HAL
open science

Collecticiel et multimodalité : spécification de l'interaction la notation COMM et l'éditeur e-COMM

Frédéric Jourde

To cite this version:

Frédéric Jourde. Collecticiel et multimodalité : spécification de l'interaction la notation COMM et l'éditeur e-COMM. Interface homme-machine [cs.HC]. Université de Grenoble, 2011. Français. NNT : 2011GRENM018 . tel-00618919v2

HAL Id: tel-00618919

https://theses.hal.science/tel-00618919v2

Submitted on 12 Sep 2011

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.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Frédéric JOURDE

Thèse dirigée par **Laurence NIGAY** et
codirigée par **Yann LAURILLAU**

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale MSTII**

Collecticiel et Multimodalité : spécification de l'interaction, la notation COMM et l'éditeur e-COMM

Thèse soutenue publiquement le 9 juin 2011,
devant le jury composé de :

Mme, Dominique, RIEU
Professeur des Universités, Université Pierre-Mendès-France - Grenoble II, Président

M, Rémi, BASTIDE
Professeur des Universités, Centre Universitaire Jean-François Champollion, Castres, Rapporteur

M, Patrick, GIRARD
Professeur des Universités, Université de Poitiers, Rapporteur

M, Didier, BAZALGETTE
Responsable du domaine scientifique Hommes et Systèmes à la DGA Recherche et Innovation, DGA Direction Générale de l'Armement, Examinateur

Mme, Angélica, LEAL
Ingénieur consultant en Ergonomie, Société BERTIN Conseil - Pôle Ergonomie et Facteurs Humains, Examinateur

M, Yann, LAURILLAU
Maitre de conférences, Université Pierre-Mendès-France - Grenoble II, Directeur de thèse

Mlle, Laurence, NIGAY
Professeur des Universités, Université Joseph Fourier – Grenoble I, Directeur de thèse

Thèse financée par la Direction Générale des Armements.

Remerciements

Au président du jury, Dominique Rieu, qui m'a fait l'honneur de présider le jury de cette thèse, dont la soutenance fut l'un des évenements les plus importants de ma vie.

A mes maitres de thèse, Laurence Nigay et Yann Laurillau, que je remercie de la confiance qu'ils ont placé en moi, et qui a été un moteur de mes travaux dans les moments les plus délicats. Ils m'ont guidé par leur conseil pendant ma thèse et lors de la rédaction.

Aux rapporteurs du jury, Girard Patrick et Rémi Bastide, pour l'interet qu'ils ont porté à mon travail et les nombreux conseils qu'ils m'ont promulgués pendant et après la soutenance ce qui m'a permit d'aboutir à cette version définitive du mémoire.

Aux membres du jury, Angélica Léal et Didier Bazalgette avec qui j'ai eu le plaisir de collaborer dans le cadre passionnant du PEA Facteur Humain et Partage d'Autorité.

A mes parents et amis qui m'ont soutenu pendant ces années de recherche. Je les remerie pour leurs écoutes, leurs conseils, leurs disponibilités et leurs patiences, qui m'ont donné la force de mener au bout mes travaux.

A Aurélie qui partage ma vie depuis plusieurs années, et qui m'a apporté un soutien sans faille et une écoute compréhensive pendant ces années de recherche.

Sommaire

Introduction 15
1 Domaine 15
2 Constat : réalisation de collecticiels multimodaux 16
3 Objectifs 17
4 Démarche adoptée 19
5 Contribution conceptuelle 19
6 Contribution logicielle 20
7 Structure du mémoire 20
Chapitre 1 : Systèmes multiutilisateurs et multimodaux 23
1 Systèmes multiutilisateurs 23
1.1 Classification espace-temps 24
1.2 Classification fonctionnelle : Modèle du trèfle 26
1.3 Activité multiutilisateur : coopération vs collaboration 27
1.4 Groupes, rôles, agents, acteurs et utilisateurs 28
1.5 Espace de conception 29
1.5.1 Conscience de groupe 29
1.5.2 Observabilité, protection de la vie privée 29
1.5.3 Réciprocité 31
1.5.4 WYSIWYS 31
1.5.5 Couplage de l'interaction 31
1.5.6 Viscosité 31
1.5.7 Synthèse 32
1.6 Etude sur le couplage de l'interaction 32
1.6.1 Camion de pompier : une activité coopérative et plusieurs IHMs 33
1.6.2 Les 4 magiciens : une IHM et plusieurs activités 37
1.6.3 Synthèse 39
1.7 Synthèse sur les systèmes multiutilisateurs 40
2 Systèmes Multimodaux 40
2.1 Définition d'une modalité et d'un système multimodal 41
2.1.1 Modalité d'interaction 41
2.1.2 Modalité en entrée ou en sortie 41

2.1.3 Modalité passive ou active ... 41
2.1.4 Système multimodal ... 42
2.1.5 WIMP et Post WIMP ... 42
2.1.6 Apport de la Multimodalité ... 42
2.2 Espace de conception : élements clefs ... 43
2.2.1 Caractérisation d'une modalité ... 43
2.2.1.1 Taxonomie de Bernsen ... 43
2.2.1.2 Modèle de modalités ... 44
2.2.2 Caractérisation des formes de multimodalité ... 44
2.2.2.1 Propriétés CARE ... 44
2.2.2.2 Espace TYCOON ... 46
2.2.3 Synthèse ... 46
2.3 Exemples de systèmes multimodaux ... 46
2.3.1 Complémentarité ... 47
2.3.2 Equivalence ... 48
2.3.3 Redondance ... 49
2.3.4 Synthèse ... 50
2.4 Synthèse sur les systèmes multimodaux ... 50
3 Systèmes multiutilisateurs et multimodaux ... 51
3.1 Exemples de systèmes multiutilisateurs et multimodaux ... 51
3.1.1 Environnement collaboratif d'édition musicale ... 51
3.1.2 Warcraft 3 collaboratif et multimodal ... 52
3.1.3 Les Sim's collaboratif et multimodal ... 53
3.1.4 Tables augmentées distantes ... 53
3.1.5 Poste de commandement pour la gestion de situations d'urgence ... 54
3.1.6 Système de visite réelle et virtuelle ... 55
3.1.7 Poste de commande de drones ... 56
3.1.8 Synthèse ... 57
3.2 Exemple illustratif : WCCM - Warcraft Collaboratif, Coopératif et Multimodal ... 58
3.2.1 Concepts de Warcraft étendu ... 59
3.2.2 Gestion des paysans : tâche individuelle et multimodale ... 59
3.2.3 Création d'unités combattantes : tâche collaborative et multimodale ... 60
3.2.4 Gestion des troupes : tâche coopérative et multimodale ... 60
3.2.5 Synthèse ... 61

4 Sujet et motivations de l'étude 61
Chapitre 2 : Etat de l'art des notations 65
1 Caractéristiques des notations 65
1.1 Caractéristiques générales 65
1.1.1 Domaine d'origine 66
1.1.2 Objectif 66
1.1.3 Nombre et types de vues 67
1.1.4 Forme des représentations 67
1.1.5 Outils logiciels 68
1.2 Caractéristiques liées à l'interaction 68
1.2.1 Description structurée de l'interaction 68
1.2.2 Relations de composition 69
1.2.3 Objet du domaine 70
1.2.4 Contexte d'exécution d'une tâche 70
1.2.5 Actions du système 71
1.3 Caractéristiques liées à la dimension multiutilisateur 72
1.3.1 Coopération et collaboration 72
1.3.2 Coopération 72
1.3.3 Collaboration 73
1.3.4 Objets partagés 73
1.3.5 Répartitions spatiale et temporelle des utilisateurs 74
1.4 Caractéristiques liées à la dimension multimodale 75
1.4.1 Modalité d'interaction 75
1.4.2 Dispositif d'interaction 75
1.4.3 Langage d'interaction 76
1.4.4 Combinaison de modalités 76
1.5 Synthèse des caractéristiques 77
2 Revue des notations existantes 78
2.1 Domaine : Génie Logiciel 79
2.1.1 UML 79
2.1.2 UML-G 87
2.1.3 BPMN 89
2.1.4 Synthèse 91
2.2 Domaine : Interaction Homme-Machine 91

2.2.1 CTT ... 91
2.2.2 K-MAD ... 93
2.2.3 TaskMODL ... 95
2.2.4 CTML ... 100
2.2.5 Web Task Model ... 101
2.2.6 SpIeLan ... 103
2.2.7 UAN ... 105
2.2.8 IOG ... 105
2.2.9 Synthèse ... 106
2.3 Domaine : Travail Coopératif Assisté par Ordinateur ... 106
2.3.1 GTA ... 107
2.3.2 MABTA ... 109
2.3.3 CIAN ... 113
2.3.4 TOUCHE ... 116
2.3.5 CUA ... 118
2.3.6 Orchestra ... 122
2.3.7 Synthèse ... 125
2.4 Domaine : Multimodalité ... 126
2.4.1 ICOM ... 126
2.4.2 ICO ... 127
2.4.3 ICARE ... 129
2.4.4 Dynamo-AID ... 131
2.4.5 CIS ... 133
2.4.6 Synthèse ... 134
2.5 Domaine : Réalité Augmentée ... 135
2.5.1 ASUR ... 135
2.5.2 IRVO ... 138
2.5.3 Modèle d'interaction mixte ... 140
2.5.4 Synthèse ... 141
3 Synthèse globale ... 142
3.1 Synthèse des caractéristiques générales ... 142
3.2 Synthèse pour la spécification de l'interaction ... 144
3.3 Synthèse pour la spécification de l'interaction multiutilisateur ... 147
3.4 Synthèse pour la spécification de l'interaction multimodale ... 148

3.5 Conclusion 150
Chapitre 3 : Notation COMM 153
1 Objectif de la notation 153
2 Définition de la notation COMM 155
2.1 Spécification de l'interaction avec un arbre de tâches 155
2.2 Spécification de l'interaction multiutilisateur 158
2.2.1 Limites des notations existantes 159
2.2.2 Notre solution 162
2.2.2.1 Tâche multiutilisateur 162
2.2.2.2 Rôle métier et nombre d'utilisateurs 164
2.2.2.3 Rôle interactif 166
2.2.3 Mise en cohérence de la notation 168
2.2.3.1 Types de tâches 169
2.2.3.1.1 Première dimension : niveau d'abstraction 169
2.2.3.1.2 Deuxième dimension : auteur(s) de la tâche 170
2.2.3.1.3 Troisième dimension : interactivité 170
2.2.3.1.4 Nouveaux types et propriétés de tâches 171
2.2.3.2 Opérateurs unaires d'itération 174
2.2.4 Synthèse 176
2.3 Spécification de l'interaction multimodale 177
2.3.1 Limite des notations existantes 177
2.3.2 Notre solution 179
2.3.2.1 Tâches modales 179
2.3.2.2 Opérateur de Allen 183
2.3.3 Mise en cohérence de la notation 188
2.3.4 Synthèse 190
2.4 Au delà de l'arbre de tâches : autres éléments de spécification 190
2.4.1 Définition et utilisation de *Template* 190
2.4.2 Spécification de contexte d'utilisation 193
2.4.3 Spécification de pré-conditions et d'effets 194
3 Synthèse de la notation COMM 195
Chapitre 4 : editeur e-COMM 199
1 Objectif et démarche 199
1.1 Objectif 199

