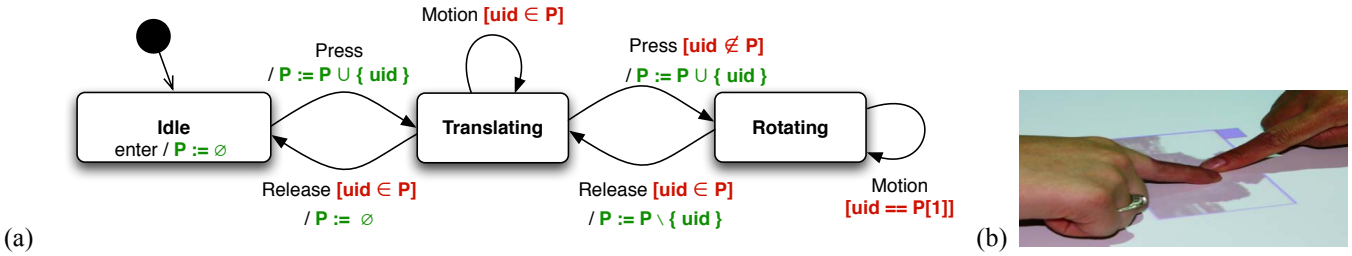


Figure 10: (a) SM model for Cooperative IAUI; (b) Cooperative gesture [17].

Problem: the UI element must consider how many users are interacting to achieve a group action. Furthermore, this UI element must remember who is interacting to take into account each user only once. As the UI element behaves differently depending on who is interacting, several states must be considered to represent the sequence of actions.

Solution: an IAUI element's state machine composed of an ordered set of states. This set corresponds to the ordered sequence of actions that the users must execute to achieve the group action. Each state is associated to different behaviors of the IAUI element. User differentiation is performed to (1) limit the number of users interacting with the IAUI element using a list similarly to a Cumulative IAUI element; (2) to associate a user for different modes of interaction using uid-based conditions (e.g. condition `[uid == P[1]]` as shown in red in Figure 10 (a)). As the number of users allowed to interact with a Cooperative IAUI element is limited, such a component may be seen as Temporarily private IAUI element.

Examples: Cooperative gesture [18] (Figure 10 (b)), Rotating shape (Figure 1).

References: TEMPORARILY PRIVATE IAUI, CUMULATIVE IAUI.

Interleaving IAUI

Context: different users are simultaneously interacting in a shared workspace on different artifacts. Some of the users may execute destructive actions (e.g. delete).

Problem: using a global mode (i.e. the same mode for all) in a shared workspace does not support parallel moded interactions. For instance, if one person is in an erasing mode, other persons cannot be in a different mode such as drawing: once the erase mode is activated, the next selected stroke would be erased.

Solution: an IAUI component's state machine managing a set of multiple instances of the same sub-state machine that are running in parallel (Figure 11 (a)). The master state machine intercepts the events and, as a proxy, dispatches events to each instance. Each instance is owned (i.e. private) by a user (e.g. conditions `[uid == user_N]` on transitions as shown in red in Figure 11 (a)) and is responsible for the management of moded interactions. Such a mechanism allows the interleaving of actions and avoids concurrent actions, even for destructive actions.

Examples: DTMap [26] (Figure 11 (b)).

References: PRIVATE IAUI.

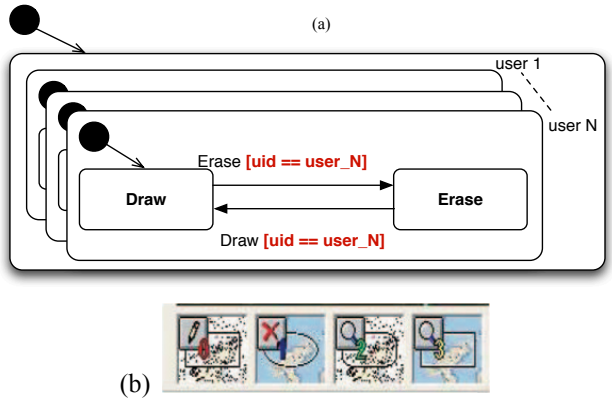


Figure 11: (a) SM model for Interleaving IAUI; (b) DTMap.

Mutually exclusive IAUI

Context: Two users are interacting simultaneously with the same UI component.

Problem: a user must wait for the first user already interacting to end up taking his/her turn and then accomplish his/her own action.

Solution: similarly to an Interleaving IAUI component, a Mutually exclusive IAUI component is based on a master state machine that manages several sub-state machines running in parallel. In addition, each sub-state machine implements an Idle/Active mechanism: the idle state is reached when a user is not interacting; the active state is reached when a user is interacting. For the latter, two sub-states are considered in order to support mutual exclusion and the fact that a user must wait his/her turn: two sub-sub-states are considered as show in Figure 12 (a): an operative state that locks the IAUI component (i.e. ownership taken) until the interaction is ended up (i.e. ownership released); a non-operative state that corresponds to a stand-by period.

Examples: Waiter's Diamondspin mechanism [28], RingMenu [8] (Figure 12 (b)).

References: TEMPORARILY PRIVATE IAUI, INTERLEAVING IAUI.

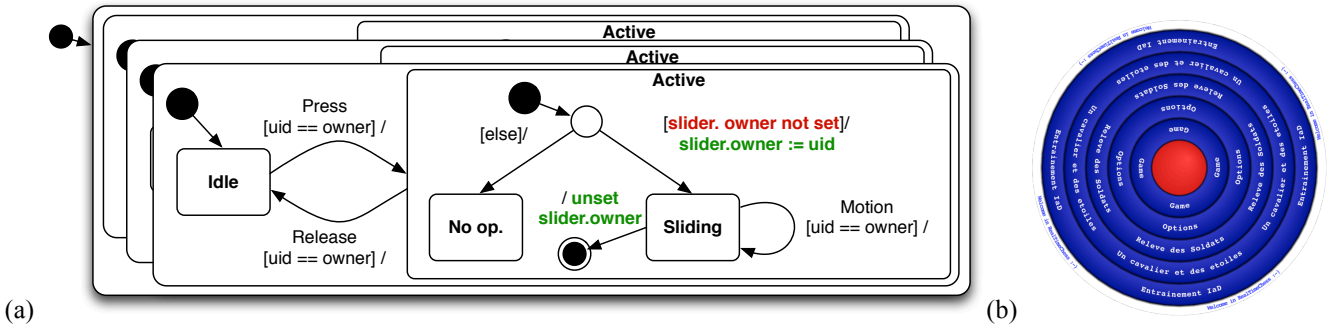


Figure 12: (a) SM model for Mutually exclusive IAUI; (b) RingMenu [7].

DISCUSSION

IOWA models

As a first evaluation, we instantiated the IOWAState models to develop eight very different IAUI components. As a second evaluation, we used our models as a framework to analyze existing implementations and to identify recurring patterns. Of course, a long-term evaluation would be clearly appropriate for a good understanding of the strengths and weaknesses of our models. In particular, we currently use our models to implement a serious game, based on a DiamoundTouch device, for the learning of cooperative practices for engineering tasks.

As the IOWA models are based on HSMs to specify the behavior of IAUIs, our approach is similar to SwingState [2], StateStream [11] or HsmTk [5] models and implementations. Although these works target single-user interfaces, our models are designed to support IA user interfaces. In particular, an IOWA component supports simultaneous user inputs and an ownership mechanism in order to allow the development of Private and Temporarily private IAUIs. In addition, our models are designed to support the parallel execution of HSMs within a MVC-like architecture in order to allow the development of Interleaving and Mutually exclusive IAUIs. The IOWA architecture model is designed to allow compositions of state machines. However this point is out of the scope of this article.

Compared to existing IA toolkits [15,23,28,32,33] widely based on a callback-based programming model inherited from traditional GUI toolkits, since our models are based on HSMs to specify the behavior of IAUIs, our several developments show it can be easily translated into code in order to produce code easier to read and to maintain, avoiding the use of a specialized and additional language. Furthermore, as we adopted an object-oriented programming approach for the implementation of the IOWAState models, we observed that the inheritance mechanism facilitates the reuse of existing HSMs. It also facilitates the creation of new behaviors with minor modifications of existing HSMs. It seems an interesting property to investigate further in order to address state explosion.

Currently, as explained previously, a first limit of our approach is the lack of long-term evaluation. Particularly, we consider another long-term evaluation with Master students following computer engineering courses, asked to implement IAUIs based on our models. Focusing only on IAUI's behavior constitutes another limit. Investigating how our models are extensible to support user-differentiation at presentation level must be considered further. Finally, we do not address the combination of two IAUI components, in particular two IAUI components having conflictual behaviors.

Design patterns

In terms of evaluation, according to [7], a pattern follows a lifecycle model composed of several steps. Currently, our patterns have reached step #5 "Pattern Gestalt" for which readers review the patterns. This article contributes to this step. The next step must be "Popular acceptance". Contributing to the evaluation as well as demonstrating the completeness of our patterns, our pattern classification covers the classification of the IDWidgets framework [26] related to behavior, and coherently integrates cooperative gestures [18]. In addition, we go one step further towards software implementation of IAUIs as we provide and detail seven design patterns. Furthermore, although CSCW literature considers UI elements' ownership as private or public, we identify a new and intermediate situation of ownership: temporary ownership.

Complementary to the conceptual IDWidgets framework [26] providing classes of IAUI widgets, our pattern classification is at implementation level and identifies classes of identity-aware user interactions. Except the fact that our patterns should reach step "Popular acceptance" (step #7), an unanswered issue is the completeness of our design patterns and the related classification. Particularly, our patterns focus on behavior only and patterns for user-differentiated presentations should be investigated further.

CONCLUSION

Focusing on the design and development of Identity-Aware User Interfaces, this article presents two main findings. First, the IOWAState models revisit the MVC architecture model to rely on hierarchical state machines in order to support identity awareness, simultaneous user inputs, and to help developers to produce code easier to read and to maintain. Another significant contribution is a classification of IAUIs based on a set of seven design patterns to specify the behavior of IAUIs using state machines.

As a perspective, we need to investigate rules to combine several IAUI components. Indeed, combining two IAUI components may lead to the combination of conflictual HSMs such as a Private IAUI component embedding a Public IAUI component. In terms of implementation, we need to investigate alternative programming languages to Python to demonstrate the generative power of the IOWAState models. Finally, we plan to extend our patterns and the state machine approach to single-user multi-touch user interfaces.

ACKNOWLEDGMENTS

This work is funded by the FI MSTIC University Joseph Fourier Grenoble 1 (TIGRE project). To Renaud Blanch for advice and hints he gave about HSMs and the permission to use his HSM-based SWIT toolkit.

REFERENCES

1. Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. In *Proc. of UIST 2011*, ACM Press (2011), 337–346.

2. Appert, C., and Beaudouin-Lafon, M. SwingStates: adding state machines to the Swing toolkit. In *Proc. of UIST 2006*, ACM Press (2006), 319–322.
3. Bérard, F., and Laurillau, Y. Single User Multitouch on the DiamondTouch: From 2x1D to 2D. In *Proc of ITS 2009*, ACM Press (2009), 1–8. http://gil.imag.fr
4. Bier, E., and Freeman, S. MMM: A User Interface Architecture for Shared Editors on a Single Screen. In *Proc. of UIST 1991*, ACM Press (1991), 79–86.
5. Blanch, R., and Beaudouin-Lafon, M. Programming rich interactions using the hierarchical state machine toolkit. In *Proc. of AVI 2006*, ACM Press (2006), 51–58.
6. Borchers, J. A pattern approach to interaction design. In *Proc of DIS 2000*, ACM Press (2000), 369–378.
7. Brown, W., Malveau, R., McCormick, H., Mowbray, T., and Thomas, S.W. The Software Patterns Criteria (1998), http://www.antipatterns.com/whatisapattern/
8. Chaboissier, J., Isenberg, P., and Vernier, F. RealTimeChess: lessons from a participatory design process for a collaborative multi-touch, multi-user game. In *Proc. of ITS 2011*, ACM Press (2011), 97–106.
9. Dietz, P., and Leigh, D. DiamondTouch: A multi-user touch technology. In *Proc. of UIST 2001*, ACM Press (2001), 219–226.
10. Druin, A., Stewart, J., Proft, D., Bederson, B., and Hollan, J. KidPad: a design collaboration between children, technologists, and educators. In *Proc. of CHI 1997*, ACM Press (1997), 463–470.
11. de Haan, G., and Post, F. StateStream: a developer-centric approach towards unifying interaction models and architecture. In *Proc of EICS 2009*, ACM Press (2009), 13–22.
12. Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc of UIST 2005*, ACM Press (2005), 115–118.
13. Hourcade, H.P., and Bederson, B. Architecture and implementation of a java package for multiple input devices (MID). *Univ. of Maryland Human–Computer Interaction Lab. (HCIL)*. Tech. report no. 99–08 (1999).
14. Kin, K., Hartmann, B., DeRose, T., and Agrawala, M. Proton: multitouch gestures as regular expressions. In *Proc. of CHI 2012*, ACM Press (2012), 2885–2894.
15. Letondal, C., Chatty, S., Phillips, G., André, F., and Conversy, S. Usability requirements for interaction-oriented development tools. *Psychology of Programming*, Maria P. D. Pérez and M.B. Rosson (2010), 12–26.
16. Marquardt, N., Kiemer, J., Ledo, D., Boring, S., and Greenberg, S. Designing user-, hand-, and handpart-aware tabletop interactions with the TouchID toolkit. In *Proc. of ITS 2011*, ACM Press (2011), 21–30.
17. Microsoft Surface, www.microsoft.com/surface/
18. Morris, M.R., Huang, A., Paepcke, A., and Winograd, T. Cooperative gestures: multi-user gestural interactions for co-located groupware. In *Proc. of CHI 2006*, ACM Press (2006), 1201–1210.
19. Morris, M.R. TeamTag: exploring centralized versus replicated controls for co-located tabletop Groupware. In *Proc. of CHI 2006*, ACM Press (2006), 1273-1282.
20. Morris, M. R. Designing Tabletop Groupware. In *Adjunct Proc. of UIST 2005*, ACM Press (2005).
21. Myers, B.A. Separating application code from toolkits: eliminating the spaghetti of callbacks. In *Proc. of UIST 1991*, ACM Press (1991), 211–220.
22. Myers, BA, Stiel, H., and Gargiulo, R. Collaboration using multiple PDAs connected to a PC. In *Proc of CSCW 1998*, ACM Press (1998), 285–294.
23. Newman, W.M. A system for interactive graphical programming. In *Proc. of AFIPS 1968*, ACM Press (1968), 47–54.
24. Partridge, G.A., and Irani, P.P. IdenTTop: a flexible platform for exploring identity-enabled surfaces. In *Ext. Abstr. of CHI 2009*, ACM Press (2009), 4411–4416.
25. Piper, A.M., O'Brien, E., Morris, M.R. and Winograd, T. SIDES: A cooperative tabletop computer game for social skills development. In *Proc. of CSCW 2006*, ACM Press (2006), 1–10.
26. Ryall, K., Esenther, A., Forlines, C., Shen, C., Shipman, S., Morris, M.R., Everitt, K., and Vernier, F. Identity-differentiating widgets for multiuser interactive surfaces. *IEEE Comput. Graph. 26*, 5 (2006), 56–64.
27. Schmidt, D., Chong, M.K., and Gellersen, H. IdLenses: dynamic personal areas on shared surfaces. In *Proc. of ITS 2010*, ACM Press (2010), 131–134.
28. Shen, C., Vernier, F., Forlines, C., and Morris, M.R. DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proc. of CHI 2004*, ACM Press (2004), 167–174.
29. Stewart, J., Bederson, B., and Druin, A. Single display groupware: a model for co-present collaboration. In Proc. of CHI 1999, ACM Press (1999), 286–293.
30. Tse, E., Greenberg, S., Shen, C., Forlines, C., and Kodama, R. Exploring true multi-user multimodal interaction over a digital table. In *Proc. of DIS 2008*, ACM Press (2008), 109–118.
31. Tse, E., and Greenberg, S. Rapidly prototyping Single Display Groupware through the SDGToolkit. In *Proc. of AUIC 2004*, Australian Computer Society (2004), 101–110.
32. Tuddenham, P., and Robinson, P. T3: A toolkit for high-resolution tabletop interfaces. In *Ext. Abstr. of CSCW 2006*, ACM Press (2006), 2237–2242.
33. Wellner, P. Statemaster: A UIMS based on statechart for prototyping and target implementation. In *Proc. of CHI 1989*, ACM Press (1989), 177–182.

Débridons l'interaction homme-machine pour une meilleure qualité des soins

Requis centrés utilisateurs et Interfaces perlées pour les systèmes d'information de santé

Laure MARTINS-BALTAR[1], Yann LAURILLAU[2], Gaëlle CALVARY[1]

1. Institut polytechnique de Grenoble, CNRS, Laboratoire LIG
41 rue des Mathématiques, BP 53, 38041 Grenoble cedex 9, France
{nom.prenom}@imag.fr

2. Université Pierre-Mendès France, CNRS, Laboratoire LIG
41 rue des Mathématiques, BP 53, 38041 Grenoble cedex 9, France
{nom.prenom}@imag.fr

RESUME. L'importance croissante des TIC en santé modifie les pratiques médicales. Le développement de solutions reposant sur le Cloud, incluant le Social Computing, accentue ce phénomène. Aussi, l'émergence de cette informatique périme la métaphore actuelle du bureau, centrée données. Il faut l'ouvrir aux acteurs, tâches et relations sociales qui les unissent. Cet article propose le concept de "perle" pour les contextes d'usage dans une approche services à la demande, appliqué à la gastroentérologie. Il présente un démonstrateur de concept dans le domaine médical et une formalisation du concept de Perle.

ABSTRACT. IT is deeply modifying the medical practices: Cloud-based solutions, including Social Computing, are growing in the medical field. However, this emergence of utility computing such as Cloud Computing makes the current desktop metaphor outdated. There is a need to open the desktop so that to, in addition to classical data, also consider actors and tasks as well as their relationships. This paper proposes the concept of "Pearl" for both denoting and representing socially augmented data. It describes a health care prototype as well as a formal definition of the related concept.

MOTS-CLES : Nuage, Réseaux sociaux, valeur sociale ajoutée, IHM Perlée, santé.

KEYWORDS: Cloud, Social Network, Socially Augmented Entity, Pearly UI, Health Care.

DOI:10.3199/JESA.45.1-n © Lavoisier 2012 AR_DOI

1. Contexte et questions de recherche

L'importance croissante des Technologies de l'Information et de la Communication (TIC) dans le domaine de la santé, comme les progiciels de santé, incitent les professionnels de la santé à produire de plus en plus de données médicales dématérialisées, à les conserver mais aussi à les traiter, à les organiser ainsi qu'à les partager : dossier médical du patient, comptes-rendus d'examen, images de scanners, résultats d'analyses biologiques, etc. Au plan national, le Dossier Médical Personnel[1] (DMP) illustre parfaitement cette tendance. L'expansion de cette **masse de données médicales** s'explique également par les évolutions des équipements d'acquisition d'images médicales et d'archivage, comme les solutions de type PACS (*Picture Archiving and Communication System*). Cependant, alors que cette pratique est devenue courante en radiologie, ces solutions restent encore compliquées à mettre en œuvre pour la plupart des autres spécialités, comme celles nécessitant d'exercer au bloc opératoire, et pour exporter l'image médicale en dehors du lieu d'acquisition. Ceci est d'autant plus complexe et critique que le praticien exerce dans de **multiples contextes d'usage**, nécessitant un **accès ubiquitaire** aux données et aux outils : dans différents lieux (au cabinet, au bloc, à domicile, etc.) ; avec différents dispositifs d'interaction (ordinateur de bureau ou portable, colonne endoscopique, téléphone ou tablette, dictaphone, etc.) (voir Morán et al., 2010) ; ou en situation d'urgence (Koufi *et al.*, 2010). Plus généralement, au-delà de la production de données médicales, la pratique médicale est en mutation sur de nombreux aspects pour intégrer cette part croissante des TIC : aide au diagnostic, partage des données médicales avec les confrères et les patients, évaluation des pratiques professionnelles, capitalisation des connaissances, etc.

Actuellement, de nombreuses solutions reposant sur le *Cloud Computing* sont proposées dans le domaine de la santé. Une infrastructure de type *Cloud* s'appuie sur des technologies certes existantes (virtualisation, nœud de calcul) mais dont l'association démultiplie leur potentiel, visant à faciliter leur exploitation : accès à une grande puissance de calcul et grande capacité de stockage, environnements virtualisés accessibles par des liens à très haut débit (Zang C. *et al.*, 2010). Cette technologie promeut un **accès ubiquitaire** à des **services à la demande** et aux **masses données** (Mell *et al*., 2009). Parmi les travaux existants dans le domaine de la santé, Many Avial-Garcia *et al* (2008) proposent une solution reposant sur le *cloud* pour apporter un service d'aide au diagnostic quant à la détection de cancer colorectal par analyse d'image. D'autres exemples de systèmes reposant sur le *cloud* sont proposés pour automatiser le processus de collecte des données médicales relatives aux patients (en situation) et traitées par des systèmes experts, accessibles ensuite via n'importe quel dispositif par le personnel médical (Rolim et *al.*, 2010). L'approche service à la demande et puissance de calcul est également traitée pour proposer des outils de visualisation d'images médicales 3D sur des dispositifs ayant

[1] www.dmp.gouv.fr

peu de ressources comme les assistants personnels (PDA pour *Personal Digital Assitant*) (Van der Burg *et al.*, 2009).

Parallèlement, les avancées en *Social Computing*, dont les solutions sont d'ailleurs souvent adossées au *cloud* (par exemple, Facebook), connaissent également un intérêt dans le domaine médical (Keckley et Hoffman, 2010), autant dans la relation avec le patient et dans la gestion de la santé publique, qu'entre professionnels de santé pour, par exemple, faciliter les échanges sur les pratiques médicales ou pour rechercher les traitements les plus adaptés (Keckley et Hoffman, 2010). Plus généralement, il s'agit de favoriser et faciliter la collaboration entre praticiens (Eysenbach, 2008) en leur permettant de développer leur **réseau socio-professionnel médical**.

Ces approches permettant un accès ubiquitaire à des services à la demande et aux masses de données en tout contexte d'usage soulèvent de nombreuses questions du point de vue de l'**interaction utilisateur**. La métaphore classique du bureau semble être obsolète vis-à-vis des avancées apportées par le *Cloud Computing* et le *Social Computing* : le schéma mental de l'utilisateur jusqu'ici construit sur la physicalité et la localisation des ressources est en rupture avec ce nouveau modèle (Odom et al., 2012) ; la génération d'applications web pour ce type d'approches conduit à une expérience utilisateur fragmentée, mettant à mal la cohérence des interfaces utilisateurs assurée localement par le gestionnaire de fenêtres (England *et al.*, 2011). Aussi, quelles propriétés nouvelles une interface utilisateur doit-elle avoir pour permettre le déploiement dynamique et ubiquitaire de tels services tout en laissant le pouvoir à l'utilisateur ? L'importance croissante des réseaux sociaux au sein des applications soulève également les questions de vie privée, de confiance et de propriétés des données (Armbrust *et al.*, 2010 ; Zang *et al.*, 2010a) ? Quel *design* d'interface utilisateur proposer pour assurer la traçabilité, le contrôle et le partage des données dans une optique de socialisation des services dans le *cloud* ainsi que les activités en ligne (Odom *et al.*, 2012 ; Zang *et al.*, 2010b) ?

Le travail présenté dans cet article aborde ces questions du point de vue de l'Interaction Homme-Machine (IHM). Le cas d'étude provient du domaine de la gastroentérologie. Précisément, nous revisitons la métaphore usuelle du bureau (le *Desktop*) et proposons une nouvelle métaphore d'interface utilisateur, la **Perle**, comme fondement aux futures générations d'interfaces utilisateur. La Perle intègre les trois pivots du *Cloud Computing* (services à la demande, accès aux masses de donnés et ubiquité) ainsi que la notion de réseau social.

Dans la suite, nous présentons tout d'abord la méthode retenue pour conduire ce travail. Notamment, dans la partie 3, nous présentons deux phases importantes de cette méthode reposant sur une étude de terrain et de cas basée sur la rétroconception d'une application patrimoniale, impliquant des acteurs du monde médical. Nous concluons cette partie sur la spécification de requis utilisateurs servant ainsi de guide à la proposition de la métaphore de la Perle. Cette dernière est présentée en partie 4 au travers d'exemples de représentations et d'une proposition d'implémentation. Avant de conclure, nous positionnons la métaphore de la Perle au regard des travaux existants.

2. Méthodologie

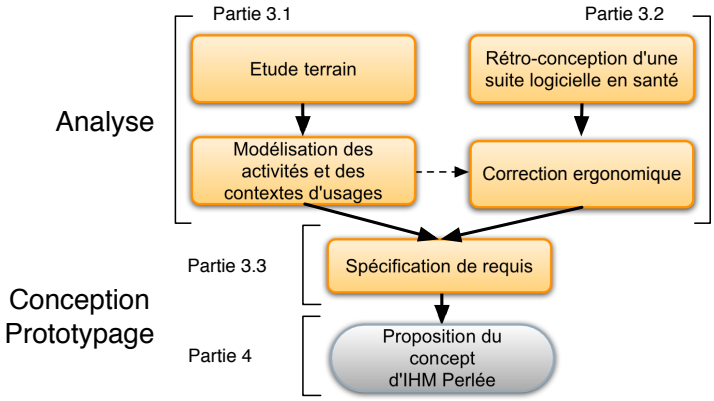


Figure 1. Principales phases de la démarche.

En partenariat avec des professionnels de la santé et des experts des technologies de l'information pour la santé, nous avons appliqué une démarche centrée utilisateur, au plus près du terrain. La phase d'analyse a consisté à mieux comprendre et formaliser les pratiques médicales pour en extraire des requis utilisateur. Deux sources de données ont été explorées : d'une part, l'observation sur le terrain de praticiens en action puis la modélisation de leurs activités et contextes d'usages ; d'autre part, l'étude d'une application patrimoniale, nommée Syseo, pour la gestion d'images médicales, de comptes-rendus médicaux et de dossiers patients en gastroentérologie. L'étude a consisté en une rétroconception de l'application puis en la proposition d'améliorations ergonomiques pour toujours mieux comprendre les tâches utilisateur attendues. Dans ces deux volets d'analyse, de nombreux modèles (des diagrammes UML) de l'activité médicale ont été produits. Ils ont tous été validés par les professionnels de la santé.

L'analyse critique des pratiques a permis de spécifier des requis utilisateur pour une meilleure qualité des soins. Ces requis ont cadré la proposition de la nouvelle métaphore : les IHM perlées dont un exemple de conception est ici relatée.

3. Requis utilisateur

Cette détaille les deux phases d'analyse du domaine de la gastroentérologie (étude terrain et étude de la suite logicielle Syséo dédiée à la gastroentérologie) pour conduire, en conclusion, à la spécification de requis utilisateurs. Ces requis constituent les germes à la proposition de la métaphore d'IHM perlées.

3.1 Etude terrain

Afin de comprendre les pratiques médicales en gastroentérologie et d'identifier les différents contextes d'usage des Systèmes d'Information (SI) de santé, nous avons conduit une étude en trois étapes : (1) interviews de médecins et secrétaires ; (2) analyse et modélisation de leurs usages, pratiques et besoins ; (3) validation des observations récoltées par les différents acteurs impliqués.

Cette étude a conduit à la réalisation de nombreux modèles (17 cas d'utilisation, 10 modèles d'activités, soit un total 29 diagrammes UML) décrivant le processus de suivi des patients en gastroentérologie. Ces modèles ont été validés par les médecins et nous fournissent une vision précise du domaine.

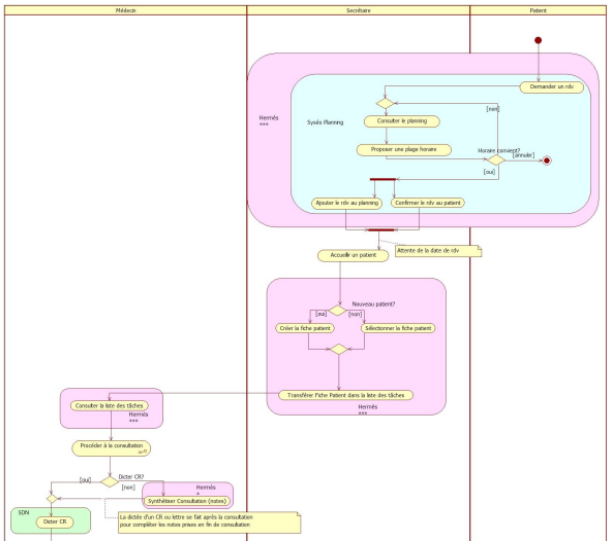


Figure 2. Exemple de modèle décrivant l'activité médicale et l'usage des TIC.

En particulier, ces modèles mettent en évidence l'importance des TIC dans l'accomplissement de l'activité médicale, ainsi que la grande variété des contextes d'usage. Par exemple, la Figure 3 est un extrait d'un modèle UML qui décrit l'activité médicale au cabinet médical. Il montre les différentes tâches et acteurs impliqués (i.e. médecin, secrétaire et patient) et montre comment les différentes activités s'articulent (i.e. *workflow*). Des rectangles aux angles arrondis et colorés (i.e. cyan, magenta et vert) indiquent les outils ou services logiciels employés pour accomplir une tâche (jusqu'à trois outils logiciels différents sont utilisés dans l'exemple montré à la Figure 2).

Cette étude a également révélé l'importance et surtout la variété des données médicales manipulées (données cliniques, rapports d'examen et images, etc.). En gastroentérologie, les images sont des données clés pour la prise de décision et

l'évaluation des pratiques, mais également pour la pédagogie et la recherche. Plusieurs besoins sont également ressortis de l'étude du terrain : le partage des données médicales, la communication entre collègues pour des expertises, et la centralisation des connaissances médicales. La socialisation de l'activité médicale est une requête importante des professionnels de santé. Lors de la prise de décision sur certains dossiers, les médecins se sentent seuls et souhaiteraient pouvoir échanger facilement avec des confrères. Cet échange existe aujourd'hui, mais reste très informel (surtout par téléphone) pour des raisons médico-légales. Ce besoin est accentué par le fait que les services et les applications logicielles restent centrés données, ignorant cette dimension sociale de l'activité médicale comme les réseaux développés par les professionnels de santé. Ce besoin fait écho à un constat posé il y a quelques années avec l'émergence de la Santé 2.0, c'est-à-dire l'application des technologies Web 2.0 au système de soin en incluant les réseaux sociaux (Keckley et Hoffman, 2010). En effet, nous avons observé que les médecins entretiennent quatre réseaux socioprofessionnels : (1) réseau des professionnels de santé, comprenant collègues, experts et amis ; (2) des universitaires et des étudiants ; (3) des acteurs médicaux concernés pour le suivi des patients, et (4) des institutions.

3.2 Etude de la suite logicielle Syséo

Dans un premier temps, nous avons étudié et rétroconçu une suite logicielle utilisée par les praticiens. Le but était de comprendre, du point de vue de l'interaction utilisateur, quelles sont les principales tâches, comment les données médicales numériques sont gérées et comment ils interagissent avec le système d'information usuel (hôpital, institutionnel ou personnel). Précisément, nous nous sommes concentrés sur la suite logicielle Syséo.

La suite logicielle Syséo couvre le parcours du patient de l'examen médical à la rédaction du rapport d'examen. Cette suite est composée de sept modules applicatifs : (1) un planning Web adapté au domaine médical ; (2) une application web pour produire et suivre des dossiers médicaux ; (3) un module dédié à la production et à l'archivage d'images endoscopiques ; (4) trois modules pour annoter, manipuler et stocker des images endoscopiques ; (5) un module pour l'évaluation des pratiques professionnelles.

La phase de rétroconception nous a conduits à (1) produire de nombreux modèles de tâches ainsi qu'à l'identification de problèmes d'ordre ergonomiques et à (2) identifier des classes de services en ligne pouvant contribuer à l'amélioration des soins : services reposant sur le réseau socio-médical des praticiens afin de mieux identifier les autres professionnels de santé impliqués dans le suivi d'un patient ; services facilitant le partage de l'image médicale tout en préservant la confidentialité ; services facilitant l'accès ubiquitaire au données médicales pour différents contextes d'usage (dispositifs d'interaction ou localisation), etc.

La nouvelle interface comporte deux éléments d'interface clés. Premièrement, comme le montre la Figure 3, la nouvelle IHM a été repensée pour éliminer les problèmes ergonomiques identifiés. La Figure 3 illustre une partie de l'IHM dédiée

à la gestion des données médicales : le dossier médical, donnée centrale de l'interface, est un regroupement de toutes les informations concernant un patient organisées autour de l'image.

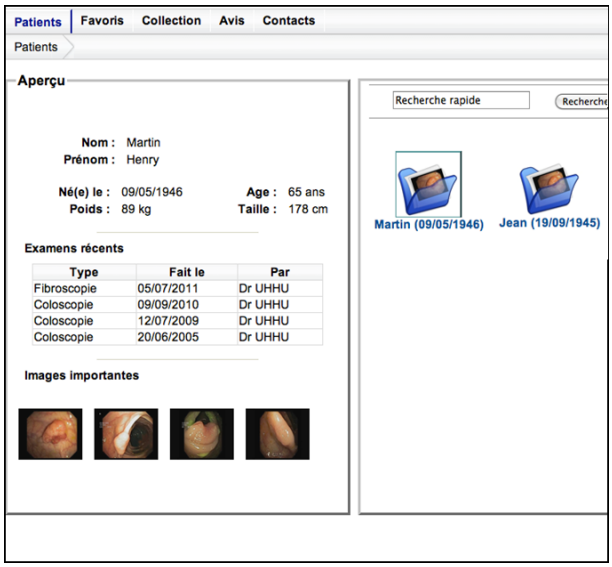


Figure 3. Exemple d'IHM issue de la rétroconception de Syséo.

Deuxièmement, le second élément clé de cette nouvelle IHM propose une nouvelle approche pour gérer les données médicales en lien avec le réseau socio-professionnel du praticien. Cette nouvelle interface, outre l'aspect gestion des données médicales, intègre des aspects du *Social Computing* grâce à une fonction de partage des données médicales, offrant ainsi un moyen aux praticiens de soumettre des demandes d'avis à des confrères. Le but premier est d'aider le praticien à formuler un diagnostic plus précis et, par conséquent, contribuer à l'amélioration des soins.

3.3 Synthèse : requis utilisateur et scénarios problème

A partir de l'étude de terrain et de la rétroconception de la suite logicielle Syséo, complétées par des interviews de professionnels de la santé, nous avons identifié trois requis de premier ordre pour les futures IHM de santé, nous concentrant sur les aspects suivants : la dimension sociale des activités médicales, l'accès ubiquitaire aux données médicales ainsi que leur gestion. Les trois requis sont :

Promouvoir la dimension sociale des activités. Les réseaux socio-médicaux sont un aspect important de l'activité médicale (Eysenbach, 2008). Des visualisations de ces réseaux semblent opportuns et permettent, par exemple,

d'observer quels sont les contacts les plus proches, les donnés médicales partagées avec des confrères, ou de visualiser qui est le médecin référent d'un patient. Une infrastructure de type *Cloud* constituerait un moyen pour mettre en œuvre des services facilitant la visualisation des relations au sein d'une même famille pour identifier des maladies héréditaires ou des systèmes d'aide au diagnostic. Qu'ils soient éphémères (relatifs à un patient) ou persistants (relatifs aux confrères), les médecins interviewés soulignent que les réseaux socio-médicaux sont des facettes importantes de l'activité médicale. (Morán *et al.*, 2010) souligne le *« besoin de coordination et de collaboration entre praticiens, l'intensité de la communication, l'usage de différents dispositifs d'interaction, et la grande mobilité du personnel médical, des patients et des documents* ».

Assurer le partage et la traçabilité. Le besoin de support pour le partage des données médicales et la communication entre professionnels de la santé devient essentiel. Ces situations d'échange sont de plus en plus fréquentes : entre praticiens pour le suivi du patient, avec les laboratoires d'analyse, les hôpitaux pour la gestion des admissions, des sorties, etc. Malgré l'importance des technologies dans le monde médical, le dossier médical est divisé en plusieurs instances : le dossier de l'anesthésiste, le dossier de l'infirmière, le dossier des médecins, et le dossier administratif. Ce manque de partage est un réel problème. En effet, les médecins n'ont pas un accès complet à l'ensemble des informations concernant un patient, ce qui est un point crucial pour connaître, par exemple, les différents traitements ou pour savoir si un patient est correctement suivi : les confrères sont souvent réticents à partager les données médicales. Ce problème peut s'expliquer par la peur de perdre le contrôle sur les données médicales, notamment dans la perspective de disposer de solutions reposant sur le *Cloud*.

Accéder facilement et rapidement aux données médicales. Au cours de sa carrière, un praticien recueille habituellement entre 15 000 et 30 000 dossiers médicaux. En effet, la crainte d'explorer une énorme quantité de données médicales est mise en avant, en particulier lorsque des conseils sont demandés par téléphone. L'ajout de la dimension sociale aux données médicales constitue une opportunité pour faciliter la navigation. Toutefois, le passage à l'échelle est un point critique : comment représenter cette masse de données sans perturber l'utilisateur ? La notion de filtrage est indispensable.

Nous illustrons ces requis sur des scénarios problème (Rosson et Carroll, 2002) élaborés en partenariat avec des médecins gastroentérologues. Ils mettent en scène le Docteur Roger, 45 ans, spécialiste de l'endoscopie hépato-gastroentérologique, exerçant en libéral. Parfois, le Dr Roger exerce également en hôpital public pour réaliser des examens au bloc opératoire, et assure des enseignements à la Faculté de Médecine. Son activité étant chronophage, le Dr Roger souhaiterait : (a) pouvoir passer plus de temps avec ses patients et moins de temps à utiliser les TIC ; (b) pouvoir facilement capturer, annoter et partager les moments clés d'un examen (images ou vidéo) ; (c) centraliser toutes ses données médicales ; (d) être plus facilement en contact avec ses confrères pour échanger à propos de pathologies complexes et sur des questions relatives au diagnostic ; (e) informer le patient.

Aujourd'hui, Dr Roger examine un patient pour retirer un polype détecté par un confrère (la Figure 5 est une planche de bande dessinée illustrant ce scénario) :

Vignette N° 1 : *besoin d'un accès facilité aux données médicales et partage simplifié des données médicales*. Avant de commencer l'examen, le Dr Roger relit rapidement le dossier médical du patient (dernier examen, pathologies détectées, antécédents, etc.). Toutefois, le dossier médical contient peu d'information et il regrette d'en avoir si peu : des images endoscopiques l'aideraient à localiser plus facilement le polype.

Vignette N° 2 : *besoin d'exporter des données médicales, de capturer et annoter des images et vidéos médicales*. Pour des raisons légales, le Dr Roger doit obligatoirement prendre des captures d'image ou des vidéos des interventions chirurgicales. Seule une colonne endoscopique permet la capture d'image. Cependant, en sortie, ces images ne peuvent être qu'imprimées et jointes au dossier médical au format papier.

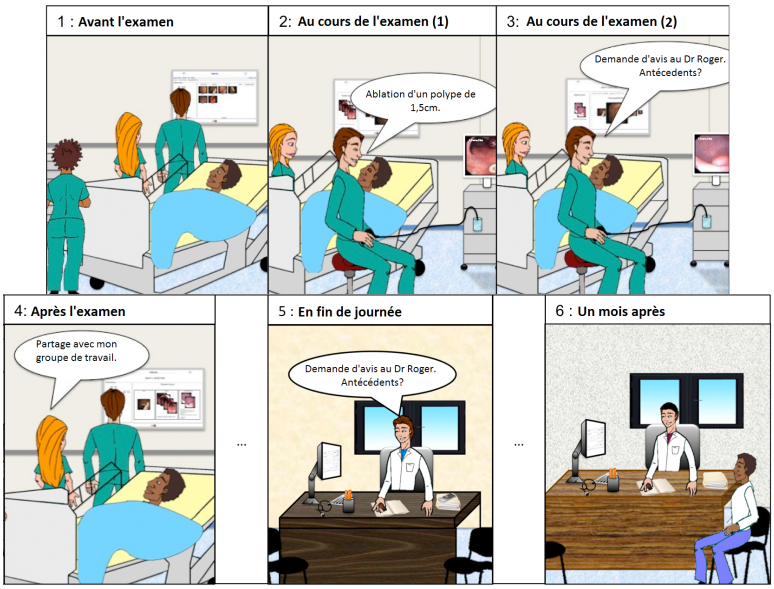


Figure 5. Illustration des requis en bande dessinée.

Vignette N° 3 : *besoin pour des services en ligne dédiés au partage de données médicales, à la collaboration entre praticiens, ainsi qu'à l'analyse de données médicales*. Le manque de données médicales pose évidemment problème pour la prise de décision. Dr Roger ne connaît pas les antécédents du patient qu'il examine, ni son suivi. Il a également besoin de pouvoir comparer les images qu'il vient de capturer avec des pathologies similaires et de demander un avis médical à un confrère avant d'établir un diagnostic.

Vignette N° 4 : *besoin pour capitaliser et rendre anonymes des données médicales*. Claire est étudiante en médecine et assiste le Dr Roger pour cet examen. Dr Roger suggère qu'elle assiste à quelques examens. Aussi, elle aimerait pouvoir partager certaines des données médicales (images, commentaires du Dr Roger) avec son groupe de travail constitué d'autres étudiants en médecine. Toutefois, pour des raisons médico-légales, elle ne le peut pas : elle n'a pas accès aux images endoscopiques et les notes qu'elle prend pendant l'intervention au bloc sont confidentielles.