1.2 Démarche 200
2 Editeurs d'arbres de tâches 200
2.1 CTTE 200
2.2 K-MADe 201
2.3 TOUCHE Case 203
2.4 EUTERPE 203
2.5 Dynamo-Aid Tool 204
2.6 CTML 205
2.7 Synthèse 206
3 L'éditeur e-COMM 207
3.1 Architecture de l'éditeur e-COMM 207
3.2 Technologies utilisées 209
3.2.1 Partie serveur 209
3.2.2 Partie RIA 210
3.2.3 Synthèse des technologies 210
3.3 IHM du portail e-COMM 211
3.3.1 Barre d'accès rapide 212
3.3.2 Menus et pages du portail 212
3.4 IHM du RIA e-COMM 216
3.4.1 Description des rôles métiers 217
3.4.2 Description des contextes 217
3.4.3 Description des arbres de tâches 218
3.4.3.1 Ajouter une tâche 218
3.4.3.2 Spécifier un opérateur CTT entre deux tâches 219
3.4.3.3 Spécifier un opérateur de Allen entre deux tâches 220
3.4.3.4 Sélectionner une tâche 221
3.4.3.5 Sélectionner plusieurs tâches 221
3.4.3.6 Déplacer des tâches 222
3.4.3.7 Manipuler la vue (facteur de zoom et position) 224
3.4.3.8 Modifier les propriétés d'une tâche 225
3.4.3.9 Associer un contexte à une tâche 226
3.4.4 Système de sauvegarde automatique 227
4 Conclusion 228
Chapitre 5 : Evaluation de COMM et e-COMM 229

1 Difficultés d'évaluation et démarche adoptée 229
2 Evaluation conceptuelle de COMM 230
3 Evaluation expérimentale : spécification de WCCM 232
3.1 Interaction individuelle et multimodale 233
3.2 Interaction coopérative et multimodale 234
3.3 Interaction collaborative et multimodale 235
3.4 Conclusion 236
4 Evaluation expérimentale par des concepteurs 236
4.1 Objectifs et protocole expérimental 236
4.2 Résultats 238
4.2.1 Qualité des spécifications 238
4.2.2 Questionnaires 239
4.2.3 Score d'utilisabilité SUS 240
4.3 Conclusion 240
5 Evaluation expérimentale dans le cadre du projet PEA FH/PA 241
5.1 Projet PEA FH/PA 241
5.1.1 Intervenants du projet 241
5.1.2 Objectif du projet 242
5.1.3 Déroulement du projet 242
5.1.4 Poste de commande de drones : terminologie 243
5.2 Spécification du second prototype 243
5.2.1 Prototype : objectif global, scénario et rôles 244
5.2.2 Prototype : partage d'autorité et définition de contextes 245
5.2.3 Prototype : maquettes graphiques 246
5.2.4 Spécification COMM de l'interaction de l'opérateur Vecteur 249
5.2.4.1 Indiquer l'état du drone tactique 250
5.2.4.2 Modifier les paramètres de vol 252
5.2.4.3 Contrôler la trajectoire du drone tactique 255
5.2.4.4 Manipuler la cartographie 257
5.2.4.5 Gérer un plan de vol 260
5.2.4.6 Gérer les zones 262
5.2.5 Spécification COMM de l'interaction pour l'opérateur Charge Utile 264
5.2.6 Spécification COMM de l'interaction collaborative de partage de la caméra 265
5.3 Conclusion 266

6 Conclusion : évaluation de COMM et e-COMM 267
Chapitre 6 : Conclusion et perspectives 269
1 Résumé des contributions 269
1.1 Analyse détaillée des notations existantes 269
1.2 Notation COMM 270
1.3 Editeur e-COMM 270
1.4 Evaluation de la notation COMM et de son éditeur 271
2 Limitations et extensions 272
2.1 Enrichir la notation 272
2.2 Améliorer l'éditeur 273
2.3 Poursuivre l'évaluation 273
3 Perspectives à long terme : ouverture 274
Table des figures 275
Publications 281
Références 283
Annexes 295
1 Annexe 1 : Couplage de l'interaction : Expérimentation 1 – camion de pompier 295
2 Annexe 2 : Couplage de l'interaction : Expérimentation 2 – les 4 magiciens 296
3 Annexe 3 : Notation COMM et éditeur e-COMM : Expérimentation 298
3.1 Questionnaire fourni aux étudiants 298
3.2 Exemples de spécifications COMM réalisées par les étudiants 299

Introduction

1 Domaine

Nos travaux de recherche s'inscrivent dans le domaine de l'Interaction Homme-Machine (IHM) que nous abordons selon deux axes complémentaires que sont le Travail Coopératif Assisté par Ordinateur (TCAO) et la MultiModalité (MM). Ainsi, nos travaux concernent la conception de systèmes interactifs multiutilisateurs et multimodaux, c'est-à-dire des systèmes permettant à plusieurs utilisateurs d'interagir selon des techniques d'interaction ou modalités d'interaction variées.

Le Travail Coopératif Assisté par Ordinateur (TCAO) a émergé en 1984 au cours d'un groupe de travail organisé par Irène Greif et Paul Cashman, auteurs notamment du terme *CSCW* (*Computer Supported Cooperative Work*), et dont la première conférence internationale a eu lieu à Austin au Texas en 1986. Les collecticiels, c'est-à-dire les systèmes interactifs multiutilisateurs, définissent de nouveaux défis par rapport aux systèmes mono-utilisateur. D'une part, la répartition de l'application sur plusieurs machines, potentiellement hétérogènes, augmente la complexité de conception logicielle : par exemple les questions de cohérence des données, de latence réseau, ou de répartition des calculs sont des éléments complexes à traiter lors de la réalisation de collecticiels. D'autre part, l'utilisation d'un système par plusieurs utilisateurs implique de concevoir des interfaces homme-machine spécifiquement dédiées aux activités de groupe. Ces interfaces doivent par exemple aider les utilisateurs à prendre conscience de la présence des autres utilisateurs et de leurs actions, ou fournir aux utilisateurs des zones privées inaccessibles ou invisibles aux autres utilisateurs. Les collecticiels se sont développés de manière importante au-delà du monde académique. Ils ont colonisé progressivement le monde industriel sous la forme de progiciels de gestion intégrée (PGI) (en anglais Enterprise Resource Planning ou ERP) tels que SAP (*Systems, Applications, and Products for data processing*). Auprès du grand public, les collecticiels se sont popularisés au travers de deux types principaux d'applications : d'une part, les jeux vidéo, avec par exemple, les jeux de rôles massivement multi-joueurs (MMORPG) tels que World of Warcraft; et d'autre part, les systèmes de mise en relation et de communication tels que les chats (MSN), la téléphonie IP (Skype) et les réseaux sociaux (FaceBook).

La MultiModalité (MM) vise à offrir plusieurs façons d'interagir avec un ordinateur. Son objectif peut être la flexibilité, l'efficacité, le caractère naturel ou encore la robustesse de l'interaction. Les travaux séminaux du domaine par Bolt [Bolt 1980] ont porté sur la combinaison de la parole avec des gestes de désignation qui ont donné lieu au paradigme du « met ça là ». La première conférence dédiée aux interfaces multimodales (ACM-ICMI : International Conference on Multimodal Interfaces) a eu lieu à Pékin en 1996, et cette ville a accueilli récemment sa douzième édition en novembre 2010. L'ordinateur de bureau s'est enrichi de nouvelles modalités d'interaction, de la manipulation directe avec une souris à la reconnaissance de la parole ou de gestes. De plus, le paradigme de l'informatique ubiquitaire a ouvert de nombreuses perspectives pour la multimodalité. Nous constatons aujourd'hui une très grande variété de modalités d'interaction disponibles, avec par exemple l'interaction gestuelle voire bi-manuelle sur une table tactile [Koiker 2009] [Fekete 2009].

De plus, le grand public à d'ores et déjà adopté plusieurs modalités innovantes, que ce soit sur téléphones mobiles (interaction en inclinant le téléphone, interaction dite incarnée "*embodied user interfaces*" [Fishkin 2000]) ou à la maison comme la console de jeu WII qui permet aux joueurs de faire des gestes 3D avec la Wiimote pour interagir. Enfin, le monde de l'entreprise intègre aussi aujourd'hui des modalités innovantes comme celles reposant sur une tablette tactile pour un vendeur.

Les axes de la TCAO et de la multimodalité sont naturellement liés. En effet, nombre de collecticiels disponibles aujourd'hui reposent sur une combinaison de modalités d'interaction, à l'instar des applications multiutilisateurs autour d'une table augmentée [Tse 2007] ou des assistants de réunion répartis sur un PDA et un écran mural [Mejia 2007]. Dans le cadre du foyer, des jeux simples et tout public ont rendu possible la collaboration à plusieurs autour des consoles de jeux, en s'appuyant sur des modalités d'interaction variées exploitant des dispositifs comme un volant ou une planche.

2 Constat : réalisation de collecticiels multimodaux

L'intégration des aspects multiutilisateur et de la multimodalité accroît la complexité de réalisation des applications interactives à la fois en termes de conception et de développement.

- En effet, lors de la conception, il faut prendre en compte les aspects liés à la dimension multiutilisateur, tels que la conscience des autres ou la conception de zones privées, ainsi que les aspects liés à la dimension multimodale tels que le choix des modalités d'interaction ou la combinaison de modalités d'interaction. L'association des dimensions multiutilisateur et multimodale conduit à de nouvelles difficultés de conception telles que le partage ou la répartition des modalités d'interaction entre les utilisateurs.
- Du point de vue du développement, les aspects multiutilisateurs rendent plus complexes le développement, par exemple la gestion de la multiplicité ou de l'hétérogénéité des plateformes, tandis que la multimodalité implique d'explorer et de développer des interactions nouvelles. Cette difficulté de développement doit être anticipée dès la conception.

Face à cette complexité accrue, la conception et le développement de collecticiels multimodaux nécessitent des outils dédiés pour la conception et le développement comme des notations de conception et des environnements de développement. Aujourd'hui peu d'outils sont disponibles et le développement de tels systèmes est réalisé de manière ad hoc, augmentant les coûts et minimisant la possibilité de capitalisation des résultats.