Vignette N° 5 : *besoin de services en ligne pour le partage de données médicales et la collaboration entre confrères*. En fin de journée, le Dr Roger retourne à son cabinet. Il doit faire le point sur les dossiers médicaux du jour, et, si cela s'avère approprié et nécessaire, appelle des confrères pour un avis médical ou pour demander l'historique médical d'un patient. Cette activité est très chronophage mais c'est une étape incontournable pour pouvoir établir un diagnostic plus précis.

Vignette N° 6 : *besoin d'un accès simple et rapide aux données médicales et pour les partager facilement*. Un mois plus tard, M. Karl, un des patients du Dr Roger, a rendez-vous avec le Dr Bernard, son médecin référent. Ce rendez-vous est l'occasion de faire le point sur le dernier examen et M. Karl a quelques questions à propos de la dernière colonoscopie. Dr Bernard lui fait un résumé des conclusions du dernier examen à partir de la copie papier qui lui a été remise et lui montre les différentes images. Il lui explique que tout est normal désormais. En complément, le Dr Bernard souhaiterait lui montrer les images d'avant le traitement mais, malheureusement, le dossier n'en contient pas.

Afin de répondre aux requis identifiés et illustrés par ce scénario, nous proposons le concept de perle comme fondement aux futures générations d'IHM.

4. Métaphore de la Perle

Nous présentons la métaphore de la Perle pour la conception d'une nouvelle génération d'IHM visant à embrasser les caractéristiques du *cloud* (masses de données, services à la demande) et du *Social Computing*. En particulier, à travers cette métaphore, nous étudions la conception d'interfaces utilisateur centrées sur la notion de service.

Cette partie détaille les deux fondements de la métaphore de la Perle : (1) sur le plan conceptuel, la métaphore repose et étend trois concepts pivots usuels (donnée, tâche, acteur) augmentés de la dimension sociale ; (2) en termes de présentation, la métaphore promeut une visualisation interactive des contextes d'usage et des relations sociales.

4.1 Entités pivots

Nous considérons des entités de type donnée, acteur et service comme fondements de la métaphore de la Perle, en support à la réalisation d'une tâche en

contexte. Ces entités font référence aux concepts usuels de tâche, d'objet du domaine et d'agent, tels qu'ils sont traités dans les domaines de l'IHM et du Travail Collaboratif Assisté par Ordinateur (TCAO) (Van Welie *et al.*, 1998). Notamment, le service est abordé comme étant un des moyens disponibles pour accomplir la tâche.

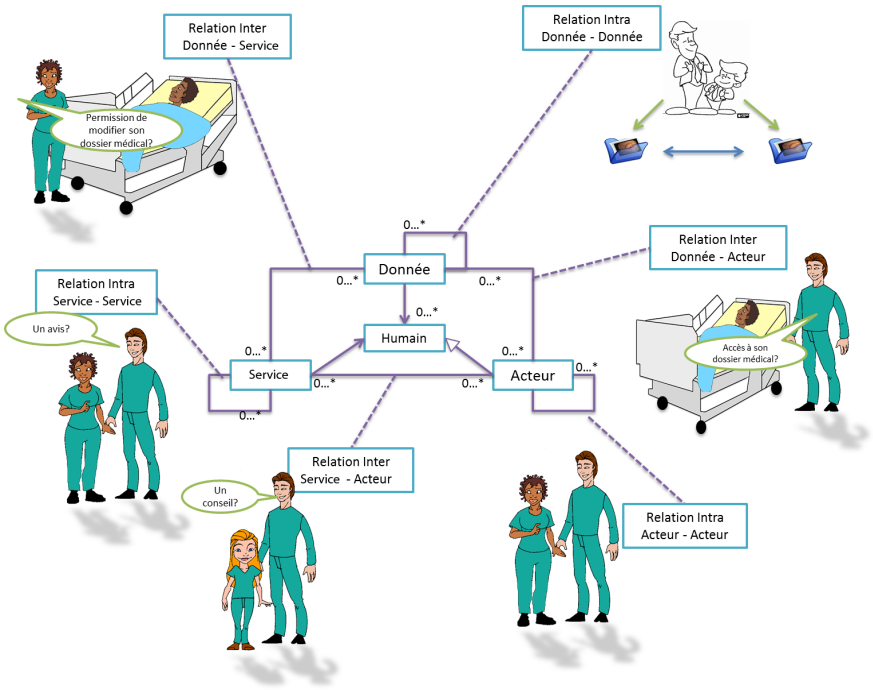


Figure 5. Taxonomie des entités pivots et à valeur sociale ajoutée (VSA).

L'entité **Donnée** dénote l'information manipulée par un acteur lorsqu'il accomplit une tâche. Dans le contexte médical, une donnée peut être aussi bien un dossier médical qu'une image endoscopique, etc. Des services en ligne, tels qu'un service d'analyse d'image ou de reconstruction 3D, sont consommateurs de données.

L'entité **Acteur** dénote les individus impliqués dans le système : ils manipulent des données et réalisent des tâches. Appliqué au médical, les acteurs peuvent être, par exemple, des médecins, des secrétaires médicales ou des étudiants en médecine. Des services en ligne peuvent être mis à leur disposition, tels qu'un système de *workflow* en santé.

L'entité **Service** dénote le moyen employé par des acteurs pour accomplir une tâche afin d'atteindre un but. Sachant qu'une tâche est classiquement décomposée en sous-tâches, cette approche s'applique également à la notion de service : des services sont également en support à l'accomplissement des sous-tâches. Appliqué

au médical, il peut simplement s'agir du suivi du patient. Cela signifie qu'un médecin gastroentérologue, afin d'accomplir ce suivi, doit réaliser de nombreuses tâches et faire appel aux services idoines pour les accomplir, par exemple : pratiquer des examens, produire des images endoscopiques, échanger avec des confrères, etc.

4.2 Entités à Valeur Sociale Ajoutée

Sur la base de ces entités pivots, nous visons à ajouter la dimension sociale. En effet, comme cela a été mis en évidence dans la partie 3, les médecins expriment le besoin et l'importance croissante d'intégrer le réseau socio-profesionnel dans les applications logicielles de santé. Ce besoin est en concordance avec les efforts récents accomplis pour faire converger le *Cloud Computing* avec le *Social Computing* (Pham, 2010).

Aussi, nous enrichissons les entités pivots pour prendre en compte leur valeur sociale et mettre en évidence les relations entre entités sur la base de cette valeur. En effet, comme l'exprime Lahire (Lahire, 2010), cette valeur d'ordre social ne se réduit pas aux interactions entre groupes. Selon Lahire, toute entité est connotée par une valeur d'ordre social dès lors qu'elle intervient dans une relation entre deux individus. Ainsi, par exemple, un document, et plus généralement une donnée, comme une trace de communication entre deux être humains est associé à une relation sociale et, dès lors, possède une valeur d'ordre social. On peut également parler de statut social. De ces constats, nous proposons une taxonomie d'entités à **Valeur Sociale Ajoutée** (VSA) enrichissant les entités pivots (Donnée, Acteur et Service), mettant en évidence leurs relations **intra**- et **extra**- (voir Figure 5).

Les relations entre les entités pivots constituent une information supplémentaire, aboutissant ainsi aux entités à Valeur Sociale Ajoutée. Les relations **intra** font référence aux relations entre entités de même nature :

Acteur-Acteur : les acteurs sont les utilisateurs du système. Dans notre cas d'étude, plusieurs acteurs ont été identifiés : les médecins, les secrétaires, les patients, et les étudiants en médecine. La notion de proximité sociale entre acteurs (par exemple, la collaboration fréquente entre deux médecins, la relation patient-médecin) peut enrichir les entités dites classiques et être considérée comme des entités VSA.

Donnée-Donnée : ces entités ne sont pas simplement liées entre elles, comme les liens hiérarchiques ou les relations dans une base de données, mais ont une réelle valeur sociale ajoutée. La décoration d'un lien par son type en est un bon exemple (relation généalogique entre le dossier médical d'un père et celui de son fils pour la prédiction de pathologie héréditaire).

Service-Service : Etiqueter les services avec les bonnes pratiques des experts est un exemple d'entité VSA. Cette information supplémentaire permettrait de proposer aux médecins des recommandations. Aussi, des services en ligne de recommandations reposant sur des systèmes experts peut s'appliquer à ces entités VSA.

Les relations **extra** font références entre entités de nature différente :

Donnée-Acteur : les acteurs produisent et manipulent les données pour atteindre leurs objectifs. Les données, quant à elles, donnent des informations sur ces différents acteurs. Le lien social est très fort entre les deux. Dans notre cas d'étude, un dossier médical (donnée) est associé au patient (acteur) et bien sûr au médecin (acteur) qui traite ce patient. Ce lien peut également donner lieu à des relations indirectes entre deux acteurs : ces acteurs ne se connaissent pas forcément mais ils partagent de l'information. Des services en ligne comme le partage d'information (confidentiel ou anonyme) ou recherchant des corrélations entre dossiers médicaux peuvent s'appliquer à ces entités VSA.

Donnée-Service : un service est socialement lié à une donnée dès lors que le statut social d'une donnée influence sur l'accomplissement d'une tâche. Par exemple, la mise à jour (tâche) d'un dossier patient (donnée) par un autre médecin que le médecin référent est possible seulement si ce dossier est partagé. A l'inverse, une tâche peut implicitement apporter une valeur sociale à une donnée, comme la production (tâche) d'une image (donnée) qui sera ajoutée au dossier médical disponible par le patient et le médecin, ou le partage (tâche) d'une image (donnée) pour une demande d'expertise. Des services comme un tableau de bord de l'activité médicale (dernier examen, dernier rendez-vous, dernière mise à jour, etc.) peut s'appliquer à ces entités VSA.

Service-Acteur : un acteur peut employer des services différents pour accomplir une même tâche, dans des contextes sociaux différents. Par exemple, dans le domaine médical, la prise de décision peut se faire lors d'une réunion pluridisciplinaire, impliquant différents spécialistes, ou lors d'un examen, impliquant seulement le chirurgien. Des services spécifiques, tels que faire une demande d'avis ou partager une donnée médicale, peut s'appliquer à ces entités VSA.

4.3 Graphes homogènes d'entités pivots

Nous considérons que les entités pivots de même nature (acteur, donnée ou service) sont organisées en graphe selon un type de relation et de métriques. Plusieurs graphes peuvent être produits à partir d'un même ensemble d'entités (i.e. une vue sur les entités) suivant le choix d'une classe d'entités, d'un type de relation et de métriques. Nous considérons donc trois types de graphes : graphe d'Acteurs, graphe de Données et graphe de Service.

L'organisation des entités en graphe permet ainsi de révéler les relations **intra**, c'est-à-dire de préciser les entités VSA entre entités de même nature. Cette organisation permet également de mettre en évidence des propriétés intéressantes :

Graphe d'acteurs. Un graphe d'acteur met en évidence les relations sociales entre acteurs : c'est la notion usuelle de réseau social. Les acteurs en sont les sommets. Les relations entre acteurs constituent les arcs et la proximité entre acteurs est déterminée selon le poids des arcs ou de la distance entre acteurs. Un graphe

d'acteurs permet ainsi d'expliciter les relations **intra** entre entités **Acteurs**, et exprime les quatre caractéristiques suivantes :

- *Sens de transmission de l'information* : la communication entre deux acteurs peut être unilatérale ou bilatérale. Cela conduit à obtenir des graphes orientés pouvant contenir des arcs non orientés (Zijlstra, 1979). Les relations entre acteurs sont définies par des fonctions binaires dépendantes de la force de la proximité sociale entre deux acteurs.

- *Structure du groupe*. Un groupe peut être organisé hiérarchiquement ou de toute autre façon. A l'extrême, un acteur peut se retrouver isolé. Ce type de structure est représenté par des graphes déconnectés ou des graphes presque fortement connectés.

- *Proximité sociale*. La proximité entre acteurs peut se définir par un poids sur les arcs reliant deux acteurs. Aussi, une telle valeur constitue un moyen pour exprimer la proximité sociale (par exemple, le nombre de courriers électroniques échangés).

- *Dynamique d'un groupe*. Un groupe varie en taille au cours du temps. Des acteurs apparaissent ou disparaissent, c'est-à-dire lorsqu'une entité acteur est ajouté ou retiré de l'ensemble des acteurs. Par conséquent, les relations sont également dynamiques et peuvent évoluer, c'est-à-dire lorsqu'un arc apparaît ou disparaît.

Le nombre de relations Acteur-Acteur exprime l'importance d'un acteur au sein d'un groupe, d'un réseau. On parle alors de **popularité**, mesurée par le degré de centralité du graphe et calculé sur le nombre de relations directes de chaque acteur.

La distance entre deux acteurs exprime l'indépendance d'un acteur. Freeman (Freeman, 1979) établit que plus un acteur est éloigné, plus il sera autonome dans ses choix et actions. Cette propriété se démontre à l'aide du degré de séparation ou la propriété de centralité proximale.

Le rôle médiateur d'un acteur est une autre propriété intéressante qui peut être mise en évidence par la propriété de centralité intermédiaire. Freeman (Freeman, 1979) établit que plus un acteur est dans une position de médiateur, plus il aura le contrôle sur la circulation de l'information entre acteurs.

Graphe de données. Un graphe de données est composé d'entités de type données constituant les sommets. Un graphe de **Données** permet ainsi d'expliciter les relations **intra** entre entités Données, et exprime les quatre caractéristiques suivantes :

- *Dépendance* : une donnée peut être indépendante, associée ou dépendante. La dépendance est liée à l'orientation du graphe : unilatérale (relations orientées comme dans une structure hiérarchique) ou bilatérale.

- *Evolution* : un graphe de données évolue dans le temps en fonction de l'environnement et de l'utilisateur. Des données sont créées ou détruites. De même, les relations évoluent dans le temps.

- *Structure* : la structure peut être totalement ou partiellement hiérarchique voire totalement désintégrée. Le graphe est alors déconnecté ou presque fortement connecté.

- *Similarité* : la similarité entre deux données s'apparente à un poids sur les arcs les reliant. Dès lors, il est possible d'exprimer un degré de similarité entre deux entités de type données comme deux images médicales relatives à une même pathologie, ou deux fichiers ayant des différences mineures dans un système de gestion de version.

Graphe de services. Un graphe de services est un graphe sont les sommets sont des entités de type service. Un graphe de Services permet ainsi d'expliciter les relations **intra** entre entités **Services**, et exprime les deux caractéristiques suivantes :

- *Dépendance* : la dépendance est relative à un graphe presque fortement connecté (totalement ou partiellement hiérarchique). Dès lors, une relation unilatérale entre deux services exprime une dépendance entre deux services.

- *Disponibilité* : la mise à disposition dynamique de service, comme au sein d'une architecture de type c*loud*, dépend de leur disponibilité selon : (1) l'utilisateur qui interagit avec le service (en fonction, par exemple, de son rôle, statut, niveau d'expertise, etc.) ; (2) de l'environnement dans lequel évolue l'utilisateur (physique, social) ; et (3) les données traitées par le service (par exemple, selon le statut social).

4.4 La perle, graphe hétérogène d'entités pivots

Du point de vue de l'IHM, une Perle représente un contexte d'usage décrit par un ensemble d'entités pivots et par les relations entre ces entités (i.e. entités VSA) en support à la **réalisation d'une tâche** : services disponibles pour accomplir la tâche (i.e. moyens), données nécessaires et accessibles, et acteurs associés.

Formellement, une Perle est une collection hétérogène d'entités pivots organisées en graphe. Dès lors, une autre définition de la Perle est la suivante : une Perle est un arc d'un hypergraphe (arc à plusieurs sommets) dont les sommets sont des entités pivots : donnée, acteur et service. Ces sommets sont donc organisés en graphe au sein de cet arc : un tel graphe entre entités de nature différentes permet ainsi d'expliciter les relations **inter** entre les entités VSA. Le choix de classes d'entités pivot (par exemple, une classe d'acteur définie par un rôle) ainsi que le choix d'un type de relations entre entités (par exemple, des images médicales relatives à des acteurs appartenant à une même famille) et les métriques (par exemple, une proximité sociale entre acteurs par rapport à une pathologie) définit la structuration en graphe de ces entités au sein de l'arc et précise ainsi le contexte d'usage définit par une Perle. L'hypergraphe représente donc un ensemble des Perles, c'est-à-dire un ensemble de contextes d'usages.

A plus gros grain, il est possible d'identifier des classes Perles. Notamment, nous précisions les quatre classes de Perles suivantes :

Perle centrée Acteur et orientées données. Cette classe de perles organise les données relatives et appartenant à un acteur : il s'agit d'une spécialisation d'un

hypergraphe hors contexte auquel une contrainte du type « données appartenant à l'utilisateur/acteur X » est appliquée. De plus, comparé à la métaphore usuelle du bureau, l'appartenance des données est augmentée du statut social. En effet, d'un point de vue social, une donnée peut être privée, confidentielle ou publique. Le degré de partage dépend alors de son statut social. Son statut social peut être calculé à l'aide de métriques appliquées à l'hypergraphe.

Perle centrée Acteur et orientées services. Cette classe organise les services pour un acteur donné : il s'agit d'une spécialisation d'un hypergraphe hors contexte auquel une contrainte du type « service disponible pour l'utilisateur/acteur X » est appliquée. Cette classe permet de définir une notion de contexte pour une tâche que doit accomplir un acteur donné.

Perle centrée données et orientées services. Cette classe est similaire à la précédente. La différence est que la spécialisation est appliquée pour une classe de données. Par exemple, cela peut être l'ensemble des services dédiés au traitement d'images médicales : comparaison de pathologie, reconstruction 3D, etc.

Perle sociale centrées données. Cette classe organise les données en fonction d'un groupe d'acteurs. La structure permet de représenter comment les acteurs sont regroupés en communautés et comment les données sont partagées entre les différents acteurs et communautés.

Une Perle peut également être un arc de l'hypergraphe dont tous les sommets sont des entités de même nature : il s'agit d'une **Perle homogène**. Nous distinguons donc trois classes de Perles homogènes :

Perle sociale. Cette instance regroupe tous les acteurs partageant une même caractéristique (par exemple, un statut, un lieu de travail, une université, etc.) et sont reliés entre eux sous forme de graphe. Il s'agit d'une forme spécialisée du graphe d'Acteurs sous la contrainte « Acteurs avec la caractéristique X ».

Perle de services. De façon similaire, une instance de Perle de services regroupe tous les services partageant une même caractéristique (par exemple, selon un type, un coût d'usage, le niveau d'expertise de l'utilisateur, etc.) et sont reliées entre eux sous forme de graphe.

Perle de données. Une instance de Perle de données regroupe toutes les données partageant une même caractéristique. Leur regroupement est similaire à l'opération de sélection dans une base de données relationnelles. De même, leur organisation s'opère sous forme de graphe.

4.5 La Perle, représentation et implémentation d'une Perle

Nous proposons des représentations conceptuelles d'interface utilisateurs basées sur la métaphore de la Perle pour illustrer la mise en œuvre du concept. Celles-ci sont appliquées à la gastroentérologie et ciblent les praticiens comme classe d'utilisateurs. La Figure 6 illustre ces représentations conceptuelles pour un médecin fictif, le Dr. Roger :

Perle centrée acteur et orientée données. La Figure 6a montre une représentation de ce type de perle : elle est centrée sur un acteur, le Dr. Roger, situé au centre de la représentation (Perle au centre) ; il est entouré par les dossiers médicaux qui lui sont propres et qu'il traite. L'organisation de ceux-ci et la distance par rapport au centre peut dépendre de différentes métriques, comme la proximité dans le temps comme les patients du jour ou le nombre de consultations. De plus, cette représentation fait apparaître le réseau socio-professionnel du Dr. Roger puisque les liens entre ses dossiers et ses confrères sont rendus visibles (Perles sociales). Grâce à la visualisation des liens, il est alors possible d'indiquer le degré de partage, et donc le statut social, des données, c'est-à-dire les dossiers médicaux.

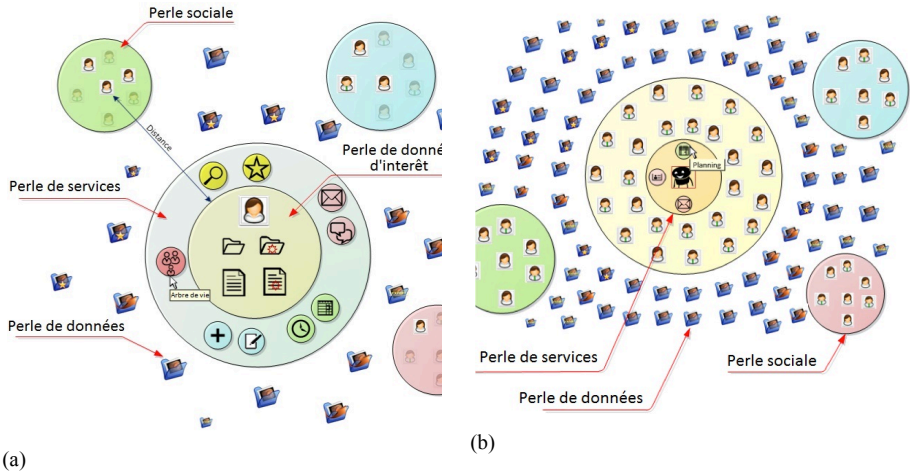


Figure 6. Représentations conceptuelles de Perles :
(a) centrée acteur orientée services ; (b) sociale orientée données.

Perle centrée acteur et orientée services. La Figure 6a et 6b illustre la mise à disposition de services (Perle de services, au centre) en fonction de son activité du moment, c'est-à-dire la tâche qu'il doit accomplir. Dans cet exemple, le Dr. Roger a pour tâche de produire un compte-rendu médical suite à un examen : plusieurs services sont à disposition comme un outil d'édition de comptes-rendus structurés ainsi que des outils de communication pour demande d'expertise à des confrères.

Perle sociale centrée données. Cette perle représente le réseau socio-professionnel du Dr. Roger pour le suivi de patients. La Figure 6b représente les différents acteurs en charge du suivi d'un patient. On peut trouver tous les acteurs du monde médical regroupés en communautés (par exemple, la communauté des médecins, des étudiants en médecine, etc.) qui représentent les différents groupes du réseau socio-professionnel du Dr. Roger (Perles sociales de couleurs différentes).

Comme le montre la Figure 7b, ce type de Perle a fait l'objet d'une implémentation dans un prototype : une visualisation interactive sous forme de

graphe représente le réseau socio-professionnel du médecin complété par l'ensemble des dossiers médicaux gérés par celui-ci. Notamment, y sont représentés les relations directes et indirectes entre les différents acteurs (lignes grises), ainsi que les diverses communautés liées au Dr Roger (médecins, étudiants, etc.).

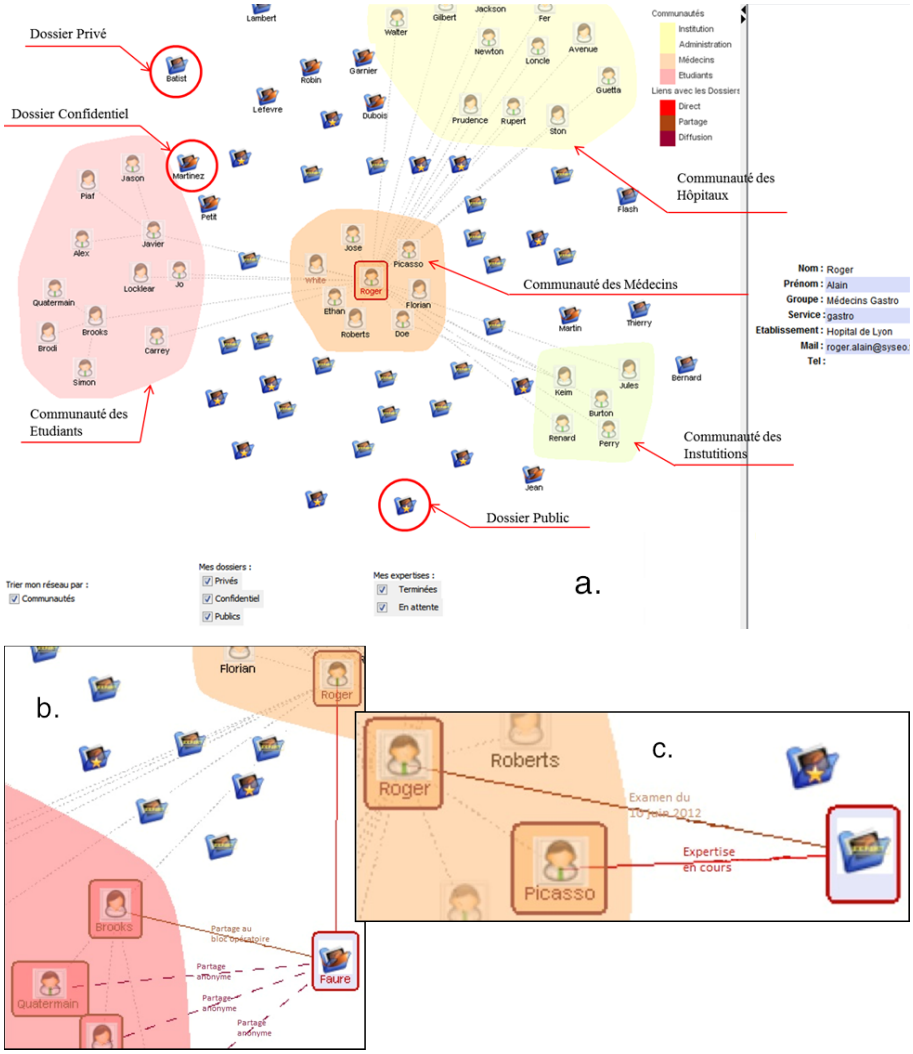


Figure 7. Perles sociales orientées données : copies d'écran du prototype.

La distance entre les icônes indique la proximité sociale entre les différents acteurs. Par exemple, la communauté des médecins (communauté orange) est la communauté qui a la plus grande proximité avec le Dr Roger. Comme le montre la Figure 7c, les liens entre les dossiers et les acteurs sont décorés par une étiquette

pour indiquer le statut de l'activité médicale en cours. Trois catégories sont considérées : (1) tâche la plus récente ou prochaine tâche à accomplir (par exemple, le dernier ou prochain examen) ; (2) nature du partage d'une donnée médicale ; et (3) traçabilité.

Perle centrée données et orientée services. Dans le contexte de la gastroentérologie, les données sont en grande partie les images médicales. Ainsi ce type perle (non représenté) peut être associée à une collection d'images médicales endoscopiques à laquelle sont associés des services dédiés au traitement et à la manipulation de l'image médicale comme des outils d'annotation, de comparaison et de recherche de pathologie, de mesure de polype, etc.

Perle de services. Dans l'exemple illustré par la Figure 6a et 6b, nous pouvons observer la présence de Perles homogènes (Perle au centre) qui regroupent différents services pour un contexte donné : outils de partage, visualisation collaborative, etc.

Perle de données. Un dossier médical peut être considéré comme étant une perle homogène élémentaire puisqu'elle est une collection d'informations médicales de nature différentes : images endoscopiques, comptes-rendus médicaux, ordonnances, résultats d'examen biologiques, etc. Comme le montre la Figure 6a et 6b et Figure 7, ce type de perle est symbolisé par une icône en forme de dossier. Le statut social des dossiers médicaux, visible sur l'icône, peut-être : (1) **privé** – seul le médecin référent et le patient y ont accès ; (2) **confidentiel** – le dossier privé est partagé avec le médecin traitant par exemple, ou pour une demande d'expertise; (3) **public** – le dossier est anonyme et publié pour un usage pédagogique. Il est possible d'y inclure la représentation des liens d'ordre social entre les différents dossiers, comme les liens entre dossiers pour des patients appartenant à une même famille.

La traçabilité est mise en œuvre au travers de la visualisation en indiquant la nature du partage sur un lien, direct ou indirect. Par exemple, le partage d'un dossier médical privé avec un confrère, un étudiant en stage ou l'administration, est direct. Inversement, la dissémination d'un dossier médical rendu anonyme (par exemple, entre deux étudiants) est un partage indirect. Ce type de partage est représenté par des lignes en pointillé. Par exemple, la Figure 7b montre que le dossier de Mr. Faure est partagé directement entre le Dr Roger et l'étudiante Brooks (ligne grise entre ces deux acteurs). Toutefois, ce dossier est partagé indirectement, donc anonyme, entre le Dr. Roger et trois autres étudiants via l'étudiante Brooks : des lignes hachurées violettes sont représentées entre le dossier médical de Mr. Faure et ces trois étudiants (par exemple, Simon). Cela indique ces étudiants ont un accès partiel à des données médicales non confidentielles car rendues anonymes. Une telle visualisation offre un moyen puissant pour représenter le réseau de diffusion. Ce point est crucial, comme l'ont confirmé les médecins partenaires, dans le processus de soin.

A l'aide de widgets dédiés (cases à cocher) situés dans la partie inférieure de l'interface, il est possible d'appliquer des filtres sur les données. Enfin, lorsque l'on sélectionne une entité (un acteur ou un dossier médical), des détails sont affichés dans le bandeau vertical à droite de l'IHM (Figure 7a). Ces fonctionnalités sont mises à disposition pour faciliter la navigation dans les données médicales couplées au réseau social du médecin.

5. Travaux existants

5.1 Approche informatique sociale

Avant l'avènement du *Cloud computing* et du *Social computing*, alors que le courrier électronique, la messagerie vocale et la messagerie instantanée étaient déjà considérés comme des outils de communication favorisant l'évolution des réseaux sociaux, Whittaker *et al.* (2001) soulignait les limites du bureau actuel face aux interfaces sociales. ContactMap est une application logicielle sociale principalement conçue pour la communication. Le bureau social est l'élément central de ContactMap: similaire aux espaces de travail partagés, il permet de structurer et de représenter visuellement l'information sociale sous forme de groupes de contacts. ContactMap est un outil alternatif basé sur la métaphore du bureau actuel, ne traitant que des activités de communication, sans considérer la dimension sociale des données.

Le bureau social, décrit par Pham (2010), revisite la métaphore du bureau en intégrant le *Cloud Computing* et le *Social Computing*, à l'aide du concept de « *user-created groups* ». Ce concept étend la métaphore du dossier pour y ajouter les fichiers, les humains et les applications. Ainsi, une instance unique d'un objet peut être inclue dans différents groupes. Ce bureau social facilite alors le partage et l'accès aux données : l'accès est implicitement accordé aux différents membres d'un même groupe, permettant le partage de fichiers. Ce travail est très proche de notre approche. Cependant, la dimension sociale n'est envisagée qu'au travers du concept de « *user-created groups* ». Semblables aux « *user-created groups* » de Pham, notre proposition est d'élargir la métaphore du dossier en intégrant la valeur sociale des données. Alors que la proposition de Pham réside en une représentation uniquement centrée sur les relations entre acteurs, notre approche, elle, fusionne deux points de vue : les données et les acteurs. De plus, la vue d'une donnée peut être restreinte ou élargie en fonction de deux critères : le statut social des acteurs et des données, et les relations sociales entre acteurs. Ainsi, alors que l'accès aux dossiers médicaux est limité pour les étudiants, un médecin référent peut avoir un accès complet à un dossier médical géré par un gastro-entérologue.

Selon Kairam *et al.* (2010), les « *réseaux sociaux en ligne sont devenus des outils indispensables pour l'échange d'informations* ». Il est pourtant encore difficile pour les utilisateurs de cibler des parties spécifiques de leur réseau. Les cercles de Google+ sont semblables au « *user-created group* » de Pham (2010) : le réseau social de l'utilisateur est structuré en cercles permettant le partage d'informations sélectives avec des parties de son réseau social. Cependant, l'utilisateur n'est pas conscient de la diffusion de ces informations transmises à ses cercles. Comme les cercles de Google+, les Perles sociales permettent un partage plus facile et plus rapide. Le concept de la perle est poussé plus loin avec cette représentation en graphe, permettant à l'utilisateur de prendre conscience de ces échanges passés et à venir, et lui donnant un sentiment de confiance.

Un autre travail est axé sur le partage social de fichiers au sein d'une entreprise (Shami *et al.*, 2010), prenant en compte leur valeur sociale. Des métadonnées sociales sont ajoutées aux fichiers en tant qu'attributs supplémentaires, permettant une organisation différente et non-hiérarchique. En effet, la métaphore courante des dossiers et leur hiérarchisation ne sont plus adaptées pour les systèmes de partage social (Odom *et al.*, 2012). Les métadonnées vont alors faciliter la navigation (c'est-à-dire, le paramétrage de la navigation basé sur les métadonnées). Pour ce faire, les auteurs ont développé un nouveau système de partage de fichiers, Cattail. Le système est capable de révéler l'activité sociale autour des fichiers. Trois degrés de partage régulent l'accès : privés, confidentiels et publics. Ce travail est également très proche de notre approche. Toutefois, la valeur sociale des données est uniquement considérée à travers les activités extérieures (c'est-à-dire, propriétés sociales des données). En outre, le système est une vision centrée sur le fichier associée à une visualisation de ces fichiers centrée sur le temps.

Plusieurs travaux ont exploré une telle approche basée sur les métadonnées afin de promouvoir une organisation différente et plus efficace des fichiers. En particulier, Dourish (2003) a introduit la notion de « *placeless document* », paradigme basé sur les propriétés du document comprenant des propriétés externes (date de création, etc.) et des propriétés internes (par exemple, une photo de mon fils et moi-même).

Comme le suggère Shami *et al.* (2010), notre proposition est d'enrichir les données classiques avec la dimension sociale pour faciliter la navigation, comme le propose Dourish (2003). Les entités VSA permettent de paramétrer la navigation. Aussi, comme le propose Muralidharan *et al.* (2012), la valeur sociale des entités est envisagée pour rendre observable l'activité médicale autour d'un dossier (par exemple le rendez-vous du patient avec un spécialiste). Une évolution possible de l'IHM serait de décorer les relations afin d'indiquer le statut des activités (dernier échange, dernier rendez-vous, etc.).

5.2 Approche à service

Le modèle centré application de la métaphore usuelle du bureau est amené à disparaître progressivement : de plus en plus, les applications et services sont déployés au sein du *cloud* et rendues accessibles sous la forme d'applications web comme le suite Office de Microsoft ou les outils d'Adobe. Toutefois, ce mouvement vers des environnements 100% web brise la garantie de disposer d'interfaces utilisateur consistantes et ne peut que mener vers une expérience utilisateur fragmentée (England *et al.*, 2011). Une possible explication de cette fêlure est que la métaphore du bureau reflète fortement la nature locale des ressources, fermement ancrée dans la notion de localisation physique des ressources et de leur organisation hiérarchique. Cette métaphore a peu évolué pour faciliter et permettre le partage et le contrôle d'accès.

Dans le domaine du CSCW, Voida *et al.* (2008) proposent de faire évoluer le bureau actuel vers un bureau centré sur les activités, regroupant les activités

collaboratives, et ajoutant une dimension sociale. Sur l'interface Giornata, une palette de contacts est mise à la disposition de l'utilisateur, lui permettant de gérer ses contacts (individus ou groupes), facilitant ainsi le partage de fichiers.

Des travaux récents étudient l'usage de services reposant sur le *cloud* pour la santé (Van der Burg *et al.*, 2009), se concentrant sur la disponibilité des services. Les auteurs observent que le monde médical bascule d'une infrastructure orientée périphériques vers un environnement orienté services, ce qui signifie que chaque outil logiciel doit être rendu disponible pour tout dispositif d'interaction. Par exemple, un outil de visualisation d'images 3D tirées des scanners (une application très gourmande en ressources CPU) peut être rendue disponible sur PDA, que le praticien soit en consultation à l'hôpital, ou sur son PC, à son cabinet médical (Van der Burg *et al.*, 2009). Koufi *et al.* (2010) considère que les services reposant sur le *cloud* sont de plus en plus importants dans le domaine de la santé, en particulier pour les urgences, du fait de leur caractère ubiquitaire. Les auteurs définissent le *Cloud computing* comme un modèle de services à la demande qui peut faciliter le partage de l'information et réduire le besoin de stockage en local. Cette évolution pose toutefois de nombreuses questions d'ingénierie logicielle ainsi que des questions en termes d'IHM notamment du point de vue de plasticité des interfaces et de l'expérience utilisateur.

5.3 Approche centrée données

Les différents entretiens avec les professionnels de santé ont permis de mettre en évidence les limites des ces métaphores centrées données : l'information est fragmentée et distribuée dans différents services, le service de messagerie pour les échanges, le système d'information de chaque établissement de santé pour la gestion des dossiers médicaux, etc. Notre instanciation des Perles au travers du premier démonstrateur semble répondre (en partie) aux attentes et besoins des professionnels de santé.

Terrenghi *et al.* (2010) propose une nouvelle métaphore du bureau pour le *Cloud*, CloudRoom, et se concentre sur les questions de stockage et de recherche de données. CloudRoom se présente en trois plans distincts pour l'organisation des données : (1) un système de stockage persistant et à long terme ; (2) un aperçu chronologique ; et (3) un stockage temporaire pour les activités en cours. Mise à part le partage entre utilisateurs (de sessions et de données), la valeur sociale des données n'est pas prise en considération. Par rapport à CloudRoom qui s'appuie sur des vues independantes pour l'accès aux données, notre métaphore révèle le contexte de données telles que les orbites sociales de Sawyer *et al.* (2012) (c'est-à-dire l'emplacement physique lié à des données telles que la salle d'opération ou un cabinet médical).

L'une des évolutions pour notre prototype serait d'ajouter cette notion temporelle de la donnée, part importante du contexte de la donnée. Pour cela, une révision de l'interaction est nécessaire afin de se conformer au mantra de Schneiderman : «*Overview, zoom and filter, details-on-demand*» : « *Nous devrions*

voir les données, les clusters, les relations et les absences de données. [...] L'utilisateur zoome sur ce qu'il veut voir, filtre ce qu'il ne veut pas et clique sur l'élément pour lequel il veut de l'information » (Schneiderman, 2010). Une première idée serait de jumeler cette métaphore centrée utilisateur avec une visualisation temporelle (ligne de temps). Cela permettrait à l'utilisateur de ne garder que ces dossiers du jour (20 à 25 consultations par jour).

6. Conclusion et perspectives

Ce travail traite de la conception d'infrastructures de santé d'un point de vue utilisateur alors qu'elles sont souvent conçues d'un point de vue organisationnel ou technique : le point de vue utilisateur, c'est-à-dire le personnel de santé, est souvent oublié. Afin d'y remédier, dans le contexte de la gastroentérologie, en partenariat avec des médecins gastroentérologues et des experts dans les technologie de l'information en santé, nous avons adopté une processus participatif pour conduire une approche montante (bottom-up) afin d'identifier des requis pour la conception de telles infrastructures. Précisément, nous traitons du manque de systèmes interactifs qui offriraient des services interactifs permettant d'améliorer les pratiques en gastroentérologie. Du point de vue de l'IHM, nous traitons ce problème en proposant une approche pour la conception d'interfaces utilisateur orientées services et appliquées à la gastroentérologie.

A partir d'une approche en trois volets qui traite à la fois des aspects médicaux et des aspects relatifs à l'interaction utilisateur, nous proposons une nouvelle métaphore d'interface utilisateur orientée service, la métaphore de la Perle. Cette métaphore repose sur le conception d'entité à Valeur Sociale Ajoutée afin d'intégrer la dimension sociale des activités. Elle repose également sur une définition formelle accompagnée de propositions de représentations conceptuelles. Nous avons appliqué cette métaphore à la gastroentérologie avec le développement d'un prototype comme preuve de concept. L'approche service constitue une opportunité intéressante pour proposer des interfaces utilisateurs sensibles au contexte d'usage en fonction des ressources interactionnelles disponibles et des tâches à accomplir. En particulier, pour la gastroentérologie, cela nous permet de proposer une application interactive qui intègre le réseau socio-professionnel des médecins.

Les premiers retours des médecins montrent que la métaphore de la Perle constitue un moyen intéressant pour traiter du partage des données médicales, de la traçabilité de l'information, de la capitalisation de l'image endoscopique et donc de contribuer à l'amélioration de la qualité des soins. En perspective, nous prévoyons de développer et d'évaluer un prototype complet ainsi que d'étudier l'adaptation de l'IHM pour différents contextes d'usage (par exemple, au bloc opératoire et au cabinet médical avec des dispositifs d'interaction différents).

Remerciements

Les recherches sont financées par le projet ANR Syséo (Technologies pour la Santé) 2010-2014. Nous remercions, en particulier, Laurent D'Orazio et Serge Torti, ainsi que les médecins A. Baetz, S. Béorchia, A. D'Orazio et T. Ponchon.

Bibliographie

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., (2010). A view of cloud computing. *Communication of the ACM*, vol. 53, nº 4, p. 50-58.

Carlin, S., Curran, K., (2012). Cloud Computing Technologies. *Computing and Services Science*, vol. 1, nº 2, p. 59-65.

Dourish, P., (2003). The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents. *Computer Supported Cooperative Work*, vol. 12, nº 4, p. 465-490.

England, D., Randles, M., Taleb-Bendiab, A., (2011). Designing Interaction for the Cloud. *Annexe des actes de la conférence ACM CHI 2011*, ACM Press, New-York, p. 2453–2456.

Eysenbach, G., (2008). Medicine 2.0: Social Networking, Collaboration, Participation, Apomediation, and Openness. *Journal of Medical Internet Research*, vol. 10, nº 3, p. 10.

Freeman, L. C., (1979). Centrality in social networks: Conceptual clarification. *Social Networks*, vol. 1, p. 215-239.

Keckley, P. H., Hoffman, M., (2010). *Social Networks in Health Care: Communication, Collaboration and Insights*. Deloitte. Référencé par le gouvernement Australien.

Kairam S., Brzozowski, M. J., Huffaker, D., Chi, E.H., (2010). Talking in Circles: Selective Sharing in Google+. *Actes de la conférence internationale ACM CHI 2012*, ACM Press, New-York, p.1065-1074.

Koufi, V, Malamateniou, F, Vassilacopoulos, G., (2010). Ubiquitous access to cloud emergency medical services. *Information Technology and Applications in Biomedicine*, p. 1-4.

Kuo, M.-H., (2011). Opportunities and Challenges of Cloud Computing to Improve Health Care Services. *Journal of Medical Internet Research*, vol. 13, nº 3, p. e67.

Lahire, B., (2010). *The Plural Actor*, Polity, Cambridge.

Many Avila-Garcia, M. S., Trefethen, A. E., Brady, M., Gleeson, F., Goodman, D., (2008). Lowering the Barriers to Cancer Imaging. *Actes de la conférence internationale IEEE eScience 2008*, IEEE Press, p. 63–70.

Mell, P., Grance, T., (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, vol. 53, nº 6, p. 50.

Morán, A. L., Rodríguez-Covili, J., Mejia, D., Favela, J., Ochoa, S., (2010), Supporting Informal Interaction in a Hospital through Impromptu Social Networking, Collaboration and Technology. *Actes de la conférence internationale CRIWG 2010*, LNCS 6257 Springer, p. 305-320.

Muralidharan, A., Gyongyi, Z., Chi, E.H., (2012). Social Annotations in Web Search. *Actes de la conférence internationale CHI 2012*, ACM Press, New-York, p. 1085-1094.

Nkosi, M.T., Mekuria, F., (2010). Cloud Computing for Enhanced Mobile Health Applications. *Actes de la conférence internationale IEEE CloudCom 2010*, IEEE, p. 629-633.

Odom, W., Sellen, A., Harper, R., Thereska, E., (2012). Lost in Translation: Understanding the Possession of Digital Things in the Cloud. *Actes de la conférence internationale CHI 2012*, ACM Press, New-York, p. 781-790.

Pham, H., (2010). User Interface Models for the Cloud. *Actes de la conférence internationale UIST 2010*, ACM Press, New-York, p. 359-362.

Rolim, C.O., Koch, F.L., Westphall, C.B., Werner, J., Fracalossi, A., Salvador, G.S., (2010). A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions. *Actes de la conférence internationale IEEE ETELEMED 2010*, IEEE, p. 95-99.

Rosson, M.B., Carroll, J.M., (2002). *Usability engineering: Scenario-Based Development of Human-Computer Interaction*, Morgan-Kaufman.

Schneiderman, B., (2010). *Information Visualization for Knowledge Discovery*.

Shami, N.S., Muller, M., Millen, D., (2010). Browse and discover: social file sharing in the enterprise. *Actes de la conférence internationale CSCW 2010*, ACM Press, p. 295-304.

Terrenghi, L., Serralheiro, K., Lang, T., Richartz, M., (2010). Cloudroom: a Conceptual Model for Managing Data in Space and Time. *Annexe des actes de la conférence internationale CHI 2010,* ACM Press, New-York, p. 3277-3282.

Van der Burg, S., de Jonge, M., Dolstra, E., Visser, E., (2009). Software deployment in a dynamic cloud: From device to service orientation in a hospital environment. *Actes de la conférence internationale ICSE 2009,* p. 61-66.

Van Welie, M., Van der Veer, G.C., Eliëns, A., (1998). An ontology for task world models. *Actes de la conférence internationale DSV-IS 1998*, Springer, p. 57-70.

Voida, S., Mynatt, E.D., Edwards, W.K., (2008). Re-framing the desktop interface around the activities of knowledge work. *Actes de la conférence internationale UIST 2008*, ACM Press, New-York, p. 211-220.

Whittaker, S., Jones, Q., Nardi, B., Creech, M., Terveen, L., Isaacs, E., Hainsworth, J., (2001). ContactMap: Organizing Communication in a Social Desktop. *Transactions on Computer-Human Interaction*, vol. 11, nº 4, p. 445-471.

Zhang, C., Wang, .M, Harper, R., (2010). Cloud mouse: a new way to interact with the cloud. *Actes de la conférence internationale ICMI-MLMI 2010*, ACM Press, New-York, p. 13.

Zhang, Q., Cheng, L., Boutaba, R., (2010). Cloud computing: state-of-the-art and research challenges. *Internet Services and Applications*, vol. 1, nº 1, p. 7-18.

Zijlstra, G. J., (1979). Networks in public policy: Nuclear energy in the Netherlands. *Social Networks*, vol. 1, nº 4, p. 359-389.

A Feasibility Study on the Use of Anthropometric Variables to Make Muscle-Computer Interface More Practical

Angkoon Phinyomark[a,b], Franck Quaine[a], Sylvie Charbonnier[a], Christine Serviere[a], Franck Tarpin-Bernard[b], Yann Laurillau[b]

angkoon.p@hotmail.com, franck.quaine@gipsa-lab.grenoble-inp.fr, sylvie.charbonnier@gipsa-lab.grenoble-inp.fr, christine.serviere@gipsa-lab.grenoble-inp.fr, franck.tarpin-bernard@ujf-grenoble.fr, yann.laurillau@imag.fr

[a] GIPSA Laboratory, CNRS UMR 5216, Control system Department, SAIGA team, University Joseph Fourier, Grenoble, France
[b] LIG Laboratory, CNRS UMR 5217, University of Grenoble, Grenoble, France

Abstract

High classification accuracy has been achieved for muscle-computer interfaces (MCIs) based on surface electromyography (EMG) recognition in many recent works with an increasing number of discriminated movements. However, there are many limitations to use these interfaces in the real-world contexts. One of the major problems is compatibility. Designing and training the classification EMG system for a particular individual user is needed in order to reach high accuracy. If the system can calibrate itself automatically/semi-automatically, the development of standard interfaces that are compatible with almost any user could be possible. Twelve anthropometric variables, a measurement of body dimensions, have been proposed and used to calibrate the system in two different ways: a weighting factor for a classifier and a normalizing value for EMG features. The experimental results showed that a number of relationships between anthropometric variables and EMG time-domain features from upper-limb muscles and movements are statistically strong (average r = 0.71-0.80) and significant ($p < 0.05$). In this paper, the feasibility to use anthropometric variables to calibrate the EMG classification system is shown obviously and the proposed calibration technique is suggested to further improve the robustness and practical use of MCIs based on EMG pattern recognition.

Keywords

Circumference; Electromyography (EMG); Feature extraction; Gesture recognition; Hand motion; Muscle size.

Corresponding author

Angkoon Phinyomark
Portal address: GIPSA-lab, Grenoble Campus, 961, rue de la Houille Blanche BP46, F-38402 Saint Martin d'Hères Cedex, France
E-mail address: angkoon.p@hotmail.com and angkoon.phinyomark@gipsa-lab.grenoble-inp.fr
Telephone number: +33 (0)6 15 92 73 80
Fax number: +33 (0)4 76 82 63 88

1. Introduction

Muscle-computer interfaces (MCIs) based on surface electromyography (EMG) recognition have been rapidly developed for various applications, i.e. prosthesis and electric-power wheelchair, in the last few years (Ahsan et al., 2010; Oskoei and Hu, 2007; Peerdeman et al., 2011; Phinyomark et al., 2011a). Nearly all previous works on EMG-based MCIs focus on classification accuracy improvement and the number of discriminated movements (Hariharan et al., 2012; Ju et al., 2011; Oskoei and Hu, 2007; Phinyomark et al., 2011a; Wojtczak et al., 2009). The success rate of these interfaces is usually higher than 80-90% based on recognizing several common tasks such as the movements of flexion/extension and abduction/adduction (Peerdeman et al., 2011). In laboratories, the number of discriminated tasks, i.e. grasping and wrist motions, has increased to cover most activities of daily living (ADLs) and to control a multiple DOF prosthesis (Cipriani et al., 2008).

While this interface holds so much potential, a few works have given attention to the context of real-world requirements such as long-term use or EMG's uncertainties, i.e. EMG electrode location shift (Tkach et al., 2010) and variation in muscle contraction between days (Phinyomark et al., 2012a), compatibility, i.e. minimal or no need for calibration and training between subjects (Saponas et al., 2010), and robustness, i.e. noise (Phinyomark et al, 2009). Among these requirements, the development of calibration and training issues is still far from being a practical one (Cannan and Hu, 2011; Saponas et al., 2010). In this study, we aimed to address these challenges with special emphasis on EMG and anthropometric techniques that can automatically or semi-automatically calibrate a system.

Due to different body compositions between users, EMG-based MCIs has never really reached the general population (Cannan and Hu, 2011). Every user has different types of muscles with varying sizes and other characteristics, so currently the systems still require new calibration and training (between users and also between days) to compensate for discrepancies. Moreover, this limitation prevents the development of standard interfaces that are compatible with almost any user.

In order to reduce the need for new training and calibration, some useful information should be added as automatic training input parameters. For a simple system that uses thresholding techniques, maximum voluntary contraction (MVC) alone is used and may be enough to adapt a system from one user to another. Based on the finding in several related works about a relationship between features of EMG and force, estimating the users MVC can be used to normalize the EMG signals or adjust the threshold values (Bolgla and Uhl, 2007; Vera-Garcia et al., 2010).

However, this relationship is strong only in an isometric or static contraction (Kamavuako et al., 2009), which can be linear or non-linear. When the muscle is free to change length and the joint is free to move as a dynamic muscle contraction, the relationship of EMG features and forces is more complicated (Oatis, 2008). Although force, or MVC, certainly plays a role, it is not significant enough to be used alone in adapting advanced EMG systems, and other additional useful information is necessary.

We know that muscle size, consisting of cross-sectional area (CSA) and length, can be employed together with EMG signal in order to determine muscle force (Hof et al., 1987). This relationship can be explained by the equation of Marras and Sommerich (1991) as shown

in Appendix A. It means that there are correlations between muscle size, muscle force, and EMG signal (Raez et al., 2006; Ray and Guha, 1983). Hence, anthropometric variables, a measurable characteristic of the body, are considered. These variables can roughly estimate some muscle-size characteristics such as estimating thigh muscle cross-sectional area by segment circumference (Housh et al., 1995). On the other hand, measuring anthropometric variables is easier and some variables can be measured directly together with EMG signal via armband (Cannan and Hu, 2011; Saponas et al., 2009).

One of the related anthropometric variables is forearm circumference, which can be measured automatically via a wearable device. Cannan and Hu (2011) used forearm circumference for estimating MVC in order to calibrate the EMG thresholding technique based on the linear relationship between grip strength and forearm circumference as presented in Anakwe et al. (2007). However, the relationship between EMG during MVC and forearm circumference is not strong as found with grip force, and further anthropometric variables are recommended to be incorporated to make a reliable adaptive system (Cannan and Hu, 2011).

In this paper, the relationship between common-used EMG time-domain features and related anthropometric variables is investigated. As mentioned, features of the EMG signal were used to find the correlation instead of EMG associated with 100% MVC because we would like to move from calibrating the simple thresholding techniques to machine learning and pattern recognition. It should be noted that feature extraction is used as an input vector for classifier to make a decision output (Phinyomark et al., 2012b). So actions associated with EMG signal, e.g. forearm pronation/supination and hand open/close (dynamic contraction), are also used instead of EMG isometric (static) contraction.

Every strong and significant association between EMG feature and anthropometric variable, found in this study, could benefit a further design of EMG systems. It could automatically adapt the setting to a wider population. Moreover, due to a rapid increased number of wearable devices, anthropometric variables would become more practical and important in the near future.

2. Material and methods

The EMG data, which were used to investigate the relationship between anthropometric variables and EMG features in this study, were recorded from eight movements, five positions and twenty subjects during four days.

Eight movements consisting of forearm pronation (FP), forearm supination (FS), wrist extension (WE), wrist flexion (WF), wrist radial deviation (WR), wrist ulnar deviation (WU), hand open (HO), and hand close (HC) were chosen based on the frequently used in MCIs studies (Oskoei and Hu, 2007; Peerdeman et al., 2011; Phinyomark et al., 2011a), as shown in Fig. 1. These are movements of hand, wrist and arm. In addition, the shoulder was positioned at 0 degree (neutral) with elbow in full extension.

EMG recordings from five muscles were selected from both upper-arm and forearm and from both flexor and extensor muscles. The amplitude shape of EMG signals acquired from all muscles was significantly different according to the direction of eight movements as shown in Fig. 2 (Phinyomark et al., 2011b). There were: extensor carpi radialis longus

(ECRL), extensor carpi ulnaris (ECU), extensor digitorum communis (EDC), flexor carpi radialis (FCR) and biceps brachii (BB), as shown in Fig. 3.

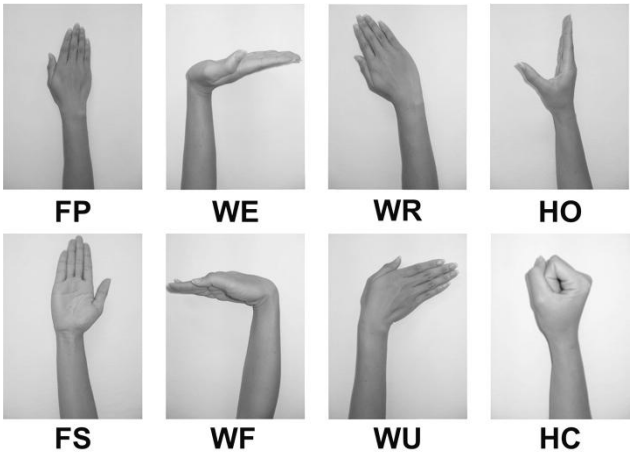


Fig. 1. Eight upper-limb movements.

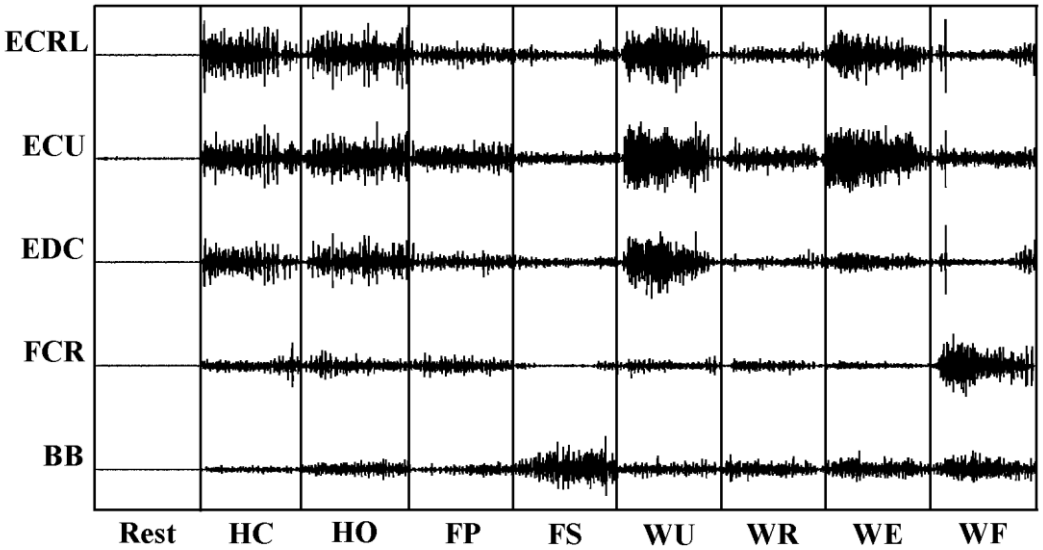


Fig. 2. The example amplitude shape of EMG signals acquired from 5 muscles and 8 movements with rest state.

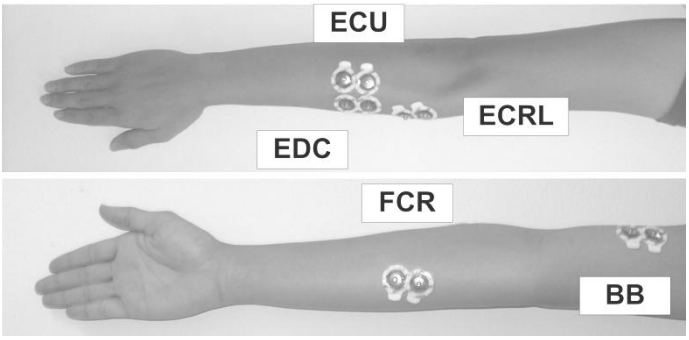


Fig. 3. Five electrode positions.

Twenty healthy subjects participated in the experiment consisting of 10 males (M1-M10) and 10 females (F1-F10). The age of the male subjects was 21.5±0.97 years and of the female subjects was 21.2±0.79 years. All subjects were dexterous with their right hands. They signed an informed consent in accordance to the University guideline. The EMG data were recorded on 4 separate days. Each day the subjects were asked to perform 15 sessions, which every movement was maintained for 2 s and the order of movements was randomized in each session. Based on a large fluctuating EMG data that was measured in this study, the effect of fluctuating EMG signals during different days was also considered.

In the experiment, the EMG data were collected from the positions on the right arm using bipolar Ag/AgCl electrodes (H124SG, Kendal ARBO) with 24 mm diameter and 20 mm inter-electrode distance. All EMG signals were amplified with a gain of 19.5x and sampled at 1024 Hz with a 24-bit resolution by a commercial wireless EMG measurement system (Mobi6-6b, TMS International B.V.). The EMG data were also passed through a band-pass filter with a cutoff frequency of 20 and 500 Hz to remove noise and unwanted signals.

2.1 Extraction of EMG features

As mentioned in the experiment above, 60 data sets with 2-s in duration in total were collected for each movement for each subject. In order to contain all EMG information, a window size of the extraction was set as a whole length of action (2-s) (Hudgins et al., 1993; Wojtczak et al., 2009). Five common used features based on Hudgins' time-domain approach (Hudgins et al., 1993) were used as the representative features in this paper: mean absolute value (MAV), mean absolute value slope (MAVS), waveform length (WL), zero crossing (ZC), and slope sign change (SSC). A set of Hudgins' time-domain features has usually been employed as a baseline feature set for comparing with a newly developed feature set, and the success of this feature set has been established in many recent studies, e.g. Li et al. (2011, 2010).

(1) MAV is commonly used as an EMG amplitude detector. It is defined as an average of the absolute value of the EMG signal amplitude in a window size L, which can be expressed as

$$MAV = \frac{1}{L}\sum_{i=1}^{L}|x_i|, \tag{1}$$

where x_i is an EMG signal amplitude at the ith sample and L is the length of the window, which is set at 2048 samples (2-s) for all features.

(2) MAVS is defined as a difference between two mean absolute values of the signal in adjacent segments, k and k+1, where the window size L is divided into sub-windows or segments for k = 1, …, K-1. The definition is given by

$$MAVS_k = MAV_{k+1} - MAV_k . \tag{2}$$

It should be noted that MAV, WL, ZC and SSC provide only one feature value per window, whereas MAVS provides *K*-1 feature values per window. To provide a fair comparison between methods, the number of feature values for each window/movement trial was set at one for all methods. Hence, the value of *K* was set at 2 in this study. In other words, only $MAVS_1$ value was yielded and so, in the rest of this paper, MAVS refers to $MAVS_1$.

(3) WL is a simple complexity measure of the EMG signal. It is defined as a cumulative length of the waveform over the time window, which can be expressed as

$$WL = \sum_{i=1}^{L-1} \left| x_{i+1} - x_i \right| . \tag{3}$$

(4) ZC is a simple frequency measure of the EMG signal. It can be obtained by counting a number of times that the EMG waveform crosses zero amplitude level. To reduce the effect of background noise, a predefined threshold *thr* was set at 10 for ZC, and also SSC. Mathematically, it is calculated as

$$ZC = \sum_{i=1}^{L-1} \left[\operatorname{sgn}\left(x_i \times x_{i+1} \right) \cap \left| x_i - x_{i+1} \right| \geq thr \right]; \quad \operatorname{sgn}(y) = \begin{cases} 1, & \text{if } y < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

(5) SSC is another measure of frequency content of the EMG signal. It is a number of times that the slope of the EMG waveform changes sign. The number of changes between the positive and negative slopes among three sequential samples is performed with the predefined threshold *thr* as used in ZC. SSC can be mathematically expressed as

$$SSC = \sum_{i=2}^{L-1} \left[f\left[\left(x_i - x_{i-1} \right) \times \left(x_i - x_{i+1} \right) \right] \right]; \quad f(y) = \begin{cases} 1, & \text{if } y \geq thr \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

As described in Phinyomark et al. (2012b), these five features in Hudgins' group share the distribution in space, and using two or three features out of them in the classification could achieve the same performance as using all features. MAV and WL have a little difference in discriminating patterns where MAV contains energy information and WL contains complexity information. MAVS is an extension version of the MAV features. ZC and SSC, two features based on frequency information, have a similar distribution in space. Hence, it is not necessary to use all features in the final feature vector. The relationship with the anthropometric variables should be discussed for each individual feature. To be flexible in the future design of EMG-based MCIs, the correlation was also summarized for each muscle position.

While the number of electrode positions should be reduced to the minimum, the number of discriminated movements should be increased as much as possible. Four movements have been used like a standard minimal number of control commands in EMG-based MCIs (Peerdeman et al., 2011; Phinyomark et al., 2011a). Hence, only the combinations between anthropometric variable, EMG feature and EMG position that have the strong and significant correlations in at least four movements will be presented in this paper.

2.2 Anthropometric variables

Anthropometry is a measurement of the dimensions of the different parts of the body and other physical characteristics. There are two types of measurement: static and dynamic dimensions. In this paper, only static dimensions were considered. Twelve related anthropometric variables were chosen: 1. Body mass (kg), 2. Standing height (cm), 3. Body mass index or BMI (kg/m^2), 4. Biceps circumference (cm), 5. Forearm circumference (cm), 6. Hand breadth (cm), 7. Hand length (cm), 8. Elbow-hand grip length (cm), 9. Elbow-fingertip length (cm), 10. Shoulder-elbow length (cm), 11. Bi-deltoid breadth (cm), and 12. Forward grip reach (cm). All variables were measured from all the subjects in the same day and from the right arm.

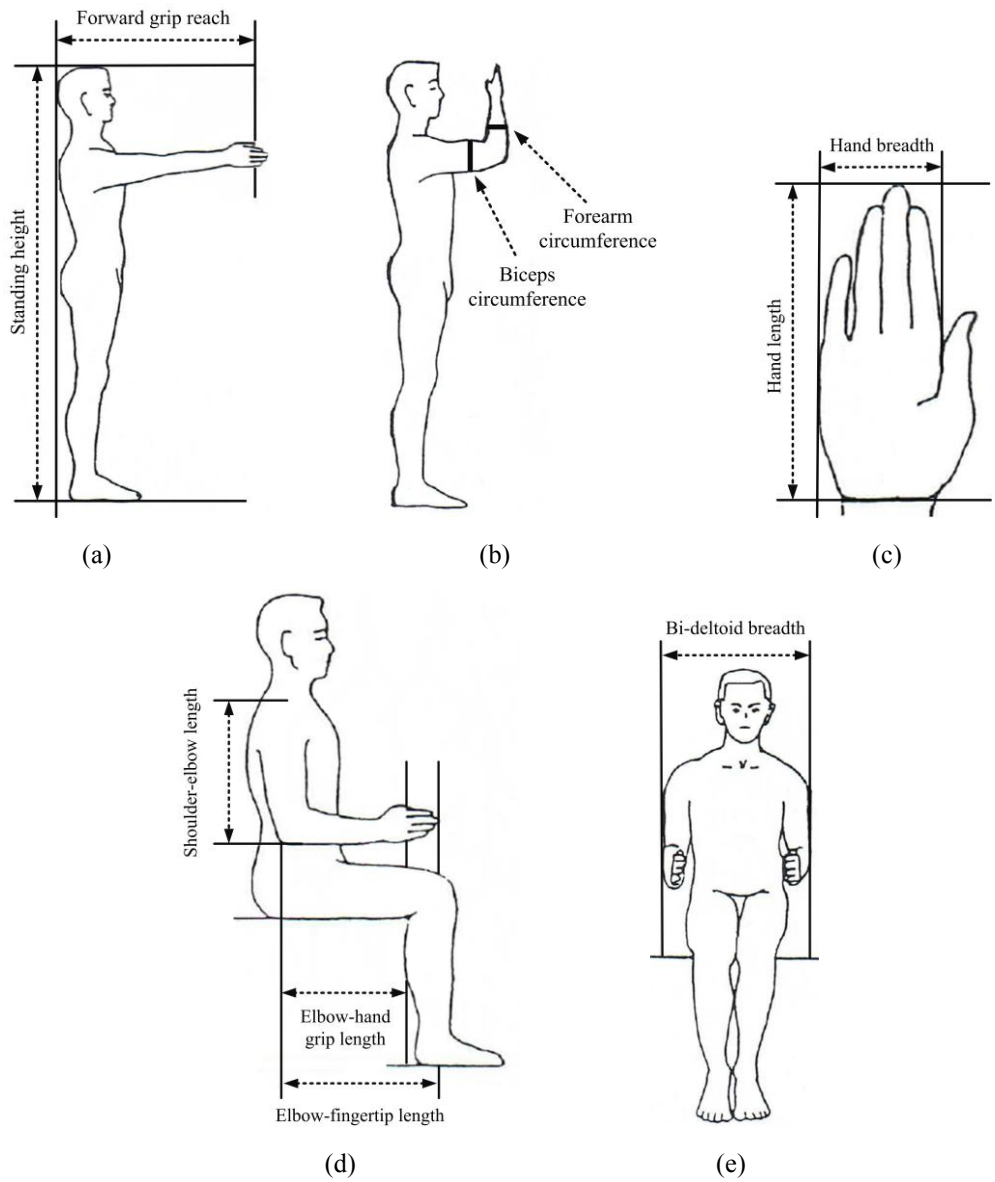


Fig. 4. Anthropometric (body) measurements (a) standing height and forward grip reach, (b) biceps circumference and forearm circumference, (c) hand breadth and hand length, (d)

elbow-hand grip length, elbow-fingertip length and shoulder-elbow length, (e) bi-deltoid breadth.

These twelve measurement techniques, as shown in Fig. 4, can be measured using different kinds of instruments (Centurion Kit, Rosscraft): (1) Balance type scales for body mass, (2) Anthropometer for standing height, (3) Tape for biceps and forearm circumferences, (4) Small bone caliper for hand breadth and hand length, (5) Wide sliding torso caliper for elbow-hand grip length, elbow-fingertip length, shoulder-elbow length and bi-deltoid breadth, and (6) Measuring block with tape measure for forward grip reach.

It should be noticed that the circumferences were measured with elbow flexed 90 degrees (the arms are abducted). For other variables, the specific locations were defined as:

(1) Hand breadth is measured between metacarpalphalangeal joint II and V.

(2) Hand length is measured from the wrist landmarks to dactylion.

(3) Elbow-hand grip length is measured from the posterior tip of the olecranon process to the center of grip during holding a pencil.

(4) Elbow-fingertip length is measured from the posterior tip of the olecranon process to dactylion.

(5) Shoulder-elbow length is measured from the right acromion landmark to the inferior tip of the olecranon process of the right elbow.

(6) Bi-deltoid breadth is measured across the body at the level of the deltoid landmarks.

(7) Forward grip reach is measured from the back wall to the tip of the thumb.

In addition, BMI, which is a roughly estimation of human body fat, is calculated as

$$BMI = \frac{body\ mass}{\left(standing\ height\right)^2}. \tag{6}$$

2.3 Evaluating functions

Correlation analysis measures a relationship or association and gives a statistic known as the correlation coefficient or *r* coefficient. This value shows the degree or strength of linear association between two measured variables. The interpretation of correlation coefficients can be defined as presented in Table 1. The *r* value contains both a magnitude and a direction (positive and negative) of the relationship. However, in this paper we reported the absolute value of average *r* or only the magnitude of correlation.

Table 1. The strength category of correlation coefficients *r* (Taylor, 1990).

Ranges of *r* in absolute value	Interpretation
$r \leq 0.35$	low or weak correlations
$0.35 < r \leq 0.67$	modest or moderate correlations
$0.67 < r \leq 1$	strong or high correlations

As we mentioned that only the combinations that have the strong correlations will be presented, it means that the value of *r* must be higher than 0.67. Moreover, the correlation coefficient has to be statistically significant too. In this study, the significant level was set at *p*

< 0.05. Due to a small samples ($n \leq 20$), t-test was employed to test the significance of a correlation coefficient, as can be defined by

$$t = r\sqrt{\frac{n-2}{1-r^2}} \,. \tag{7}$$

Note that the degrees of freedom for entering the t-distribution is n-2.

3. Results

The different results of the relationship of muscle size/force and anthropometric variable between male and female subjects are found in several studies (e.g. Anakwe et al., 2007; Holzbaur et al., 2007). Therefore, the relationships of anthropometric variables have usually been investigated and discussed for each gender: male and female. All anthropometric variables of male and female subjects, which were measured in the experiment, are reported respectively in Table 2 and Table 3.

Table 2. Anthropometric variables of ten male subjects (M1-M10) with the mean and the standard deviation (SD) of each variable.

Variables \ Subjects	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	Mean	SD
Body mass	49.0	62.0	65.0	74.0	73.0	58.0	57.0	55.0	63.0	54.0	61.0	8.1
Standing height	166.5	170.0	173.0	177.0	172.0	170.0	170.0	167.5	167.0	164.0	169.7	3.7
Body mass index	17.7	21.5	21.7	23.6	24.7	20.1	19.7	19.6	22.6	20.1	21.1	2.1
Biceps circumference	23.1	26.2	27.5	32.6	32.4	30.7	26.1	23.6	26.7	25.6	27.5	3.4
Forearm circumference	22.5	22.4	25.2	29.1	28.2	26.8	23.4	23.1	24.3	22.7	24.8	2.5
Hand breadth	7.4	7.2	8.4	9.1	8.9	12.6	11.7	7.5	7.8	7.6	8.8	1.9
Hand length	17.2	17.8	15.5	18.2	18.9	19.1	19.3	17.4	18.3	17.1	17.9	1.1
Elbow-hand grip length	34.4	35.1	37.6	39.3	39.7	33.9	33.8	36.0	38.5	38.4	36.7	2.3
Elbow-fingertip length	46.8	49.7	52.2	52.2	52.6	46.2	46.2	47.2	50.0	49.7	49.3	2.5
Shoulder-elbow length	37.7	33.6	36.9	37.8	38.2	35.7	38.4	35.4	38.3	36.6	36.9	1.6
Bi-deltoid breadth	39.3	46.4	44.6	54.7	47.3	37.4	30.4	42.1	45.9	47.4	43.6	6.7
Forward grip reach	75.6	74.5	80.7	89	80.4	74.3	75.7	72.6	83.5	78.1	78.4	5.0

Table 3. Anthropometric variables of ten female subjects (F1-F10) with the mean and the standard deviation (SD) of each variable.

Variables \ Subjects	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
Body mass	47.0	45.0	53.0	46.0	54.0	45.0	43.0	56.0	50.0	49.0	48.8	4.4
Standing height	150.0	160.0	156.0	155.0	160.0	146.0	159.0	163.0	167.0	162.0	157.8	6.3
Body mass index	20.9	17.6	21.8	19.1	21.1	21.1	17.0	21.1	17.9	18.7	19.6	1.8
Biceps circumference	24.4	19.8	27.6	24.2	24.8	25.5	22.6	24.9	22.3	21.5	23.8	2.2
Forearm circumference	21.6	19.1	22.9	21.5	22.2	21.9	19.7	22.7	20.6	20.9	21.3	1.2
Hand breadth	7.4	6.4	7.3	7.5	6.9	6.6	6.9	8.1	7.4	7.2	7.2	0.5
Hand length	15.6	16.7	16.8	16.6	16.4	14.7	16.1	15.2	17.1	17.2	16.2	0.8
Elbow-hand grip length	32.8	35.0	34.8	34.6	34.4	32.6	34.3	35.2	38.4	36.2	34.8	1.6
Elbow-fingertip length	43.3	46.8	46.4	46.4	46.2	39.1	46.2	47.1	49.0	47.7	45.8	2.8
Shoulder-elbow length	34.9	35.3	36.1	35.8	33.1	34.2	33.0	37.1	37.8	36.0	35.3	1.6
Bi-deltoid breadth	41.9	36.8	43.4	41.2	41	37.8	36.9	37.0	37.2	38.7	39.2	2.5
Forward grip reach	65.1	72.0	60.8	68.1	75.7	65.2	70.2	63.3	74.4	70.6	68.5	4.9

The differences between parameters of male and female subjects were highly significant at $p < 0.01$ for 8 parameters and significant at $p < 0.10$ for the remainders: BMI, elbow-hand grip length, shoulder-elbow length, and bi-deltoid breadth. However, the difference between ages of male and female subjects was not significant ($p = 0.458$) in this study (subjects of the same age).

The strong and significant correlations between EMG features and anthropometric variables for each muscle are presented in Table 4 for male subjects and Table 5 for female subjects. From Table 4, two features (MAV and WL) from different two muscles (ECU and BB) showed a strong relationship with bi-deltoid breadth for 4-5 movements. On the other hand, from Table 5, four features (MAVS, WL, ZC, and SSC) from two muscles (FCR and BB) have high correlations with a number of anthropometric variables i.e. hand length, biceps circumference, and bi-deltoid breadth. The correlation coefficients r ranged between 0.69 and 0.87. It should be emphasized that all features were computed from the EMG data recorded from 4 separate days, thus the effect of fluctuating EMG signals between days was also included in the finding results. In other words, the correlated anthropometric variables with EMG features could be used to calibrate the system even though the EMG feature values have been changed from one day to another day. Additionally, the correlations from each single day are similar to that from all days. The interesting result is in the case of the third day and the fourth day. The correlations between features and anthropometric variables improved a litter bit. It may be due to an increasing experience of the subjects to perform the stable movements. The performance of EMG pattern classification has been found that it improves when the data used for training a classifier are recorded from the third day and the fourth day (Phinyomark et al., 2012a; Zhang et al., 2008).

Table 4. Correlation coefficients r between anthropometric variables and EMG features in cases of strong and significant relationships, at least 4 movements for a muscle, based on 10 male subjects.

Feature	Anthropometric variable	Position	Movements	Average $\|r\|$ (min-max)
MAV	Bi-deltoid breadth	ECU	FP, WE, WF, WU, HC	0.77 (0.73-0.82)
WL	Bi-deltoid breadth	BB	WE, WR, WU, HC	0.71 (0.69-0.73)

Table 5. Correlation coefficients r between anthropometric variables and EMG features in cases of strong and significant relationships, at least 4 movements for a muscle, based on 10 female subjects.

Feature	Anthropometric variable	Position	Movements	Average $\|r\|$ (min-max)
MAVS	Hand length	FCR	FS, WE, WF, WR, HC	0.77 (0.70-0.86)
MAVS	Hand length	BB	FS, WE, WF, WU	0.80 (0.75-0.87)
WL	Bi-deltoid breadth	BB	FP, FS, WF, WR, HC	0.75 (0.72-0.83)
ZC	Biceps circumference	BB	FP, FS, WF, HO	0.76 (0.69-0.85)
ZC	Bi-deltoid breadth	BB	FP, FS, WF, WU, HC	0.77 (0.70-0.84)
SSC	Bi-deltoid breadth	BB	FP, FS, WF, WU, HC	0.80 (0.70-0.86)

To observe the results, for instance, the average MAV feature extracted from ECU muscle and WF movement of each male subject was plotted in space with the bi-deltoid

breadth value of that subject, and the linear line of best fit representing the association was also shown in Fig. 5. It showed that both measured variables have an inverse relationship (one variable increased when another one decreased) and the variable points closed to a straight line ($r = 0.82$ at $p < 0.05$).

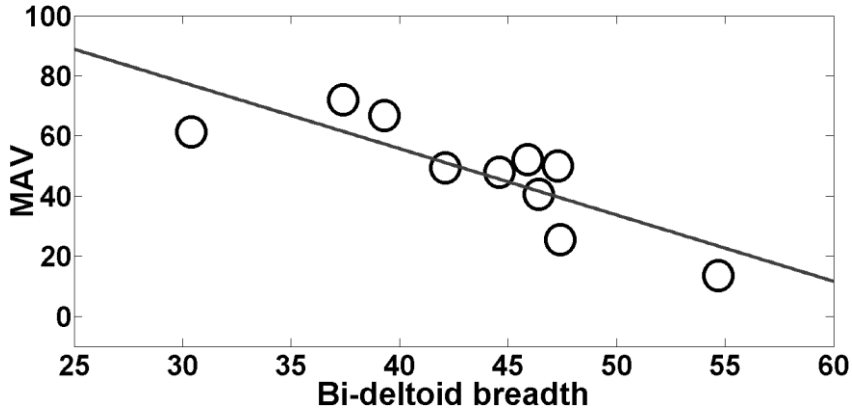
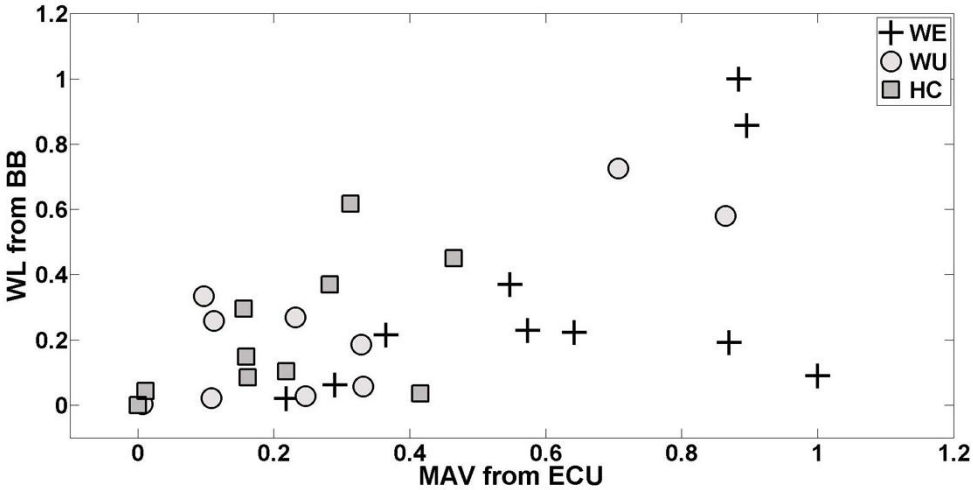
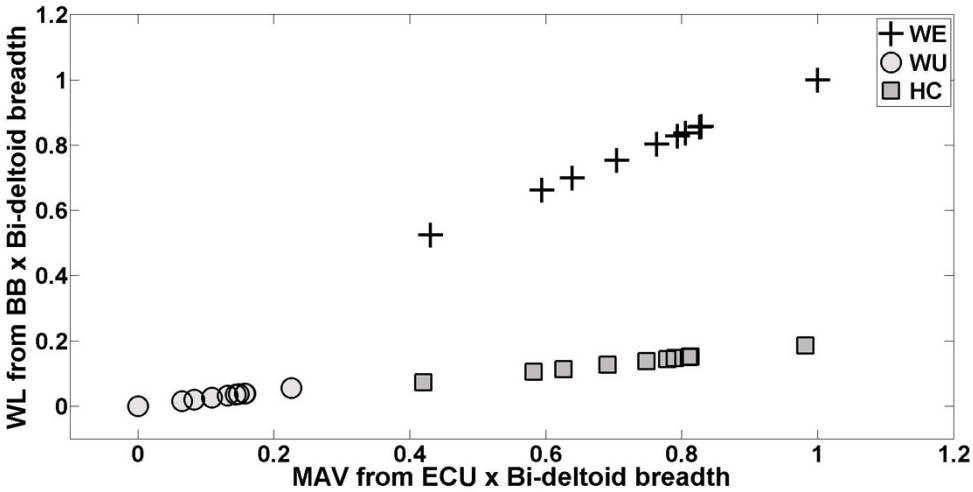


Fig. 5. The relationship between average MAV features extracted from ECU muscle and bi-deltoid breadth values for 10 male subjects with a line of best fit applied for WF movement.



(a)



(b)

Fig. 6. Scatter plots of MAV features extracted from ECU muscle and WL features extracted from BB muscle with 3 movements (WE, WU, and HC) from 10 male subjects (a) original features (b) normalized feature. Note that all features were normalized to be [0, 1].

In practice, we can use the correlated anthropometric variables to calibrate the EMG classification system in two different ways: a weighting factor for classifier and a normalizing value for EMG features. This topic is out of scope for this paper because the approach depends on the types of the classifier. However, in order to show the feasibility to use this anthropometric variable, two original features of 3 movements (WE, WU and HC) of 10 male subjects from Table 4 were plotted in space as shown in Fig. 6(a) and the normalized EMG features (original EMG feature multiply by bi-deltoid breadth value of the same subject) were also plotted as shown in Fig. 6(b). By comparing Figs. 6(a) and 6(b), it is clear that the distribution in space of each movement of the original features is very poor. It is difficult to classify the movements because of the large variation of each feature between the subjects. If we train the classification system using EMG features extracted from one of the subjects and test the system with features from the remaining subjects, we would get a low classification accuracy. On the other hand, the discrimination between three movements of the normalized features, in Fig. 6(b), is very good. It could be possible to achieve a high classification accuracy by training the EMG system using features extracted from one of the subjects and testing with features calculated from the remaining subjects.