- Pour la dimension multiutilisateur, il existe des outils pour la conception, comme des méthodes de conception [Molina 2006] et des notations de spécification [Mori 2002], mais aussi quelques outils dédiés au développement comme la boîte à outils [Roseman 1992] pour les collecticiels. Malgré ces outils, la conception et le développement de collecticiels restent complexes.
- Pour la multimodalité, nous trouvons un ensemble de cadres conceptuels pour la conception (CARE [Coutaz 1995], TYCOON [Martin 1999]) ainsi que des outils pour le développement,

comme des environnements de réalisation logicielle (ICARE [Bouchet 2006], OpenInterface [OpenInterface 2009], Squidy [König 2009]) et des moteurs de fusion réutilisables.

La conjonction des deux aspects place les concepteurs et les développeurs face à une difficulté supplémentaire : il s'agit d'intégrer les connaissances et les outils de TCAO et de multimodalité. En effet, la conception et le développement de collecticiels multimodaux n'ont pas été étudiés conjointement. Les études existantes adoptent soit le point de vue de l'apport de l'aspect multimodal au système interactif monoutilisateur, soit celui de l'apport des aspects multiutilisateur. Dans nos travaux nous visons l'étude conjointe des deux facettes, collecticiel et multimodalité, d'un système interactif afin de répondre aux problèmes de conception spécifiques que soulève ce type de systèmes interactifs.

3 Objectifs

Dans ce contexte, et face à l'absence d'outils conceptuels et logiciels pour la conception de systèmes interactifs multiutilisateurs et multimodaux, notre objectif est de définir une notation de spécification de collecticiels multimodaux. Comme première étape de travail à l'intersection des deux domaines de la TCAO et de la Multimodalité, nous considérons donc les phases amont de la réalisation d'un collecticiel multimodal.

Pour bien cerner notre sujet et la portée de nos contributions, nous nous référons aux étapes classiques d'un projet de développement logiciel. Le cycle de vie en V de la Figure 1 fixe les grandes étapes d'un projet : analyse des besoins, spécification, conception architecturale, conception détaillée, développement, test unitaire, test intégration, test de validation et recette. Dans ce contexte, nous nous focalisons sur la phase de spécification. La notation visée doit permettre de spécifier des solutions d'interaction multiutilisateur, qu'elle soit abstraite ou concrète - dépendante des modalités d'interaction choisies.

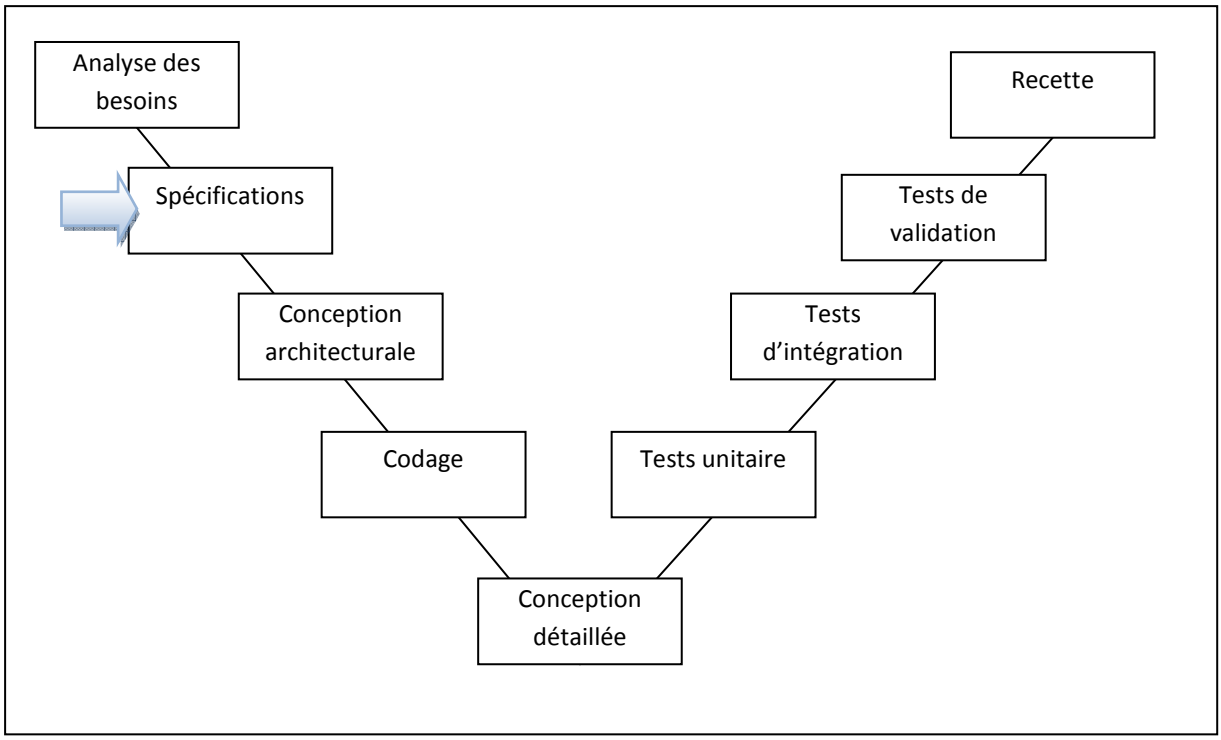


Figure 1 : Objectif de nos travaux au sein d'un cycle de vie en V : phase de spécification.

Nous visons l'aide à la conception en fournissant aux concepteurs un modèle pour l'interaction abstraite et concrète sous la forme d'une notation de spécification dédiée aux collecticiels multimodaux et d'un outil d'édition prenant en charge la spécification de solutions d'interaction selon le modèle défini par la notation :

- Le modèle défini par la notation fixe un cadre à l'expression des idées du concepteur, structurant ainsi son activité mentale. La matérialisation des idées sous la forme de schéma fournit un support pour l'analyse de l'espace des possibilités et un vecteur pour sa communication. Il s'agit d'un outil tendant à améliorer et enrichir les solutions de conception par le biais de son expression et de son raffinement.
- Le support d'un outil logiciel pour une notation rend efficace l'utilisation d'une notation à une large échelle, fédérant l'usage des utilisateurs autour des concepts de la notation, et évitant ainsi une dérive vers le savoir faire individuel de chaque concepteur. Avec un tel outil, il est possible de créer, modifier, et partager des spécifications à moindre coût. Ces spécifications ainsi saisies peuvent servir de support à la communication interne et externe au projet, ainsi qu'à des analyses automatiques.

L'étape suivante à ce travail serait alors de s'intéresser aux phases de réalisation logicielle (les étapes de conception architecturale et détaillée ainsi que de codage de la Figure 1) en définissant des outils logiciels de développement (boîte à outils, UIMS, UIDE [Myers 1995] [Myers 2000]), qu'il conviendrait de lier à notre notation de spécification afin de garantir un passage fluide entre les étapes de spécification et de développement (Figure 1).

4 Démarche adoptée

Notre démarche de travail est schématisée par le diagramme de la Figure 2. Dans un premier temps, nous avons identifié les méthodes et les notations de conception existantes, aussi bien pour les collecticiels que pour les systèmes multimodaux. Nous avons mené une étude comparative de ces notations afin d'identifier leurs concepts saillants, leurs complémentarités, mais également leurs limites et manques. Sur la base de cette étude des notations existantes, nous avons identifié deux requis que notre notation doit satisfaire pour répondre à notre objectif : (1) distinguer les interactions collaboratives et coopératives et (2) exprimer le lien entre l'interaction abstraite et l'interaction multimodale concrète.

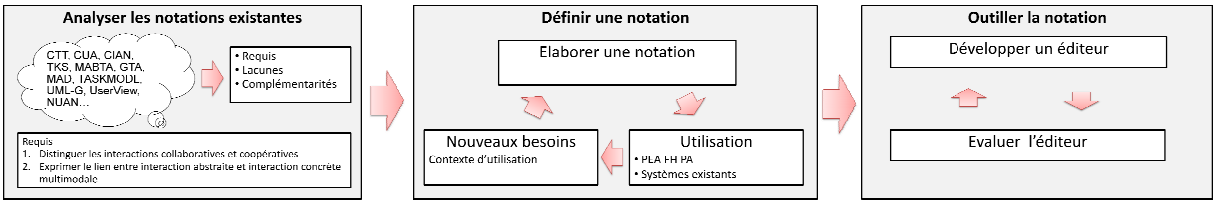


Figure 2 : Démarche de travail : trois grandes étapes.

A l'issue de cette analyse, nous avons élaboré notre notation comme un ensemble cohérent de concepts pour décrire des collecticiels multimodaux, en s'appuyant sur les concepts des notations existantes et en les étendant, mais également en ajoutant de nouveaux concepts. La notation produite a ensuite été raffinée par itération successive grâce à son utilisation dans le cadre du Projet d'Etude Amont (PEA) Facteurs Humains et Partage d'autorité (FH/PA) et à son application à la spécification de l'interaction de nombreux systèmes existants.

Enfin, nous avons conçu et développé un outil logiciel permettant d'éditer des spécifications selon notre notation. L'éditeur et la notation sur laquelle il repose ont ensuite été évalués par son utilisation et sa diffusion.

5 Contribution conceptuelle

Afin de contribuer à l'espace de conception des systèmes interactifs multiutilisateurs et multimodaux, nous proposons un modèle d'interaction intégrant les dimensions collaboratives et multimodales de l'interaction. Ce modèle permet au concepteur de spécifier des solutions d'interaction, ainsi que de comparer plusieurs solutions décrites. Son originalité provient de la prise en compte conjointe de la collaboration entre utilisateurs et de la multimodalité. Ce modèle prend la forme d'une notation semi-formelle d'arbre de tâches, appelée COMM (COllaborative et MultiModale). L'utilisation de la notation COMM nécessite donc la compréhension des concepts liés à l'analyse de la tâche et des modèles de tâches existants.

La notation COMM est dédiée à la spécification de solutions de conception, et n'est pas prévue pour aider à l'analyse des besoins (première phase à la Figure 1). Ce positionnement implique de ne pas considérer plusieurs concepts existant dans d'autres notations et strictement relatifs à l'analyse des besoins tels que la fréquence de réalisation d'une tâche K-MAD [Lucquiaud 2005, Baron 2006]. La

notation COMM inclut des concepts originaux et spécifiques aux systèmes collaboratifs et multimodaux comme ceux de rôle interactif et de tâche modale. De plus reprenant plusieurs concepts de la notation d'arbres de tâche CTT [Paterno 1997], la notation COMM peut être considérée comme une extension de CTT pour les systèmes collaboratifs et multimodaux.