4. Discussion and future works

Based on the results of both tables, Table 4 and Table 5, we can observe that bi-deltoid breadth and BB muscle usually have a strong correlation and should be paid more an interest in further study. However, the bi-deltoid breadth cannot automatically and/or directly be measured from the EMG wearable device like the forearm or biceps circumference. On the other hand, biceps circumference had a strong correlation only with ZC features extracted from BB muscle. It is not enough for achieving very high classification accuracy for many gestures. However, a strong correlation between anthropometric variables and EMG features was obviously present, and further studies should be done in two different ways.

First, for an automatically calibrated system using measures directly obtained from the EMG armband, the relationship between biceps/forearm circumference and other EMG features should be investigated and used together with ZC to make a useful feature vector. Based on the results in this paper, only one out of five features has a strong relationship and the proposed time-domain features share mathematical definition and information with most of time-domain features (Phinyomark et al., 2012b). Hence, maybe only a few features will have a strong correlation with circumference variable, as found with the EMG MVC in Cannan and Hu (2011). However, the relationship with other types of EMG features like frequency-domain or time-scale features have not been evaluated yet; such feature types may have a high association.

Second, for a semi-automatically calibrating system, bi-deltoid breadth parameter showed a strong association with 4 out of 5 features and the remaining feature MAVS has low classification performance and was not used in several previous EMG systems e.g. Li et al.

(2011). Thus the bi-deltoid breadth could be useful in future works to be used as a weighting factor for classifier or a normalizing parameter for EMG features.

However, more attention should be paid to two limitations in the future: (1) the number of discriminated movements and (2) gender. In the experiments, eight movements were performed but only 4-5 movements had strong relationships. Other kinds of upper-limb movements may be needed in order to increase the ability of the system such as grasping and finger movements. In addition, strong correlations between anthropometry and EMG features were not found across the gender (both male and female subjects) in this study. The maximum *r* coefficient across the gender was 0.64, which was calculated between ZC and bi-deltoid breadth from BB muscle based on 8 movements.

In future works, the evaluation of the relationship between useful anthropometric variables and other types of features, movements and also muscle locations should be done. An EMG classification system that can automatically or semi-automatically calibrate should be implemented and its classification accuracy measured to evaluate the performance of the calibration. The system should be trained using EMG features extracted from one or a few subjects and tested with EMG features of the remaining (many subjects) like a 10-fold cross-validation between the subjects.

5. Conclusion

This paper presented the feasibility study on the use of anthropometric variables to make EMG-based MCIs easy to use in general population. The relationship of twelve anthropometric variables with five popular well used time-domain features have been evaluated based on five muscle positions and eight movements of upper-limb. Some strong associations between anthropometric variables and EMG features have been found and can be used to calibrate the EMG classification system automatically or semi-automatically.

Acknowledgements

This work was supported by the “FI UJF MSTIC”.

References

Ahsan, Md.R., Ibrahimy, M.I., Khalifa, O.O., 2010. Advances in electromyogram signal classification to improve the quality of life for the disabled and aged people. J. Comput. Sci. 6, 705–715.

Anakwe, R.E., Huntley, J.S., Mceachan, J.E., 2007. Grip strength and forearm circumference in a healthy population. J. Hand Surg. Eur. 32, 203–209.

Bolgla, L.A., Uhl, T.L., 2007. Reliability of electromyographic normalization methods for evaluating the hip musculature. J. Electromyography Kinesiol. 17, 102–111.

Cannan, J.A.R., Hu, H., 2011. Automatic circumference measurement for aiding in the estimation of maximum voluntary contraction (MVC) in EMG systems. In: Proceedings of the 4th international conference on Intelligent Robotics and Applications, pp. 202–211.

Cipriani, C., Zaccone, F., Micera, S., Carrozza, M.C., 2008. On the shared control of an EMG–controlled prosthetic hand: Analysis of user–prosthesis interaction. IEEE Trans. Rob. 24, 170–184.

Hariharan, M., Fook, C.Y., Sindhu, R., Ilias, B., Yaacob, S., 2012. A comparative study of wavelet families for classification of wrist motions. Comput. Electr. Eng. 38, 1798–1807.

Hof, A.L., Pronk, C.N.A., van Best, J.A., 1987. Comparison between EMG to force processing and kinetic analysis for the calf muscle moment in walking and stepping. J. Biomech. 20, 167–178.

Hudgins, B., Parker, P., Scott, R.N., 1993. A new strategy for multifunction myoelectric control. IEEE Trans. Biomed. Eng. 40, 82–94.

Holzbaur, K.R.S., Murray, W.M., Gold, G.E., Delp, S.L., 2007. Upper limb muscle volumes in adult subjects. J. Biomech. 40, 742–749.

Housh, D.J., Housh, T.J., Weir, J.P., Weir, L.L., Johnson, G.O., Stout, J.R., 1995. Anthropometric estimation of thigh muscle cross-sectional area. Med. Sci. Sports Exerc. 27, 784–791.

Ju, Z., Zhu, X., Liu, H., 2011. Empirical copula-based templates to recognize surface EMG signals of hand motions. Int. J. Humanoid Rob. 8, 725–741.

Kamavuako, E.N., Farina, D., Yoshida, K., Jensen, W., 2009. Relationship between grasping force and features of single-channel intramuscular EMG signals. J. Neurosci. Methods. 185, 143–150.

Li, G., Li, Y., Yu, L., Geng, Y., 2011. Conditioning and sampling issues of EMG signals in motion recognition of multifunctional myoelectric prostheses. Ann. Biomed. Eng. 39, 1779–1787.

Li, G., Schultz, A.E., Kuiken, T.A., 2010. Quantifying pattern recognition–based myoelectric control of multifunctional transradial prostheses. IEEE Trans. Neural Syst. Rehabil. Eng. 18, 185–192.

Marras, W.S., Sommerich, C.M., 1991. A three-dimensional motion model of loads on the lumbar spine: I. model structure. Hum. Factors. 33, 123–137.

Oatis, C.A., 2008. Kinesiology: the mechanics and pathomechanics of human movement, second ed. Lippincott Williams & Wilkins.

Oskoei, M.A., Hu, H., 2007. Myoelectric control systems–A survey. Biomed. Signal Process. Control. 2, 275–294.

Peerdeman, B., Boere, D., Witteveen, H., Huis, in `t Veld R., Hermens, H., Stramigioli, S., Rietman, H., Veltink, P., Misra, S., 2011. Myoelectric forearm prostheses: State of the art from a user–centered perspective. J. Rehabil. Res. Dev. 48, 719–738.

Phinyomark, A., Phukpattaranont, P., Limsakul, C., 2012a. Investigating long–term effects of feature extraction methods for continuous EMG pattern classification. Fluctuation Noise Lett. 11, 1250028.

Phinyomark, A., Phukpattaranont, P., Limsakul, C., 2012b. Feature reduction and selection for EMG signal classification. Expert Syst. Appl. 39, 7420–7431.

Phinyomark, A., Phukpattaranont, P., Limsakul, C., 2011a. A review of control methods for electric power wheelchairs based on electromyography (EMG) signals with special emphasis on pattern recognition. IETE Tech. Rev. 28, 316–326.

Phinyomark, A., Phukpattaranont, P., Limsakul, C., 2011b. Electromyography (EMG) signal classification based on detrended fluctuation analysis. Fluctuation Noise Lett. 10, 281–301.

Phinyomark, A., Limsakul, C., Phukpattaranont, P., 2009. A novel feature extraction for robust EMG pattern recognition. J. Comput. 1, 71–80.

Raez, M.B.I., Hussain, M.S., Mohd-Yasin, F., 2006. Techniques of EMG signal analysis: Detection, processing, classification and applications. Biol. Proced. Online. 8, 11–35.

Ray, G.C., Guha, S.K., 1983. Relationship between the surface e.m.g. and muscular force. Med. Biol. Eng. Comput. 21, 579–586.

Saponas, T.S., Tan, D.S., Morris, D., Turner, J., Landay, J.A., 2010. Making muscle–computer interfaces more practical. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems, pp. 851–854.

Saponas, T.S., Tan, D.S., Morris, D., Balakrishnan, R., Turner, J., Landay, J.A., 2009. Enabling always-available input with muscle-computer interfaces. Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, pp. 167–176.

Taylor, R., 1990. Interpretation of the correlation coefficient: A basic review. J. Diagn. Med. Sonography. 6, 35–39.

Tkach, D., Huang, H., Kuiken, T.A., 2010. Study of stability of time–domain features for electromyographic pattern recognition. J. NeuroEng. Rehabil. 7, 21.

Vera-Garcia, F.J., Moreside, J.M., McGill, S.M., 2010. MVC techniques to normalize trunk muscle EMG in healthy women. J. Electromyography Kinesiol. 20, 10–16.

Wojtczak, P., Amaral, T.G., Dias, O.P., Wolczowski, A., Kurzynski, M., 2009. Hand movement recognition based on biosignal analysis. Eng. Appl. Artif. Intell. 22, 608–615.

Zhang, X., Chen, X., Zhao, Z.Y., Li, Q., Yang, J.H., Lantz, V., Wang, K.Q., 2008. An adaptive feature extractor for gesture SEMG recognition. Proceedings of the 1st International Conference on Medical Biometrics, pp. 83–90.

Appendix A

Equation A.1 (Marras and Sommerich, 1991) is used to determine a muscle force level at each sampling point in time. It sums up all of the modifications made to the EMG signal and can convert the empirical EMG signal into muscle force.

$$Muscle\ force = gain \times \frac{EMG}{EMG_{\max}} \times area \times LS\ factor \times V\ ratio, \tag{A.1}$$

where *Muscle force* = muscle tension associated with EMG;
gain = factor that includes maximum muscle force per unit of area;
EMG = measured EMG value at a particular time;
$EMG_{\max}$ = maximum EMG value for specific muscle at specific angle of operation;
area = muscle cross-sectional area;
LS factor = length-strength modulation factor;
V ratio = velocity modulation factor.

EMG Feature Evaluation for Improving Myoelectric Pattern Recognition Robustness

Angkoon Phinyomark[a,b], Franck Quaine[a], Sylvie Charbonnier[a], Christine Serviere[a], Franck Tarpin-Bernard[b], Yann Laurillau[b]

angkoon.p@hotmail.com, franck.quaine@gipsa-lab.grenoble-inp.fr, sylvie.charbonnier@gipsa-lab.grenoble-inp.fr, christine.serviere@gipsa-lab.grenoble-inp.fr, franck.tarpin-bernard@imag.fr, yann.laurillau@imag.fr

[a] GIPSA Laboratory, CNRS UMR 5216, Control System Department, SAIGA team, University Joseph Fourier, Grenoble, France
[b] LIG Laboratory, CNRS UMR 5217, University of Grenoble, Grenoble, France

Abstract

In pattern recognition-based myoelectric control, high accuracy for multiple discriminated motions is presented in most of related literature. However, there is a gap between the classification accuracy and the usability of practical applications of myoelectric control, especially the effect of long-term usage. This paper proposes and investigates the behavior of fifty time-domain and frequency-domain features to classify ten upper limb motions using electromyographic data recorded during 21 days. The most stable single feature and multiple feature sets are presented with the optimum configuration of myoelectric control, i.e. data segmentation and classifier. The result shows that sample entropy (SampEn) outperforms other features when compared using linear discriminant analysis (LDA), a robust classifier. The averaged test classification accuracy is 93.37%, when trained in only initial first day. It brings only 2.45% decrease compared with retraining schemes. Increasing number of features to four, which consists of SampEn, the fourth order cepstrum coefficients, root mean square and waveform length, increase the classification accuracy to 98.87%. The proposed techniques achieve to maintain the high accuracy without the retraining scheme. Additionally, this continuous classification allows the real-time operation.

Keywords

Electromyography (EMG); feature extraction; linear discriminant analysis; myoelectric control; sample entropy.

Name and address for correspondence

Angkoon Phinyomark
Portal address: GIPSA-lab, Grenoble Campus, 961, rue de la Houille Blanche BP46,
F-38402 Saint Martin d'Hères Cedex, France
E-mail address: angkoon.p@hotmail.com, angkoon.phinyomark@gipsa-lab.grenoble-inp.fr
Telephone number: +33 (0)6 15 92 73 80, Fax number: +33 (0)4 76 82 63 88

1. Introduction

Myoelectric control systems (MCSs) have been used to control assistive and rehabilitation devices for many years by conducting the classified patterns of surface electromyography (EMG) signal (Oskoei & Hu, 2007; Peerdeman et al., 2011; Zecca, Micera, Carrozza & Dario, 2002). Most of related literature on EMG pattern recognition focuses on the improvement of classification accuracy and the number of discriminated motions (Oskoei & Hu, 2007; Zecca et al., 2002). Although accuracy has achieved above 90% for multiple discriminated motions using various combinations of advanced techniques in feature extraction, dimensionality reduction and pattern classification (Peerdeman et al., 2011), the usability of MCSs is still challenged by some issues (Boschmann, Kaufmann, Platzner & Winkler, 2010; Chen, Geng & Li, 2011; Fougner, Scheme, Chan, Englehart & Stavdahl, 2011; Kaufmann, Englehart & Platzner, 2010; Tkach, Huang & Kuiken, 2010; Young, Hargrove & Kuiken, 2011, 2012; Zhang et al., 2007). These problems need to be solved for realizing practical applications of MCSs, such as effects of electrode location shift (Tkach et al., 2010; Young et al., 2010, 2011), variations in muscle contraction effort (Tkach et al., 2010), variations in limb position (Chen et al., 2011; Fougner et al., 2011), and changes in EMG patterns over time (Boschmann et al., 2010; Kaufmann et al., 2010).

Recently, the effect of long-term/prolonged usage has been emphasized in a few works (Boschmann et al., 2010; Kaufmann et al., 2010; Phinyomark, Phukpattaranont & Limsakul, 2012a; Zhang et al., 2007). The conditions controlled for collecting training and testing data in most of related literature are only from one or a few days. On the other hand, EMG data measured in one day are relatively different from that in another day even on the same subjects (Jain, Singhal, Smith, Kaliki & Thakor, 2012). In addition, the EMG data recorded in the laboratory are likely different from what would be expected in the real-world usage. Therefore, the classification accuracies that are measured and reported in the published literature and in the clinically observed results tend to be different.

To demonstrate the real-world use of MCSs, EMG data recorded under realistic conditions are required in order to provide the real classification accuracy. The long-term effect of fluctuating EMG signals in state-of-the art classification algorithms, with training and testing EMG data recorded from 21 days is presented clearly in Kaufmann et al. (2010). The results show that the classification accuracies decrease with increasing time difference between training and testing data and decrease gradually (>10%), if not being retrained, for four effective algorithms: multi-layer perceptron neural networks (MLP-NN) (Hudgins, Parker & Scott, 1993), support vector machine (SVM) (Oskoei & Hu, 2008), k-nearest neighbor (KNN) (Kim, Choi, Moon & Mun, 2011), and decision tree (DT) (Geethanjali & Ray, 2011) using Hudgins' time domain (TD) feature set (Kaufmann et al., 2010). On the other hand, the classification accuracy drops only 3.6% for linear discriminant analysis (LDA) classifier, thus LDA is recommended as the robust classifier and it has been employed in several recent literatures (Tkach et al., 2010; Young et al., 2010, 2011) for developing robust pattern recognition-based MCSs.

However, the classification accuracy achieved with LDA and Hudgins' TD feature set is approximately 80%. Moreover, due to a lack of evaluating the optimal robust EMG features (Boschmann et al., 2010; Kaufmann et al., 2010; Zhang et al., 2007), this paper proposes and

investigates the behavior of fifty TD and frequency domain (FD) features to classify ten upper limb motions using EMG data recorded during 21 days, in order to fulfill the completion of the previous study (Kaufmann et al., 2010) on finding the most stable combination between EMG feature extraction and classification algorithm. All features were evaluated with LDA when trained on the initial data, most recent data, and all preceding data. In addition, the optimum configuration of MCSs for the robust features is explored.

2. Background

In the past decade's literature on MCSs (Englehart, Hudgins & Parker, 2001), myoelectric pattern recognition is often divided to three components: feature extraction, dimensionality reduction, and pattern recognition. The success of this system to achieve a high classification performance depends almost entirely on the selection of EMG features (Oskoei & Hu, 2007, 2008).

2.1 EMG Feature Extraction

Feature extraction is a technique to transform raw input data into a reduced representation set of features, which is called a feature vector. A suitable feature vector should contain the useful information and discard the irrelevant/noise information (Zardoshti-Kermani, Wheeler, Badie & Hashemi, 1995). Based on the literature, EMG features can be separated into several groups: TD, FD or spectral domain, and time-scale or time-frequency domain (TFD) (Oskoei & Hu, 2007). In other words, features of EMG can be computed based on both linear and nonlinear analysis.

TD features are extracted directly from raw EMG time series without any transformation, thus features in this group are easy to implement and require low computational load. On the other hand, FD features are usually statistical properties of power spectral density of EMG signal.

Fifty features are proposed in this study. They are based on TD and FD and also linear and nonlinear analysis, as shown in Table 1 with the specific parameters and references for mathematical definition. All of these features have been previously used in the analysis of surface EMG signal. TD features are presented in the first 33 rows in Table 1 and remaining 17 rows in the table are features in FD.

The optimal parameters for the EMG features that were implemented in this study are based on the suggestion of related works and the preliminary experiments, as presented in Table 1. For instance, the m and r parameters of approximation entropy and sample entropy are based on the suggestion in the literature (Zhao, Jiang, Cai, Liu & Hirzinger, 2006a; Zhao, Xie, Jiang, Cai, Liu & Hirzinger, 2006b), or the threshold value which is used in slope sign change and Willison amplitude is based on the preliminary experiments in this study.

However, features in the TFD group, e.g. discrete wavelet transform (DWT) and wavelet packet transform (WPT), are not included in this study based on two reasons:

(1) TD features achieve higher accuracy than TFD features for the LDA classifier, whereas TFD features achieve higher accuracy than TD features for the SVM classifier (Lorrain, Jiang & Farina, 2011; Phinyomark et al., 2012a). Due to a robustness of LDA over

SVM as described (Kaufmann et al., 2010), the development of EMG pattern recognition should be based on the LDA classifier.

(2) TD features show a better performance than TFD features in the classification of both transient and steady-state EMG signal using LDA (Lorrain et al., 2011). Due to a drawback of classifying only transient signal or only steady-state signal, the development of EMG pattern recognition should be based on the continuous classification of the both signal types (Yang, Zhao, Jiang & Liu, 2012).

In addition, it is clearly shown that raw wavelet coefficients are not possible to be used if additional feature extraction and/or feature projection is not applied. For instance, feature extraction that is applied for wavelet coefficients is usually TD techniques (Hariharan, Fook, Sindhu, Ilias & Yaacob, 2012; Phinyomark, Nuidod, Phukpattaranont & Limsakul, 2012b). Principle component analysis (PCA) and uncorrelated LDA (ULDA) are two popular used feature projection in EMG signal classification (Englehart et al., 2001; Phinyomark et al., 2012a). The classification performance of these features also depends on wavelet parameters such as mother families and decomposition level (Hariharan et al., 2012; Phinyomark et al., 2012b).

2.2 Dimensionality Reduction and Pattern Classification

Dimensionality reduction is a technique to reduce the dimension of an original feature vector, while preserving the most discriminative information and removing the remaining irrelevant information, for reducing the computational time in a classifier. This technique is mostly applied for TFD features (Englehart et al., 2001) which provide a high vector dimension. Additionally, it improves small accuracy of classification for the TD and FD features (Phinyomark et al., 2012a). Hence, the dimensionality reduction technique is not applied in this study.

LDA is used as a pattern classification algorithm in this study. In addition to the robust property of LDA, this classifier was chosen based on several reasons (Englehart et al., 2001; Jain et al., 2012; Kaufmann et al., 2010), i.e. it is a simple statistical approach and does not require any parameter adjustment. LDA is also a computationally efficient real-time operation. The classification performance of LDA is similar to more complex classification algorithms such as MLP-NN and SVM (Kaufmann et al., 2010; Oskoei & Hu, 2008).

3. Data Collection

The EMG dataset used to test the proposed EMG features was provided by the University of Paderborn in Germany. It is the same EMG dataset that is used in the evaluation of EMG classification algorithms (Kaufmann et al., 2010). The data were obtained using a portable EMG data recording device (NeXus-10, Mind Media BV). The EMG data were collected from a male non-amputee subject from four electrode-pair positions on the top, bottom, medial, and lateral sides of the forearm (ch0-ch3) with the reference (r) at the wrist, as shown in Fig. 1. These signals were sampled at 2048 Hz with a high resolution of 24 bits.

The EMG data were collected as the volunteer performed eleven upper limb motions including (m1) extension, (m2) flexion, (m3) ulnar deviation, (m4) radial deviation, (m5)

pronation, (m6) supination, (m7) open, (m8) close, (m9) key grip, (m10) pincer grip, and (m11) extract index finger. Because, the energy of the pincer grip grows considerably over the trials, thus only ten motions (m1-m9, m11) were included for the data analysis (Kaufmann et al., 2010).

To investigate the long-term EMG effect, the EMG data were recorded for 21 days with five to six trials per day. In total, 121 trials are obtained from 21 days. Within each trial, all motions were performed in the same sequence by the subject and each motion is maintained for five seconds. At the start, the end, and between motions, the rest periods were introduced as can be observed from the example of one trial in Fig. 2. Each contraction period can be divided roughly into a one-second onset phase and a four-second subsequent steady-state phase. Therefore, only the data of the steady-state phase were used for the classification experiments (Kaufmann et al., 2010).

It should be emphasized that the exact electrode positions were marked specifically for the subject in order to be able to reestablish the experimental setup on different days. Furthermore, to approximate realistic conditions of MCSs, the recordings were performed under various subject conditions, such as after sleep and meals, and after periods of high and low physical activity.

4. Data Analysis

4.1 Investigation of Robust Single EMG Feature

All EMG features in Table 1 were used to investigate the effect of long-term usage of the pattern recognition-based myoelectric control. Three experiments that are proposed in the evaluating EMG classification algorithms (Kaufmann et al., 2010) were used.

The fluctuating EMG data, which are used as training and testing data, contained 121 trials in total from 21 days. All experiments used the EMG trials (i = 2-121) as the testing set, whereas the training set trials are different in each experiment: (1) training on the initial data (only the first day), (2) training on the most recent data (five recent preceding trials), and (3) all preceding data. For a test trial i, $2 \leq i \leq 121$, the indices of the training set can be defined respectively as follows:

1) $i, \ldots, \min(s, i-1)$,

2) $\max(i-s, 1), \ldots, i-1$,

3) $1, \ldots, i-1$.

where the number of trials s is set at 5, because five trials corresponded to the EMG data collected from one day and were enough to gain high classification accuracy for the classifiers (Kaufmann et al., 2010).

The classification accuracies were obtained from LDA with a segment length of 250 ms (512 samples). An overlapped segmentation with an increment of 125 ms (256 samples) was applied.

The selection of the most robust EMG features is based on the result of the first experiment. The stable feature should succeed in maintaining high classification accuracy (>80%) (Peerdeman et al., 2011), whether the time difference between training and testing

data increased. Furthermore, the differences between accuracies obtained from three experiments can be observed in order to find the suitable training approach.

4.2 Investigation of Robust Multiple EMG Feature Sets

Using a single feature for the continuous classification may make it difficult to reach a high accuracy. Multiple feature sets have been employed and succeeded in the classification of multiple motions-based EMG signal (usually more than 4 motion classes) in many recent researches (Oskoei & Hu, 2007; Peerdeman et al., 2011).

The possible combinations of two to four features were applied to the LDA classifier. The number of multiple features in this study is based on the finding result in previous study (Phinyomark, Phukattaranont & Limsakul, 2012c) that many TD and FD features provide similar information and patterns in feature space. Thus, increasing the number of redundant features would improve a little classification performance and four features are enough to gain high classification accuracy. The features were computed from overlapped segments with a length of 250 ms and an increment of 125 ms. The possible combinations of two, three, and four multiple features from a total of 50 features are 1225, 19600, and 230300 feature sets.

4.3 Optimum Configuration of MCSs for Robust Features

The optimum configuration of robust features based MCSs was investigated consisting of the classification algorithm and the data segmentation.

For the classification algorithm, LDA shows more robust when not being trained iteratively compared to MLP-NN, SVM, KNN, and DT using the same EMG dataset (Kaufmann et al., 2010). In this study, two new state-of-the art classifiers, namely random forests (RFS) (Breiman, 2001) and quadratic discriminant analysis (QDA) (Kim et al., 2011), were compared with LDA.

The QDA classifier can be seen as a more general version of LDA, which separates the classes of objects by a quadratic surface instead of linear, as implemented in LDA classifier. The RFS classifier is closely related to DT. It is an ensemble classifier of DTs. RFS also shows better performances than LDA, QDA, KNN, MLP-NN and SVM in the classification of grasping of objects (Liarokapis, Artemiadis, Katsiaris, Kyriakopoulos & Manolakos, 2012).

For the data segmentation, a segment length which is used to compute features in the previous experiments is set at 250 ms. This segment length contains enough information to estimate a motion and can performs real-time operation (Oskoei & Hu, 2008). However, a segment smaller than 250 ms could reduce the computation time, while leads to increase the variation in estimated feature. On the other hand, a segment larger than 250 ms could decrease the variance of estimation, while it leads to increase the computational load.

To use a larger segment, the overlapped segmentation is applied instead of the disjoint segmentation. The accuracy of classification with a segment length of 125, 250, and 500 ms (256, 512, and 1024 samples) and an increment of 62.5, 125, and 250 ms (128, 256, and 512 samples) were examined for both types of segmentation. The increment can be defined as an

interval of time between two consecutive segments and should be less than 300 ms to avoiding failure in real-time operation.

In total, eight options of segmentation (segment length, increment) were proposed: (125 ms, 62.5 ms), (125 ms, 125 ms), (250 ms, 62.5 ms), (250 ms, 125 ms), (250 ms, 250 ms), (500 ms, 62.5 ms), (500 ms, 125 ms), and (500 ms, 250 ms).

5. Results and Discussion

5.1 Robust Single EMG Feature

The averaged classification accuracies obtained from LDA of all test trials for 50 features for 3 experiments are presented in Table 2. SampEn has the highest classification accuracy, followed by ApEn and MFL. Without retraining scheme, the accuracy of SampEn is 93.37%. It drops only by 3.24% and 2.45% compared with training in recent preceding data and all preceding data, respectively. The SampEn provides 8.69% accuracy higher than ApEn, the second rank of the 50 features. The accuracy of ApEn is 84.68%. Furthermore, the results clearly show that the best single feature, SampEn achieves better classification performance than Hudgins' TD feature set i.e. MAV, WL, ZC, and SSC. The classification accuracy increases from 78.73% to 93.37% (Kaufmann et al., 2010).

The SampEn also has better classification performance than the other EMG features using five recent preceding trials for training, followed closely by the MFL and DASDV. On the other hand, the MFL has a bit improved classification accuracy than the SampEn using all receding trials for training. However, the accuracy of MFL and DASDV degrades with rising time difference between training and testing data (when test trial $i > 30$ or > 6 days), as can be seen in Fig. 3. Note that the horizontal axis in Fig. 3 shows the test trial i and the vertical axis shows the corresponding test accuracy.

Based on the discriminating ten upper limb motions using four forearm EMG channels, the SampEn performs the best in both classification accuracy and stability.

There are several interested issues from the experimental results that could be discussed as follows:

(1) The difference of classification accuracies when trained in recent preceding five trials and all preceding trials, as shown in Table 3, is not statistically significant in 40 features and the accuracy of using five recent trials (T1) for training is higher than the another scheme (T2) for 35 features. It means that it is not necessary to use all preceding data as a training set and we can reduce the size of training set by using only the recent EMG data. This may be due to the variation in muscle contraction effort and the difference in subject conditions between days, thus the excluded trials would have provided less relevant information for the classification system.

(2) SampEn shows a success as the robust muscle activity onset detection against the spurious spikes in surface EMG signal by using global tolerance r (Zhang & Zhou, 2012). The global tolerance scheme is also recommended to use in the estimation of grasping force (Kamavuako, Farina, Yoshida & Jensen, 2012) compared to the local tolerance scheme or standard SampEn. In this study, the global tolerance scheme is applied. This scheme sets a uniform tolerance for the signal of all the analysis segments instead of each analysis segment,

as implemented in the local tolerance scheme. This parameter is critical in reducing the effect of large variations in motions-based EMG signal.

The tolerance r is set as $0.20 \times \sigma$, in this study, where σ is the standard deviation of the EMG time series. This value is based on the suggestion of Richman and Moorman (2000). The global tolerance r, which was set as $0.20 \times \sigma$, also provides good performances in the classification of finger motions-based surface EMG signal (Zhao et al., 2006a) and the estimation of grasping force-based intramuscular EMG signal (Kamavuako et al., 2012). More details about the theory of SampEn are presented in the Appendix.

(3) As mentioned above that SampEn with global tolerance scheme showed a success as both the onset detection method (compared to the amplitude thresholding and Teager-Kaiser Energy operation-based methods (Zhang & Zhou, 2012) and force estimation method (compared to the global discharge rate and RMS methods (Richman & Moorman, 2000), the SampEn feature would be useful not only in continuous classification, as presented in this study, but also in discrete classification. In other words, it would be useful to classify and estimate both type and force level of movements simultaneously, while improve the robustness to noise.

(4) The stable single features i.e. SampEn, ApEn, and MFL are nonlinear signal processing methods (Arjunan & Kumar, 2010; Phinyomark, Phukpattaranont & Limsakul, 2012d; Zhang & Zhou, 2012; Zhao et al., 2006a, 2006b). As mentioned in several studies, one of the major properties of surface EMG signal is nonlinearlity. Nonlinear analysis techniques can extract the real hidden information from surface EMG data and these techniques are robust to the low-level muscle contraction (Phinyomark et al., 2012d) and noise (Zhang & Zhou, 2012). These features contain the complexity information and can be used to highlight bursts of surface EMG signal.

(5) TD features imposed relatively high performance of classification and low load of computation compared to FD features. All FD features provided weak performance during classification (< 67%), except three FD features (i.e. MNP, TTP, and SM), that have the same discrimination in feature space as TD features. It means that the patterns of FD features are not sufficiently discriminative to identify the upper limb motions. However, the patterns of the TD and FD features are different, and multiple TD and FD features may increase the class separability, as can be observed in Table 4, such as SampEn + MNF and SampEn + FR.

5.2 Robust Multiple EMG Feature Sets

Based on the results in the previous study (Phinyomark et al., 2012c), most of TD and FD features are redundant. In other words, they share similar properties and information. Hence, it does not guarantee that combinations of several good robust single features would provide good robust multiple-feature sets. For instance, both SampEn and ApEn are entropy estimation techniques (Richman & Moorman, 2000), or TD features like IAV, MAV, and RMS are based on the energy information (Phinyomark et al., 2012c).

In this study, the classification performance obtained with all the possible combinations of 50 features to make two, three and four multiple-features are evaluated and the best ten combinations in each multiple-feature type are presented in Table 4.

Classification accuracies obtained from the multiple feature sets increase compared to the single features. They reach 97-98%, while the variation in the accuracy during test trials decreases (<3%) for the robust multiple feature sets. The best feature sets for two, three, and four features are respectively SampEn + CC, SampEn + CC + MAV2, and SampEn + CC + RMS + WL/AAC.

Based on the discriminating ten upper limb motions using four forearm EMG channels, the best feature set (i.e. SampEn, CC, RMS and WL) provides 97.75% accuracy with the real-time operation and stability of classification performance. It should be noted that WL is chosen instead of AAC because WL and AAC provide the same feature space when WL requires less computational cost than AAC. The dimension of feature vector is 7 (one value from each of SampEn, RMS and WL, and four values from CC). Moreover, the patterns in space of four features in the best robust set are different. These four features provide the useful information based on the similarity, the energy, the complexity, and the predicting model (Phinyomark et al., 2012c).

5.3 Optimum Configuration for Robust Features

The accuracies of LDA with the robust single feature and multiple-feature sets are compared with RFS and QDA. The results are shown in Fig. 4. The features were computed with a segment length of 250 ms and an increment of 125 ms and all classifiers were trained in the first five trials or only the first day. It can be clearly seen that the LDA classifier yields better performance than the RFS and QDA classifiers. Among the proposed classifiers, LDA performs the best in classification accuracy, stability, and computational load.

Table 5 shows the performance of classification in different segment lengths. The results show that accuracy increases by increasing the segment length from 125 ms to 500 ms. This is because a larger segment provides additional information and yields small bias and variance in the estimation of feature. The optimal segment length is 500 ms. The results also show that it does differ significantly by changing the segment length from 125 ms to 250 ms (about 2-3%), but it does not largely differ by changing the segment length from 250 ms to 500 ms (<1%). In addition, the performance of the overlapped segmentation has no improvement of classification accuracy, but it is significant to employ the large segment (greater than 200 ms) in order to avoiding a delay time.

6. Conclusion

For realizing practical applications of MCSs, the effect of long-term usage or reusability is carefully considered in this study. Fifty EMG features in TD and FD are evaluated by using EMG recorded from 21 days. From the results, four main points are summarized as in the following:

(1) SampEn is the best robust single feature and SampEn + CC + RMS + WL set is the best robust multiple-feature set.

(2) Features based on TD show a better performance than FD features for robust EMG pattern classification.

(3) LDA shows a better performance in the classification of fluctuating EMG signals compared to several classifiers i.e. RFS, QDA, MLP-NN, SVM, KNN, and DT.

(4) The suitable data segmentation is the overlapped segment with a length of 500 ms and an increment of 125 ms. This segment condition can be operated in real-time and provides high accuracy.

(5) The performance of the fifty EMG features proposed in this paper has been reported before in the literature, but it is difficult to compare their performances due to the difference in the experiments. The performance of the fifty features is compared under the same EMG dataset in this paper. This could be useful in the selection of EMG features for MCSs in future research.

The findings in this study can be applied in many EMG applications. In human computer interaction (HCI) (Saponas et al., 2009), for instance, a novel fusion of an interactive surface and EMG signal (Benko, Saponas, Morris & Tan, 2009) would provide useful interaction capabilities if the EMG system can be used in many days without retraining schemes as same as the interactive surfaces that do not require the retraining procedure.

Appendix

SampEn is feature extracted similarity information in time series. This entropy feature is developed from the ApEn in order to avoid the bias caused by self-matching. SampEn can be expressed as

$$\mathrm{SampEn}\left(m,r,N\right)=-\ln\left[A^{m}\left(r\right)/B^{m}\left(r\right)\right], \tag{A1}$$

where $B^m(r)$ and $A^m(r)$ are the probability that two series will match for m and m+1 points, respectively. Hence, the SampEn is the negative natural logarithm of an estimate of the conditional probability that the patterns of the time series that are similar to each other within a predefined tolerance r will remain similar for the next comparison point (Richman & Moorman, 2000).

Acknowledgements

The authors gratefully acknowledge the support of Prof. Paul Kaufmann from the University of Paderborn, Germany, for providing EMG data. This work was supported by the “FI UJF MSTIC”.

References

Anatomy of the human body [Image] (1918). Retrieved November 4, 2012, from Wikipedia.

Arjunan, S. P., & Kumar, D. K. (2010). Decoding subtle forearm flexions using fractal features of surface electromyogram from single and multiple sensors. *Journal of NeuroEngineering and Rehabilitation*, *7*(53).

Benko, H., Saponas, T. S., Morris, D., & Tan, D. (2009). Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of ACM International Conference on Interactive Tabletops and Surfaces* (pp. 93–100).

Boschmann, A., Kaufmann, P., Platzner, M., & Winkler, M. (2009). Towards multi-movement hand prostheses: Combining adaptive classification with high precision sockets. In *Proceedings of 2nd European Conference Technically Assisted Rehabilitation*.

Breiman, L. (2001). Random forests. *Machine Learn*ing, *45*(1), 5–32.

Chen, L., Geng, Y., & Li, G. (2011). Effect of upper-limb positions on motion pattern recognition using electromyography. In *Proceedings of 4th International Congress on Image and Signal Processing* (pp. 139–142).

Ehtiati, T., Kinsner, W., & Moussavi, Z. K. (1998). Multifractal characterization of the electromyogram signals in presence of fatigue. In *Proceedings of IEEE Conference on Electrical and Computer Engineering* (pp. 866–869).

Englehart, K., Hudgins, B., & Parker, P. A. (2001). A wavelet-based continuous classification scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 48(3), 302–311.

Fougner, A., Scheme, E., Chan, A. D. C., Englehart, K., & Stavdahl, Ø. (2011). Resolving the limb position effect in myoelectric pattern recognition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *19*(6), 644–651.

Geethanjali, P., & Ray, K. K. (2011). Identification of motion from multi-channel EMG signals for control of prosthetic hand. *Australiasian College of Physical Scientists and Engineers in Medicine*, *34*(3), 419–427.

Gitter, J. A., & Czerniecki, M. J. (1995). Fractal analysis of the electromyographic interference pattern. *Journal of Neuroscience Methods*, *58*(1–2), 103–108.

Gupta, V., Suryanarayanan, S., & Reddy, N. P. (1997). Fractal analysis of surface EMG signals from the biceps. *International Journal of Medical Informatics*, *45*(3), 185–192.

Hariharan, M., Fook, C. Y., Sindhu, R., Ilias, B., & Yaacob, S. (2012). A comparative study of wavelet families for classification of wrist motions. *Computers & Electrical Engineering*, 38(6), 1798–1807.

Hudgins, B., Parker, P., & Scott, R. N. (1993). A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, *40*(1), 82–94.

Jain, S., Singhal, G., Smith, R. J., Kaliki, R., & Thakor, N. (2012). Improving long term myoelectric decoding, using an adaptive classifier with label correction. In *Proceedings of 4th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics* (pp. 532–537).

Kamavuako, E. N., Farina, D., Yoshida, K., & Jensen, W. (2012). Estimation of grasping force from features of intramuscular EMG signals with mirrored bilateral training. *Annals of Biomedical Engineering*, *40*(3), 648–656.

Kaufmann, P., Englehart, K., & Platzner, M. (2010). Fluctuating EMG signals: investigating long-term effects of pattern matching algorithms. In *Proceedings of 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 6357–6360).

Kendell, C., Lemaire, E. D., Losier, Y., Chan, A., & Hudgins, B. (2012). A novel approach to surface electromyography: An exploratory study of electrode-pair selection based on signal characteristics. *Journal of NeuroEngineering and Rehabilitation*, *9*(24).

Khushaba, R. N., Al-Ani, A., & Al-Jumaily, A. (2010). Orthogonal fuzzy neighborhood discriminant analysis for multifunction myoelectric hand control. *IEEE Transactions on Biomedical Engineering*, *57*(6), 1410–1419.

Kim, K. S., Choi, H. H., Moon, C. S., & Mun, C. W. (2011). Comparison of *k*-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions. *Current Applied Physics*, *11*(3), 740–745.

Liarokapis, M. V., Artemiadis, P. K., Katsiaris, P. T., Kyriakopoulos, K. J., & Manolakos, E. S. (2012). Learning human reach-to-grasp strategies: towards EMG-based control of robotic arm-hand systems. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 2287–2292).

Lorrain, T., Jiang, N., & Farina, D. (2011). Influence of the training set on the accuracy of surface EMG classification in dynamic contractions for the control of multifunction prostheses. *Journal of NeuroEngineering and Rehabilitation*, *8*(25).

Nazarpour, K., Al-Timemy, A. H., Bugmann, G., & Jackson, A. (2013). A note on the probability distribution function of the surface electromyogram signal. *Brain Research Bulletin*, *90*, 88–91.

Oskoei, M. A., & Hu, H. (2007). Myoelectric control systems–A survey. *Biomedical Signal Processing and Control*, *2*(4), 275–294.

Oskoei, M. A., & Hu, H. (2008). Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Transactions on Biomedical Engineering*, *55*(8), 1956–1965.

Peerdeman, B., Boere, D., Witteveen, H., in ‘t Veld, R. H., Hermens, H., Stramigioli, S., …, Misra, S. (2011). Myoelectric forearm prostheses: State of the art from a user-centered perspective. *Journal of Rehabilitation Research and Development*, *48*(6), 719–737.

Phinyomark, A., Phothisonothai, M., Phukpattaranont, P., & Limsakul, C. (2011). Critical exponent analysis applied to surface electromyography (EMG) signals for gesture recognition. *Metrology and Measurement Systems*, *18*(4), 645–658.

Phinyomark, A., Phukpattaranont, P., & Limsakul, C. (2012a). Investigating long-term effects of feature extraction methods for continuous EMG pattern classification. *Fluctuation and Noise Letters*, *11*(4), 1250028.

Phinyomark, A., Nuidod, A., Phukpattaranont, P., & Limsakul, C. (2012b). Feature extraction and reduction of wavelet transform coefficients for EMG pattern classification. *Elektronika ir Elektrotechnika*, 122(6), 27–32.

Phinyomark, A., Phukpattaranont, P., & Limsakul, C. (2012c). Feature reduction and selection for EMG signal classification. *Expert Systems with Applications*, *39*(8), 7420–7431.

Phinyomark, A., Phukpattaranont, P., & Limsakul, C. (2012d). Fractal analysis features for weak and single-channel upper-limb EMG signals. *Expert Systems with Applications*, *39*(12), 11156–11163.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology*, *278*(6), H2039–H2049.

Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., & Landay, J. A. (2009). Enabling always-available input with muscle-computer interfaces. In *Proceedings of 22nd Annual ACM Symposium on User Interface Software and Technology* (pp. 167–176).

Talebinejad, M., Chan, A. D. C., Miri, A., & Dansereau, R. M. (2009). Fractal analysis of surface electromyography signals: a novel power spectrum-based method. *Journal of Electromyography and Kinesiology*, *19*(5), 840–850.

Tkach, D., Huang, H., & Kuiken, T. A. (2010). Study of stability of time-domain features for electromyographic pattern recognition. *Journal of NeuroEngineering and Rehabilitation*, *7*(21).

van den Broek, E. L., Schut, M. H., Westerink, J. H. D. M., van Herk, J., & Tuinenbreijer, K. (2006). Computing emotion awareness through facial electromyography. In *Proceedings of International Conference on Computer Vision in Human-Computer Interaction* (pp. 52–63).

Yang, D., Zhao, J., Jiang, L., & Liu, H. (2012). Dynamic hand motion recognition based on transient and steady-state EMG signals. *International Journal of Humanoid Robotics*, *9*(1), 1250007.

Young, A. J., Hargrove, L. J., & Kuiken, T. A. (2011). The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift. *IEEE Transactions on Biomedical Engineering*, *58*(9), 2537–2544.

Young, A. J., Hargrove, L. J., & Kuiken, T. A. (2012). Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration. *IEEE Transactions on Biomedical Engineering*, *59*(3), 645–652.

Zardoshti-Kermani, M., Wheeler, B. C., Badie, K., & Hashemi, R.M. (1995). EMG feature evaluation for movement control of upper extremity prostheses. *IEEE Transactions on Rehabilitation Engineering*, *3*(4), 324–333.

Zecca, M., Micera, S., Carrozza, M. C., & Dario, P. (2002). Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, *30*(4–6), 459–485.

Zhang, X., Chen, X., Zhao, Z. Y., Li, Q., Yang, J. H., Lantz, V., Wang, K. Q. (2007). An adaptive feature extractor for gesture SEMG recognition. In *Proceedings of 1st International Conference on Medical Biometrics* (pp. 83–90).

Zhang, X., & Zhou, P. (2012). Sample entropy analysis of surface EMG for improved muscle activity onset detection against spurious background spikes. *Journal of Electromyography and Kinesiology*, *22*(6), 901–907.

Zhao, J., Jiang, L., Cai, H., Liu, H., & Hirzinger, G. (2006a). A novel EMG motion pattern classifier based on wavelet transform and nonlinearlity analysis method. In *Proceedings of IEEE International Conference on Robotics and Biomimetics* (pp. 1494–1499).

Zhao, J., Xie, Z., Jiang, L., Cai, H., Liu, H., & Hirzinger, G. (2006b). EMG control for a five-fingered underactuated prosthetic hand based on wavelet transform and sample entropy. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3215–3220).

Figure captions

Fig. 1. The electrode placement used in the four-channel EMG acquisition (*Anatomy of the human body*, 1918).

Fig. 2. The example of EMG data from one trial from ch0.

Fig. 3. Test classification accuracy (%) in the trial i, where $2 \leq i \leq 121$, of (a) SampEn, (b) ApEn, (c) MFL, (d) DASDV, when trained in the first five, recent preceding five, and all preceding trials.

Fig. 4. Averaged test classification accuracy (%) of the best single feature and multiple-feature sets, when trained in the first five trials using the LDA, RFS and QDA classifiers with a segment length of 250 ms and an increment of 125 ms. The error bars show the standard deviation across trials.

COMM Notation for Specifying Collaborative and MultiModal Interactive Systems

Frédéric Jourde, Yann Laurillau, Laurence Nigay
University of Grenoble, CNRS, LIG
Domaine Universitaire
38000 Grenoble Cedex 9, France
{frederic.jourde, yann.laurillau, laurence.nigay}@imag.fr

ABSTRACT
Multi-user multimodal interactive systems involve multiple users that can use multiple interaction modalities. Although multi-user multimodal systems are becoming more prevalent (especially multimodal systems involving multitouch surfaces), their design is still ad-hoc without properly keeping track of the design process. Addressing this issue of lack of design tools for multi-user multimodal systems, we present the COMM (Collaborative and MultiModal) notation and its on-line editor for specifying multi-user multimodal interactive systems. Extending the CTT notation, the salient features of the COMM notation include the concepts of interactive role and modal task as well as a refinement of the temporal operators applied to tasks using the Allen relationships. A multimodal military command post for the control of unmanned aerial vehicles (UAV) by two operators is used to illustrate the discussion.

Author Keywords
Groupware, multimodal interaction, specification notation.

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Computer-supported supported cooperative work. H.5.2. User Interfaces: Input devices and strategies. D.2.2. Design Tools and Techniques: User interfaces.

General Terms
Human Factors; Design; Algorithms.

INTRODUCTION
We present the COMM (Collaborative and MultiModal) notation dedicated to the specification of multi-user multimodal interactive systems. An interactive system is multimodal if users can interact with the system using at least two (input or output) interaction modalities either operating in parallel or not. Several multi-user interactive systems (e.g. groupware) are multimodal such as the interactive systems described in [3, 4, 6, 12, 25, 35]. Furthermore, there is a growing interest in the CSCW community for multi-user multimodal interactive systems based on interactive multitouch surfaces [19, 22, 28, 30, 31]. Although multi-user multimodal systems are becoming more prevalent, their design is still ad-hoc without properly keeping track of the design process. The design of such interactive systems gives rise to new challenges due to the multiplicity of users with roles as well as the availability of different modalities and combinations of them. Addressing this issue of a lack of design tools for multi-user multimodal systems, we present the COMM notation and its editor for specifying interactive systems that are both multi-user and multimodal.

The specification step is one of the fundamental steps in any software development process regarding the waterfall [23] or the V lifecycle models [14]. It is essential to have specification tools that help the designer to precisely analyze and describe the user interface and to specify the interaction process [9]. A specification should also help the designer to detect usability problems at an early stage before implementing the interactive systems.

For specifying multi-user multimodal systems, we adopt a task-based notation. As pointed out in [33], task analysis is used in various steps of the development process and we focus here on the specification of a system to be then developed. In a previous study [10], we have reviewed several task-based notations, such as CTT [17], GTA [32, 33], MABTA [13], CUA [21] or UML-G [24] and we have highlighted the lack of support to describe the multimodal aspects of a multi-user interface. Addressing this issue, we present a notation, named COMM, that enables the designer to specify collaborative and multimodal aspects of an interactive system. Rather than creating a brand new notation from scratch, using a completely new graphical language to be learned, the COMM notation reuses and coherently articulates concepts of existing notations, such as CTT [17]. To describe the multimodal aspects of the interaction, we base the notation on the OpenInterface conceptual model [27] for multimodality (definition of a modality [18] and temporal constraints [34]).

We identify two requirements that the notation must satisfy: (1) to describe at a finer grain the group activities in order to clearly identify the cooperative and the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
EICS'10, June 19–23, 2010, Berlin, Germany.
Copyright 2010 ACM 978-1-4503-0083-4/10/06...$10.00.

collaborative activities; (2) to explicitly describe the link between the abstract User Interface (UI) specified as group tasks and the concrete UI involving multiple interaction modalities. In order to satisfy the first requirement, we introduce the concept of **interactive role**, in contrast to the concept of role usually used in the CSCW literature, that we will denote in this paper as business role[1] for clarity. To satisfy the second requirement: first, we introduce the concept of **modal task,** which denotes an atomic user action performed on a physical device; second, we refine CTT temporal operators using the temporal relationships between modalities defined in [34] and based on Allen relationships [2].

Figure 1: Two military operators controlling UAVs.

The paper is organized as follows: we first present a state of the art of the existing notations and we focus on their ability to specify concrete user interaction. We then describe the main elements of the COMM notation before presenting the e-COMM editor, an on-line application for specifying interactive systems using the COMM notation. We illustrate our notation by considering the design of a multi-user multimodal military command post. Indeed the COMM notation is currently used in the context of a French Army Research Department project, in partnership with the firms BERTIN, SAGEM, EADS and PY Automation, to design military command posts for the control of two unmanned aerial vehicles (UAV) by two operators, as shown in Figure 1.

RELATED WORK

"*Task analysis is the process of understanding the user's task thoroughly enough to help design a computer system that will effectively support users in doing the task.*" [11]. Beyond analyzing current or envisioning user's tasks, task-based notations are also commonly used to design and specify the user interface (UI) [33]. Finally some task-based specifications are also used as inputs for UI generation. In our study, we focus on task-based specification of the user interface and we do not cover the UI generation aspects. Therefore we do not consider UML-based notations, such as UML-G [24], which are best suited for software implementation and UI generation or transformation.

Several task-based notations have inspired our work. We reviewed the ones dedicated to the specification of multi-user interactive systems. In the field of group work task analysis, there exist several notations dedicated to the description of group activities such as GTA [32], MAD [26], MABTA [13] or CTT [17]. Some of these notations, mostly CTT and CTT-based notations such as CIAN [15], TaskMODL [29] or UserView [20], are used to design and specify the user interface. We analyse these notations in light of our two requirements exposed in the introduction section: capacity to support the specification of (1) cooperative and collaborative activities (2) multimodal interaction.

First the notation should support the specification of both kinds of group work task: collaborative and cooperative. Dillenbourg et al. [5] define the two types of tasks as follows:

> *"Collaboration" is distinguished from "cooperation" in that cooperative work "... is accomplished by the division of labor among participants, as an activity where each person is responsible for a portion of the problem solving...", whereas collaboration involves the "... mutual engagement of participants in a coordinated effort to solve the problem together."*

For multi-user multimodal interaction, this requirement is important because cooperative versus collaborative tasks lead to design of different concrete multimodal UIs. Indeed, cooperative tasks imply a predefined task allocation to business roles (division of labor). At the concrete UI level, it implies that, for each business role, the designer assigns a predefined set of modalities and therefore decides at design time which interaction modalities should be used. In contrast, for collaborative activities, task allocation is performed at runtime. Consequently, when the designer specifies the concrete UI, s/he cannot assign modalities to each business role. Thus, GTA, MAD, MABTA, CTT support only cooperative tasks. Furthermore, considering CTT, a group work task is described with a cooperative task tree and a set of individual task trees for each involved business role. Only CIAN and CUA explicitly identify the two types of tasks, cooperative and collaborative, in addition to the existing types of tasks: user, system, interactive. Nevertheless CIAN and CUA focus on abstract interaction only.

The second requirement is to support the description of concrete multimodal interaction. Existing notations mainly focus on abstract UI: As stated in [16], the notations in particular any CTT-based notations such as CIAN [15], enable the specification of abstract UI and may serve as input for designing a concrete/final UI. Nevertheless we note that GTA is based on (N)UAN [8] which is well suited

[1] Business role: Representing role in the organization

to specify concrete interaction in a textual/tabular form while MABTA makes references to a graphical sketch of the UI.

A notation should support a seamless transition from the specification of the abstract UI to the specification of the concrete multimodal UI and vice-versa. Moreover multimodal interaction requires the specification of temporal constraints between modalities [34]. Temporal operators such as the ones in CTT-based notations (e.g., sequence, parallelism, interleaving) cannot express such qualitative temporal constraints at a fine grain. For example, we consider a multimodal interaction for navigating in a map; a user may say ”here”, using voice recognition, while pointing to the map. The temporal constraints can be defined by the designer in several ways: For instance (1) the user must first speak and, when finished, point to the map; or (2) the user must start to point to the map and, after say “here”, and then finally release her/his finger. For the first example, modalities are used in a sequential way. For the second example, based on Allen relationships [2] there is a temporal coincidence between modalities as defined in [34]: Indeed the speech modality is used in the context of the direct-manipulation modality. This is a particular case of parallelism, not covered by CTT operators.

Motivated by the fact that existing notations only partly satisfy the two above requirements for specifying multi-user multimodal interaction, we define the COMM (Collaborative and MultiModal) notation, an extension of the CTT notation. We chose to reuse CTT because it is a popular and widely used notation for UI specification. Therefore usual concepts, such as business role, object, agent, task or event, are included in the COMM notation. In the following sections, we first describe the key features of the COMM notation before presenting the e-COMM editor that supports the COMM notation.

COMM NOTATION

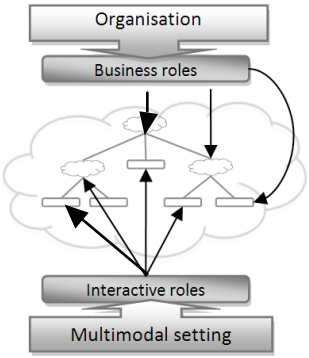


Figure 2: Business Role / Interactive Role.

The COMM notation extends CTT by introducing two concepts, namely **interactive role** and **modal task,** and by considering the five Allen relationships as temporal operators applied to concrete tasks.

- An **interactive role** differs from the usual concept of (business) role in the CSCW literature. As schematized in Figure 2, an interactive role identifies users that interact with the system using modalities, while a business role denotes a role in an organisation such as a technician, an operator, etc.
- A **modal task** is an additional type of task corresponding to the concrete interaction level: A modal task denotes an atomic user action performed on a physical device.
- Allen relationships are used to refine existing temporal operators applied to tasks in order to specify multimodal concrete interaction. Since it is a refinement of CTT temporal operators for abstract interaction, COMM supports a seamless transition from abstract UI to concrete multimodal UI and vice-versa.

We detail these features of the COMM notation by considering an existing multi-user multimodal interactive system presented in [30] as an illustrative example: co-located players are involved in a team play by interacting with a multitouch table and by issuing voice commands. The game is to play Warcraft: two players cooperate to control troop units in order to win the game.

Cooperation vs collaboration

Let us first consider the following cooperative multimodal task implemented in the Warcraft multi-user system [30]: moving a group of troop units. This is a cooperative task since two business roles are clearly identified: the first business role, named *sergeant*, has the responsibility of grouping troop units while the second business role, named *chief*, has to move the group to the newly specified location. As shown in Figure 3, this cooperative task is represented with CTT by two individual interactive task trees, one per business role, and one cooperative task tree. Using COMM, as shown in Figure 4(a), the tasks are represented using a unique task tree: a label decorates the root node to indicate which business roles are involved (i.e., Chief, Sergeant). Similarly, the interactive task is decorated with labels that indicate which business role (i.e., Chief or Sergeant) must perform the concrete task.

Let us now envision the following collaborative task, based on an existing task available in Warcraft, and inspired from [30]: create a unit such as a knight. For performing this task, the first user selects the mode [barrack] (i.e., a building dedicated to the creation of units) and then selects the kind of units to be created; the second user has to confirm the creation of the unit. Such a task cannot be described with CTT since it only supports the description of cooperative tasks. Indeed specifying the constraint that at least two users involved in this collaborative task must interact with different modalities will be complex using existing notations. This will lead to a non-compact and therefore not easily readable specification.

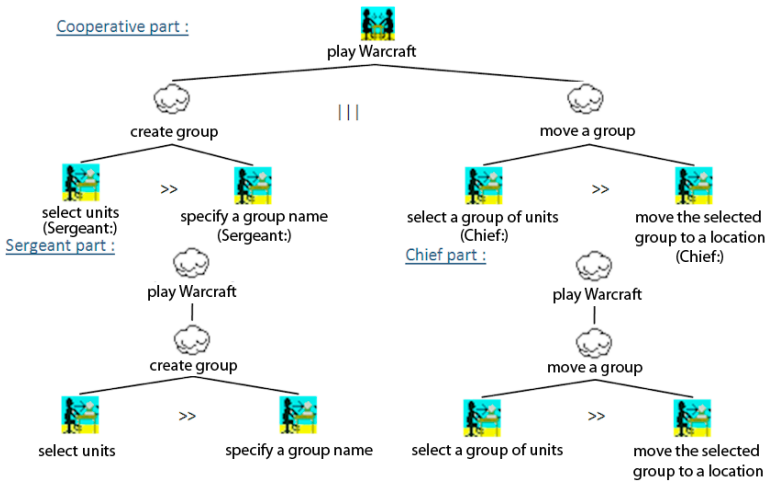


Figure 3: CTT cooperative task.

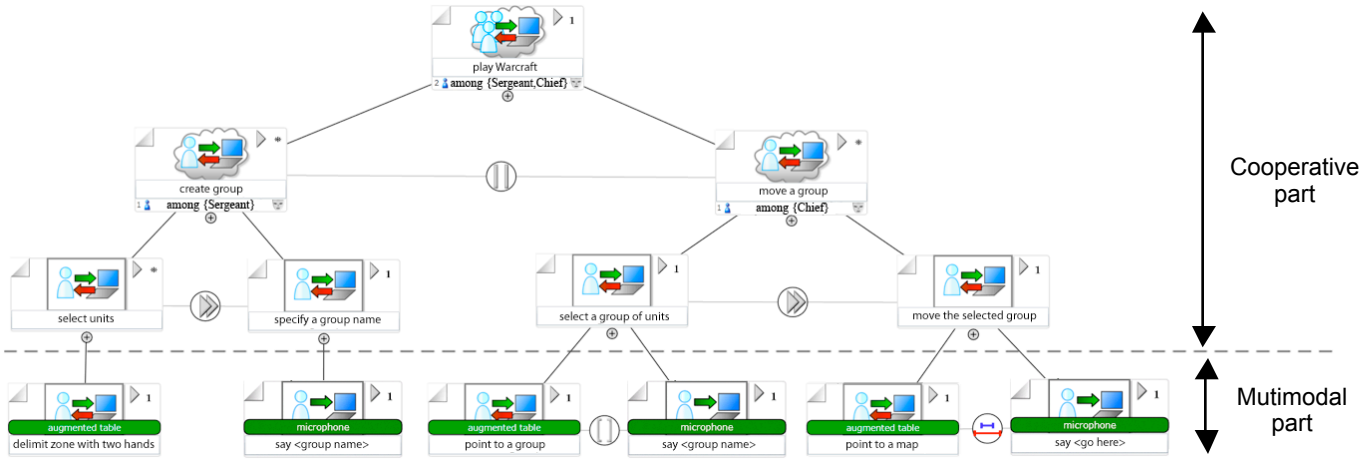


Figure 4 : (a) COMM cooperative task.

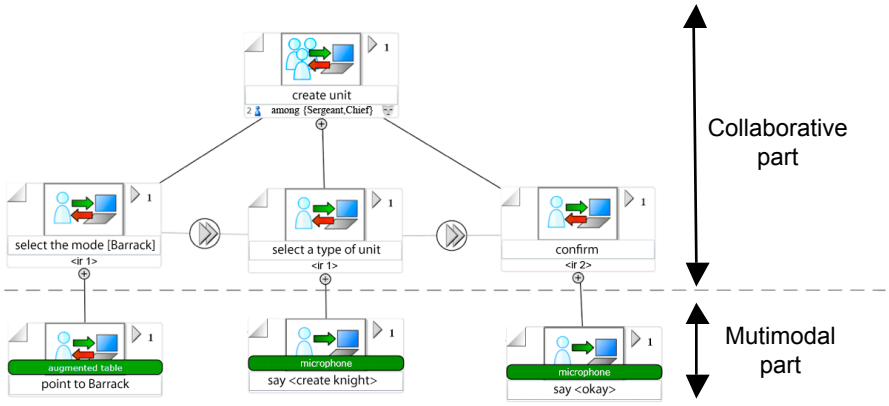


Figure 4: (b) COMM collaborative task.

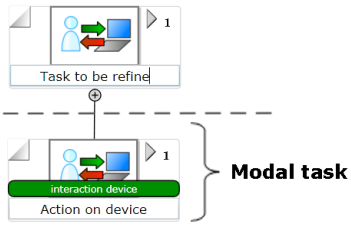


Figure 5: Modal task.

Using COMM, as shown in Figure 4(b), we use the concept of **interactive role** (i.e., <ir1> and <ir2> in Figure 4(b)). An interactive role is associated with each elementary task that composes the unit creation task. We associate the role <ir1> to the two first subtasks and <ir2> to the subtask of confirming the unit creation. This implies that the same role should perform the two first subtasks (specify the unit creation) while the third subtask (confirm the unit creation) should be performed by a different role. Furthermore this specifies that both interactive roles are instantiated at runtime and should be performed by any role (sergeant or chief). To fully describe the collaborative concrete task, the subtasks labelled by interactive roles must then be described in terms of interaction modalities: in the example of Figure 4(b), the two modalities - direct manipulation on the multitouch table and speech commands - for <ir1> and speech commands for <ir2>. For clarity, due to a limited space, we have only represented input modalities.

Interactive roles allow us to specify how tasks are dynamically allocated to business roles and how such tasks can be performed using multiple modalities (i.e., direct manipulation and speech). An interactive role is an ephemeral role played by an actor, dynamically assigned to a business role until the task is completed. As shown in Figure 2, while a business role rules the division of labor, which corresponds to a top-down approach (abstract to concrete), an interactive role is a means to specify how group activities are influenced by user interaction constraints, which corresponds to a bottom-up approach (concrete to abstract). In other words, business and interactive roles make explicit the tension between group work activity requirements and concrete interaction requirements.

Specifying concrete multimodal interaction

In order to fully specify the concrete multimodal interaction, elementary tasks (leaves of the task tree) must be refined in order to describe which modalities are used and how the concrete interaction is performed. To do so, we decompose an elementary task as a composition of **modal tasks** (or monomodal tasks). Based on the definition of an interaction modality as the coupling of a physical device with an interaction language [18], an atomic task is defined as an association of an interactive physical device with an atomic user action that is covered by the interaction language.

In the COMM notation, as shown in Figure 5, a modal task is symbolized by two overlapping rounded rectangles (white and green), each containing a textual label. The first label describes the interaction device and the second one describes the user action (interaction language) performed on the device. For example, the elementary concrete subtask *select the mode [barrack]* of Figure 4(b) is made of the modal task that corresponds to the modality, put her/his finger on the touch table. This elementary concrete subtask is defined by the modal task (<Augmented table, point to barrack>). Another choice of modalities and in particular the choice of devices (for instance a mouse) could lead one to define different modal tasks for each of the subtasks.

A task performed using multiple modalities is described as a combination of modal tasks. This is the case of the task *create unit* in Figure 4(b). When multiple modalities are used at the same time, the temporal constraints between the modalities must be specified. This temporal constraint drives the fusion mechanism in order to detect when fusion must be performed. We consider the task of moving troop units of Figure 4(a). While one user (*sergeant*) can group units by sliding her/his hands on the touch table (left/right), another user (*chief*) can move another group in parallel. To do so the user (*chief*) will select an existing group of units and will then point to the new location on the table while saying <go here>. In the latter case, the temporal constraint that links these two concrete subtasks is specified by a *Coincidence* operator [34] inherited from the Allen relationships [2]: Indeed the speech modality is used in the context of the direct-manipulation modality. Such an operator [34] is used to specify a temporal constraint between modal tasks at a fine grain of description, while we use the parallel operator of CTT to specify temporal relationships between non-modal tasks as in Figure 4(b). We therefore keep CTT operators (i.e., parallelism and sequence) to describe the relationships between tasks at a higher level of abstraction.

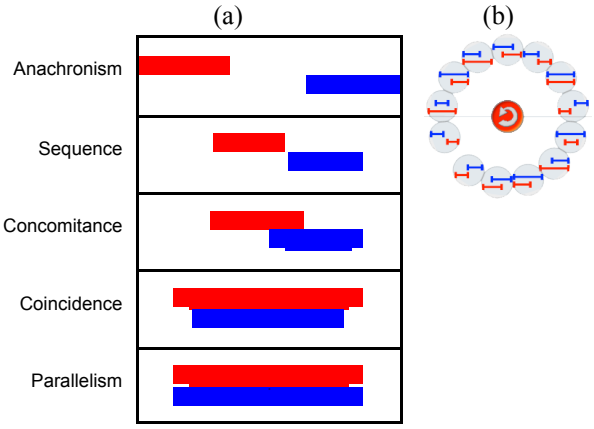


Figure 6: (a) Temporal combinations between modalities. (b) Palette of temporal combinations in the e-COMM editor.

To specify the temporal combinations of modal tasks, we use the five temporal operators [34] described in Figure 6(a). Figure 6(b) shows how the temporal relationships can be specified between two modal tasks using the e-COMM editor. In Figure 6(a), each line corresponds to the time interval in which a modality is used. Sequential and parallel combinations are used in CTT. We add here three new temporal combinations described in [34]. Two modalities are combined anachronously if there is a temporal gap between their usage. Two modalities are concomitant when one modality replaces another one with a time interval during which the two modalities coexist. The coincidence of two modalities is when one modality is only used in the context of another one. This is the case of the example of Figure 4(a), when speech modality is used in the context of

the direct-manipulation modality. The two CTT operators (i.e., parallelism and sequence) are then refined by one of the five temporal relationships defined in [34]. This enables us to ensure continuity between the description of the abstract UI, independent of the modalities and the description of the concrete multimodal UI.

Other elements of the COMM notation

In comparison with CTT, additional features are included in the COMM notation. In particular we consider new types of tasks:

- *User interactive* task (Figure 7-5): A *User interactive* task implies that interaction occurs through a combination of input and output modalities that are described at the level of the **modal tasks** (Figure 7-6). Each user interactive task can be decorated with a role (business role or interactive role). For example in Figure 7-4, the user interactive task is decorated by an interactive role <ir 1> which means that this task can be exclusively performed by the *Pilot* or the *Payload operator* (see our example of UAV command post).
  We also consider the *Action*, *Presentation* and *System* types of tasks. The first one is based on input modalities only without feedback, the second one is based on output modalities only. The last one denotes an internal computation without feedback.
- *Group interactive* task (Figure 7-1): This type of task includes the *Cooperation* task of CTT. But, as explained above, a group task may be cooperative or collaborative. This type of task implies multiple *User interactive* tasks assigned to different roles. A *Group interactive* task is decorated (Figure 7-2) with the list of business roles that could perform the task and with the minimal number of users required to perform the task; this number may be greater than the number of roles: for example, a task may be performed by two pilots and one payload operator (see our example of UAV command post).
- *Group task*: This type of task extends the *User* task of CTT. A *User* task denotes internal and cognitive action by one user in CTT. We generalize it to *Group* task that denotes a decision/discussion by the group without interacting with the system.

Finally, and in addition to the refinement of the CTT temporal operators described previously, we also extend the iteration operator of CTT. As shown in Figure 7-3, the iteration operation, represented by a *, is further detailed in order to specify how the iteration occurs. The iteration operation is preceded by a || or a |> symbol which means that the iteration can be performed in parallel or sequentially.

e-COMM: COMM EDITOR

As highlighted by van der Veer et al [32] and by Molina et al. [16], a notation is fully usable if it exists as a tool for the designer that supports the notation. For instance, we may say that CTTE (CTT editor) has contributed to the success of CTT. Towards this goal, we have developed the e-COMM application that supports the COMM notation. This application is implemented as an **on-line editor** (iihm.imag.fr/demo/ecomm) developed in C# and based on the Microsoft SilverLight technology. The source code will be publicly available and released under the GPL Licence. The e-COMM graphical user interface (Figure 7) emphasizes direct manipulation of the COMM concepts in order to avoid forms, menus or toolbars as much as possible.

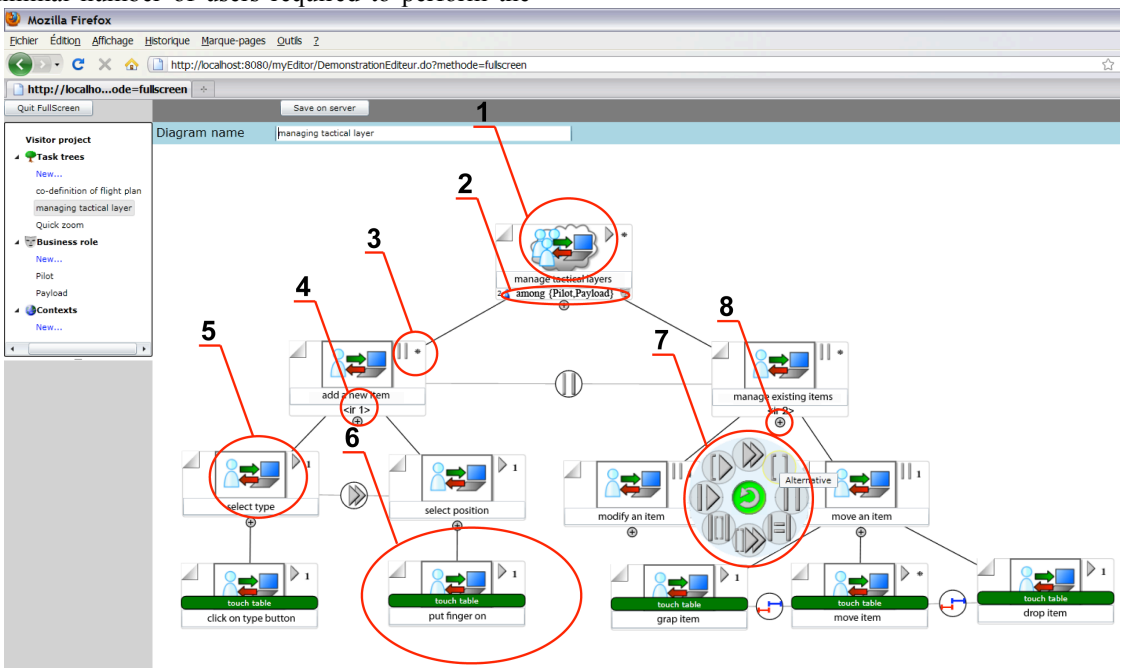


Figure 7: Screenshot of the e-COMM editor.

Indeed, we think that the plethora of filling form popup windows makes a tool less usable and does not encourage focus on the specification under development. Using e-COMM, when a new task tree is created, there is only one node, the root node. Clicking on the (+) symbol (Figure 7-8) leads to the creation of a new subtask. By default, the sequential operator is selected to link the newly created subtasks. The operator can be modified by clicking on the operator symbol and selecting another one: to do when clicking on the operator, the operator wheel (Figure 7-7) appears and shows the available operators. The same interaction technique is used to change the attributes of a task: name, iteration operator and type. Multiple task types are available such as the ones available in CTT and the multiuser task types. Furthermore in order to efficiently navigate in a large task tree the zoom (in/out) action is controlled by the mouse wheel. It is then easy and smooth to focus on a particular part of the tree or to have a global view of the tree. The resulting COMM task tree specification can be saved in different formats including XML, pdf and image formats in order to be easily included in any specification document. In the following section, the trees shown in Figure 8 and 9 have been defined using the e-COMM editor.

CONCRETE EXAMPLE: UAV COMMAND POST

In the context of a French Army Research Department project, in partnership with the firms BERTIN, SAGEM, EADS and PY Automation, the COMM notation is used to specify military command posts for the control of two unmanned aerial vehicles (UAV) by two operators (Figure 1).

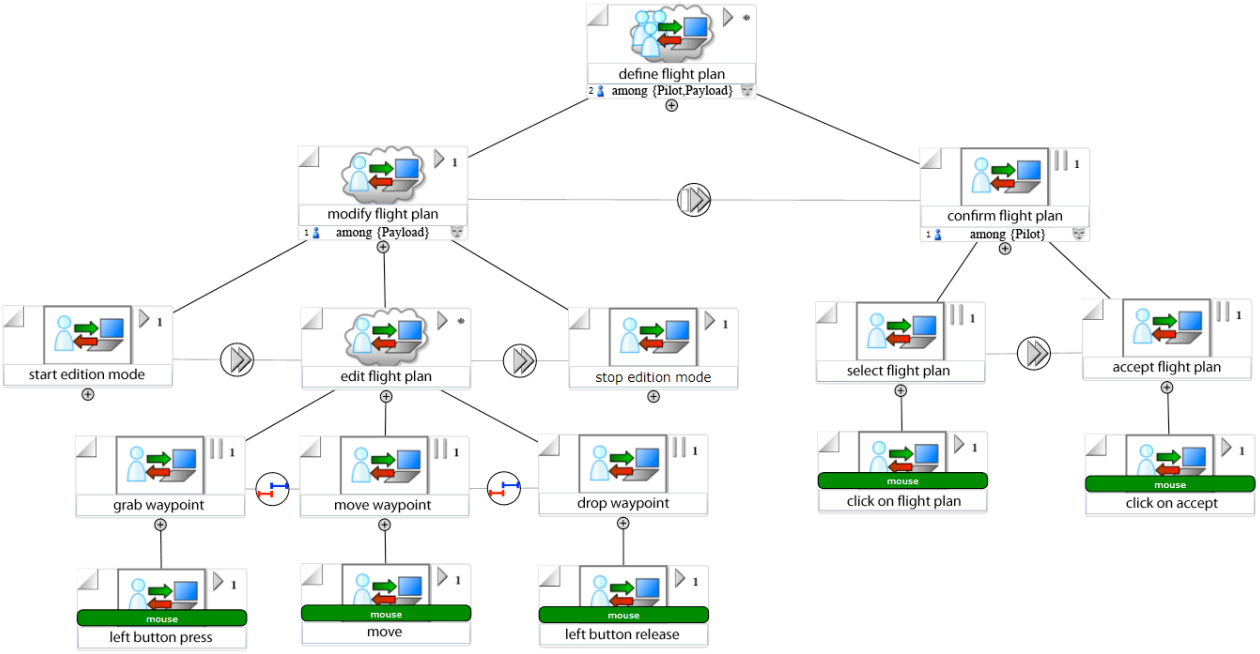


Figure 8: Cooperative task for the modification of a flight plan.

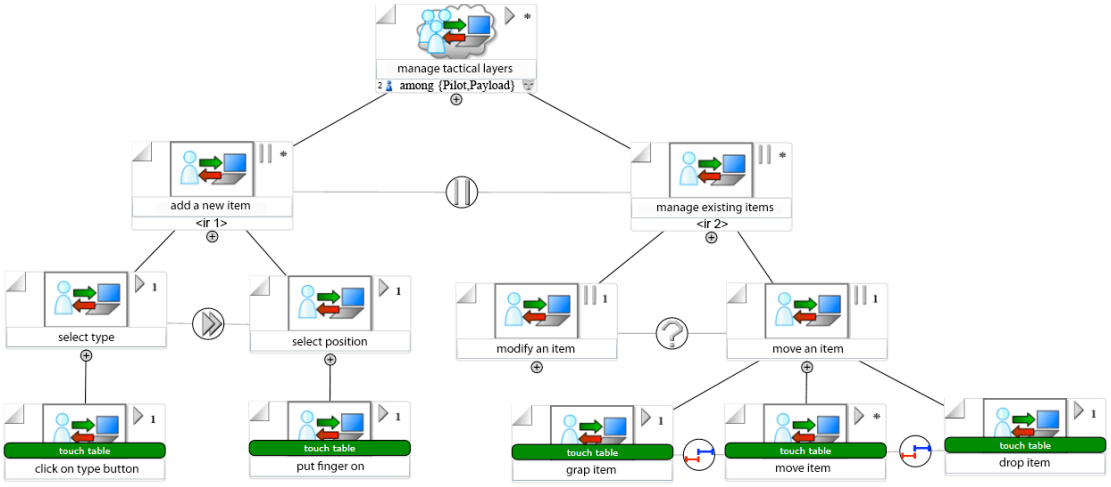


Figure 9: Collaborative task for managing tactical layers.

In this project we adopted an iterative design of the multi-user multimodal interaction to envision the future command posts, and we illustrate the COMM notation by some specifications done for the first iteration using the e-COMM editor. Two operators are involved in the control of UAVs. The first operator has to control the UAV (navigation, guidance, speed, etc). The second one controls embedded devices (payload) such as an infrared camera or a 360º degree camera. In the following, we denote the first operator by the Pilot role and the second one by the Payload operator role. We present three of the specified tasks along with the designed multimodal interaction: (1) to modify a flight plan dynamically and cooperatively; (2) to collaboratively manage tactical layers; (3) to control a multimodal zoom.

Cooperative modification of a flight plan

We consider the first interactive task. During the flight, if required, the flight plan may be modified. Indeed, the Payload operator may observe an event, such as enemy troop movements along the trajectory, detected by the embedded devices, which implies redefining of the flight plan for safety reasons. A flight plan is defined as a set of waypoints linked with straight lines. Then, the Payload operator interactively defines a new trajectory by selecting and moving existing waypoints on a map. When finished, the modified flight plan is submitted to the Pilot. He/she may accept or not the new flight plan. If the new plan is not accepted, the current flight plan is still active and the new plan is cancelled. If the new plan is accepted, the effective UAV trajectory is modified in order to follow the new flight plan. As shown in Figure 8, we describe this interaction by a cooperative task. Indeed, there is a clear separation of work between the two roles. The cooperative task is composed of two individual sub tasks. The first one is performed by the Payload operator: the main interaction is the composition of three modal tasks (selecting a waypoint, moving the waypoint and dropping the waypoint on a map) that must be executed sequentially. The modal tasks are performed using a mouse. When the operator decides to terminate the task, it triggers the individual task *confirm flight plan* with an object as a parameter, an instance of the flight plan class. Then, the Pilot must first select the new flight plan and validate or not the modification.

Collaborative management of tactical layers

For the second interactive task, we consider a set of stacked layers that contains geographical or military information represented on a map such as tanks, buildings or landmarks. For simplicity, we consider that there is only one layer that contains items representing military vehicles or buildings. The content of the layer can be modified by both roles (Payload operator and Pilot). The information contained in these layers are used by the system to compute new trajectories as a function of the environmental context such as enemy vehicle movements. As shown in Figure 9, the two main interactive subtasks are: *add a new item* and *manage an existing item*. Both subtasks may be executed simultaneously by both operators (the task is decorated with the label *among {Pilot, Payload}*) but the execution is exclusive. Indeed, the first subtask must be performed only by the first interactive role `<ir1>` while the second one must be performed by the interactive role `<ir2>`. It means that if the Pilot decides to perform the task *add a new item*, the interactive role `<ir1>` is instantiated with the Pilot role. It also means that if the Payload operator role decides to perform the task *Manage tactical layers*, only the task *manage existing items* is available until the Pilot has terminated her/his task. The corresponding modal tasks are based on finger-based selection of buttons and items displayed on a touch table that offers user identification (e.g., DiamondTouch). In this design, the two users can work in parallel but at a given time, they cannot perform the same task. One user can create a new item by (1) selecting the type of item in a palette and then (2) selecting a location on the map, while the second user is moving an existing item. But the two users cannot work in parallel on the same task: for instance one user cannot select the item to be created in the palette while the other one is selecting the location of the new item on the map. Such a design solution is concisely and precisely specified using COMM by using interactive roles.

Multimodal zoom

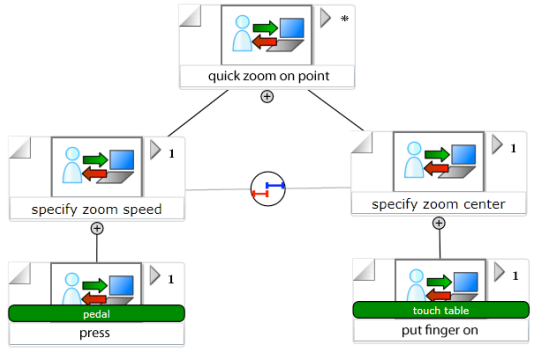


Figure 10: Concrete multimodal zoom task.

The last example focuses on the task of zooming in/out in the map described in the previous section. As shown in Figure 10, the task is specified as a combination of two elementary tasks that involve different modal tasks: one modality is based on a pedal which must be pressed to control the zoom speed, while the other one is based on the touch table to specify the zoom center. The modality based on the pedal is active only if the zoom center is specified, in other words only if the user is touching the touch table using her/his finger. Such a temporal constraint is expressed by a Coincidence [34] (C label) operator. Since no interactive role is specified, the two modalities can be used in parallel by the two operators, one using the pedal while the other one is specified with her/his finger the zoom center.

CONCLUSION AND FUTURE WORK

We have presented the (Collaborative and MultiModal) COMM notation, an extension of CTT for specifying multi-user multimodal interaction. The salient features of the notation are: (1) the concepts of **interactive role** and (2) **modal task**, (3) a refinement of the temporal operators applied to tasks using the Allen relationships. CTT temporal operators are used for abstract tasks and are translated in terms of the five relationships between modalities identified in [34] and based on Allen relationships, when describing temporal constraints between modality dependent tasks, i.e, modal tasks. This introduces a smooth continuity between the specification of the abstract UI and that of the concrete multimodal UI.

Furthermore e-COMM is a fully operational on-line editor that is used in a military project involving industrial partners (BERTIN, EADS, SAGEM and PY automation). The workflow specification is also implemented in the e-COMM editor but not described in this paper.

As future work, we plan to further evaluate the COMM notation and its editor.

Empirical validation, i.e. by experience, is the first type of validation we assessed. The model has been used to specify existing multi-user multimodal systems such as the ones described in [1, 30]. Furthermore we are using the notation for specifying new multi-user multimodal systems as part of the on-going military project. The iterative approach adopted in this project provides a means to compare the result of the design step specified using COMM with the implementation done by another partner, and, then, to evaluate the precision and power of expression of the notation. Moreover in the project, the COMM specifications are resulting from a joint effort from different partners, not only the authors of the notation.

Conceptual validation is the second kind of validation we started to assess. To do so we consider the Notational Dimensions of the Cognitive Dimensions of Notations framework [7]. For instance *low viscosity* (resistance to change in the specification) can be studied when a designer changes a modality. In addition to changing the corresponding modal task, s/he only has to adapt the neighboring modal tasks, only if they are parts of a multimodal task (e.g., changing the temporal constraints). Furthermore interactive roles may be impacted by changing a modality. We plan to further evaluate the COMM notation based on the Cognitive Dimensions of Notations analysis framework and in particular the *Low premature commitment* and the *high visibility* dimensions by conducting experiments with 16 master students (Master Course on Groupware) using the e-COMM editor.

ACKNOWLEDGMENTS

The work presented in the article is partly funded by DGA (French Army Research Dept.) under contract PEA FH/PA. Special thanks to G. Serghiou for reviewing the article.