6 Contribution logicielle

Comme cela a été souligné par [Mori 2002] et [Beaudoin 2004], un modèle pour l'interaction nécessite des outils dédiés, pour permettre son utilisation la plus large possible. Dans le but de rendre utilisable notre modèle à une large échelle et de favoriser sa diffusion, nous avons étudié les outils existants et proposé un outil appelé e-COMM qui prend en charge l'édition de diagrammes selon la notation COMM. Avec e-COMM, les concepteurs peuvent produire des spécifications détaillées pour les systèmes interactifs multiutilisateurs et multimodaux. Ces spécifications de l'interaction servent de référence pour les phases suivantes de développement (Figure 1).

7 Structure du mémoire

La structure de ce mémoire, qui présente nos contributions tant conceptuelles que pratiques, reflète la démarche de travail adoptée en trois grandes étapes.

Le Chapitre 1 situe et motive nos travaux sur les notations de spécification au sein des domaines du Travail Coopératif Assisté par Ordinateur et de la Multimodalité.

Nous consacrons le Chapitre 2 à l'étude de l'état de l'art des notations pour la spécification de l'interaction de systèmes multiutilisateurs et multimodaux. Cette analyse fine des notations existantes repose sur une grille d'analyse des caractéristiques des notations, ainsi que sur leur utilisation sur un cas d'étude unique intitulé WCCM (Warcraft Collaboratif Coopératif et Multimodal). En vue de la définition d'une nouvelle notation, nous concluons ce chapitre par les concepts nécessaires à intégrer et à étendre au sein d'une notation pour la spécification de systèmes multiutilisateurs et multimodaux.

Nous présentons au Chapitre 3 la notation COMM dédiée à la spécification de systèmes multiutilisateurs et multimodaux. La notation COMM est une extension de notations existantes présentées au Chapitre 2. COMM introduit de nouveaux concepts tels que les rôles interactifs et les tâches modales. Nous justifions chacune de nos propositions de manière détaillée, en nous appuyant sur le cas d'étude WCCM et l'état de l'art établi au Chapitre 3.

Correspondant à la troisième étape de notre démarche de travail, le Chapitre 4 est dédié à la présentation de l'outil e-COMM, un éditeur de spécifications COMM que nous avons conçu et développé pour permettre l'utilisation et la diffusion de la notation COMM.

Enfin, le Chapitre 5 est consacré aux différentes formes d'évaluation de nos contributions : la notation COMM et l'éditeur e-COMM. Il présente trois formes d'évaluation. La première consiste à illustrer le pouvoir d'expression de la notation en considérant le même cas d'étude WCCM utilisé

pour comparer les notations existantes. La deuxième forme d'évaluation est expérimentale : elle consiste à proposer à des utilisateurs/concepteurs extérieurs à l'équipe IIHM du LIG d'utiliser la notation COMM pour concevoir et spécifier un système. Enfin la troisième forme de validation a consisté à étudier le passage à l'échelle dans l'usage de la notation et a montré qu'au delà des cas d'école et prototypes de recherche considérés, la notation COMM peut effectivement être utilisée dans un projet de taille conséquente : pour cela nous avons appliqué la notation COMM dans le cadre du projet PEA FH/PA de spécification de l'interaction du poste de commande de drones.

Nous concluons ce mémoire par une synthèse de nos contributions, conceptuelles et techniques. Nous élargissons ensuite le cadre d'étude en définissant des perspectives conceptuelles et techniques par une mise en perspective de nos contributions dans le cadre du cycle de vie du logiciel.

Trois annexes sont incluses à la fin du mémoire et fournissent des compléments d'information sur les protocoles et résultats des trois études expérimentales que nous avons menées.

Chapitre 1 : Systèmes multiutilisateurs et multimodaux

Nos travaux de recherche traitent de la conception de l'interaction de systèmes multiutilisateurs et multimodaux, c'est-à-dire des systèmes permettant à plusieurs utilisateurs d'interagir selon des techniques d'interaction variées. Ces travaux s'inscrivent donc dans le domaine de l'Interaction Homme-Machine (IHM) et y abordent deux axes de recherche complémentaires que sont le Travail Coopératif Assisté par Ordinateur (TCAO) et la MultiModalité (MM). Dans ce cadre de recherche, notre objectif est de définir une notation de spécification de systèmes interactifs multiutilisateurs et multimodaux.

Pour cerner le cadre de notre étude et fixer la terminologie adoptée, ce chapitre présente les deux domaines de nos travaux que sont la Travail Coopératif Assisté par Ordinateur et la Multimodalité. En particulier nous décrivons les enjeux de conception qui sont spécifiques aux systèmes multiutilisateurs et multimodaux. Le chapitre est structuré en trois parties principales : nous considérons d'abord les systèmes multiutilisateurs (section 1) en mettant l'accent sur le couplage de l'interaction par le biais d'une étude expérimentale que nous avons menée. Nous présentons ensuite le domaine de la multimodalité (section 2). Les deux domaines définis, nous décrivons à la section 3 plusieurs systèmes multiutilisateurs et multimodaux et en particulier le système Warcraft Collaboratif, Coopératif et Multimodal que nous utilisons tout au long de ce manuscrit comme étude de cas. Nous concluons enfin ce chapitre en motivant le sujet de nos travaux de recherche.

1 Systèmes multiutilisateurs

Un système multiutilisateur ou collecticiel constitue l'objet d'étude du domaine de Travail Coopératif Assisté par Ordinateur (TCAO). La définition la plus couramment adoptée pour caractériser un collecticiel précise qu'il s'agit d'un système informatique assistant un groupe de personnes engagées dans une tâche commune (ou un but commun) et fournissant une interface à un environnement partagé[1] [Ellis 1991].

S'inscrivant dans le domaine de l'IHM, le domaine de la TCAO consiste à adapter et à étendre les méthodes et outils du domaine de l'IHM pour la réalisation de systèmes mono-utilisateur à la réalisation de systèmes multiutilisateur. Le domaine de la TCAO s'intéresse en particulier à étudier les enjeux supplémentaires que pose la réalisation d'un système multiutilisateur par rapport à un système mono-utilisateur, et ce lors de toutes les étapes du cycle de vie logiciel. Nous focalisons nos travaux sur les étapes de conception (de l'interaction) et plus précisément sur la spécification de solutions d'interaction conçues.

La réalisation de systèmes multiutilisateurs définit de nouveaux défis par rapport aux systèmes mono-utilisateur qui doivent être relevés en amont de leur développement, et notamment lors de la

[1] Definition originale : *"Computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment."* [Ellis 1991]

conception de ces systèmes. Nous illustrons ces défis en soulignant la diversité des collecticiels au travers de la présentation d'une part de la classification espace temps [Ellis 1991, Grudin 1994] et du modèle clover [Salber 1995]. Nous présentons ensuite un ensemble d'éléments à prendre en compte lors de la conception de collecticiels. Enfin nous présentons une étude expérimentale que nous avons menée pour mettre en relief les formes de couplage de l'interaction multiutilisateur.

1.1 Classification espace-temps

La classification Espace - Temps introduite dans [Ellis 1991] et étendue dans [Grudin 1994] repose sur deux axes (espace et temps) pour caractériser un système multiutilisateurs. Elle est illustrée au sein du Tableau 1.

		Temps		
		Même moment	**Moments différents (prévisibles)**	**Moments différents (imprévisibles)**
Espace	**Même lieu**	(1) assistant de réunion	(2) enchaînements	(3) salle de travail en équipe
	Lieux différents (prévisibles)	(4) vidéoconférence	(5) *workflow*	(6) intranet
	Lieux différents (imprévisibles)	(7) messagerie instantanée	(8) édition partagée	(9) mail, forum, réseaux sociaux

Tableau 1 : Classification espace-temps

Le premier axe de cette classification est le temps. Il s'agit de caractériser la relation temporelle existant entre deux utilisateurs d'un système. Ainsi, deux utilisateurs peuvent utiliser le système au même moment, ce que l'on qualifie également d'usage synchrone, ou à des moments différents, ce que l'on qualifie d'usage asynchrone.

Le deuxième axe de cette classification est l'espace. Cet axe vise à décrire la relation spatiale existant entre deux utilisateurs d'un système. Les utilisateurs peuvent ainsi utiliser le système en étant co-localisés au sein d'un même lieu ou l'utiliser à distance l'un de l'autre depuis des lieux différents.

Dans [Grudin 1994], l'auteur étend cette classification avec une distinction sur l'imprévisibilité de l'usage synchrone et de la localisation des utilisateurs d'un système. Ainsi, chacun des axes est défini selon trois cas, deux cas prévisibles : même lieu et lieu différent pour l'axe Espace, et un cas imprévisible. Nous illustrons les neufs types de collecticiels ainsi définis dans la classification Espace – Temps étendue.

1. Un assistant de réunions tel que décrit dans [Meja 2007] permet aux utilisateurs de partager des documents médicaux informatisés (par exemple une radio) qui servent de support à la discussion. Dans cette exemple, les utilisateurs sont des médecins dans un même lieu (co-localisation) et interagissent pendant une même période de temps : la durée de la réunion, ce qui constitue un usage synchrone.
2. Un poste de travail informatisé pour le travail en équipe par exemple pour une chaîne de production implique une localisation unique des utilisateurs (le poste de travail) et une répartition du travail prévu dans le temps (travail en équipe en trois-huit).

3. Une salle de travail en équipe dédiée par exemple à l'*extreme programming* peut être informatisée pour permettre le travail de l'équipe. Les membres de l'équipe travaillent dans un lieu unique (la salle), mais leur présence dans la salle n'est pas prévisible.
4. Les systèmes de vidéoconférence sont utilisés pendant une même période de temps donnée par plusieurs utilisateurs localisés (éventuellement par groupe) dans des lieux différents.
5. Un système de *workflow* vise à coordonner les actions des utilisateurs de manière prévisible et peut être localisé dans un ensemble de lieux prévisibles. C'est le cas par exemple pour un système de gestion d'une chaîne de production ou d'assemblage. Le travail est réparti entre les opérateurs sur la base de postes de travail.
6. Les systèmes d'intranet visent à offrir à un ensemble d'utilisateurs, généralement les employés d'une société, un ensemble de services, tel qu'un outil pour faire une demande de congés, que les employés peuvent utiliser quand ils le souhaitent (moments différents et imprévisibles), mais accessible uniquement depuis un de leurs postes de travail (lieux différents mais prévisibles).
7. Les systèmes de messagerie instantanée constituent un exemple d'application requérant la présence simultanée des utilisateurs pouvant être répartis dans des lieux parfaitement imprévisibles.
8. Les outils d'édition partagée sont des systèmes où le lieu d'édition n'a pas d'importance, mais où les utilisateurs doivent se coordonner dans le temps pour planifier leurs échanges et leurs travaux sur un document.
9. Enfin les mails, les forums, ou les réseaux sociaux sont des exemples de systèmes où la localisation des utilisateurs est imprévisible et dont l'utilisation peut se faire de manière asynchrone et imprévisible.

La classification Espace-Temps ne vise pas à classer les systèmes de manière absolue, mais plutôt à caractériser leur usage. En effet, comme le souligne [Grudin 1994], de nombreux collecticiels sont utilisés dans des situations (espace + temps) variées. Ainsi par exemple un jeu multi-joueurs sur console de salon que l'on peut classer a priori dans la catégorie 1, c'est-à-dire regroupant plusieurs utilisateurs dans un même lieu et pendant un même moment, peut également comporter un mode permettant de jouer avec d'autres joueurs sur internet. Le jeu peut alors être utilisé entre plusieurs utilisateurs distants ou entre plusieurs groupes distants.

Par ailleurs, cette classification ne prend pas en considération la mobilité des utilisateurs lors de l'usage d'un collecticiel, ce qui peut par exemple entraîner le passage d'une situation d'interaction en des lieux différents, à un lieu unique : ainsi, deux collégiens, communiquant par messagerie instantanée avec leur téléphone, situés respectivement dans les bus qui les conduisent à leurs collèges, peuvent poursuivre leur conversation une fois arrivés dans la salle de classe (même lieu). L'étude de la mobilité pour les collecticiels fait l'objet du chapitre 2 de la thèse de P. Renevier [Renevier 2004] : il introduit une nouvelle valeur sur l'axe Temps notée « Temps d'Utilisation Composés » et considère sur l'axe Espace la notion de lieu confiné versus lieu vagabond pour obtenir quinze classes de systèmes multiutilisateurs mobiles.

Enfin, les systèmes de messageries instantanées ou de réseaux sociaux peuvent être utilisés de manière presque continue puisqu'ils sont accessibles depuis de nombreux dispositifs (Smartphone, tablette et ordinateur fixe et portable). Il devient alors plus difficile de qualifier l'usage synchrone ou asynchrone puisque la déclaration d'état de disponibilité et l'envoi de messages s'effectuent selon

des modes aussi bien compatibles avec un usage synchrone (conversation sous forme d'échanges rapides de messages) et asynchrone (envoi de messages et réponses différées).

1.2 Classification fonctionnelle : Modèle du trèfle

Une forme complémentaire de classification des collecticiels consiste à identifier leurs fonctions. Dans [Salber 1995], l'auteur présente le modèle du trèfle des collecticiels illustré à la Figure 3. Celui repose sur l'identification de trois classes de fonctionnalités que remplit un collecticiel :

- **Production** : un collecticiel peut viser à produire des œuvres partagées en permettant des actions communes ou individuelles.
- **Coordination** : un collecticiel permet d'organiser les tâches des utilisateurs sur la base des rôles qu'ils jouent.
- **Communication** : un collecticiel permet ou améliore la communication entre les utilisateurs (communication homme-homme médiatisée).

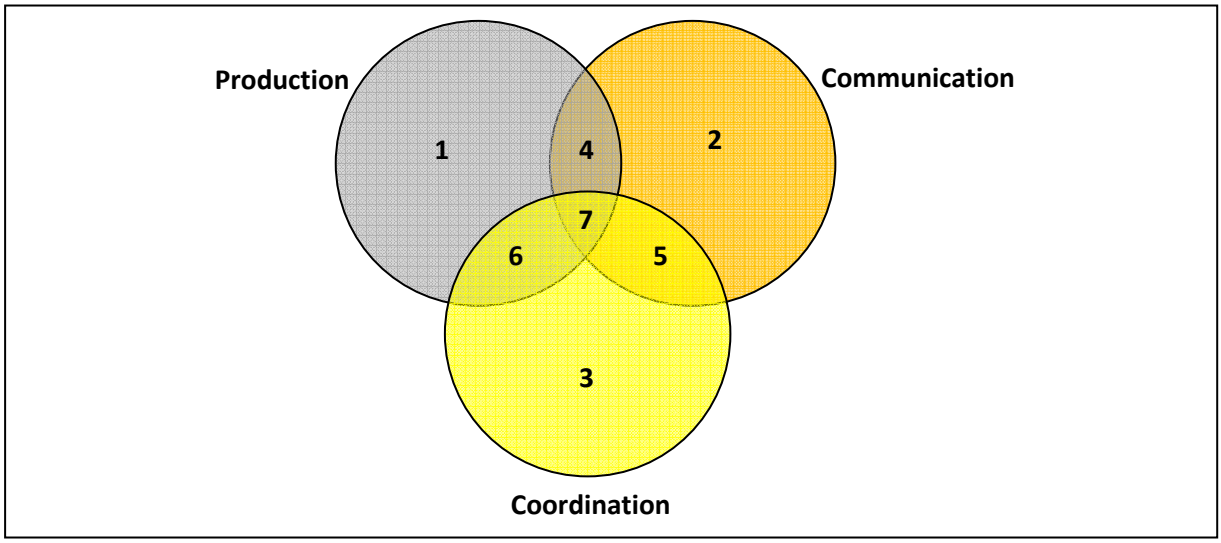


Figure 3 : Trois fonctions d'un collecticiel.

Un collecticiel peut couvrir une ou plusieurs de ces classes de fonctionnalités, c'est pourquoi les cercles représentant ces fonctionnalités au sein de la Figure 3 se recouvrent. Ceci permet d'identifier 7 classes de collecticiels couvrant plus ou moins les trois classes de fonctionnalités.

Certains collecticiels couvrent une unique classe de fonctionnalités, comme dans les cas 1, 2, et 3 de la Figure 3 que nous illustrons :

1. Un jeu vidéo sur une console de salon constitue une activité de production. Les fonctions de communication et de coordination sont généralement assurées par les joueurs eux-mêmes par une communication directe.
2. Un système de téléphonie mobile constitue un exemple d'outil de communication.
3. Le système doodle[2] par exemple est un outil de coordination qui permet à plusieurs utilisateurs d'identifier des dates de disponibilités communes pour fixer un rendez-vous.

[2] http://www.doodle.com/

Les trois fonctions (production, coordination, et communication) peuvent également être conjuguées au sein d'un même système multiutilisateur, ce qui correspond aux cas 4, 5, 6 et 7 de la Figure 3 que nous illustrons :

4. Dans le contexte de la conduite automobile, il existe des systèmes de détection de radar de vitesse [3] qui reposent sur une communauté d'utilisateurs. Ceux-ci produisent dynamiquement une cartographie des contrôles radars mobiles (Production) qui est diffusée aux autres utilisateurs du système (Communication).
5. Les systèmes de messagerie instantanée[4] constituent un exemple d'outils de communication entre les personnes. De plus, ils assurent la coordination des utilisateurs pour la tenue d'échanges synchrones (conversation), ou d'échanges asynchrones (messages, puis réponses différées) notamment grâce à l'affichage permanent des utilisateurs amis connectés, et de leur état affiché tel qu'en ligne, occupée ou absent.
6. Les outils d'édition partagée permettent d'une part la production de contenu commun, mais peuvent également assurer la coordination de l'édition entre les différents utilisateurs. Ainsi, le système de wiki comme Wikipedia.org permet à des milliers d'utilisateurs de réaliser une encyclopédie commune en assumant chacun l'écriture, la lecture et la correction des contributions.
7. Un système de forum coordonne la production (et la vérification) de connaissances partagées par les utilisateurs, et constitue un moyen de communication de ces connaissances.

L'identification des classes de fonctionnalités (production, coordination, et communication) que l'on souhaite offrir aux utilisateurs avec un collecticiel constitue une première étape de l'analyse des besoins. Pour chacune des classes de fonctionnalités, il s'agit ensuite d'identifier les fonctionnalités réelles dont les utilisateurs ont besoin. Cette classification permet ainsi de diviser les préoccupations lors de l'analyse des besoins. Les réponses à ces préoccupations pourront être intégrées de manière cohérente par le concepteur lors de la conception et la spécification du collecticiel.

1.3 Activité multiutilisateur : coopération vs collaboration

Dans la littérature, les termes de coopération et de collaboration sont souvent utilisés indistinctement pour décrire des activités multiutilisateurs ou de groupe. Ces deux termes recouvrent pourtant des réalités différentes qui apparaissent clairement lorsque l'on s'attache à décrire finement les interactions multiutilisateurs.

Dans [Dillenbourg 2006], l'auteur propose de dissiper la confusion entre ces termes en considérant la manière de diviser le travail entre les participants :

"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."

[3] Par exemple, le système coyote : http://www.moncoyote.com/
[4] Par exemple, le logiciel Skype : *www.skype.com*

Ainsi, la coopération et la collaboration couvrent deux aspects des activités de groupes. La coopération implique une répartition du travail à priori entre les participants, tandis que la collaboration met en jeu une répartition dynamique du travail (non décidée à priori). Nous adoptons la définition de Dillenbourg pour délimiter la frontière entre coopération et collaboration.

1.4 Groupes, rôles, agents, acteurs et utilisateurs

Les notions de groupes, rôles, d'agents, d'acteurs et d'utilisateurs sont largement utilisées dans la littérature du domaine de la TCAO.

Ainsi l'ontologie de [Veer 2000] illustrée à la Figure 4, sur laquelle repose la notation GTA (*Groupware Task Analysis*) définit les concepts de rôle et d'agent comme les concepts de base pour la modélisation de collecticiels. Comme l'illustre la Figure 4, un rôle est défini par la responsabilité de la réalisation d'un ensemble de tâches.

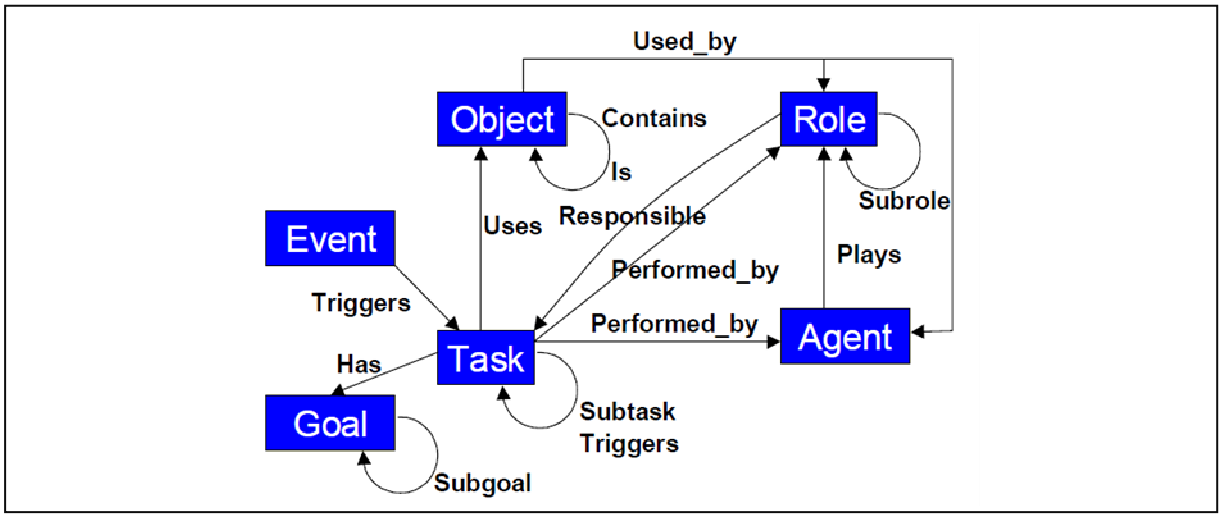


Figure 4 : Ontologie pour la description des tâches (illustration issu de [Veer 2000]).