REFERENCES

1. Ajaj, R., Vernier, F., Jacquemin, C., Navigation Modes For Combined Table/Screen 3D Scene Rendering, in *Proceedings of ITS'09* (Banf, Canada, November 2009), ACM Press, 141-148
2. Allen, J. Maintaining Knowledge about Temporal Intervals. *Comm. of the ACM*, 1983, 26(11), 832-843, ACM Press.
3. Brown, B., MacColl, I., Chalmers, M., Galani, A., Randell, C., and Steed, A., Lessons from the lighthouse: collaboration in a shared mixed reality system, in *Proceedings of CHI'03* (Ft. Lauderdale FL, April 2003), ACM Press, 577-584.
4. Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. QuickSet: multimodal interaction for distributed applications, in *Proceedings of MULTIMEDIA'97* (Seattle WA, November 1997), ACM Press, 31-40.
5. Dillenbourg, P., Baker, M., Blaye, A., and O'Malley, C. The evolution of research on collaborative learning. *Learning in Humans and Machine: Towards an interdisciplinary learning science*, 1996, 189-211, Elsevier.
6. Etter, R., and Röcker, C., A Tangible User Interface for Multi-User Awareness Systems, in *Proceedings of TEI'07* (Baton Rouge LA, February 2007), ACM Press, 11-12.
7. Green, T., Instructions and descriptions: some cognitive aspects of programming and similar activities, in *Proceedings of AVI'00* (Palermo, Italy, 2000), ACM Press, 21-28.
8. Hix, D., and Hartson, H. *Developing User Interfaces: Ensuring Usability Through Product & Process*, 1998, Wiley & Sons.
9. Jacob, R. J. K., A Specification Language for Direct Manipulation User Interface. *ACM Transactions on Graphics*, 1986, 5(4), ACM Press, 283-317.
10. Jourde, F., Laurillau, Y., Morán, A., and Nigay, L., Towards Specifying Multimodal Collaborative User Interfaces: A Comparison of Collaboration Notations, in *Proceedings of DSV-IS'08* (Kingston, Canana, July 2008), Springer, 281-286.
11. Kieras, D. Task Analysis and the Design of Functionality. *The Computer Science and Engineering Handbook (2nd Ed)*, 1997, 46-1-25, Chapman & Hall.
12. Kraut, R.E., Fussel, S.R., and Siegel, J. Visual Information as a Conversational Resource in Collaborative Physical Tasks. *Human-Computer Interaction Journal*, 2003, 18(1), 13-49, Lawrence Erlbaum.
13. Lim, Y.K., Task models for groupware and multitasking: Multiple aspect based task analysis (MABTA) for user requirements gathering in highly-

contextualized interactive system design, *in Proceedings of TAMODIA'04* (Prague, Czech Republic, 2004), ACM Press, 7-15.

14. McDermid, J., and Ripkin, K. *Life Cycle Support in the ADA environment*, 1984, Cambridge University Press.
15. Molina, A.I., Redondo, M.A., and Ortega, M., A conceptual and methodological framework for modelling interactive groupware applications, in *Proceedings of CRIWG'06* (Medina del Campo, Spain, September 2006), Springer, 413–420.
16. Molina, A.I., Redondo, M.A., and Ortega, M. A Review of Notations for Conceptual Modeling of Groupware Systems. *New Trends on Human–Computer Interaction*, 2009, 75-86, Springer.
17. Mori, G., Paternó, F.,Santoro, C., CTTE : Support for Developing and Analyzing Task Models for Interactive System Design. *TOSE Journal*, 2002, 28(8), 797-813, IEEE Press.
18. Nigay, L., Coutaz, J., A Generic Platform for Addressing the Multimodal Challenge, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 98-105.
19. Pauchet, A., Colfedy, F., Lefebvre, L., Picard, S., Bouguet, A., Perron, L., Guerin, J., Corvaisier, D., and Collobert, M., Mutual Awareness in Collocated and Distant Collaborative Tasks Using Shared Interfaces, in *Proceedings of INTERACT'07* (Rio de Janeiro, Brazil, September 2007), Springer, 59-73.
20. Penichet, V.M.R., Paternò, F., Gallud, J.A., and Lozano, Maria D., Collaborative Social Structures and Task Modelling Integration, in *Proceedings of DSV-IS'06* (Dublin, Ireland, July 2006), ACM Press, 66-80.
21. Pinelle, D., Gutwin, C., and Greenberg, S. Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. Transactions on Computer-Human Interaction (TOCHI), 2003, 10(4), 281-311, ACM Press.
22. Piper, A., O'Brien, E., Morris, M.R., and Winograd, T., SIDES: a Cooperative Tabletop Computer Game for Social Skills Development, in *Proceedings of CSCW'2006* (Baff, Alberta, Canada, November 2006), ACM Press, 1-10.
23. Royce, W., Managing the development of large software systems, in *Proceedings of WESCON'70*, IEEE Press, 1-9.
24. Rubart, J., and Dawabi, P. Shared Data modelling with UML-G. *International Journal of Computer Applications in Technology*, 2004, 19(3-4), Inderscience, 231-243.
25. Sallnäs, E., Rassmus-Gröhn, K., and Sjöström, C., Supporting Presence in Collaborative Environments by Haptic Force Feedback. *ACM TOCHI*, 2001, 7(4), 461-476, ACM Press.
26. Scapin, D. L., and Pierret-Golbreich, C. Towards a method for task description. *Work with Display Unit*, 1990, 371-380, Elsevier.
27. Serrano, M., Juras, D., Nigay, L. A Three-dimensional Characterization Space of Software Components for Rapidly Developing Multimodal Interfaces, in *Proceedings of ICMI'08* (Chania Crete, Greece, October 2008), ACM Press, 149-156.
28. Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S, Collaborative Coupling over TableTop Displays, in *Proceedings of ACM CHI'06* (Montréal, Québec, Canada, April 2006), ACM Press, 1181-1190.
29. Treatteberg, H. Model-based User Interface Design. *PhD Thesis*, 2002, Norwegian University of Science and Technology.
30. Tse, E., Shen, C., Greenberg, S., and Forlines, C., Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop, in *Proceedings of AVI'06* (Venezia, Italy, May 2006), ACM Press, 336-343.
31. Tse, E., Greenberg, S., Shen, C., Forlines, C., and Kodama, R., Exploring True Multi-User Multimodal Interaction over a Digital Table, in *Proceedings of DIS'08* (Cape Town, South Africa, February 2008), ACM Press, 109-118.
32. Van der Veer, G., Welie, M., Task Based Groupware Design: Putting Theory into Practice, in *Proceedings of DIS'00* (New York NY, August 2000), ACM Press, 326-337.
33. Van der Veer, G., Welie, M, Chisalita, C., Introduction to Groupware Task Analysis, in *Proceedings of TAMODIA'02* (Bucarest, Romania, July 2002), ACM SIGCHI, 32-39.
34. Vernier, F., Nigay, L., A Framework for the Combination and Characterization of Output Modalities, in *Proceedings of DSV-IS'00* (Limerick, Ireland, June 2000), Springer, 32-48.
35. Voida, A., and Mynatt, E.D., Challenges in the Analysis of Multimodal Messaging, in *Proceedings of CSCW'2006* (Baff, Alberta, Canada, November 2006), ACM Press, 427-430.

Towards Specifying Multimodal Collaborative User Interfaces: A Comparison of Collaboration Notations

Frédéric Jourde[1] , Yann Laurillau[1], Alberto Moran[2], Laurence Nigay[1]

[1] Grenoble Informatics Laboratory (LIG)
38042, Grenoble, France
{fjourde,laurilla,nigay}@imag.fr

[2] Facultad de Ciencias, UABC,
Ensenada, Mexico
alberto_moran@uabc.mx

Abstract. Interactive systems including multiple interaction devices and surfaces for supporting the collaboration of a group of co-located users are increasingly common in various domains. Nevertheless few collaborative and multimodal interface specification notations are proposed. As a first step towards a notation for specifying a design solution prior to its software design and development, we adopt an empirical approach. We applied and compared four existing notations for collaborative systems by considering a case study, namely, a system for supporting informal co-located collaboration in hospital work. This paper reports the conclusions from this empirical comparison.

Keywords: CSCW, multimodality, multi-devices, specification notation.

1 Introduction

The multimodal domain, including multi-surface and multi-device areas, has expanded rapidly. Significant achievements have been made in terms of both modalities and multimodal applications especially for Computer-Supported Cooperative Work such as co-located collaboration in a smart room. Real collaborative multimodal systems are now built in various domains [12] including the medical one [10]. Moving away from research prototypes, we now observe the need for specifying such interactive systems especially in the context of industrial projects. In this article, we address this problem of specification of multimodal collaborative User Interfaces (UI).

Specifying user interfaces is a well-established discipline and various notations have been proposed for specifying the tasks, the dialog elements, the sequences of interaction, concrete UI elements, dynamics of group behavior and so on. Such a variety of notations both in terms of their descriptive qualities, their syntactic structures and the amount of support that they offer according to the development phases has already been highlighted ten years ago in [5]. In [9], the review of notations for interaction design underlines that the most common interaction representational needs are covered by four models: task, domain, abstract and concrete UI. Many of these notations are dedicated to single user WIMP interfaces and we are interested in studying the proposed extensions of these notations and more

recent notations dedicated to collaborative and multimodal UI. To do so, our approach for studying existing notations for specifying multimodal collaborative UI is in the first instance empirical: we start from existing collaboration notations and we apply them for specifying a case study: a system for supporting informal co-located collaboration in hospital work.

2 Empirical Comparison of Collaboration Notations

The relationships between collaborative and multimodal interaction open a vast world of possibilities that has not been systemically explored in terms of specification notations. We aim at going further than considering multimodal aspects such as the CARE properties [2] for the concrete UI and collaborative aspects for the abstract UI. In our empirical comparison, we first focus on existing notations for specifying collaborative UI. Since we are interested in also modeling multimodal interaction, while studying collaborative UI specification notations, we also examine the power of expression of the notations for specifying concrete UI. A complementary approach to ours would be to start from multimodal UI specification notations.

2.1 Scope of the Comparative Study

In [1], three dimensions for evaluating an interaction model are described: descriptive power (i.e., ability to describe a UI), evaluative power (i.e., ability to help assess multiple design alternatives) and generative power (i.e., ability to help designers create new designs). As a starting point for our comparative study, we are focusing on the descriptive power of UI specification notations. Their impact on the design including their evaluative and generative powers will be studied afterwards. Moreover, our study does not aim at evaluating the selected notations that can be studied in light of the criteria identified in [5] and of the notational dimensions of the framework "cognitive dimensions of notations" [4]. Since the selected notations differ in their descriptive qualities, some focusing on collaborative tasks while others on the users' roles and on collaborative situations, our goal is to assess their complementary aspects and their projected ability to specify a multimodal collaborative user interface.

2.2 Rational for the Selected Notations

Our review of existing collaboration notations highlights the fact that the notations such as CTT [11], CUA [4], GTA [15] and MABTA [8], mainly focus on both individual and collaborative tasks. Such notations aim at accommodating several aspects of collaborative work situations into a task specification and thus extend task specification with contextual information. Some notations also focus on other aspects than individual and collaborative tasks such as TKS [7] which focuses on users' knowledge involved in task behavior and UML-G [14] on modeling shared data.

Amongst the existing notations, a first way for selecting the ones to be applied to our case study would be based on the syntactic structure of the notations (i.e., graphical, tabular, textual approaches) as in [6]. This solution was not satisfying since most of the notations imply several types of representations. Moreover since our study focuses on the specification of a collaborative user interface, we did not consider the notations that are not dedicated to interaction tasks and system behaviors, although they may be complementary to the other notations. We therefore exclude TKS. Moreover although CUA is focusing on individual and collaborative tasks in the context of scenarios, its main focus is on modeling the tasks for the needs of groupware evaluation. As a conclusion, we selected four notations, CTT, GTA, MABTA and UML-G that involve different background disciplines. UML-G is an extension of a standard in Software Engineering. CTT is a well established notation for task analysis in Human-Computer Interaction, while GTA and MABTA aims at extending task analysis with elements from Social Sciences (social psychology, sociology) in order to capture key elements of the nature of groupworking.

3 Specifications Based on the Selected Notations

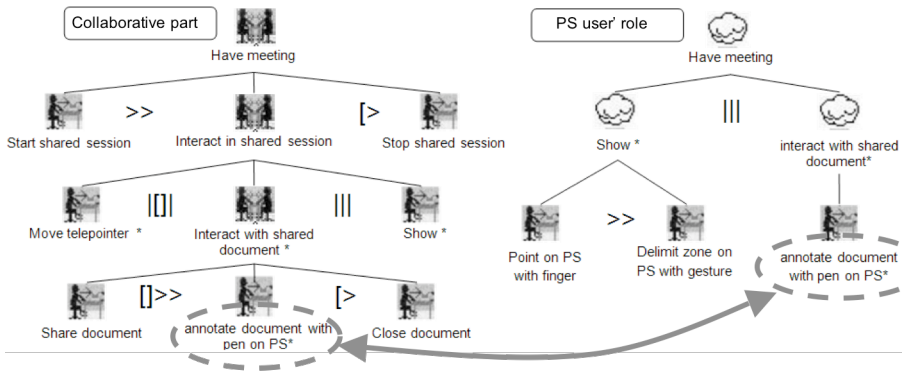


Fig. 1. CTT collaborative and public display (PS user's role) individual task trees.

We apply the four selected notations to specify a collaborative multi-surface [10] that provides a support for informal co-located collaboration by allowing multiple users to study medical documents. As highlighted by the field study described in [10], hospital medical workers including physicians and medical interns are very mobile and need to opportunistically and informally establish co-located collaboration while focusing on a particular patient. Using the system, two physicians can share extracts from a patient medical displayed both on the large screen (i.e., Public Screen) and on the PDA screen (i.e., Private Screen): (i) on the public screen, physicians can only annotate the medical information using a virtual pen ; (ii) on the PDA, a physician can initiate a shared session, select documents to share, edit documents and stop a session. For illustrative purpose, we partly present the CTT specification of our case study

while the complete specifications along the four notations are available at [16]. Applying CTT, collaborative activities are described at a high-level of abstraction through a collaborative task tree; individual task trees (one per role) describe concrete tasks. A collaborative task tree contains collaborative tasks and high-level individual tasks, related to one of the multiple individual task trees. Figure 1 shows the collaborative task tree for our case study and one individual task tree associated with the public screen user's role. The CTT notation includes five types of tasks: system, mental, abstract, individual (user) and collaborative tasks. A collaborative task is an abstract task that must be composed of individual tasks. Relation operators between tasks are inherited from LOTOS. In particular, Figure 1 highlights the coupling between the PDA and PS (|[]| operator) and the document sharing between users ([]>> operator).

Difficulties or limitations identified by applying CTT include the fact that the links between the tasks of different trees are not explicit (i.e., no role is specified for a task) and are only deduced from the task identifiers. Moreover to specify the modalities of a concrete task, the only means is to use the task identifier (e.g., *Annotate document with pen on PS*). In addition, the notation does not provide any means of representing shared objects and of specifying a policy for the sharing. For example, we are able to specify that the telepointer is controlled by the PDA but we are not able to specify that the telepointer can be observed by both roles.

4 Conclusions from the Comparative Study and Future Work

About role specification, the four notations explicitly support user's roles specification in working group. While GTA and MABTA advocate a dedicated representation to roles and relationships between users, we describe roles using UML-G with a class diagram and the involved objects for each role. As opposed to UML-G that focuses on the manipulated objects per role, CTT describes the tasks per role in individual task trees (Figure 1).

About group and individual work specification at abstract level, on the one hand, CTT and MABTA advocate a dedicated representation that combines collaborative tasks, and individual tasks that take part directly in the group work coordination, such as *start shared session* in Figure 1. However, CTT operators refine the MABTA "influence" relation for the case of temporal interdependencies only. As pointed out in [3], in addition to temporal interdependencies related to the activity level, interdependencies are related to the object level and describe the multiple participants' access to the same set of objects. In [13], they define a set of generic mechanics of collaboration as elementary abstract tasks for such coordination issues surrounding how objects are assessed. Such elementary abstract tasks are generic since they are common to a variety of social and organizational work groups. For example one abstract task "Obtain resource" could be part of the CTT and MABTA group work representations.

On the other hand, GTA and UML-G represent together group and individual work. Using GTA, it is possible to annotate each task with the roles and the manipulated objects. As for GTA, with UML-G, group work is implicitly described

within the class diagram by different roles manipulating the same object. Collaborative activities are further described in the UML-G activity diagram that highlights the relationships between the individual tasks over time.

Individual tasks are described hierarchically in CTT, MABTA and GTA for each role. MABTA refines the work group tasks into sub-tasks while maintaining the columns for describing the roles and adding new individual tasks that are not related to the group work. GTA advocates only one representation for both group and individual works. Links between tasks of different roles can be specified by triggered task and hence corresponds to the "influence" relation of MABTA. As opposed to MABTA and GTA, CTT does not explicitly describe the links between the tasks of different roles. This link is deduced from the group work representations that share tasks with the individual task trees (Figure 1). In contrast to the hierarchical refinement approach of CTT, MABTA and GTA, the activity diagram in UML-G shows individual work and interdependencies with respect to time and roles. Moreover only GTA and UML-G enables us to represent task flows respectively in terms of activity diagram and sequence diagram.

Finally, about group and individual work specification at concrete level, CTT and MABTA advocate the same representation for abstract and concrete tasks. The GTA elementary abstract tasks are described using NUAN which enables a precise description of both users' actions, system feedback and dialogue states. For UML-G, concrete tasks can be described by sequence diagrams along with state-transition diagrams. For each object, the users' actions on it as well as its reactions are described. Nevertheless such a specification would be extremely tedious for a complete user interface.

To conclude, by applying four existing notations for specifying a simple groupware where two users are working on a medical image using a PDA and a public display enables us to identify some complementary aspects in the induced representations as well as some missing aspects. We underline three key issues from this empirical study. Firstly, the distinction between group work and individual work (per role) is useful in a specification for describing at different level of detail (i.e., abstract and concrete) a collaborative user interface from its two facets, the group and the users. However a unified representation of group and individual work enables us to depict interdependencies between users with respect to time and roles. Classical hierarchical representations such as CTT are suitable for individual tasks, while group work representations need to include specific aspects of collaboration such as in MABTA where tasks are decorated with concepts from coordination theory. Secondly, temporal relationships between tasks for describing group work are not sufficient: Temporal interdependencies are at the activity level and interdependencies related to the object level are required for describing the multiple users' access to the same set of objects. UML-G focusing on shared objects can be used for describing such interdependencies. Thirdly, the specification of concrete multimodal interaction as concrete tasks involves extending the selected notations dedicated to WIMP user interfaces. For example, it was not possible to explicitly specify the redundancy (one of the CARE properties of multimodality [2]) of the display (PDA and public display) of our case study. Further studies must be done on the description of tightly coupled multimodal interaction (a concrete multimodal group task corresponding to an abstract group task) and on loosely coupled multimodal interaction (concrete

multimodal individual tasks corresponding to abstract individual tasks that define a composed abstract group task).

As further work, we plan to experiment on the complementary usage of the studied notations on another case study, namely a collaborative and multimodal military command post. The focus will be on studying the links between the activity (task) and shared resource (object) aspects and on extending the notations in order to depict multimodal interaction. For multimodal interaction, distinguishing abstract/concrete tasks as well as group/individual tasks allows us to identify: (1) tasks that require tightly coupled multimodal interaction when two users are continuously engaged with the accomplishment of physical actions for realizing a concrete group task. (2) Tasks that require loosely coupled multimodal interaction when two users are performing actions along different modalities for realizing two concrete individual tasks that define an abstract group task. For specifying these two types of multimodal group tasks, one of our research avenues is to study extensions of the ICARE notation [2].

References

1. Beaudouin-Lafon, M.: Designing Interaction, not Interfaces. AVI'00, pp. 15-22, ACM.
2. Bouchet, J., Nigay L., Ganille, T.: ICARE Software Component for Rapidly Developing Multimodal Interfaces. ICMI'04, pp. 251-258, ACM.
3. Ellis, C. A., J. Wainer, J.: A Conceptual Model of Groupware. CSCW'94, pp. 79-8, ACM.
4. Green, T.: Instructions and descriptions: some cognitive aspects of programming and similar activities, AVI'00, pp. 21-28, ACM.
5. Johnson, C.W.: The Namur Principles: Criteria for the Evaluation of User Interface Notations. DSVIS'96, Springer.
6. Johnson, C.W.: The Evaluation of User Interface Design Notations. DSVIS'96, pp. 188-206, Springer.
7. Johnson, H., Hyde, J.: Towards Modeling Individual and Collaborative Construction of Jigsaws Using Task Knowledge Structures. TOCHI, 2003, 10 (4), pp. 339-387, ACM.
8. Lim, Y.K.: Task models for groupware and multitasking: Multiple aspect based task analysis (MABTA) for user requirements gathering in highly-contextualized interactive system design. TAMODIA'04, pp. 7-15, ACM.
9. Markopoulos, P. and Marijnissen, P.: UML as a representation for Interaction Design. OZCHI'2000, pp. 240-249.
10. Mejia, D.A., Morán, A.L, and Favela, J., Supporting Informal Co-located Collaboration in Hospital Work. CRIWG'07, pp. 255-270, Springer.
11. Mori, G., Paterno, F.,Santoro, C.: CTTE : Support for Developing and Analyzing Task Models for Interactive System Design. TOSE, 2002, 28 (8), pp.797-813, IEEE Press.
12. Oviatt, S. et al.: Designing the user interface for multimodal speech and gesture applications. HCI, 2000, 15 (4), pp. 263-322, Taylor & Francis.
13. Pinelle, D., Gutwin, C. Greenberd, S.: Task Analysis for Groupware Usability Evaluation, TOCHI, 2003, 10 (4), pp.281-311, ACM.
14. Rubart, J., Dawabi, P.: Shared data modeling with UML-G. IJCAT, 2004, 19 (3/4), pp. 231-243, Inderscience.
15. Veer, G., Welie, M.: Task Based Groupware Design: Putting Theory into Practice. DIS'00, pp. 326-337, ACM.
16. http://iihm.imag.fr/laurillau/four-notations-comparison.pdf

Supporting museum co-visits using mobile devices

Yann Laurillau, Fabio Paternò

Consiglio Nazionale delle Ricerche (CNR-ISTI),
Via G. Moruzzi 1, 56100 Pisa, Italy
{Yann.Laurillau, Fabio.Paterno}@isti.cnr.it

Abstract. The goal of this work is to provide tools that promote social interactions between visitors through cooperative and educational games. In this paper, we describe how to support collaborative learning in museum visits and show an example application based on mobile palmtop systems. To this end, we have developed a system that is able to support collaborative and independent activities, and offer context-aware content.

1 Introduction

The wide dissemination of mobile technologies, such as cell phones or handheld personal digital assistants (PDAs), offers a good opportunity to get groupware applications out of the laboratories and develop new kinds of groupware applications which are no longer reserved for professionals and desktop computers. Mobile devices are becoming real social media, particularly in terms of communication. Such technologies are means to explore collaborative activities and move groupware applications to public settings, such as museums.

The museum visit is usually perceived as an individual experience. Furthermore, electronic guides or interactive systems in museums are not designed to promote social interaction among visitors. However, the museum experience, according to Falk and Dierking [3], is influenced by the social context, which includes interactions between visitors. In addition, many studies have highlighted the fact that interactions with the exhibit, as well as communication and social interaction between visitors are also key points of a successful learning environment [6] [7].

The research on social interaction and collaboration using new technologies is quite recent. Interest has grown with the evolution of mobile devices. In addition, there has been a change in the design of museum exhibitions in an increasing number of projects: little by little, the museum experience is considered as a collaborative activity and, more and more, museums are designed to support and encourage group interactions.

In this work, our goal is to promote interaction and communication between visitors through cooperative and interactive educational games, based on sharing, and using handheld PDAs. In this context, museum interactive systems are embedded in an electronic companion rather than being static fixtures in the museum. In addition, interactivity is considered at the visit level and not only at the artwork level: visitors are able to pace the visit and interact in the museum according to their desires. Furthermore, educational games are an interesting and entertaining way to initiate and

promote collaboration between visitors. For example, the Ghost ship project [5] shows that playing and exploring artwork may help visitors to initiate collaboration. However, in this project, the Ghost Ship is a single interactive artwork and the approach should be extended to all the artworks in the museum.

In order to obtain a new solution, we have developed and deployed an interactive system, the collaborative extension of the portable Cicero [2], dedicated to supporting the visit of the Marble Museum of Carrara. This system enables communication, sharing and collaboration among visitors, and also offers context-aware and personalized content. In the rest of the paper, we detail the main ideas of this project and provide a short review of related work. We then introduce our approach to support co-visiting in museum environments through PDAs. In the third part, we describe our system, the portable Cicero system.

2 Museum Co-Visiting

Museum co-visiting has been considered in a number of projects. The Sotto Voce project [4], developed at Xerox PARC, is a mobile companion, based on the iPaq technology, that provides audio content of artwork descriptions and acts as an audio media space between visitors, which offers a mean for awareness and sociability. The authors have identified four kinds of activity: (i) *shared listening*, in order to promote interaction and communication between companions; (ii) *independent use*, in order to enable temporarily or entirely the switching off of the shared listening, in particular when visitors do not want to engage social interactions; (iii) *following*, when a companion is in charge of driving, implicitly or explicitly, the tour; (iv) *checking in*, which is a short activity, to maintain and update the shared context.

The City project [1], part of the Equator project, takes place at the lighthouse in Glasgow, a museum dedicated to the work of the designer Mackintosh. The system considers three kinds of technology: (i) a real visit using a PDA; (ii) for the virtual reality visit in a 3D world; (iii) for the Web visit. With this system, visitors are able to share their museum experience visit and navigate jointly through mixed realities: the Web, the virtual and physical reality. Information is provided about each visitor location and orientation. In addition, they may communicate through audio channels. The authors have observed that voice interaction, location and orientation awareness, and mutual visibility are essential to the success of museum co-visiting between remote users.

The Ghost Ship project [5], compared to the previous projects, is more oriented to an artistic experience of the museum co-visiting. The goal of this work is to analyse and consider informal and social interactions between visitors through video interaction recordings. The Ghost Ship installation is a dedicated room of the SOFA exhibition containing a wood painted ship, wooden figures, a simulated desk and an "inside the ship" area. Some of the ship portholes are video portholes which record and show visitor's behaviours and interactions with the ship. In addition, microphones capture visitor comments about their actions and about what they can see on the video portholes. The authors observed that the Ghost Ship helps visitors to break the ice more easily and to play with and explore collaboratively the ship.

Compared to the Equator City project, we consider "physical" visitors moving in the real museum while they consider a mixed visit combining the real museum and a virtual representation of the museum (in a 3D representation or through a Web site). Some of the existing projects consider pure collaborative virtual visits as Web co-visiting such as the Van Gogh museum. The authors of the Equator City project, as in the Sotto Voce project, have noted that information about location and orientation of the companions (*checking in* task) is essential in a cooperative visit in order to maintain group awareness. This point has been considered in our project, as detailed in the next section: visitors are able to check for their companion and are aware about the state of the cooperative game.

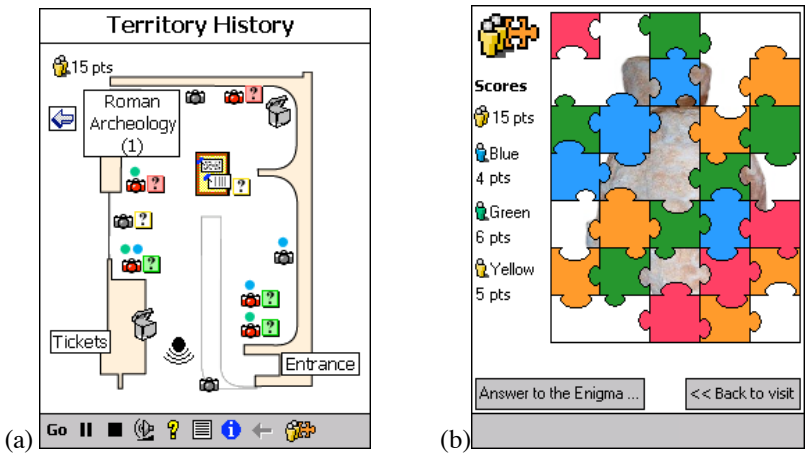
3 The Cicero Project

Cooperative visiting through educational games, as explained in the introduction and highlighted by the authors of the Ghost Ship project, is an interesting way to promote co-visiting and to engage visitors to share their museum experience. In addition, it may also preserve the individual aspects of the museum experience as highlighted by the authors of the Sotto Voce project. However, audio sharing, as described and implemented in Xerox's project, may lead too much to a passive collaboration between companions such as the following or checking in tasks.

In this project, our goal is to design a new interactive and multi-user system for mobile devices that is able to support and promote social interactions in museums through collaborative activities (CSCW) based on educational games. We consider the gaming approach as means to learn things about the museum in an entertaining context. However, our work is driven by HCI and CSCW issues rather than educational issues. In our system, we consider two kinds of cooperative games: one to support explicitly cooperation and sharing; and one to support implicitly cooperation through individual activities. The first kind of cooperative game is similar to enigma or treasure hunt games: visitors have to gather clues and to solve cooperatively enigmas in order to find the solution and to find a particular artwork in the museum. This game needs visitors to share and to debate about what they have seen and learned during the visit. The second kind is a collection of educational games to discover the museum, at the individual level, all along the visit and to gather clues. Indeed, solving an individual game would provide clues to solve the shared enigma (solving a puzzle and answering to a quiz) and would provide awareness information about each visitor's activity. In addition, each visitor can pace its own visit: the group interaction is not highly coupled and the system supports mixed synchronous and asynchronous modes. A scenario is provided at the end of the section, in order to illustrate both kinds of games and how explicit and implicit collaboration is supported.

During the visit, at any time, visitors can use the museum map and the peripheral information about other visitors in order to share their clues and try to solve the shared enigma. In addition, visitors are able to submit solutions, which are validated when visitors need to meet each other and to discuss about the solution. However, in order to stimulate visitors to play with our interactive electronic guide, and their name appear in the fame list in function of the number of points accumulated during the visit: if they cooperate a lot, they receive a proportional number of points.

Designing a user interface in the context of the museum visit is not an easy task because that kind of software will be used only one time by a visitor for one or two hours: the interface must be highly intuitive and affordable at first sight. It is, in one sense, a throw-away interface. For this reason, we have tried to avoid a cognitive and visual overload of the new user interface, and we added only few icons, as shown in Figure 1(a), in order to provide information about other users and the group activity, as well as about the available games. In the new interface, visitors are identified by their name and by a coloured bullet. In addition, coloured bullets indicate what items had been seen by other visitors. An icon representing a symbol indicates that an interactive game is associated with the related artwork, which had not been already solved. The following icon 15 pts, representing two little men, indicates the current score of the group. Finally, a click on the button in the command bar, at the bottom of the screen and representing two little men and a piece of jigsaw, leads to the shared enigma screen described below.



(a) (b)

Fig. 1. (a) View of the current room (b) Puzzle part of the shared enigma.

In Figure 1(b), we present an example of shared enigma: the goal is to find which artwork is hidden by puzzle pieces and to answer to a quiz about this artwork. Each time a visitor finds a clue for the shared enigma, a puzzle piece is removed and a piece of information about the artwork is made available. On the left part of the screen, the system indicates the current score and how many points each visitors have gained. In addition, visitors are able to answer questions about this artwork even if there are still puzzle pieces hiding parts of the image. At any time, the visitors can decide together to provide and validate a common answer to solve the shared enigma, based on the set of clues gathered during the visit. To illustrate this, let us consider the following scenario: Fabio (blue player), Yann (green player) and Carmine (yellow player) are visiting the Marble Museum and have decided to play together during the visit; the goal of the shared enigma is to find the artwork representing the statue of goddess Luni and to answer some questions about it, such as “who was Luni ?” (a goddess, protecting the colony of Luni, living near the town of Carrara). During the visit,

Fabio is playing some educational games. For example, one is to associate a type of marble with the right picture; another one is to play with letters in order to find the author of Vicarius' epigraph. Fabio has solved these games and has gained two clues that are automatically shared with Yann and Carmine: two pieces of information about the statue ("it represents a goddess" and "she was the protector of a colony"); in addition, two puzzle pieces, as shown in Figure 1(b), are removed and the middle part of the statue is now visible. At that point, Yann and Carmine are aware that new clues have been found, as indicated by the "two little men" icon 15 pts, which is updated to indicate the new score, as shown in Figure 1(a). Yann has found which artwork is hidden by the puzzle using the clues discovered by Fabio, but the questions still remain unsolved. Yann asks Fabio and Carmine if they have any idea. Based on the clues, they discovered that the sculpture represents a goddess, protector the colony of Luni: the shared enigma is solved.

5 Conclusion and future work

In this paper, we have presented a system that enables and supports co-visiting at the Marble museum of Carrara. Compared to the existing works, the novelty of our project is to promote communication and social interactions between visitors based on interactive and cooperative educational games embedded in mobile devices such as iPaq PDAs. Furthermore, with this system, we consider the individual museum experience and visitors can regulate the pace of their own visits more flexibly than in solutions as those proposed by the Sotto Voce project where companion visitors share the audio comments. Finally, early feedback from the users has been encouraging. However, more formal tests are being currently conducted in order to improve our system.

References

1. Brown, B., MacColl, I., Chalmers, M., Galani, A., Randell, C., and Steed, A., Lessons from the Lighthouse: Collaboration in a Shared Mixed Reality System, in Proc. ACM CHI'03, 2003, p. 577-584, ACM Press.
2. Ciavarella, C., and Paternò, F., Design Criteria for Location-aware, Indoor, PDA Applications, in Proc. of Mobile HCI'03 Conference, 2003, p. 131-144, Springer Verlag.
3. Falk, J.H., and Dierking, L.D., Learning From Museums, Visitor Experiences and the Making of Meaning, 2000, 288 p., Rowman & Littlefield.
4. Grinter, R., Aoki, P., Hurst, A., Szymanski, M., Thornton, J., and Woodruff, A., Revisiting the Visit: Understanding How Technology Can Shape the Museum Visit, in Proc. ACM CSCW'02, 2002, p. 146-155, ACM Press.
5. Hindmarsh, J., Heath, C., Vom Lehn, D., and Cleverly, J., Creating Assemblies: Aboard the Ghost Ship, in Proc. CSCW'02, 2002, p. 156-165, ACM Press.
6. Hindmarsh, J., Heath, C., Vom Lehn, D., Ciolfi, L., Hall, T., and Bannon, L., Social Interactions in Museums and Galleries, in Deliverable of the Shape Project, D2.1, 2001, Chapter 2, p. 9-23, Disappearing Computer Project.
7. Leinhardt, G., and Crowley, K., Objects of Learning, Objects of Talk: Changing Minds in Museums, in S. Paris (Editors), Multiple Perspectives on Object Centered Learning, 2002, p. 301-324, Lawrence Erlbaum Associates.

CoCicero : un Système Interactif pour la Visite Collaborative de Musée sur Support Mobile

Yann Laurillau

Laboratoire CLIPS-IMAG, équipe IIHM
Domaine Universitaire
38000 Grenoble cedex
Yann.Laurillau@imag.fr

Fabio Paternò

CNR-ISTI
Via G. Moruzzi 1
56124 Pise, Italie
Fabio.Paterno@isti.cnr.it

RESUME

Ce projet porte sur la mise en œuvre d'un système interactif, le système CoCicero, dédié à la visite collaborative de musée sur support mobile à partir de jeux éducatifs et collaboratifs. Dans cet article, nous présentons notre système et détaillons son implémentation. En particulier, nous montrons comment notre système est suffisamment générique pour être adapté à tout musée. Notre propos est illustré par un cas concret d'exploitation au sein du musée du Marbre de Carrara en Italie.

MOTS CLES : Collecticiels, visite collaborative de musée, jeux collaboratifs, supports mobiles, PDA.

ABSTRACT

The goal of this work is to provide tools that promote social interactions between visitors through cooperative and educational games. In this paper, we describe how to support collaborative learning in museum visits and show an example application based on mobile palmtop systems. To this end, we have developed a system that is able to support collaborative and independent activities, and offer context-aware content.

CATEGORIES AND SUBJECT DESCRIPTORS: D.2.11 [Software Engineering]: Software Architectures, H.5.2 [Information Interfaces and Presentation]: User Interfaces -Prototyping.

GENERAL TERMS: Theory, Human Factors.

KEYWORDS: Groupware, CSCW, museum co-visiting, collaborative games, mobile devices, PDA.

Réserver cet espace pour la notice de copyright

INTRODUCTION

La technologie mobile, tels que les téléphones cellulaires ou les assistants personnels numériques (PDA), est de plus en plus répandue et devient un support incontournable pour la production de systèmes interactifs et, a fortiori, des collecticiels. En effet, contrairement aux stations de travail traditionnelles, ces systèmes mobiles offrent une perspective formidable pour le domaine des collecticiels, pour diffuser les systèmes collaboratifs en dehors des laboratoires et pour proposer au grand public des logiciels de moins en moins réservés aux professionnels. Ce type de supports constituent d'excellents vecteurs pour la communication et le partage. Par ailleurs, cette technologie est tout à fait adaptée pour proposer des activités de groupe dans des espaces publics. En particulier, nos travaux portent sur les musées et sur la visite collaborative.

La visite de musée est souvent considérée comme une expérience individuelle et les guides électroniques sont rarement conçus pour favoriser les activités groupe. Au contraire, la visite est un acte intrinsèquement social [4] : notre façon d'apprécier une exposition ou une œuvre est fortement influencée par le contexte social [7][8], c'est-à-dire le comportement de chacun. Aussi, petit à petit, les musées proposent de nouvelles activités, notamment grâce aux technologies mobiles, favorisant la visite collaborative et les activités de groupe.

Dans ce contexte, l'objectif de notre projet est de proposer un système interactif sur support mobile dédié à la visite collaborative de musée, dans le but de promouvoir les interactions, le partage et la communication entre les visiteurs. Notre approche repose sur l'exploitation de jeux éducatifs et collaboratifs comme moyen simple et ludique pour faciliter les échanges entre les visiteurs. Par exemple, le projet ghost ship [6] montre que le jeu et l'exploration ludique des œuvres aident les visiteurs à initier plus facilement une collaboration. Cependant, ce bateau fantôme constitue une seule œuvre mise à disposition des visiteurs, ce qui n'est pas le cas dans le reste de l'exposition.

Aussi, afin de proposer une solution alternative et complémentaire, nous avons développé un système interactif, le système CoCicero, permettant à tout visiteur de mener des activités de groupe et de faire partager son expérience. D'une part, nous avons conçu ce système pour qu'il soit suffisament générique et adaptable à tout type de musée. D'autre part, ce système est en mesure d'offrir des contenus personnalisés et adaptés en fonction du contexte de la visite. Actuellement, ce système est exploité dans le cadre du musée du Marbre de Carrara en Italie. Dans cet article, nous abordons essentiellement les aspects de mise en œuvre du système. Toutefois, dans la suite, nous exposons brièvement les motivations de ce projet et notre positionnement par rapport aux travaux existant. Dans la troisième partie, nous décrivons les principales caractéristiques de l'interface utilisateur du système CoCicero tel qu'il est exploité dans le cadre du musée du Marbre. Ceci nous permet ensuite d'illustrer sa mise en œuvre logicielle détaillée dans la quatrième partie.

LA VISITE COLLABORATIVE DE MUSEE

La visite collaborative de musée est un sujet abordé par de nombreux projets. Par exemple, le projet Sotto Voce [5], développé par Xerox, est un guide électronique, basé sur la technologie iPaq, proposant des commentaires audio sur les œuvres, partagés entre un visiteur et son compagnon via un Media Space audio. Ce dernier a pour objectif de renforcer les liens sociaux et de favoriser la communication. Les auteurs ont identifié quatre types d'activité : (i) l'écoute partagée (*shared listening*), afin de synchroniser les visiteurs et ainsi de faciliter le partage ; (ii) l'indépendance (*independent use*) permettant à un compagnon de mener temporairement ou entièrement la visite individuellement, notamment pour ceux qui ne souhaitent pas s'engager dans une activité de groupe ; (iii) suivre (*following*), quand un compagnon se charge de mener la visite ; (iv) vérifier (*checking in*), activité très courte qui consiste à s'assurer que son compagnon de visite est toujours "présent".

Le projet City [2], une instance du vaste projet Equator, a pour objectif d'étudier la visite collaborative en combinant différentes formes d'interaction : (i) visite à l'aide d'un support mobile avec présence physique dans le musée, (ii) visite virtuelle par immersion dans une reconstitution 3D du musée et (iii) visite virtuelle sur le web. Avec ce système, les visiteurs peuvent partager leur expérience et naviguer conjointement au sein du musée suivant trois paradigmes d'interaction. En outre, le système, qui met en œuvre un système de localisation, indique aux visiteurs la position de chacun au sein du musée et permet aux visiteurs de communiquer à l'aide de canaux de communication audio. Les auteurs ont remarqué que la communication, la localisation et l'orientation, ainsi que la l'observabilité mutuelle, sont des éléments indispensables pour le succès d'une visite collaborative.

Le projet ghost ship [6], comparé aux deux autres projets, est avant tout une démarche artistique. Toutefois, l'objectif de cette étude est d'observer et d'analyser les interactions entre les visiteurs à travers des séquences videos enregistrées et mettant en scène ces derniers. Le système a été installé lors de l'exposition SOFA (Sculpture, Objects and Functional Art, à Chicago) et est constitué d'un bateau peint sur des supports de bois, de faux personnages, d'un faux pont et d'une zone "à l'intérieur du bateau". Certains hublots du bateau sont en fait des écrans vidéos qui restituent les séquences vidéos des interactions entre visiteurs et avec le bateau. Notamment, les auteurs ont observé que l'exploration et le jeu permettaient aux visiteurs de nouer plus facilement des contacts et d'explorer ensemble les différentes facettes du bateau.