Pour certains, comme dans l'ontologie de [Veer 2000], un rôle peut être endossé par un agent [Veer 2000], ce dernier pouvant être un utilisateur (acteur) ou le système (agent logiciel). Pour d'autres, les rôles ne peuvent être endossés que par des utilisateurs [Penichet 2006].

Dans de nombreux travaux, la notion de groupe permet d'englober un ensemble d'utilisateurs [Penichet 2006] [Molina 2006], qui peut lui-même être responsable de la réalisation d'un ensemble de tâches.

Dans la suite de ce mémoire, nous introduisons la notion de rôle « interactif » en opposition avec la notion traditionnelle de rôle « métier » utilisé dans la littérature. Dans la suite de ce mémoire, nous utiliserons le terme de « rôle » pour indiquer la notion de « rôle métier » et nous spécifierons explicitement les cas de « rôle interactif ».

Nous avons souligné dans les sections précédentes la diversité des systèmes multiutilisateurs en présentant deux classifications des collecticiels : la classification Espace-Temps et la classification fonctionnelle. De plus nous avons introduit plusieurs concepts clefs du domaine des systèmes multiutilisateurs : coopération, collaboration, rôle, agent, utilisateur et groupe. Nous abordons maintenant la conception de l'interface homme-machine des systèmes multiutilisateurs.

1.5 Espace de conception

Lors de la conception de l'interface d'un collecticiel, il convient de prendre en compte les contraintes fonctionnelles mais également veiller à son utilisabilité. Dans ce but, de nombreuses recommandations ergonomiques ou propriétés ergonomiques ont été introduites, telles les propriétés d'observabilité et d'honnêteté de [Bastien 1992]. La prise en compte de ces recommandations ou propriétés ergonomiques en amont de la réalisation d'un système interactif et leur contrôle en aval visent à augmenter son utilisabilité. Par exemple la propriété ergonomique d'observabilité de [Bastien 1992] indique que le système doit rendre perceptible à l'utilisateur son état interne pertinent pour la réalisation de la tâche : ainsi dans le cas où un système est occupé à réaliser un calcul, il convient d'indiquer à l'utilisateur que le système est occupé et le temps restant pour finir le calcul (barre de progression).

Des propriétés ergonomiques dédiées aux collecticiels ont été proposées pour prendre en compte les problèmes spécifiques que posent les systèmes multiutilisateurs. Ces propriétés sont décrites en détail dans [Salber 1995], [Laurillau 2002] et [UsabilityFirst]. Parmi ces propriétés, nous présentons la conscience de groupe, l'observabilité et la protection de la vie privée, la réciprocité, le WYSIWIS, le couplage de l'interaction, et la viscosité, afin d'illustrer les problèmes spécifiques que pose la conception de l'interface de collecticiels.

1.5.1 Conscience de groupe

Au cours d'une activité de groupe, les participants conservent une attention particulière à ce que font les autres membres du groupe afin d'anticiper, d'agir ou de réagir à leurs actions. Ainsi par exemple, un participant d'une réunion entend celui qui parle, et voient les autres participants. Cela lui permet de prendre la parole lorsqu'il anticipe un blanc dans le discours, ou de solliciter un participant quand celui-ci est dans un état d'attention correspondant à son attente, par exemple lorsqu'il a terminé de noter quelque chose.

Lorsque plusieurs utilisateurs utilisent un même système, la conscience de groupe peut être rompue. Cela peut être le cas lors de l'édition d'un document avec un système d'édition partagée où les utilisateurs sont distants. Le but d'un collecticiel est de permettre l'activité de groupe, et doit ainsi veiller à maintenir chez les utilisateurs cette conscience des autres et de leurs actions. La conscience de groupe implique de rendre observable l'état de l'interaction pour chacun des autres utilisateurs : ce qu'ils ont fait, et ce qu'ils sont en train de faire. Les moyens mis en œuvre pour rendre observable cet état à un utilisateur ne doivent pas le gêner dans ses propres tâches, mais plutôt être réalisés de manière périphérique, d'où le terme conscience des autres. Par exemple, c'est dans ce but que des widgets tels qu'une barre de défilement multiutilisateur [Hill 2004] ont été développés. Outre le fait qu'elle permet de faire défiler un document, elle renseigne sur la position de lecture ou d'édition de chacun des utilisateurs.

Cette propriété de conscience de groupe est liée à plusieurs autres propriétés ergonomiques, notamment l'observabilité et protection de la vie privée que nous abordons dans la section suivante.

1.5.2 Observabilité, protection de la vie privée

L'observabilité est une propriété ergonomique définie pour les systèmes mono-utilisateur [IFIP 1996]. Elle traduit la capacité d'un système à rendre perceptible à l'utilisateur son état interne afin

que celui-ci puisse interagir au mieux avec le système. Pour les systèmes multiutilisateurs, l'observabilité est étendue aux informations concernant les autres utilisateurs et leurs activités.

L'observabilité des actions des autres utilisateurs doit toutefois respecter la vie privée des utilisateurs. Un système de vidéoconférence tel que Skype ou Live Messenger transmet la vidéo des utilisateurs, cependant, cette vidéo peut contenir des informations sur l'environnement qui entoure les utilisateurs. Par exemple un utilisateur peut ne pas souhaiter montrer aux autres qu'il se trouve à la terrasse d'un café, plutôt que dans son bureau, ce qu'il ne pourra pas faire avec ces systèmes. Ces systèmes ne protègent donc pas particulièrement la vie privée des utilisateurs. C'est dans ce cadre qu'a été définie la propriété de protection de la vie privée qui définit la capacité d'un système à protéger la vie privée de ses utilisateurs.

Dans le but de protéger sa vie privée, un utilisateur peut vouloir décider quelles sont les informations qui seront présentées aux autres utilisateurs. Cela peut se faire par la publication d'un statut comme dans le cas de la messagerie instantanée (en ligne, occupé, absent) ou encore par des restrictions sur l'accès à des données (personnelles ou professionnelles). Dans [Laurillau 1999a], l'auteur définit trois formes d'observabilité :

- Une information peut être observable et donc perçue par les autres utilisateurs.
- Une information peut être publiable c'est-à-dire rendue observable par un choix de l'utilisateur.
- Une information peut également être filtrable et donc modifiée par l'utilisateur (forme et contenu de l'information) avant publication. Par exemple, dans un système messagerie, le fait de publier un état « occupé » indique que l'utilisateur ne souhaite pas être dérangé. Il s'agit d'un filtre sur l'activité en cours. En effet, les autres utilisateurs peuvent savoir que l'utilisateur est occupé, mais ne peuvent pas savoir à quoi l'utilisateur est occupé.

Naturellement, si un système propose un moyen de filtrer une information, il s'agit également que ce système rende observable l'information qui peut être filtrée à l'utilisateur. Par exemple dans les systèmes de vidéoconférence, la propre image filmée de l'utilisateur est visible. Ce dernier peut alors décider d'autoriser ou non l'envoi de sa vidéo aux autres utilisateurs.

Ce filtrage peut également être réalisé par le système, comme dans le système de *media space* [Coutaz 1999] où certains gestes d'un utilisateur sont éliminés de la vidéo envoyée aux autres utilisateurs, afin de respecter sa vie privée. La Figure 5 illustre ceci avec le cas d'un geste intempestif : l'utilisateur se gratte le nez (image de gauche), geste qui est éliminé de la vidéo observable par les autres utilisateurs (image de droite).

Figure 5 : Filtrage d'un geste intempestif dans le cadre du respect de la vie privée [Coutaz 1999].

1.5.3 Réciprocité

La réciprocité est également liée à l'observabilité. Elle exprime la capacité d'un système à fournir aux utilisateurs des moyens d'observation mutuelle. Une observabilité réciproque permettra à un utilisateur dont l'image filmée est diffusée à un autre de voir lui-même l'image de l'autre utilisateur. A l'inverse un système de vidéosurveillance ne respecte pas la réciprocité, puisque la personne filmée ne voit pas la personne qui la surveille.

1.5.4 WYSIWYS

La propriété de WYSIWIS (*What You See Is What I See*) [Stefik 1987] traduit le principe d'une vue identique pour plusieurs utilisateurs et où toutes modifications effectuées par un utilisateur sont répercutées aux autres utilisateurs.

Le WYSIWIS est caractérisé de strict quand les utilisateurs partagent une IHM identique : toute action d'un utilisateur sur son IHM locale produit des effets identiques sur les IHMs des autres utilisateurs. Ainsi, dans un système de cartographie par exemple, si un utilisateur déplace la vue, celle-ci est déplacée également pour les autres utilisateurs.

A l'inverse, le WYSIWIS est défini comme relâché lorsque les utilisateurs ont une vue sur des données similaires mais chaque vue n'est pas forcément identique. Dans ce cas, les actions des utilisateurs n'induisent pas systématiquement un effet sur les IHMs des autres utilisateurs. Ainsi, pour l'exemple d'un système de cartographie, une tâche articulatoire de déplacement effectuée par un utilisateur n'implique pas que les vues de la carte des autres utilisateurs soient modifiées.

1.5.5 Couplage de l'interaction

Le couplage de l'interaction [Dewan 1995] caractérise le délai plus ou moins long qui peut intervenir entre une action sur une donnée par un utilisateur et la répercussion de cette action pour les autres utilisateurs.

Le couplage peut être fort auquel cas, on considère que le délai est court entre l'action et la répercussion, ou faible et dans ce cas, le délai est plus important.

La question du couplage est fortement liée à celle du WISIWIS. En effet, le cas du WYSIWIS strict implique un couplage fort, tandis que le cas du WISIWIS relâché indique un couplage faible, dans le sens où les modifications de la vue d'un utilisateur suite à une action d'un autre utilisateur est moins prioritaire que la tâche en cours de cet utilisateur.

1.5.6 Viscosité

La viscosité définie dans [Green 1990] indique l'incidence que peut avoir une action d'un utilisateur sur les activités des autres utilisateurs. La viscosité est étroitement liée au niveau de couplage adopté pour l'interaction. Pour l'exemple d'un système de cartographie, en considérant un couplage fort dans un cas de WISIWIS strict, lorsqu'un utilisateur déplace la vue, cette dernière est modifiée pour tous les autres utilisateurs : le système peut être alors considéré de visqueux, si cette modification répercutée de la vue empêche un utilisateur de poursuivre sa tâche et implique qu'il déplace à nouveau la vue.