Dans ce contexte, comparé au projet City, nous considérons des visiteurs menant une visite "réelle" alors que le projet City mise l'exploitation de systèmes mixtes. D'ailleurs, un grand nombre de travaux portant sur la visite collaborative de musée portent sur la visite virtuelle sur le web comme c'est le cas pour le musée Van Gogh [11]. Ou bien comme d'autres travaux, comme le projet HIPS, qui concentrent leurs efforts pour offrir des contenus personnalisés en fonction de leur position GPRS à des groupes de touristes. De plus, les auteurs du projet City, tout comme ceux du projet Sotto Voce, ont noté que les informations relatives à la localisation des visiteurs est un élément essentiel de la visite collaborative. Cet aspect est pris en compte dans notre projet puisque, comme nous le détaillons dans la section suivante, le système rend observable la position des visiteurs au sein du musée.

Démarche de travail

Pour aborder ce projet de visite collaborative de musée, nous misons sur l'exploitation de jeux éducatifs et collaboratifs qui, comme cela a été souligné en introduction et par les auteurs du projet ghost ship, constituent une approche intéressante pour promouvoir les interactions et le partage entre les visiteurs. De plus, une telle approche permet de préserver le caractère individuel de la visite, qui est, comme l'ont remarqué les auteurs du projet Sotto Voce, un aspect important. Cependant, le partage audio, tel qu'il est implémenté dans ce dernier, conduit assez souvent à une visite passive, telle que l'activité following, alors que nous envisageons une participation active entre visiteurs.

Aussi, dans ce projet, nous considérons deux types de jeux collaboratifs : le premier met en œuvre une collaboration explicite entre les compagnons tandis que le se-

cond, reposant sur une activité individuelle, participe indirectement à l'activité de groupe. Le premier type de jeu est la résolution d'énigme commune, présentée sous forme de piste au trésor : les visiteurs ont pour objectif de récolter une série d'indices permettant de découvrir une œuvre mystère. Cette énigme se caractérise par un quiz et un puzzle représentant l'œuvre à découvrir. Aussi, ce jeu nécessite que les visiteurs partagent et communiquent sur ce qu'ils ont vu. Le second type de jeu est une série de jeux, tel que le quiz, qui permettent à chacun de récolter un indice dès que la solution a été trouvée. Cet indice, qui se matérialise sous la forme d'une bribe d'information textuelle ou d'un morceau de puzzle, doit permettre au groupe de résoudre ultérieurement l'énigme commune. Ce type de jeu permet à chaque utilisateur de réguler sa visite et de ne pas être soumis au mettre rythme que ses compagnons. Nous proposons dans la partie suivante un scénario d'utilisation.

A l'issue de la visite, et afin d'encourager les visiteurs a essayer la version collaborative du système, le noms des participants s'affichent sur le tableau des scores sachant que le score sera d'autant plus élevé que la collaboration aura été importante.

VISITE COLLABORATIVE DU MUSEE DU MARBRE

Dans cette partie, afin d'illustrer les détails d'implémentation de notre système, nous présentons les principaux aspects de l'interface utilisateur dans le cas du musée Marbre de Carrara. Dans le premier paragraphe, nous rappelons brièvement les principales fonctionnalités de la version mono-utilisateur pour aborder ensuite celles de la version collaborative.

Le projet original

Le système CoCicero repose sur le système Cicero [3], une application mono-utilisateur qui vise à offrir des contenus numériques personnalisés en fonction de la localisation du visiteur, et des informations complémentaires à propos des œuvres présentées dans le musée. Cette application est développée sur la technologie iPaq. Pour la conception de l'interface utilisateur, trois types de tâches ont été considérées :

- S'orienter dans le musée : trois niveaux d'information sont disponibles : une carte du musée, une carte des sections du musée et un plan de la pièce en cours de visite, affichant une série d'icônes représentant les œuvres présentes dans la pièce, comme le montre la Figure 1. En sélectionnant une icône, le système affiche une photo de l'œuvre accompagnée d'informations factuelles et d'un commentaire audio. L'apparition de la photo a pour objectif de permettre à l'utilisateur de repérer l'œuvre dans la pièce et d'associer les commentaires avec la bonne œuvre. La présence de détecteurs à infrarouge dans chaque pièce du musée permet au système de mettre à jour automatiquement le plan de la pièce en cours de visite et d'activer les commentaires audio associé.
- Contrôler l'interface en agissant, par exemple, sur les paramètres relatifs aux commentaires audio.
- Accéder aux informations générales sur le musée selon trois niveaux de granularité (musée, section et pièce).

A tout moment, le système est capable d'indiquer à l'utilisateur où il se trouve. Toutefois, le visiteur peut modifier le contenu et choisir d'afficher le plan d'une autre pièce.

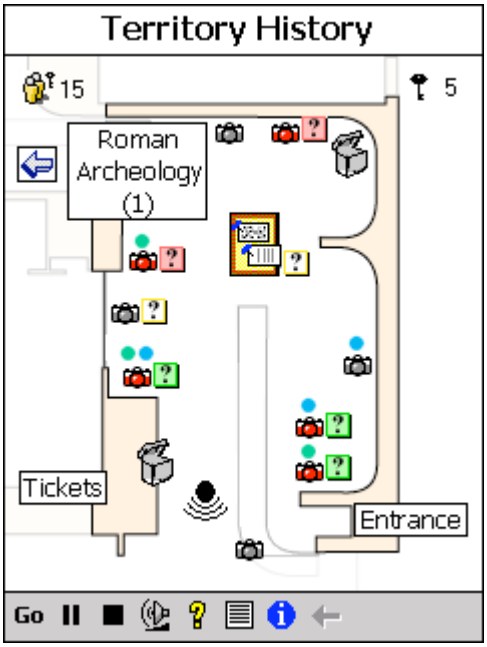


Figure 1. Visite d'une pièce du musée.

L'interface du système CoCicero

La conception d'une interface utilisateur pour les guides électroniques dédiés à la visite d'un musée n'est pas une tâche facile puisque ce type d'interface, que l'on peut qualifier d'interface jetable, est en général destinée à être utilisée une seule fois pendant une heure ou deux. Aussi, une telle interface doit être hautement intuitive et utilisable après une très courte période d'apprentissage. Pour cette raison, nous avons cherché à proposer une interface la plus légère possible, ce qui est d'autant plus difficile qu'il s'agit d'un collecticiel. En particulier, par rapport à la version mono-utilisateur, nous avons ajouté que quelques icônes, comme le montre la Figure 1, relatives à l'activité de groupe et aux jeux interactifs. Dans cette nouvelle interface, les visiteurs sont identifiés par une couleur unique et une puce colorée ●● qui indique les œuvres qui ont été vues par nos compagnons. L'icône représentant un ? indique qu'un jeu est associé à une œuvre et qui n'a pas encore été résolu. Quand, l'utilisateur a terminé le jeu, cette icône change de couleur indiquant s'il s'agit d'un succès (couleur verte) ou d'un échec (cou-

leur rouge). Dans la partie droite de l'écran, en haut, une icône représentant une clé 5 indique le nombre d'indices que le visiteur a gagné pour le groupe. Ces indices sont systématiquement partagés avec les autres visiteurs et utilisables pour résoudre l'énigme commune. Enfin, un clic sur l'icône, représentant deux bonhommes 15 et indiquant combien le groupe a gagné d'indices, fait apparaître l'écran relatif à l'énigme commune décrit ci-dessous.

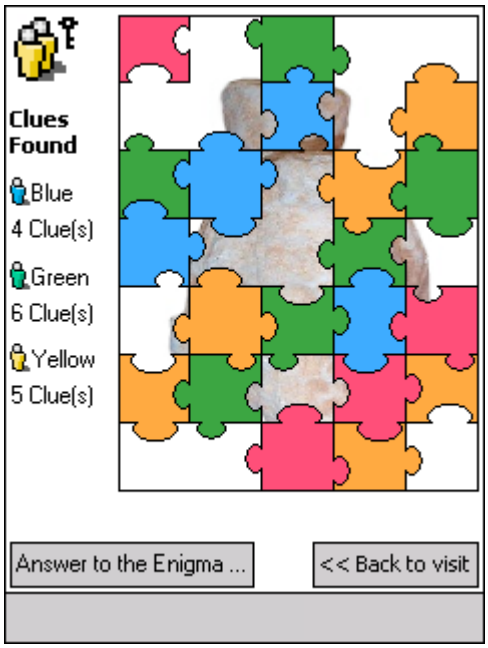


Figure 2. Partie puzzle de l'énigme commune.

La Figure 2 représente la partie puzzle de l'énigme commune : l'objectif est de réagencer le puzzle pour découvrir l'œuvre mystère et ainsi répondre à un quiz sur cette œuvre. A chaque fois qu'un jeu individuel est résolu, une pièce du puzzle est disponible ainsi qu'un indice qui permettra aux joueurs de répondre au quiz. Dans la partie gauche de l'écran, le système indique combien d'indices ont été collectés par chaque participant. Enfin, les joueurs ont la possibilité de se consulter à tout moment pour tenter de répondre au quiz, sans forcément avoir obtenu toutes les pièces du puzzle. Toutefois, les visiteurs obtiendront plus de points si le puzzle est terminé.

Pour illustrer l'utilisation du système, considérons le scénario suivant : Jean (joueur bleu), Anne (joueur verte) et pierre (joueur jaune) visitent le musée du Marbre et ont décidé de jouer ensemble pendant la visite ; l'objectif de l'énigme commune est de trouver la statue représentant la déesse Luni et de répondre à des questions telles que "Qui est Luni ?" (la déesse protectrice de la colonie de Luni). Au cours de la visite, Jean joue à une série de jeux : l'un d'eux, comme le montre la Figure 3, est de retrouver et d'associer pour chaque type de marbre le nom qui le caractérise ; un autre consiste à jouer avec des lettres pour retrouver le nom de l'auteur de l'épigraphe de Vicarius. Jean a trouvé la solution de ces jeux et a gagné deux nouveaux indices qui sont automatiquement partagés avec Anne et Pierre : "il s'agit d'une statue" et "elle protège une colonie". De plus, deux nouvelles pièces du puzzle sont disponibles. A ce moment précis, Anne et Pierre peuvent observer que de nouveaux indices ont été gagnés : l'icône représentant deux bonhommes 15 est mise à jour et indique le nouveau total du nombre d'indices collectés. Anne pense avoir trouvé quel est l'œuvre mystère, en particulier grâce aux indices collectés, mais le quiz n'a pas encore été résolu. Anne propose alors à Jean et à Pierre de résoudre le quiz : il s'agit d'une sculpture qui représente une déesse, protectrice de la colonie de Luni ; l'énigme est résolue.

Figure 3. Exemple de Jeu.

MISE EN ŒUVRE LOGICIELLE

Dans cette section, nous détaillons l'implémentation du système. Le premier paragraphe détaille les principaux composants de l'architecture logicielle. Le second paragraphe porte sur le moteur de rendu de chaque pièce du musée. Le dernier porte sur la partie collaborative de l'application.

Architecture logicielle

Tout comme la version d'origine, la nouvelle version du système CoCicero a été développée en C++ sur un iPaq. Le système exploite également une série de détecteurs à infrarouge pour la localisation et un réseau sans fil pour la communication entre PDA. Il est à noter qu'une contrainte technique a été imposée, à savoir de maintenir

une version mono-utilisateur indépendante de la version collaborative.

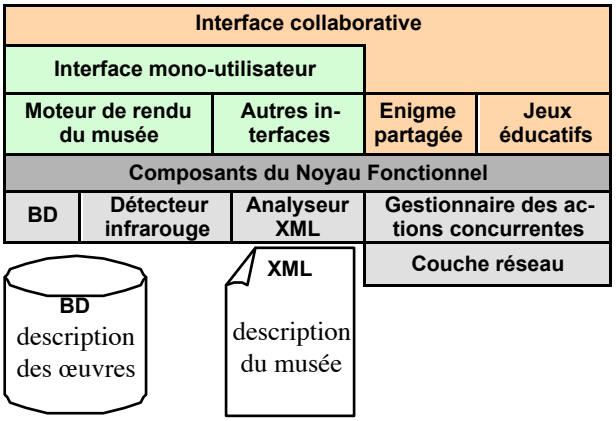


Figure 4. Principaux composants du système.

Comme le montre la Figure 4, au niveau de l'interface utilisateur, deux types de composants ont été développés :

- La partie mono-utilisateur de l'interface repose sur un composant essentiel chargé du rendu graphique des pièces du musée et devant gérer les interactions de l'utilisateur avec les œuvres représentées par des icônes. Ce composant est lui-même basé sur trois composants du noyau fonctionnel : une base de données contenant tous les détails sur les œuvres contenues dans le musée ; un composant gérant la détection à infrarouge, chargé de localiser l'utilisateur dans le musée et de mettre à jour automatiquement le plan de la pièce en cours de visite ; un analyseur XML chargé de lire le fichier XML contenant la description du musée et la répartition des œuvres au sein des pièces, et de produire des structures de données manipulées par la suite par le système. Afin de disposer d'un code modulaire et flexible, la description du musée et la position des icônes associées à chaque œuvre dans chaque pièce est contenue dans un fichier XML séparé. Cet aspect est intéressant parce qu'il contribue à rendre notre système suffisamment générique pour pouvoir être adapté à tout musée.
- La partie collaborative de l'interface utilisateur repose sur la partie mono-utilisateur et repose sur des composants spécifiques à la visite collaborative du musée. En particulier, cette partie gère les composants dédiés à l'énigme commune et aux jeux éducatifs. De plus, les jeux éducatifs ont été développés sous forme de pluggins chargés au démarrage de l'application. Cette partie repose aussi sur deux composants du noyau fonctionnel chargés de gérer les actions concurrentes et la communication à travers le réseau.

Moteur de rendu

La fonction première de ce composant, élément central de l'application, est d'assurer le rendu graphique du plan de chaque pièce du musée et de gérer le dialogue entre l'utilisateur et l'application, en particulier, lorsque l'utilisateur sélectionne une œuvre afin d'obtenir un commentaire audio. De plus, grâce à ce composant logiciel, la coexistence entre la version collaborative et la version mono-utilisateur est rendue possible, en partie.

Ce moteur de rendu repose sur deux principaux composants :

- Une base de donnée qui contient toutes les informations relatives aux œuvres telles que le titre de l'œuvre, son auteur, sa date de création, etc.
- Un analyseur XML, chargé de récupérer les données relatives à la description du musée et à la position des œuvres au sein du musée, l'enchaînement de la visite entre les différentes pièces du musée, la traduction dans plusieurs langues des informations textuelles (actuellement en italien et en anglais), le nom des images utilisées pour afficher les différentes cartes du musée et les différentes icônes, les noms de fichier des commentaires audio et des vidéos. En sortie, l'analyseur génère une série de structures de données représentant l'organisation du musée en mémoire. Ces structures sont manipulées par la majorité des composants logiciels des deux versions du système. Dans la suite, nous décrivons comment les fichiers XML doivent être structurés.

Du point de vue de l'implémentation, les icônes et le texte affichés à l'écran sont considérés comme étant des objets identifiés par une valeur entière et enregistrés dans une liste. Aussi, quand un utilisateur clique sur un des objets, il est très facile de retrouver l'objet sélectionné, comme le font de nombreuses librairies graphiques. De plus, ce mécanisme est suffisamment souple pour ajouter ou retirer des objets. Cette fonctionnalité est amplement utilisée par la version collaborative.

Propriétés du moteur du rendu

Ce composant dispose de propriétés logicielles qui ont facilité le développement des deux versions de l'application :

- *Souplesse (i):* Premièrement, du point de vue du développeur, le moteur de rendu est indépendant du système et n'est aucunement lié avec ses caractéristiques telles que la taille de l'écran ou le nombre de couleurs affichables. De plus, ce composant est suffisamment souple pour être employé sur une large gamme de supports mobiles. Actuellement, l'application fonctionne pour la technologie iPaq et pour les ordinateurs de bureau. Deuxièmement, la réalisation actuelle de ce composant nous a permis de

développer la version mono-utilisateur de l'application indépendamment de la version collaborative. Aussi, afin d'ajouter les éléments d'interface relatifs à la partie collaborative, il n'est pas nécessaire d'accéder directement au moteur de rendu ; il suffit juste de manipuler les structures de données qui ont été crées par l'analyseur XML. De plus, tous les événements d'interaction générés par ces objets sont redirigés vers les composants logiciels spécifiques de la version multi-utilisateur sans interférer avec la version mono-utilisateur.

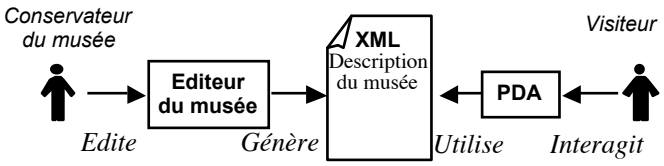


Figure 5. Processus de gestion du musée.

- *Souplesse (ii):* l'organisation des œuvres au sein d'un musée est rarement définitive et le conservateur du musée est souvent amené à modifier cette organisation, en particulier, lorsqu'une exposition temporaire est mise en place. Cela se traduit souvent par le déplacement, l'ajout et la suppression d'œuvres. De plus, les personnels du musée ne sont pas nécessairement des experts en informatique ; aussi, il est impératif que notre système soit suffisamment souple pour prendre en compte les changements et refléter l'organisation réelle du musée. Pour ces raisons, comme le montre la Figure 5, nous avons développé un éditeur qui permet aux membres du musée de modifier le fichier XML contenant la description du musée et qui est directement exploité par notre guide électronique. Cet éditeur repose sur des techniques de manipulation directe en émulant l'interface du guide et qui génère un nouveau fichier XML à chaque fois que l'organisation du musée est modifiée. Bien sûr, les membres du musée n'accèdent pas directement au contenu du fichier XML et n'ont pas besoin de connaître la technologie XML.
  Avec cette approche, notre système et le moteur de rendu ne sont aucunement liés avec le musée du Marbre et peuvent être exploités pour tout type de musée : il suffit de créer un nouveau fichier XML.
- *Régionalisation :* à l'aide de ce fichier XML, nous offrons aussi un support simple et efficace pour la régionalisation de l'interface. Dans la version actuelle, nous proposons une interface en italien et en anglais, et il est tout à fait aisé d'ajouter une nouvelle langue telle que le français. Ce mécanisme est largement exploité par de nombreuses applications comme le navigateur web Mozilla [10].

Notre approche, du point de vue de l'implémentation, diffère de projets existants qui exploite la technologie XML comme moyen pour décrire entièrement l'interface utilisateur tels que la plate-forme Made [1] qui utilise le langage HTML ou XUL (XML-based User Interface Language) pour le projet Mozilla [10]. Dans notre cas, nous ne cherchons pas à décrire l'interface utilisateur mais à proposer un système capable de mettre en œuvre la visite interactive de musée. En particulier, comme nous venons de le détailler, la technologie XML est exploitée pour refléter l'organisation des œuvres au sein du musée.

Détails de la structure du fichier XML

```
<museum>
 <map>
  <first>1</first>
  <sections>
   <section>
    <id>1</id>
    <title>1</title><image>1</image>
    <orientation>LEFT</orientation>
    <coords>
     <X>471</X><Y>757</Y>
     <W>156</W><H>233</H>
    </coords>
    <audio>StoriaTerr</audio>
    <items>
     <item>
      <id>10</id><type>artwork</type>
      <coords>
       <X>10</X><Y>105</Y>
      </coords>
      <parameters>
       <image>37</image>
       <image2>74</image2>
       <resource>724</resource>
       <audio>1</audio>
       <video>PiazzaAlberica</video>
      </parameters>
     </item>
     [...]
    </items>
   </section>
   [...]
  </sections>
 </map>
 [...]
</museum>
```

Figure. 6. Exemple d'utilisation de la technologie XML pour décrire l'organisation du musée du Marbre.

Toutes les données relatives à l'organisation du musée sont stockées dans un fichier XML dont voici les détails de sa structure. Comme le montre la Figure 6, une pièce est décrite par les données contenues dans le abel <sections>. Chaque pièce est identifiée par un identifiant unique qui est exploité pour la localisation des visiteurs au cours de leur déplacement. Une pièce est définie par un titre et une image pour le plan. Nous fournissons aussi une information sur l'orientation de la pièce par rapport à la position du visiteur lorsque celui y entre pour la première fois. Par exemple, comme le montre la Figure 1, le plan de la pièce a été orienté à 270° par rapport au plan du musée et représente la pièce tel que le visiteur la découvre lorsqu'il entre dans celle-ci. Le champ <coords> contient les coordonnées de la pièce par rapport la carte

du musée. Le champ <audio> contient le nom de fichier du commentaire audio sur la pièce en cours de visite.

Les artefacts, réels et virtuels, contenus dans une pièce (champ <items>) sont décrit par le champ <item>. Dans notre exemple, la pièce contient un artefact qui se trouve être une œuvre (artwork) comme l'indique le champ <type>. Chaque artéfact est identifié par un unique identifiant et par des coordonnées permettant de le localiser au sein de la pièce. Des informations supplémentaires sont disponibles pour chaque type d'artefact, contenues dans le champ <parameters>. Dans notre exemple, l'artefact est représenté par deux icônes qui indiquent si l'œuvre à été vue ou non. Les champs <audio> et <video> indiquent si un commentaire audio et une vidéo sont disponibles pour cette œuvre.

Enfin, comme le montre la Figure 6, chaque champ relatif à une portion de texte ou à une image sont représentés par une valeur entière. En effet, ces informations sont stockées séparément dans les sections <languages> et <images> du fichier XML (absentes de la Figure 6).

Couches logicielles collaboratives

Comme nous l'avons souligné dans le paragraphe précédent, le moteur de rendu est un composant important des deux versions de l'application. En particulier, les aspects dynamiques de ce composant ont été exploités intensivement dans la mise en œuvre de la visite collaborative du musée. En effet, il est facile de modifier, à tout moment, le contenu de chaque pièce afin d'ajouter ou de modifier les éléments d'interface liés aux aspects collaboratifs. En particulier, cette fonctionnalité est exploitée pour :

- *L'interaction avec une œuvre :* quand une œuvre devient le centre d'intérêt d'un visiteur, des puces de couleur apparaissent à l'écran, comme le montre la Figure 1.
- *L'activité de groupe :* les icônes relatives à l'énigme commune et à l'évolution de la visite (5 et 15), comme nous l'avons décrit dans la troisième partie, sont affichées pour indiquer combien d'indices ont été collectés par le groupe. Les valeurs affichées par ces icônes sont mises à jour automatiquement dès que l'un des participants trouve la solution à l'un des jeux.
- *Les jeux éducatifs :* les jeux éducatifs sont développés sous forme de modules indépendants et sont chargés dynamiquement lors du démarrage de l'application, sous forme de librairie dynamique (ces modules sont dans le format DLL sous le système Windows CE). Quand l'application démarre, le système balaye un répertoire prédéfini dans lequel sont stockés les différents types de jeux et des structures de données spécifiques sont crées. Les données exploitées par ces jeux sont eux aussi stockés dans des fichiers XML. Chaque jeu est associé à une œuvre en se basant sur l'identifiant de ce dernier. Aussi, pour chaque jeu disponible, une icône ? est affichée à côté de l'œuvre associée comme le montre la Figure 1. Quand un jeu est sélectionné, le moteur de rendu est en mesure d'envoyer l'événement interactif au module correspondant.

Du point de vue du réseau, la communication entre les PDA repose sur le protocole multicast et ainsi chaque PDA ne nécessite pas de configuration particulière ni de connaître à l'avance chaque adresse IP. Toutefois, afin d'initier une visite collaborative, un PDA est élu en tant que "serveur" dont la tâche unique est d'assigner un identifiant de joueur à chaque nouvel arrivant. Comme le montre la Figure 4, ces aspects sont gérés par un composant dédié au-dessus duquel a été implémenté un composant chargé de gérer les actions concurrentes, en particulier, lorsque les visiteurs tentent de résoudre l'énigme commune.

CONCLUSION

Dans cet article, nous avons présenté les détails techniques de mise en oeuvre d'un système générique dédié à la visite collaborative de musée. En particulier, nous avons fait apparaître ses qualités logicielles, notamment, sa grande souplesse qui lui permet d'être adapté à tout type de musée. Actuellement ce système fait l'objet d'une utilisation concrète au sein du musée du Marbre de Carrara. Plus globalement, l'objectif de ce projet est d'offrir un outil favorisant l'interaction et la coopération entre visiteurs, à base de jeux interactifs et collaboratifs sur supports mobiles.

REFERENCES

1. Bellotti, F., Riccardo, B., De Gloria, A., et Margarone, M., Margarone, Supporting Efficient Design of Mobile HCI, actes de la conférence Mobile HCI'03, 2003, p. 241-255, Springer-Verlag.
2. Brown, B., MacColl, I., Chalmers, M., Galani, A., Randell, C., et Steed, A., Lessons from the Lighthouse: Collaboration in a Shared Mixed Reality System, actes de la conférence ACM CHI'03, 2003, p. 577-584, ACM Press.
3. Ciavarella, C., et Paternò, F., Design Criteria for Location-aware, Indoor, PDA Applications, actes de la conférence Mobile HCI'03, 2003, p. 131-144, Springer Verlag.
4. Falk, J.H., and Dierking, L.D., Learning From Museums, Visitor Experiences and the Making of Meaning, 2000, 288 p., Rowman & Littlefield.

5. Grinter, R., Aoki, P., Hurst, A., Szymanski, M., Thornton, J., and Woodruff, A., Revisiting the Vi-sit: Understanding How Technology Can Shape the Museum Visit, actes de la conférence ACM CSCW'02, 2002, p. 146-155, ACM Press.

6. Hindmarsh, J., Heath, C., Vom Lehn, D., and Cleverly, J., Creating Assemblies: Aboard the Ghost Ship, actes de la conférence ACM CSCW'02, 2002, p. 156-165, ACM Press.

7. Hindmarsh, J., Heath, C., Vom Lehn, D., Ciolfi, L., Hall, T., and Bannon, L., Social Interactions in Museums and Galleries, dans Deliverable of the Shape Project, D2.1, 2001, Chapitre 2, p. 9-23, projet Disappearing Computer.

8. Leinhardt, G., and Crowley, K., Objects of Lear-ning, Objects of Talk: Changing Minds in Mu-seums, dans S. Paris (Editeur), Multiple Perspecti-ves on Object Centered Learning, 2002, p. 301-324, Lawrence Erlbaum Associates.

9. Musée Lighthouse, http://www.thelighthouse.co.uk

10. Projet Mozilla, http://www.mozilla.org

11. Visite virtuelle du musée Van Gogh, http://www.vangoghmuseum.nl

Clover Architecture for Groupware

Yann Laurillau
Laboratoire CLIPS/IMAG
University of Grenoble
Domaine Universitaire, BP 53
38041 Grenoble, FRANCE
Yann.Laurillau@imag.fr

Laurence Nigay[1]
University of Glasgow
Department of Computer Science
17 Lilybank Gardens
Glasgow G12 8QQ
laurence@dcs.gla.ac.uk

ABSTRACT

In this paper we present the Clover architectural model, a new conceptual architectural model for groupware. Our model results from the combination of the layer approach of Dewan's generic architecture with the functional decomposition of the Clover design model. The Clover design model defines three classes of services that a groupware application may support, namely, production, communication and coordination services. The three classes of services can be found in each functional layer of our model. Our model is illustrated with a working system, the CoVitesse system, its software being organized according to our Clover architectural model.

Keywords

Conceptual Software Architecture, Clover Design Model.

INTRODUCTION

People's interest in collaborating with others is growing daily and in particular on the internet. This is not only in order to read web pages or download files but also to communicate, to play multi-user games or to exchange data such as music files. In a nutshell, to interact with others. This leads to an outbreak of a multitude of multi-user systems (or groupware) designed to chat, such as ICQ, to play, such as Quake or to share mp3 files, such as the well-known Napster (peer-to-peer). Despite the multiplicity of existing groupware, their development still remains complex and unsystematic. It is widely recognized that, although the adhoc development of software is acceptable for throw-away prototypes, architectural design of complex systems can no longer simply emerge from craft skills. In this paper we address the architectural design of groupware by defining a conceptual software architectural model for groupware.

A conceptual software architecture is an organization of computational elements and the description of their interactions. A conceptual software architecture model defines a vocabulary of design elements, imposes configuration constraints on these elements, determines a semantic interpretation that gives meaning to the system description, and enables analysis of the system properties [22].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CSCW'02, November 16-20, 2002, New Orleans, Louisiana, USA.
Copyright 2002 ACM 1-58113-560-2/02/0011…$5.00.

For example, Dewan's generic architectural model [8] structures a groupware application as a stack of shared and replicated layers which communicate with each other by exchanging events. A conceptual software architecture is then mapped into an implementation architecture dependent on the software tools available. Software tools for the construction of groupware such as the Groupkit interaction toolkit [21] or the COCA [16] and DragonFly [1] platforms will not eliminate conceptual architectural issues as long as the construction of these systems requires programming. Clearly, a conceptual model for identifying and organizing the components of groupware is still a necessity. As explained in [5], without an adequate architectural framework, the construction of groupware and in general interactive systems is hard to achieve, the resulting software is difficult to maintain and iterative refinement is impossible.

Several conceptual software architectural models for groupware exist, such as Dewan's generic architecture [8], Clock [11] and PAC* [5]. Our Clover architectural model is complementary to the existing models by defining at a finer grain how functionalities should be structured at each level of abstraction in the architecture. To do so, we base our architectural model on the Clover design model [9][23]. The Clover design model defines three classes of services: production, communication and coordination services. During the system design phase, the three types of services must be identified and their access harmoniously combined in the user interface. Our Clover architectural model makes such system design concepts explicit within the software. Indeed, because software architecture modeling is a design activity at the interface between the system design field and the programming field, software architecture must take into account attributes of both fields: these attributes include properties of the designed system and of the future software to be implemented [17].

The structure of the paper is as follows: first, we introduce the Clover design model and stress its impact on the design of the system as well as on the design of the user interface. We then present the main characteristics of three architectural models, Arch, Dewan's architecture and PAC*, all three of which our Clover metamodel relies on. Our Clover architectural model is a metamodel i.e. several Clover architectural models can be derived from it. We first present one Clover architectural model and then the generalization, the metamodel. We close with the presentation of the clover architecture of a running system, CoVitesse, which enables collaborative navigation on the World Wide Web. The discussion will be illustrated with

1. On sabbatical from the University of Grenoble, CLIPS Laboratory, BP 53, 38041 Grenoble Cedex 9, France.

two systems, CoVitesse system, whose main features are presented in the next section.

AN ILLUSTRATIVE EXAMPLE: COVITESSE SYSTEM

We applied our Clover architectural model to the software design of the CoVitesse system.

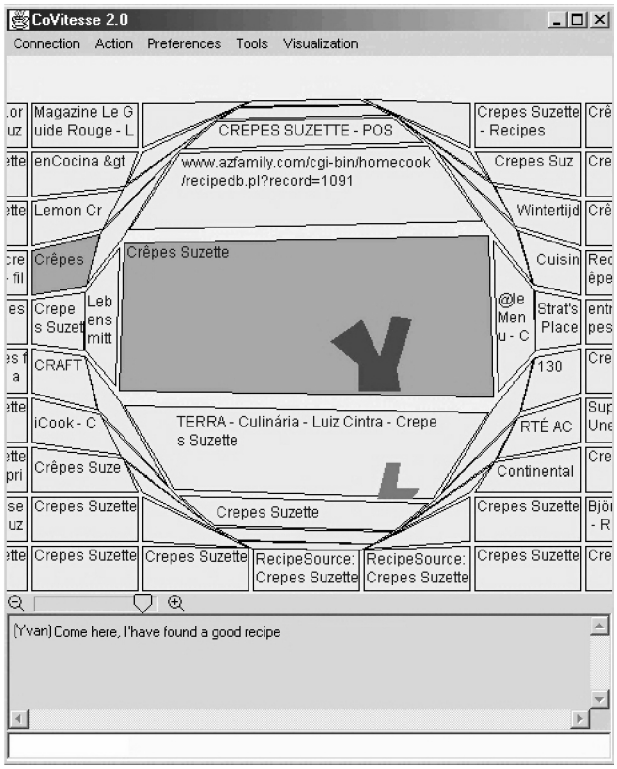


Figure 1: Snapshot from the CoVitesse system.

CoVitesse [7][14] is a groupware application that enables collaborative navigation on the WWW. Users sharing the same information space can cooperate to seek information. This system is built on the Vitesse system [19], a single-user application that visualizes the results of a query submitted to a search engine on the WWW. As shown in Figure 1, CoVitesse displays the overall set of results: each retrieved page (node) is displayed by a polygon. Users are represented by colored shapes (a red Y for Yann and a blue L for Laurence). When a user wants to visit a page, she/he double-clicks on the corresponding polygon and the web page is downloaded in a separate browser. Darker polygons indicate visited web pages. At any time, a user can see all other users moving in the space, use the chat box (which is below the information space) to communicate with other users, or organize her/his own caddy which contains the marked pages.

CoVitesse allows users to create groups. A group is identified by a name and a color. For example, in Figure 1, Yann, represented by the shape Y, is a member of a group identified with a red color. In the current version of the system, four kinds of groups, i.e. collaborative navigational tasks, are available: guided tour, opportunistic navigation, shared navigation and cooperative navigation. According to the group type, different functionalities are available. For example, within a group defined as an opportunistic navigation group, a member can take control of the navigation of the other members; such a functionality is not possible with a Shared navigation group. Moreover, access rights to data are different according to group types; rights are imposed on the group caddy (i.e. the pages selected by the group) as well as on the group preferences. Group preferences include the information related to the group, the choice or not to publish the gathered results and the publication filter applied on the results. For example, any member of an opportunistic navigation can modify the group preferences. More details about these navigation types can be found in [14].

CoVitesse provides persistent access to all the data, modified or produced during a session. These data include information about users and groups such as the avatar shape, the gathered results and the preferences. These data are protected with a simple mechanism of username/ password. Then, when a user starts a new session, she/he recovers her/his own private data.

CLOVER DESIGN MODEL

Description

The Clover design model [9][23] provides a high level partitioning for reasoning about the collaborative services a groupware application may support. As shown in Figure 2, a groupware application covers three kinds of services: production, communication and coordination.

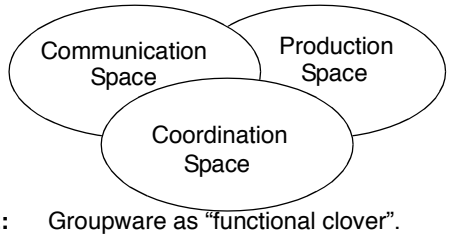


Figure 2: Groupware as "functional clover".

- The production space refers to the objects produced by a group activity or to the objects shared by multiple users. In the CoVitesse system, the production is the set of results gathered by users and groups during a session.
- The communication space refers to person-to-person communication such as e-mail, relay chat, mediaspace. The CoVitesse system provides a chat room for users navigating in the same information space.
- The coordination space covers activities dependencies including temporal relationships between the multi-user activities. It also refers to the relationships between actors and activities. In the CoVitesse system, coordination tasks include creating, joining, leaving a group or accepting a new member. Looking at the avatars moving within the information space is also a coordination task.

However, not all the existing groupware support the three facets of the Clover design model. This point is further discussed in the following section.

CSCW Design Issues

The Clover design model is useful for the design of groupware during the functional specification phase because it defines three classes of functionalities that a groupware application may support. For example, the

CoVitesse system supports the following three types of functionalities:

- *Production*: the system manages a caddy of collected results for each user and group. It is possible to modify the group caddy according to the defined access rights.
- *Communication*: the system provides a simple chat box. Currently, the communication group is composed of all the users.
- *Coordination*: the system provides services to join/leave a session, to create/join/leave a working group. In addition, coordination is done when a user moves her/his avatar in the information space or when s/he modifies the preferences.

As we mentioned in the previous section, not all the existing groupware support the three types of functionalities. For example, a mediaspace, a system dedicated to informal communication to enhance team awareness, is built upon communication functionalities and few coordination functionalities such as joining or leaving a session. In this example, there is no support for production. In addition some approaches treat the coordination functionalities as special in contrast to the production and communication functionalities: coordination functionalities are separated from the system itself. For example in [8] the session manager dedicated to coordination is modeled as an external software component. Another example is the COCA platform [16]. The platform is dedicated to coordination and is able to manage tools including conference tools or whiteboards based on a set of rules defined in a prolog-like language. These rules dedicated to coordination are interpreted at the runtime by the COCA virtual machine. The key idea of this platform is to extract the coordination functionalities from the system. The platform therefore enables the reuse of non-collaborative systems, but it provides no support for production and communication. Moreover an existing groupware application may include its own coordination model which may not match with the COCA one.

When the Clover model defines three types of functionalities, it also defines three classes of data structures manipulated by these functionalities. For example let us consider the implementation of the notion of group in a chat box used to manage a lecture. We consider two types of actors: the lecturer and the audience. The system implements coordination functions, to join a session and to assign roles (i.e., lecturer or audience) and enables communication between the lecturer and the audience. In order to avoid interference generated by the audience chatting, we should consider two communication channels: one channel for communication between the lecturer and the audience and another channel between the members of the audience excluding the lecturer. The system therefore maintains two data structures related to the notion of group:

- one structure containing the representation of a lecturer and an audience, for coordination and communication,
- one structure containing the representation of an audience with its members and excluding the lecturer, dedicated to communication only.

The example illustrates the fact that data structures can be classified according to the three Clover facets. In addition the example shows that the same data structure can be shared and manipulated by different Clover functionalities. The data structure therefore belongs to one of the intersections of the three facets in Figure 2. To further illustrate this last point we consider CoVitesse: when a user moves her/his avatar in the information space:

- the shape is drawn at the new coordinates in each user's view: a coordination functionality;
- the URL is added to the group history: a production functionality.

One data structure is used for modeling the position of an avatar in the information space and several Clover functionalities manipulate this unique structure.

In conclusion, the Clover design model can be used as a guideline in the design of groupware for specifying the functionalities as well as the data structures. The usefulness of the model for the design of the user interface is more problematic.

HCI Design Issues

While the Clover model is useful for identifying the services provided by groupware, it should not be used as a guideline for designing the user interface. For example, having separate windows or panels dedicated to each Clover space will lead to a complex interaction. The first design of the CoVitesse user interface was based on the Clover design model. The following three windows are part of the first CoVitesse user interface:

- a window dedicated to the production containing the information space and the marked web pages,
- a window dedicated to communication that includes textual chat, and,
- a third window dedicated to coordination that allows the user to create and edit a group.

Informal tests with groups of four users having to perform a scenario (i.e., collect a list of the ten most relevant web sites about human-computer interaction), lead us to conclude that the information space and the chat box should be combined at the user interface (combination of production and communication). Indeed, at the end of the test sessions, users complained about a lack of awareness: they were not able to perform a navigational task and to look at chat exchanges at the same time.

In addition to our own experience with CoVitesse, several groupware systems combine the services provided by each of the three Clover spaces, making them accessible and observable through appropriate user interface composite objects. For example, as shown in Figure 3, the collaborative scrollbar introduced in [5] and based on the multi-user scrollbar of the SASSE editor [2] covers functionalities of the three spaces. The right scrollbar,

dedicated to production, is an ordinary scrollbar for scrolling the text. The left scrollbar, dedicated to communication and coordination, contains multiple elevators, one per remote user who can be seen in a video thumbnail inside the elevator. On the one hand, from a functionality viewpoint, it is easy to observe that the coordination is covered by the current position of remote users in the text and communication by a video service. On the other hand, from a user interface viewpoint, a single interaction object (a scrollbar) is used for coordination as well as communication.

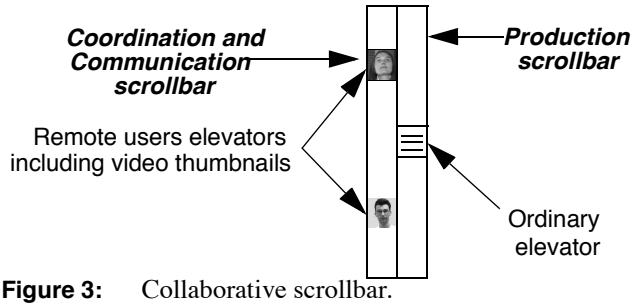


Figure 3: Collaborative scrollbar.

In conclusion, our own experience as well as existing groupware widgets underline the fact that the Clover design model is useful in identifying the funtionalities that the user interface should provide but should not be applied as a guideline for designing the user interface itself. Such conclusions have a direct impact on our Clover architectural model. Before presenting our model, we show the main characteristics of three architectural models on which our model relies.

SOFTWARE ARCHITECTURES

This section contains the description of three architectural models, Arch, Dewan's generic architecture and PAC*. The three models serve as foundations for our Clover architectural metamodel.

Arch model

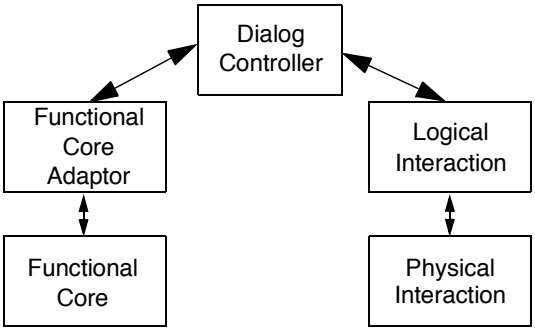


Figure 4: Arch model.

Software architectures for groupware are often based on architectural models for single-user interactive systems. In this field, the Arch model [3], an extension of the Seeheim model [10], is a reference model that defines a canonical functional decomposition of an interactive system. Arch advocates domain dependent components as well as user interface components. As shown in Figure 4, the Arch model breaks up an interactive system into five layers or components:

- The Functional Core component implements domain-specific concepts and functions, in a presentation independent way. The data structures managed by this component are domain objects.
- The Functional Core Adapter component serves as a mediator between the Dialog Controller and the Functional Core. Data exchanged with the Functional Core are the domain objects that the Functional Core exports to the user. Data exchanged with the Dialog Controller are conceptual objects. These define perspectives on domain objects intended to match the user's mental representation of domain concepts.
- The Physical Interaction component denotes the underlying platform, both software and hardware (interaction devices). It supports the physical interaction with the user and corresponds to the services of a User Interface toolkit.
- The Logical Interaction component implements the perceivable behavior of the application for outputs as well as inputs. It relies on the Physical Interaction component. The Logical Interaction component is usually compared to an abstract User Interface toolkit. It serves as an adapter between the Physical Interaction component and the Dialog Controller.
- The Dialog Controller component is the keystone of the arch. It has the responsibility for task-level sequencing. It manages both conceptual objects and presentation objects.

Dewan's Generic Architecture for Groupware

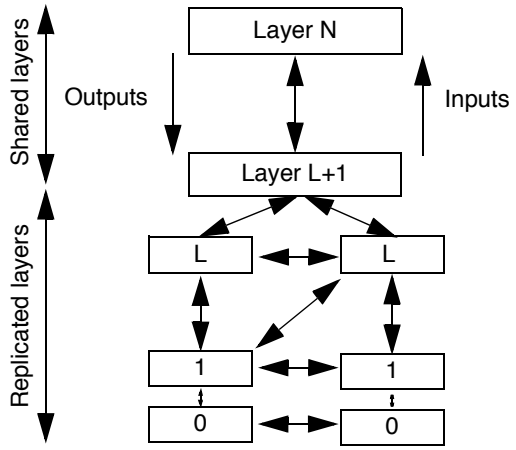


Figure 5: Dewan's generic collaborative architecture.

Dewan's generic collaborative architecture [8] can be seen as a generalization of the "zipper model" [20] and the Arch model. Dewan's architecture structures a groupware application into a variable number of layers from the domain-dependent level to the hardware level. A layer is a software component corresponding to a specific level of abstraction. As shown in Figure 5, the top-most layer corresponds to the semantic layer while the bottom-most layer corresponds to the hardware layer. To make a parallel with the Arch model, the top-most layer coincides with the Arch Functional Core component and the bottom-most

layer coincides with the Arch Physical Interaction component.

The overall structure is composed of a stem and several branches. The stem, as shown in Figure 5, is composed of shared layers (layers L+1 to N). A branch is composed of replicated layers (layers 0 to L). The layer L of Figure 5 defines the branch point of replicated or versioned layers [8]. A branch contains private layers for each user of the application. Moreover the objects managed by layers of a branch are all private objects of a user. Conversely, shared layers and shared objects are public. The layers communicate with each other using two types of events: interaction and collaboration events. Interaction events denote events sent up (input events) and down (output events) between layers. Events sent between layers of different branches are collaboration events.

In the rest of the discussion, we shall often use the terms layer and component interchangeably. Although a software component can encompass several layers, for simplicity we assume that there is one layer per component.

PAC* model

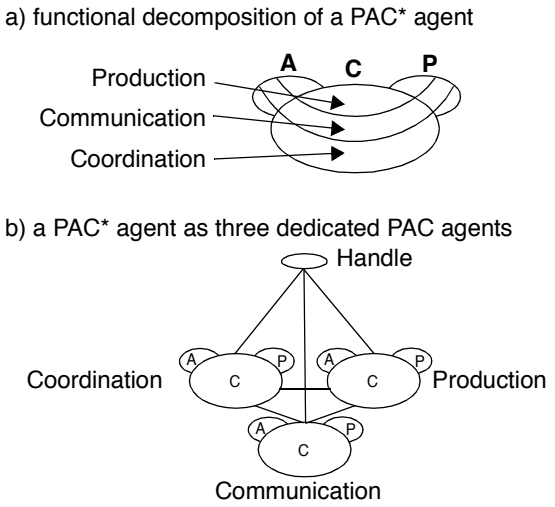


Figure 6: PAC* architectural model.

PAC* [5] is the collaborative extension of the PAC-Amodeus hybrid architectural model [18]. PAC-Amodeus reuses the main components of the Arch model and populates the Dialog Controller with PAC agents. A PAC agent is composed of three facets:

- a Presentation facet which defines the user interface of the agent,
- an Abstraction facet which manages the domain concepts, and,
- a Control facet which manages the links and constraints between its two surrounding facets (i.e., the Presentation and the Abstraction facets) as well as its relationships with other agents.

A PAC* agent, as shown in Figure 6 (a), is a PAC agent in which each facet is structured along with the three functional classes of the groupware clover. Alternatively, a PAC* agent can be seen as a cluster of three PAC agents, each agent dedicated to one functional class of the groupware clover. The cluster of PAC agents represented in Figure 6 (b) is a hybrid form of a PAC* agent: the three clover agents can communicate with each others via their Control facets and a single handle is in charge of communication with the external world (i.e., other agents).

CLOVER ARCHITECTURE

The Clover architectural metamodel relies on the three above models. From our metamodel, several Clover architectural models can be derived. In this section we present one Clover architectural model derived from the metamodel, the latter being presented in the next section.

Description

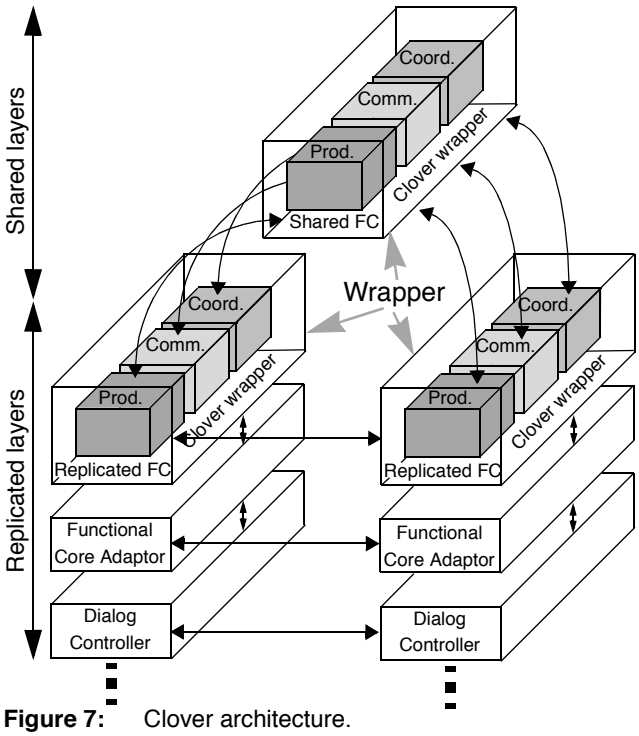


Figure 7: Clover architecture.

The Clover architectural model of Figure 7 is derived from Dewan's model using the functional layers of the Arch model [3]. Nevertheless instead of the five layers of the Arch model, our model advocates six layers: the Arch Functional Core (FC) component is split into two components: the shared and replicated FCs. In Figure 7, to avoid a visual overload, we have not drawn the Arch Logical and Physical Interaction components, which are located below the Dialog Controller in the stack. In addition, some arrows, between layers of the two branches are not drawn.

Having defined the canonical component decomposition, let us now consider the global architecture made of shared and replicated components. All the Arch components are replicated components except the shared FC. In this model, we focus on the functional components and we have made no assumptions about the Dialog Controller and the Presentation Components. Compared to the PAC* model, in which the FC is a completely shared layer, the FC is split here in two layers: one is replicated and private; the other is shared and public. The latter provides shared domain-

dependent objects and functions that are manipulated by all the users during the interaction. Conversely, the replicated FC maintains the state of a single user and manages private domain-dependent objects. As in Dewan's model, the collection of shared and user's private objects defines the interaction state for a particular user. This point is illustrated with the example of the software design of CoVitesse in the last section.

The originality of this architecture comes from the Clover decomposition of the Shared and Replicated FCs. The FCs are made of a wrapper which embeds three sub-FCs, each sub-FC being dedicated to one of three functional classes of the groupware clover. Each Clover FCs manipulates semantic objects dedicated to one of the Clover functionalities and performs specific processing functions on their objects. The wrapper approach is similar to the software component oriented programming in CORBA. The design rationale for the wrapper is threefold:

- The wrapper encapsulates the three Clover FCs and acts as a functional glue. The wrapper is also responsible for communication with other layers. Because the wrapper is a software interface that hides the Clover partitioning, it enables communication with non-clover aware layers. For example the shared FC wrapper serves as a mediator between the three Clover shared FCs and the Functional Core Adapter. The wrapper allows us to stack a Clover partitioned component with a component that is not collaboration aware or Clover aware.

- The wrapper maintains a common state, i.e. the common semantic objects, among the production, communication and coordination FCs. Such objects common to the three Clover spaces have been identified and illustrated in the section entitled "Clover design model".

- The wrapper provides functionalities including the system services and single-user functionalities, which are not intrinsically collaboration aware or Clover aware.

The communication between layers in the stack and between peer layers follows the same rules as described in Dewan's model. Moreover interaction and collaboration events are categorized into three classes of events: production, communication and coordination events. This enables direct communication between Clover FCs. Nevertheless the model maintains a general event for communication between layers that are not Clover aware.

Finally, we have explained in a previous section that the Clover model should not be used as a guideline for designing the user interface. As a consequence, we only partitioned the FC component according to the three Clover functional classes and not the presentation components (Logical and Physical Interaction components). Furthermore it is important to note that the model is one example of a model derived from our metamodel: in this model we have unzipped the architecture at the FC layer, the branching point being the Replicated FC. Different branching points are possible. Fixing the branching point is a design issue and defines the versioning/replication architectural degree, as defined in [8]. In our Clover architecture, the Replicated FC being the branching point, the corresponding replication degree is high. The Clover model therefore allows relaxed WISIWYS (What I See Is What You See). Indeed a low replication (i.e. an architecture unzipped at the Physical or Logical Interaction layers) implies strict WISIWYS.

An Extension of PAC*

Our Clover architectural model is complementary to the PAC* model. Indeed in the PAC* model, a PAC* agent of the Dialog Controller exchanges events with the Functional Core via the Functional Core adapter. In our Clover model, if the Dialog Controller is populated by PAC* agents (Figure 6), each Clover Abstraction facet of a PAC* agent will communicate with the corresponding Clover Replicated FCs. But for a high modifiability of the code, the communication will not be direct but will be via the Functional Core Adapter that is not necessarily Clover aware, and then via the Replicated FC wrapper.

Moreover in Figure 6(b), a PAC* agent contains a handle. This latter is responsible for managing the communication with the external world, i.e. the other agents. The role of the handle is one of the roles fulfilled by a wrapper in our Clover model. The wrapper manages communication between the three Clover FCs and their surrounding layers.

CLOVER METAMODEL

Having presented one Clover architectural model, we now describe the generalization: the Clover metamodel.

Description

The Clover architectural metamodel of Figure 8 structures a groupware application into a variable number of layers, as opposed to the previously presented Clover model which is based on the five layers of the Arch model. Compared with the previous Clover model of Figure 7, Layer L corresponds to the replicated FC and Layer L+1 to the Shared FC.

The originality of this metamodel comes from the Clover decomposition of the software units that compose the branches and the stem. As shown in Figure 8, each unit of the model contains a wrapper that encapsulates three Clover sub-components: production, communication and coordination sub-components. Interaction and collaboration events exchanged between layers are subsequently categorized into three classes of events: production, communication and coordination events. The metamodel nevertheless authorizes a general event for communication between layers that are not Clover aware. In Figure 8, in order to avoid a visual overload, we have not drawn every arrow corresponding to an exchange of events between layers.

As explained in the previous section, one role of a layer wrapper, which encompasses a Clover's structure, is to hide the Clover breakdown from the other layers. Communication between a Clover aware layer and a non-Clover aware layer is thus possible. Indeed, a layer does not necessarily implement a collaboration semantic and is therefore not necessarily decomposed according to the three Clover functionalities. This is usually the case for the lowest layers, which are dedicated to the physical and logical interaction with a user. For example, as shown in Figure 8, Layer 0 (the leaf of the branch) which

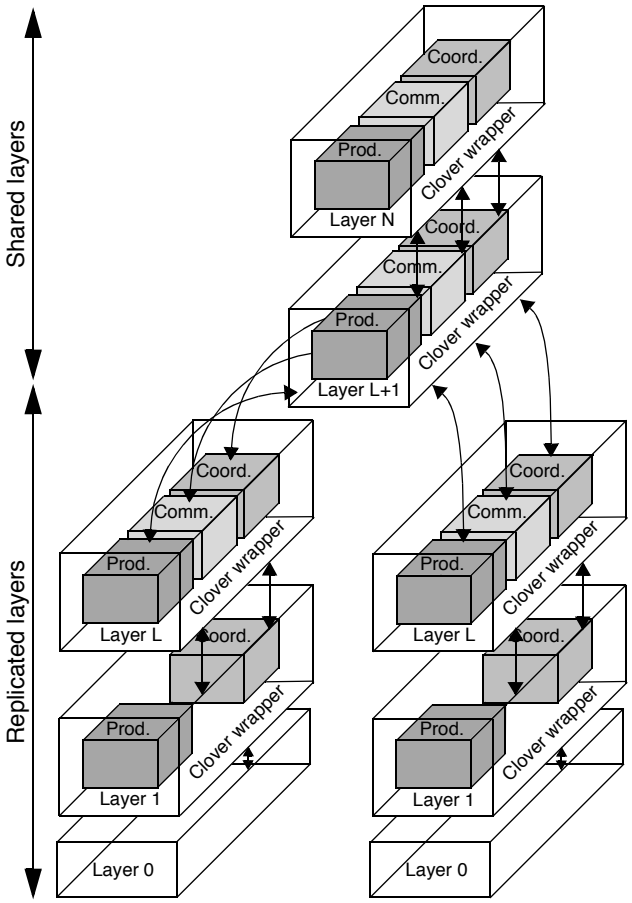


Figure 8: Generalized Clover architecture.

corresponds to the hardware layer, is typically a non-Clover aware layer.

Moreover the full decomposition of a layer according to the three Clover functionalities is not mandatory. A layer may include Production and Coordination functionalities and no Communication functionality: this is for example the case for Layer 1 in Figure 8.

BENEFITS AND PROPERTIES

The Clover functional partitioning establishes a direct mapping between the design concepts and the software architecture modeling. This partitioning provides a clear guide in the organization of the functionalities identified during the design phase. In addition it is complementary to the traditional partitioning of functionalities into components, where each of them corresponds to one level of abstraction as for instance in Dewan's architecture or Arch model. Indeed, for a component corresponding to one level of abstraction, the Clover metamodel advocates three sub-components dedicated to production, communication and coordination. Such a highly modular implementation satisfies the modifiability, reusability and extensibility properties. These properties are crucial for supporting a user-centered design. For example, in the CoVitesse system, it would be possible to add a video service as a new communication channel by modifying the Shared and Replicated Functional Core dedicated to communication without endangering the other components. Moreover modularity is a well-known software engineering mechanism for reducing the cost of development. Based on our own experience, applying the Clover architectural model to the CoVitesse software design made it easy to develop parts of the system without having a global view of the overall code. For example, the three Shared Functional Cores and the three Replicated Functional Cores, described in the following paragraph, have been developed and tested independently.

The metamodel does not fix the number of layers: layers can be used to increase the separation of functionalities i.e. to increase the modularity of the code. This approach is orthogonal to the Clover breakdown of a layer.

The Clover breakdown of a layer can be partial. The metamodel therefore allows a pile of several layers with different Clover breakdowns. For example, instant messaging (communication functionality) is usually used for managing coordination among users [24]. To implement such a coordination which is based on communication, a shared layer dedicated to coordination is stacked on top of a shared layer that contains communication functionalities only.

In our metamodel the coordination, production and communication functionalities are all treated the same way. This is in contrast to previous studies [8][16] where the coordination functionalities are treated as special whereas the production and communication functionalities are not. For example, in Dewan's architecture [8], the session manager is a special component that should be located outside the architecture. This external component manages the users and the groups during a session. This component is therefore responsible for creating the tree and for connecting the branches with the stem. In the Clover metamodel, the session manager is a coordination component located in a stem layer. Indeed we argue that there is always a shared layer in the conceptual software architecture. The conceptual architecture is then mapped into an implementation architecture by assigning processes to components; the processes in turn must then be assigned to hosts. Different approaches to distribution can be applied to a conceptual architecture. Distribution and layer decomposition (or software component decomposition) are two orthogonal mechanisms. In particular, a shared layer at the conceptual level does not imply a central site at the implementation level.

Finally the Clover metamodel extends Dewan's generic architectural model [8]. An architecture based on Dewan's generic model is described by an awareness degree. The awareness degree is equal to the level of the highest layer that is collaboration-aware. Extending the awareness degree, we define a production-aware degree, a communication-aware and a coordination-aware degree. Such extensions will allow a more precise classification of existing systems from an architectural point of view, than the one presented in Table 9.1 in [8].

CLOVER MODEL IN PRACTICE: COVITESSE SOFTWARE DESIGN

In this section, we show how the Clover model can be used in practice by presenting the Clover software design solution of CoVitesse.

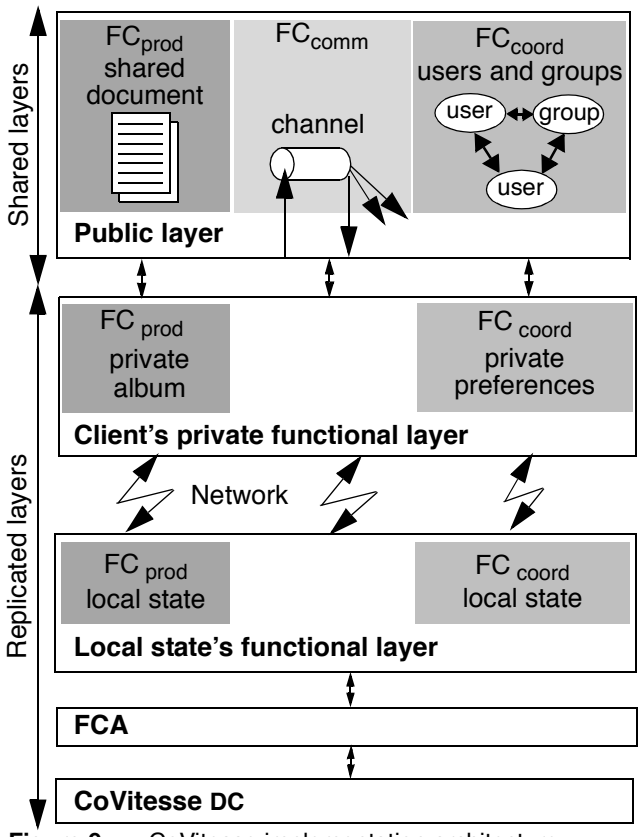


Figure 9: CoVitesse implementation architecture.

Figure 9 illustrates the application of the Clover model to the software design of CoVitesse. At the conceptual level, we applied the Clover model of Figure 7 that includes a replicated Functional Core and a shared Functional Core. At the implementation level, CoVitesse is a client/server application written in Java. The shared components are on the server side and the replicated components on both sides (client and server). The communication between clients and the server is based on network streams.

We describe the CoVitesse architecture from the highest layer (top of Figure 9) to the lowest. The highest layer, the Shared Functional Core (FC), is made of a FCprod, a FCcomm, a FCcoord, and a meta-server. The meta-server is a wrapper: it encompasses the three Clover FCs and manages the access to the underlying network.

- *Shared Production Functional Core (FCprod)*:
  This component manages concepts related to production, represented as a document icon in FCprod of Figure 9. These concepts are document, reference, album and catalogue. A document is an elementary unit that is manipulated by a group. A reference is a pointer on a document. An album is a container, created and managed by authors: an album contains documents and references. A catalogue is a set of references. These four concepts are implemented as Java interfaces. There are several functions available including: the creation or obtaining of a catalogue, creation of an album, addition or removal of an author from an album, addition of a document in an album, modification of the content of an album. In CoVitesse, a reference is an URL, a document is the content of a web page, i.e. an HTML document, a catalogue is a set of URLs resulting from a query submitted to a search engine and an album is a set of URLs created by authors.

- *Shared Communication Functional Core (FCcomm)*:
  This component manages communication channels that are represented by a tube in FCcomm of Figure 9 The communication channel is the only concept manipulated by FCcomm. A client can subscribe or unsubscribe to a communication channel and can post a message on the channel. A channel is implemented in Java. In CoVitesse, the unique communication channel is a chat channel, which can be compared to a mailing list: when a user posts a message on this channel, the message is broadcast to all the subscribers.

- *Shared Coordination Functional Core (FCcoord)*:
  This component has the responsibility for coordinating users and groups. It first hosts a database of users and groups. Therefore this component allows persistent access to the description of the users and the groups. Moreover, this component provides mechanisms for concurrency control and for consistency control. For instance, it synchronizes a concurrent access for two members to modify group preferences. This component also notifies all the clients, for consistency purposes, when a user modifies her/his own private preferences. Consequently the concepts manipulated by FCcoord are related to users and groups; these concepts are implemented as Java interfaces.

  - *User*: Two functions are implemented to read and modify information about a user as well as two functions to read and modify the access rights to those pieces of information. This implementation allows the developer to define the user's description according to the nature of tha groupware application. In CoVitesse, a user is described by a name, a password, a geometrical shape, filtering preferences, an E-mail address, etc. The user is identified by her/his username and password.

  - *Group*: A Group is defined by, a set of members, rights for joining a group and a minimal set of functions. These functions include setting or obtaining information about a group, creating a group, joining a group (a user sends a request to all the members who accept or do not accept the newcomer), leaving a group. In CoVitesse, a group is described by: a group name, a color, filtering preferences, a navigation type and navigation preferences.

As shown in Figure 9, the layer under the Shared Functional Core is the Replicated Functional Core. At the implementation level, the Replicated Functional Core is split into two parts: one part on the server side and one part on the client side. The part on the client side maintains a duplicated version of the current session, which is originally saved on the server side. This mechanism of caching data is used to increase performance at runtime. The Shared Functional Core (on the server side as well as on the client side) is made of two Clover FCs, namely FCprod and FCcoord. These two Clover components

maintain private data owned by the client. This design solution enables the server part to provide persistent access to those private data.

- Replicated FCcoord maintains the user's private preferences and communicates with other clients through the Shared Functional Core. Both replicated and shared components dedicated to coordination manipulate the same data about a user. Nevertheless only the replicated component is allowed to modify private data such as the username and password, while the shared component can read the private data about a user when it is published.
- Replicated FCprod hosts the user's private album containing the collected URLs. Moreover FCprod has access to the group album managed and protected by the Shared Functional Core dedicated to production.

In the current implementation of CoVitesse, there is no Replicated Functional Core dedicated to communication. Consequently the wrapper transfers communication events directly, in a bidirectional way, between the shared FCcomm and the Functional Core Adapter (FCA in Figure 9). However, future developments are planned for implementing this missing facet. For example, private aliases of recipients such as mailing-lists can be managed by the shared FCcomm. Moreover, we have planned to extend the shared FCcomm in order to allow private channels for chatting within a defined group of users.

In the rest of the architecture, the components do not implement a Clover structure. These components are the Functional Core Adapter (FCA), the Dialog Controller (DC) and the Logical and Physical Interaction components. In Figure 9, the Logical and Physical Interaction components are not drawn. They should be drawn under the CoVitesse DC.

VALIDATION OF THE CLOVER METAMODEL

Software tools for the construction of groupware will not eliminate architectural issues as long as the construction of groupware requires programming. Developers and maintainers of groupware need to rely on architectural models:

- for identifying software components,
- for organizing their interconnections,
- for reasoning about components and interconnections,
- for modifying and maintaining them in a productive way,
- for verifying ergonomic and software properties.

Nevertheless an architectural design is neither inherently good nor bad, but is conformant (or not) to a set of specific properties [4]. The SAAM method shows how to assess an architectural design along these lines [13]. Shaw et al. demonstrate that different software architecture models have different strengths and weaknesses and therefore lead to different architectural design solutions with significantly different software properties [22]. Therefore, software architecture models should be chosen in accordance with the system requirements and a software architecture model is suitable for a sub-set of properties. By “suitable” we mean that the model helps to either verify or assess a property. In this paper, we presented a set of properties that the Clover architectural metamodel verifies. In particular, the Clover metamodel accommodates style heterogeneity by allowing a pile of several layers with different Clover breakdowns, modifiability, reusability as well as extensibility.

Another avenue to validate an architectural model is to show that it corresponds to an implicit software practice: the model therefore makes explicit programmers savoir-faire. To do so, in [15], we studied the code of three groupware applications: a mediaspace developed in our team [6], a shared text editor NetEdit [25] and a collaborative ping-pong game [12]. By reverse engineering based on the code of the three applications, we showed that the modules of the existing code are organized according to our Clover metamodel.

Finally the generality of the metamodel (and in particular the variable number of layers and the possible partial Clover breakdown of a layer) is a required property for embracing the diversity and the novelty of the technical problems raised by groupware. Indeed the generality of an architectural model stems from its capacity to adapt to a variety of constraints according to a sound design rationale. So far the Clover metamodel has proven useful in triggering the right software design questions and in providing operational answers for several cases, both for forward design and engineering as well as for reverse design and engineering. On the one hand, the Clover metamodel guides the development of a future system such as CoVitesse; on the other hand it helps in understanding the organization of existing code. We do feel however that Clover metamodel patterns and heuristic rules need to be devised to respond to recurrent problems in multi-user systems. Indeed the unavoidable generality of the metamodel makes its application difficult. As done for our PAC-Amodeus software architectural model [17], providing patterns and rules will serve as guidelines for applying the Clover architectural metamodel.

CONCLUSION

We have presented a new conceptual architectural metamodel, the Clover architectural metamodel. The Clover metamodel results from the combination of the layer approach of Dewan's generic architecture with the functional decomposition of the Clover design model. The Clover architectural model has the following properties:

- By applying a Clover functional breaking up of each layer of our metamodel, we make system design issues explicit within the software architecture. Indeed the software architecture modeling is a design activity at the turning point between two worlds: the system design field and the programming field. Because of its interlinking location, software architecture must take into account design issues and properties of the two worlds.
- The Clover metamodel results from a motivated combination of existing architectural models selected for the complementarity of their properties. In particular our metamodel inherits the generic property from the layer approach of Dewan's generic architecture. The metamodel also refines each layer and event according to the three

Clover facets. This Clover breakdown of the layer increases the modularity of the code.

One architectural model derived from the Clover metamodel has been applied to the software design of the CoVitesse system. This derived model (Clover model), that has been presented in the paper and illustrated using CoVitesse, refines the Functional Core in terms of replicated and shared components. Moreover the Functional Core components are partitioned into three Clover sub-components dedicated to production, communication and coordination, the three sub-components being encapsulated by a wrapper.

Future developments consist of completing a Clover platform for developing groupware. The platform illustrates our Clover architectural metamodel by providing software components dedicated to the three Clover facets. The platform implements reusable encapsulated mechanisms that facilitate the implementation of any groupware. The CoVitesse system has both implemented using our platform. We are currently testing the robustness of the platform.

ACKNOWLEDGMENTS

This work has been partly supported by the French Ministers of Research and Industry Contract SIRII. Special thanks to G. Serghiou for help with style and English Grammar.

REFERENCES

1. Anderson, G., Graham, T.C.N., Wright, T., Dragonfly: Linking Conceptual and Implementation Architectures of Multiuser Interactive Systems, in *Proceedings of ICSE'00*, 2000, p. 252-261, ACM Press.
2. Baecker, R.M., Nastos, D., Posner, L.R., Mawlby, M.K., The user-centered iterative design of collaborative writing, in *Proceeding of the Workshop on Real Time Group Drawing and Writing Tools, CSCW'92*, 1992.
3. Bass, L., and al., A Metamodel for the Runtime Architecture of an Interactive System, The UIMS Tool Developers Workshop, in *SIGCHI Bulletin*, vol. 24(1), 1992, p. 32-37, ACM Press.
4. Bass, L., Clements, P., Kazman, R., *Software Architecture in Practice*, 1998, 452 pages, Addison Wesley.
5. Calvary, G., Coutaz, J., Nigay, L., From Single-User Architectural Design to PAC*: a Generic Software Architecture Model for CSCW, in *Proceedings of CHI'97*, 1997, p. 242-249, ACM Press.
6. Coutaz, J., Bérard, F., Carraux, E., Astier, W., CoMedi: Using Computer Vision to Support Awareness and Privacy in Mediaspaces, in *Proceedings of CHI'99,* 1999, p. 13-14, ACM Press.
7. CoVitesse system, http://iihm.imag.fr/demos/CoVitesse/
8. Dewan, P., Architectures for Collaborative Applications, in Beaudouin-Lafon (eds.), *Computer Supported Cooperative Work*, 1999, p. 169-194, John Wiley & Sons Ltd.
9. Ellis, C., Wainer, J., A Conceptual Model of Groupware, in *Proceeding of CSCW'94*, 1994, p. 79-88, ACM Press.
10. Green, M., Report on Dialogue Specification Tools, in *User Interface Management Systems, Eurographics Seminars*, 1985, p. 9-20, Springer Verlag.
11. Graham, T.C.N., Urnes, T., Nejabi, R., Efficient distributed implementation of semi-replicated synchronous groupware, in *Proceedings of UIST'96*, 1996, p. 1-10, ACM Press.
12. java, http://www.javasoft.com/
13. Kazman, R., Bass, L., Abowd, G., Webb, M., SAAM: A Method for Analyzing the Properties of Software Architectures, in *Proceedings of ICSE'94*, 1994, p. 81-90, ACM Press.
14. Laurillau, Y., Synchronous Collaborative Navigation on the WWW, in *Extended Abtracts of CHI'99*, 1999, p. 308-309, ACM Press.
15. Laurillau, Y., Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe : le modèle et la plate-forme Clover, *Ph.D. dissertation*, University of Grenoble, France, 2002, 264 pages.
16. Li, D., Muntz, R., COCA: Collaborative objects coordination architecture, in *Proceedings of CSCW'98*, 1998, p. 179-188, ACM Press.
17. Nigay, L., Coutaz, J., Software architecture modelling: Bridging Two Worlds using Ergonomics and Software Properties, *Formal Methods in Human-Computer Interaction*, 1997, p. 49-73, Springer Verlag.
18. Nigay, L., Coutaz, J., A Generic Platform for Addressing the Multimodal Challenge, in *Proceedings of CHI'95*, 1995, p. 98-105, ACM Press.
19. Nigay, L., Vernier, F., Design Method of Interaction Techniques for Large Information Spaces, in *Proceedings of AVI'98*, 1998, p. 37-46, ACM Press.
20. Patterson, J.F., A taxonomy of Architectures for Synchronous Groupware Applications, in *Workshop on Software Architectures for Cooperative Systems of CSCW'94*, 1994, p. 79-88, ACM Press.
21. Roseman, M., Greenberg, S., Building Real-Time Groupware with GroupKit, A Groupware Toolkit, in *ACM ToCHI*, vol. 3(1), 1996, p. 66-106, ACM Press.
22. Shaw, M., Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*, 1996, 242 pages, Prentice Hall.
23. Salber, D., De l'interaction homme-machine individuelle aux systèmes multi-utilisateurs. *Ph.D. dissertation*, University of Grenoble, France, 1995, 305 pages.
24. Whittaker, S., Nardi, B., Bradner, E., Interaction and Outeracion Instant Messaging In Action, in *Proceedings of CSCW'00*, 2000, p. 79-88, ACM Press.
25. Zafer, A., NetEdit: a Collaborative Editor, *Master of Science*, University de Virginia, USA, 2001, 82 pages.

Synchronous Collaborative Navigation on the WWW

Yann Laurillau
Laboratoire CLIPS/IMAG
BP 53, 38041 Grenoble Cedex 9, FRANCE
+33 4 76 51 44 40
Yann.Laurillau@imag.fr

ABSTRACT
We describe an ongoing research effort in designing synchronous collaborative navigational techniques on the WWW. The goal of the research program is to provide usable tools to collaboratively search information on the WWW. We begin by studying collaborative navigational techniques in the search results space defined by a WWW search engine. We first present our design method. We then characterize four types of collaborative navigation that are embedded in the current prototype, Co-Vitesse.

Keywords
Collaborative/Social Navigation, CSCW, Design Method.

INTRODUCTION
Seeking information through the World Wide Web is difficult. Users are "lost in hyperspace" [1] and have difficulties "in gaining an overview of the material" [3]. In addition, navigational behavior on the web often relies on the expertise of other users. One typical social behavior on the web consists of asking a colleague about information we assume the other has the pointer to: Such observed social behaviors show that web users are striving for collaboration. Several collaborative tools, including chat room, mail or news for exchanging URLs [2], can help the users in their information retrieval but are not integrated in browsers. Moreover most of these tools only support asynchronous collaboration.

Our research program consists of designing and building synchronous collaborative tools for seeking information on the web. Indeed, as motivated above, the need for systems on the web that support coordination, communication and shared production between and within groups increases markedly. We begin by focusing on synchronous navigation in the search results space: Such tools are dedicated to a group of users searching information. Examples of usage include:

- A group of master students have to write a bibliography about a particular subject.
- A group of workers are responsible for monitoring technological development in a firm.

DESIGN METHOD
As shown in Figure 1, our design method is based on a conceptual study of navigational tasks as well as

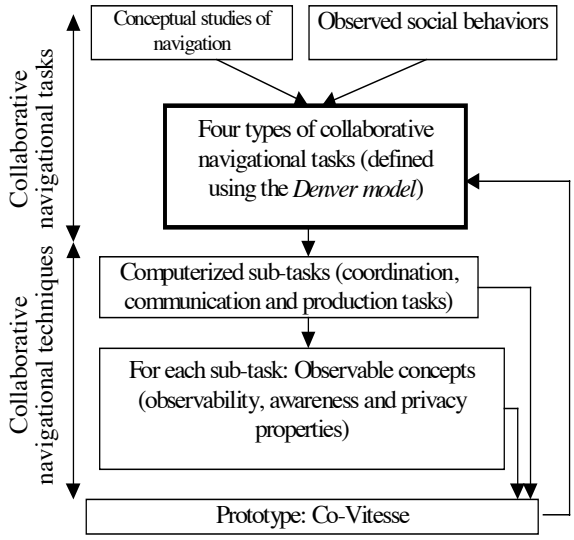


Figure 1: Design method.

observed social behavior of users while searching for information. The results of these studies and observations enable us to identify four types of collaborative navigational tasks that are described in the following paragraph. In the next step, we refine the four types of navigation in terms of sub-tasks that are organized according to the three spaces: communication, coordination and production spaces. We then address the awareness and privacy issues by studying relevant concepts to be made observable by the users for each task.

COLLABORATIVE NAVIGATIONAL TASKS
Our work uses the observed collaborative activities described in [6]. In that study, work-related collaborations between users of OPAC terminals in Lancaster University Library were observed. From these observed social behaviors, we identify four types of navigational tasks by applying the Denver model [5], which involves characterizing the interactive situations and protocols of each navigational task:

- Relaxed navigation: It defines an open group without a leader. No objective is clearly defined and it is completely informal. Users navigate independently. Nevertheless at any time the user can give the control to someone else if she/he needs help.

- Cooperative navigation: There is again no leader, but members have a common objective that was

planned in advance. Each member has to explore a predefined section of the space. All the members can accept or decline a newcomer. Each member automatically receives the current findings of the group (marked web pages) when leaving the group.

- Planned navigation: The leader decides the objective of the session and has the responsibility for assigning a section of the shared information space to each member. The member can individually work on her/his assigned section until the leader decides to gather the findings of all the members. In addition the leader can group together (" teleport") all the members in a particular location of interest for the group.

- Guided tour: The guide freely navigates and the members follow the guide. A user can join a guided tour at any time. The guide is by definition an expert of the information space and/or the domain.

These four types of navigation are implemented in the Co-Vitesse system that is described below.

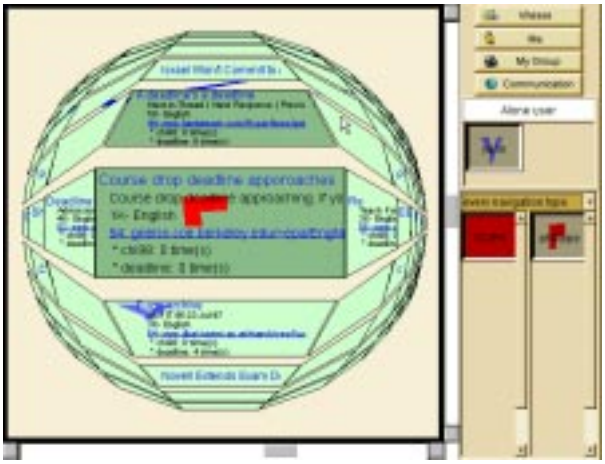


Figure 2: A snapshot from the Co-Vitesse system: Main window.

APPLICATION

The Co-Vitesse system enables the users to navigate synchronously on the WWW. The four types of navigational tasks are explicitly available to the users. Co-Vitesse is based on a single-user application, Vitesse [4]. The VITESSE system visualizes the results of a query submitted to a search engine on the WWW. As shown in Figure 2 (left part), the overall graph structure of the results is displayed (six visualization techniques are available): Each retrieved page (node) and their links are displayed. One retrieved page or node is displayed by a polygon. The selection of a node (double click) enables the user to access the web page.

When starting a session, a Co-Vitesse user defines his/her avatar by a shape and a name. The user then either selects an information space or specifies a query that will be sent to a selected search engine. The results of the query define a new information space. The user can then navigate in the information space, observe other users (in Figure 2, two users are navigating), create or join a group.

On the right side of the space, two palettes are displayed. One palette displays all the single users and the groups in the information space. Selecting a group will make the corresponding members observable on the information space and in the second palette. The user can then opt to only observe some of the members of the group by selecting their corresponding icons. A group is represented by a color and a name. If a user belongs to a group, her/his shape will be displayed in the color of the corresponding group; else a predefined color is automatically assigned to a user.

Additional windows are available through the menu in the top right corner. The windows are organized according to three sets of tasks: single-user tasks, group tasks and communication tasks (chat room). One of the main single-user tasks is the creation of a group: At any time, a single user can create a group, its objective and its style of collaborative work by selecting one of the four types of navigation.

At the end of the session each user collects the findings of the group gathered in his/her bookmark.

CONCLUSION

The next step in our work is to experimentally test Co-Vitesse in order to evaluate the four types of navigation. We plan to ask a group of master students of our University to perform scenarios while having wizards serving as guides (guided tour) and leaders (planned navigation).

ACKNOWLEDGMENTS

This work is supported by the French Ministers of Research and Industry Contract SIRII. Many Thanks to L. Nigay my Ph.D. supervisor, F. Vernier and P. Renevier for their help in developing the Vitesse and Co-Vitesse systems.

REFERENCES

1. Conklin, J. Hypertext : an introduction and survey. *IEEE Computer*, 20(9), September 1987, 17-41

2. Dieberger, A. Supporting social navigation on the World Wide Web. *International Journal of Human-Computer Studies*, 46, 1997, 805-825.

3. Hammond, N., Allinson, L. Extending Hypertext for Learning : An investigation of Access and Guidance Tools, in *Proceedings of HCI'89* (Nottingham, Sept. 1989), Cambridge University Press, 293-304.

4. Nigay, L., Vernier, F. Design Method of Interaction Techniques For Large Information Spaces, in *Proceedings of AVI'98* (Aquila, May 1998), ACM Press, 37-46.0

5. Salvador, T., Scholtz, J., Larson, J. The Denver Model for Groupware Design. *SIGCHI Bulletin*, 28(1), Jan. 1996.

6. Twidale, M., Nichols, D., Paice, C. Browsing is a collaborative process. *Information Processing & Management*, 33(6), 1997, 761-83.

Résumé

Avec les avancées des réseaux et les progrès en miniaturisation, nous passons d'une informatique monolithique à un large choix de services et de dispositifs dont la disponibilité varie dans le temps et dans l'espace. Par leur succès auprès du grand public, nous avons dépassé le stade de l'acceptation : les systèmes interactifs, à l'image du smartphone, sont de nouveaux couteaux suisses ; ils nous assistent dans de nombreuses activités. Cependant, cette grande variété et variabilité conduit à une expérience utilisateur fragmentée.

Mes travaux de recherche relèvent du domaine de l'ingénierie de l'Interaction Homme-Machine. Ils traitent le problème de la fragmentation sous l'angle des systèmes symbiotiques, systèmes capables d'augmenter nos capacités humaines (par exemple, cognitives, physiques, sensori-motrices) tout en gagnant en intelligence et en connaissance sur nous-mêmes par leurs capacités perceptives et décisionnelles. Mon défi est d'assurer l'ingénierie de la fluidité interactionnelle, une propriété des systèmes symbiotiques assurant une interaction cohérente en tout contexte d'usage et basée sur une capacité de polymorphisme et de polyvalence.

Mes contributions se destinent aux concepteurs des systèmes symbiotiques. J'étudie l'ingénierie de la fluidité interactionnelle sous l'angle de la flexibilité logicielle. Sur un espace de conception génératif, je classe mes contributions (des notations, modèles, et boîtes à outils logicielles) et montre qu'elles couvrent l'interaction multiutilisateur, multimodale et contextuelle. Les perspectives sont nombreuses.