1.5.7 Synthèse

Nous avons présenté un ensemble de propriétés ergonomiques dédiées aux systèmes multiutilisateurs. Toutes ces propriétés mettent l'accent sur la difficulté de concevoir une interface partagée entre les utilisateurs, qui reste utilisable pour chacun, et permette d'enrichir la collaboration entre les utilisateurs. Plus précisément ces propriétés traitent :

- des relations d'observation mutuelle qu'entretiennent les utilisateurs dans le cadre de l'utilisation d'un collecticiel (conscience de groupe, observabilité, publication, filtrage, réciprocité), et,
- de l'impact sur chaque utilisateur de l'usage collectif d'un même système (WISIWIS, couplage, viscosité).

Dans le but d'illustrer la difficulté de concevoir des interfaces multiutilisateurs, et notamment le point central du couplage de l'interaction à considérer lors la conception, nous présentons une étude expérimentale que nous avons menée sur la définition de niveaux de couplage intermédiaires aux couplages forts et faibles.

1.6 Etude sur le couplage de l'interaction

Dans cette section, nous présentons une étude expérimentale que nous avons menée afin de montrer l'existence de plusieurs niveaux de couplage de l'interaction, que nous définissons selon deux dimensions : la dimension de l'activité et la dimension de l'interaction concrète. Aussi, à partir de ces deux dimensions, nous proposons une matrice pour décrire différents types et niveaux de couplage. Nous montrons dans cette étude que l'utilisateur perçoit ces différents niveaux de couplage et qu'il est nécessaire de considérer le couplage dès les étapes de conception d'un système multiutilisateur.

Dans [Jourde 2009], nous présentons une matrice pour caractériser les niveaux de couplage de l'interaction, reproduite au Tableau 2. Notre caractérisation repose sur deux dimensions. La première dimension est la dimension de l'activité où trois niveaux sont proposés : non couplé (activité individuelle), faiblement couplé (activité coopérative) et fortement couplé (activité collaborative). La deuxième dimension décrit l'interaction utilisateur concrète. Elle est définie selon plusieurs niveaux : fortement couplé, intermédiaire, faible et non couplé, dont les niveaux fort et faible correspondent respectivement au WYSIWIS strict et au WYSIWIS relâché.

Dans [Tang 2006], les auteurs identifient par l'observation six degrés de couplage, du plus fortement couplé au plus faiblement couplé, pour des activités collaboratives menées autour d'une table interactive : même problème, même zone (SPSA) ; l'un travaille, l'autre regarde tout en étant impliqué (VE) ; même problème, zones de travail différentes (SPDA) ; l'un travaille, l'autre regarde avec une faible implication (V) ; non impliqué : l'un travaille, l'autre est désengagé (D) ; problèmes différents (DP).

Comme le montre le Tableau 2, ces six degrés se retrouvent dans notre matrice. Pour les trois premiers, les utilisateurs sont impliqués dans la même tâche avec le même but mais avec des interactions plus ou moins couplées (communication ou zone de travail différente). On retrouve donc la hiérarchie observée par Tang et al. Pour les trois derniers degrés, le degré de couplage est faible voire inexistant que ce soit pour un couplage au niveau interaction (V et D) qu'au niveau activité (D

et DP). Le degré DP correspondrait à notre degré le plus faible où il n'y a aucun couplage au niveau interaction et ni au niveau des tâches puisqu'il n'y a aucune collaboration. Enfin, nous n'avons pas considéré les niveaux marqués d'un "?" car une application ne requérant pas de couplage au niveau de l'activité ne nécessite pas de couplage au niveau de l'interaction concrète.

Degrés de couplage		Dimension activité		
		Non couplé *individuel*	Faible *coopératif*	Fort *collaboratif*
Dimension Interaction Utilisateur	Non couplé	DP	D	D
	Faible	?	V	SPDA
	Intermé-diaire	?	V	VE
	Fort	?	V	SPSA

Tableau 2 : Matrice des degrés de couplage de l'interaction, illustration issue de [Jourde 2009]

Dans le but de montrer l'existence de ces différents niveaux de couplage, nous avons conçu deux expériences utilisateurs. La première expérience repose sur un jeu de conduite coopératif et explore les trois cases V de la matrice des degrés de couplage de l'interaction : une activité coopérative, et trois interactions utilisateur différentes. Par cette expérience, nous visons à montrer que les sujets ressentent différemment le couplage de l'interaction selon l'IHM qui leur est proposée. La deuxième expérience repose sur un jeu de tir multiutilisateur. Dans cette expérience, nous adoptons une IHM unique pour les différents utilisateurs et les mêmes interactions avec le système. Nous faisons en revanche varier l'activité entre les utilisateurs. Elles peuvent être collaboratives, coopérative, ou individuelle. Nous montrons ainsi l'impact de ce changement d'activité sur la perception du couplage entre les utilisateurs et incidemment sur l'intérêt que portent les sujets au jeu.

Dans les sections suivantes, nous présentons les jeux multiutilisateurs que nous avons développés pour réaliser ces expériences, avant de décrire le protocole expérimental, les résultats des expérimentations et nos conclusions.

1.6.1 Camion de pompier : une activité coopérative et plusieurs IHMs

Le premier collecticiel que nous avons développé est inspiré du camion de pompier [Stuckel 2008]. Comme pour la version originale, notre version met en jeu deux joueurs incarnant respectivement le conducteur d'un camion de pompier et le pilote de la lance à incendie situé sur le toit du camion. L'objectif des joueurs est d'éteindre des feux à l'aide de l'eau projetée par la lance à incendie.

Dans notre version, nous proposons un parcours (circuit) sur lequel sont repartis des feux. Le circuit doit être parcouru trois fois par le camion pour le terminer. Dans ce jeu, l'objectif du conducteur est de terminer le circuit le plus rapidement possible tandis que l'objectif du pilote de la lance est d'éteindre les feux grâce à l'eau projetée par la lance. Pour cela, il doit maintenir le jet d'eau sur le feu pendant un court laps de temps (environ 1 seconde et demie).

L'activité proposée aux utilisateurs est coopérative selon la définition que nous avons adoptée. En effet, les tâches sont réparties a priori entre les utilisateurs : l'un des utilisateurs conduit le camion, tandis que l'autre pilote la lance à incendie. Avec ce démonstrateur, nous avons cherché à montrer l'impact de l'IHM sur la perception du couplage par les utilisateurs. C'est-à-dire que pour une activité donné, les utilisateurs perçoivent différemment le couplage entre eux, selon l'IHM qui leur est proposée. Pour cela, nous avons développé trois versions de l'interface en sortie correspondant à trois vues différentes sur le circuit. Les trois vues sont illustrées à la Figure 6 :

- Une vue de dessus (Figure 6 A) qui est centrée sur la position du camion.
- Une vue liée au camion (Figure 6 B) qui s'oriente selon l'orientation du camion.
- Une vue liée à la lance (Figure 6 C) qui s'oriente selon l'orientation de la lance.

La lance à incendie est orientable selon un angle maximal de 30° sur la droite ou la gauche de l'axe du camion. Puisque la lance est fixée sur le toit du camion, toutes modifications de l'orientation du camion a un impact fort sur l'orientation du jet d'eau, et ce, notamment lorsqu'il s'agit de viser les feux répartis sur le circuit. La conduite du conducteur a donc un impact important sur la capacité du pilote de la lance à incendie à éteindre des feux. Nous avons représenté cela sur les trois vues de la Figure 6. Sur la vue de dessus, la vue camion est orientée selon un angle θ_{camion} par rapport au référentiel de l'écran (0, x, y). La lance s'oriente elle-même selon un angle θ_{lance} autour de l'axe du camion. Ainsi l'orientation du jet d'eau est la combinaison des angles θ_{camion} et θ_{lance}. Dans le cadre de la vue liée au camion (B), le camion est fixe par rapport à l'écran. En effet, seule la lance est orientable. Aussi, l'angle θ_{camion} est égal à zéro. L'orientation du jet d'eau est ainsi égale à θ_{lance}.

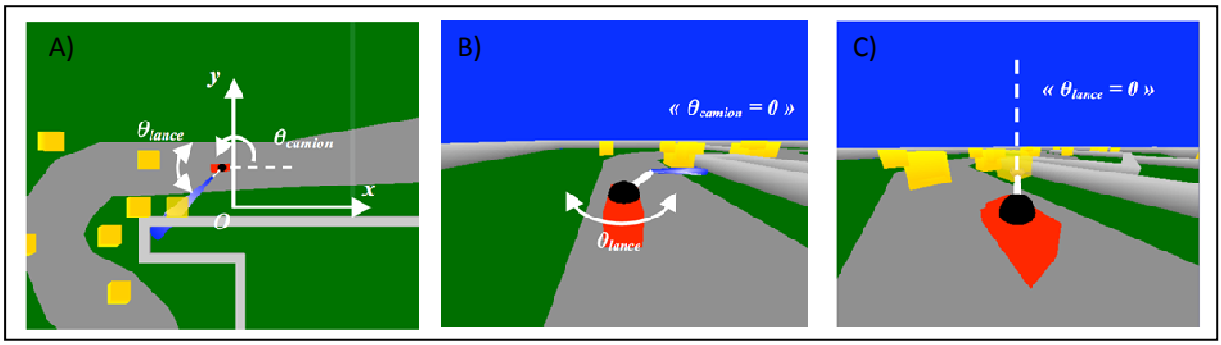


Figure 6 : Vues (A) de dessus, (B) liée au camion et (C) liée à la lance.

Avec ces trois vues de l'interface concrète, nous pouvons proposer trois scénarios correspondant aux trois niveaux de couplage de l'interaction :

- Couplage fort : les deux joueurs jouent chacun avec la vue de dessus. Il s'agit d'un cas de WYSIWIS strict.
- Couplage faible : le conducteur utilise la vue de dessus, et le pilote de la lance la vue liée au camion. Les vues étant différentes, il s'agit d'un WYSIWIS relâché. De plus, pour le pilote, l'orientation de la lance ne dépend plus de l'orientation du camion, puisque celui-ci est fixe à l'écran (θ_{camion} = 0).
- Couplage intermédiaire : le conducteur utilise la vue de dessus et le pilote la vue liée à la lance. La vues étant différentes, il s'agit d'un WYSIWIS relâché, cependant, nous considérons qu'il s'agit d'un niveau intermédiaire, car toutes les actions du conducteur ont un impact

direct sur la vue du pilote. L'orientation de la vue est définie par l'orientation de la lance qui est elle-même liée à l'orientation du camion.

Nous avons mené l'expérimentation selon la méthode du magicien d'Oz pour deux raisons principales :

- D'une part, seul le rôle de pilote de la lance subit les effets du couplage, aussi, interroger le conducteur sur le couplage a peu de sens.
- D'autre part, un des critères d'évaluation est la performance du pilote de la lance mesurée en fonction du nombre de feux éteints. Ce critère peut être faussé selon la conduite du conducteur.

Ainsi, lors de l'expérimentation, nous simulons la conduite du conducteur par un enregistrement de conduite. Nous utilisons le même enregistrement quel que soit la vue utilisée par le sujet afin d'éviter de fausser les résultats selon la conduite. Aucun sujet ne nous a déclaré s'être rendu compte que la conduite du camion était simulée. Nous avions prévu ceci en choisissant un enregistrement de conduite sans particularité évidente (comme une sortie de route). De plus les sujets doivent réussir leur tâche qui requiert suffisamment d'attention pour ne pas se préoccuper de la conduite. Enfin, le sujet et le magicien sont disposés face à face, afin d'éviter au sujet d'identifier les incohérences possibles entre les actions du magicien et l'enregistrement de conduite, comme cela est illustré à la Figure 7.

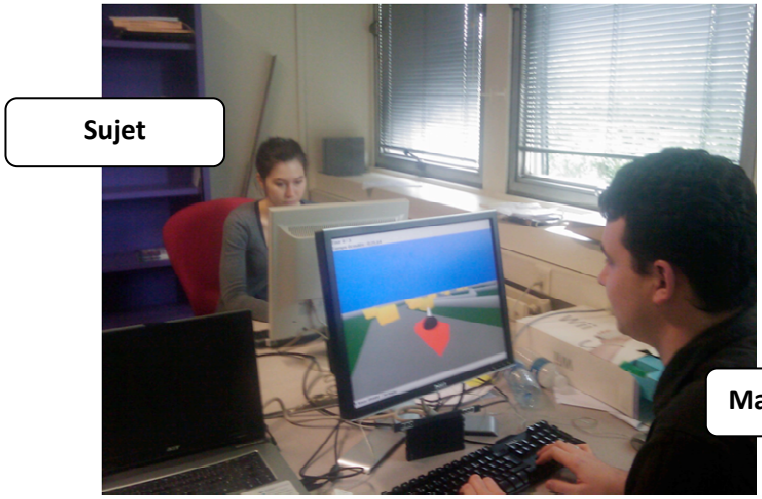


Figure 7 : Disposition du pilote de la lance (sujet) et du conducteur du camion (Magicien d'oz) pour l'expérimentation.

Nous avons mené l'expérimentation avec 16 sujets âgés de 20 à 45 ans dont 12 sujets masculins et 4 sujets féminins, 4 sujets jouant très peu, 5 sujets jouant régulièrement et 13 sujets travaillant dans le domaine de l'informatique.

Chaque sujet a dû utiliser une fois les trois vues. L'ordre de passage des vues a été modifié pour chaque sujet afin d'éviter l'effet d'apprentissage sur les résultats.

Enfin, pour familiariser les sujets aux différentes vues proposées, nous avons réalisé un circuit d'essai contenant seulement trois feux. Nous avons fait jouer ce circuit aux sujets avant chaque utilisation

du grand circuit avec une vue donnée. Pour ce circuit d'essai, la conduite du camion est réalisée en direct par le complice qui incarne ensuite le magicien d'Oz.

Nous avons soumis les sujets à un questionnaire à l'issue de l'utilisation de chaque vue (Annexe 1), ainsi qu'à l'issue de l'expérimentation. Après l'utilisation d'une vue, nous avons demandé aux sujets de noter de 1 (faible) à 10 (fort), l'impact de la conduite sur leur tâche, ainsi que la jouabilité de la vue. A l'issue de l'expérience, nous avons demandé aux joueurs d'indiquer avec quelle vue ils avaient ressenti le plus de contraintes, et avec quelle vue ils en avaient ressenti le moins. Enfin, nous avons questionné les sujets sur l'application elle-même en dehors des considérations graphiques en guise de commentaires libres.

Les résultats de l'expérience de la Figure 8 montrent une différence de perception du degré de dépendance du pilote de la lance à incendie par rapport au conducteur du camion selon la vue utilisée. Ainsi, 62,5% des utilisateurs (10 sur 16) déclarent que la vue liée au camion est la moins contraignante, tandis que 50% des utilisateurs (8 sur 16) déclarent la vue liée à la lance comme étant la plus contraignante. Enfin, la vue de dessus est jugée par 50% des utilisateurs (8 sur 16) comme la vue présentant un niveau de contrainte intermédiaire aux deux autres vues.

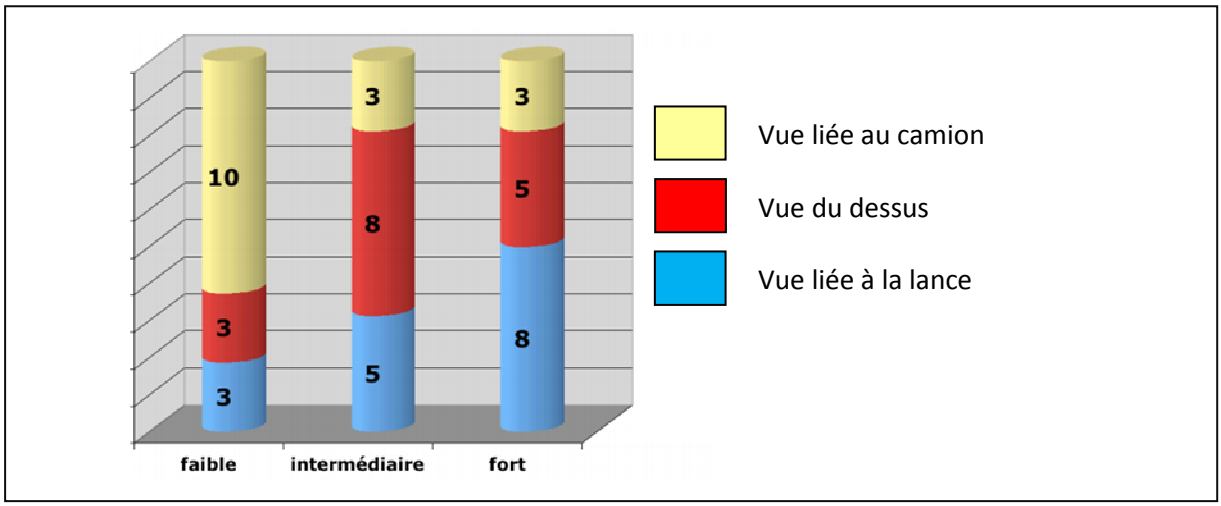


Figure 8 : Perception de la dépendance du pilote selon la vue utilisée.

Par ailleurs, aucun des sujets n'a mentionné que la tâche de viser les feux avec la lance était liée à la tâche de conduite du conducteur, ce qui souligne le fait qu'il s'agit d'un couplage faible au niveau de l'activité. En effet, 50% des sujets indiquent même qu'ils se sont sentis indépendants du conducteur : « on oublie le conducteur », « on ne sent pas assez la présence de l'autre ». Enfin, une large majorité des sujets a conclu que la contrainte émanait principalement de l'interaction : « difficile de s'adapter au mouvement », « quand le camion bouge on ne peut pas le prévoir […] », « on est déstabilisé par le changement d'orientation ». Ces réponses confirment que les tâches coopératives que nous avons mises en œuvre dans le prototype sont perçus comme faiblement couplées.

Les résultats obtenus semblent confirmer notre hypothèse d'existence de trois niveaux de couplage et de leur hiérarchie. Cependant, notre approche expérimentale et exploratoire ne permet de conclure que pour le cas du couplage faible, et il serait nécessaire d'élargir l'échantillon pour confirmer les tendances qui se dégagent de cette expérimentation.

Enfin, cette expérience montre l'influence que peut avoir le choix d'une IHM sur le sentiment de collaboration entre les utilisateurs. La différence de perception du couplage qui a été mise en relief dans cette expérimentation est intéressante dans le cas d'un jeu, mais elle peut aussi se révéler être un problème dans le cadre d'une application de travail. Le choix d'une présentation en sortie et du mode de couplage pour chaque interaction est donc particulièrement crucial et doit être réfléchi en amont de la réalisation.

1.6.2 Les 4 magiciens : une IHM et plusieurs activités

Le deuxième collecticiel que nous avons développé est un jeu de tir à plusieurs, nommé « les 4 magiciens ». Dans ce jeu, plusieurs joueurs (de 2 à 4) peuvent participer à la défense d'une ville incarnant chacun un magicien capable de lancer des sorts contre des ennemis venus pour détruire la ville. L'objectif des joueurs est de tenir le plus longtemps possible.

Les 4 magiciens proposent 3 modes de jeu. Dans les trois modes, l'interaction est identique. En effet, l'interface est partagée, et l'interaction concrète d'un utilisateur consiste à viser et à tirer sur un ennemi pour le détruire. La différence entre les modes réside dans la combinaison d'actions élémentaires qui permet de détruire un ennemi. Les modes de jeu sont les suivants :

- Mode collaboratif, où chaque joueur peut lancer des sorts sur les ennemis pour les détruire. Les effets des sorts de chaque joueur se cumulent sur un ennemi donné provoquant plus de dommages et donc une destruction plus rapide de l'ennemi. Il s'agit d'un mode d'interaction faiblement couplée au niveau de l'activité.
- Mode coopératif, où, pour détruire un ennemi, les deux joueurs doivent impérativement combiner leur sort sur cet ennemi. L'ennemi ne subit en effet des dommages que si les zones de sort de tous les joueurs sont localisées sur lui. Il s'agit d'un mode fortement couplé au niveau de l'activité.
- Mode individuel (ou exclusif), où, chaque joueur peut lancer des sorts sur un ennemi pour le détruire mais les effets des sorts des différents joueurs ne se cumulent pas. Il s'agit d'un niveau de couplage intermédiaire.

L'expérimentation a pour objectif de montrer la différence de perception du couplage chez l'utilisateur selon les trois modes proposés.

Le jeu se joue devant un écran unique partagé entre les joueurs. Il s'agit donc d'un WYSIWIS strict puisque les effets des actions d'un utilisateur sont répercutés sur l'interface graphique à destination de tous les utilisateurs. Pour jouer, chaque joueur dispose d'une WiiMote (console Wii de Nintendo).

L'interface du jeu illustrée à la Figure 9 correspond à un cas d'utilisation à deux joueurs. Elle est composée deux zones. La zone supérieure est la prairie qui s'étend devant la ville à défendre. La zone inférieure représente les remparts de la ville. L'interface contient plusieurs indications d'état du jeu. A gauche et à droite, sont représentées deux jauges indiquant le niveau d'énergie de chaque joueur (l'énergie permet au magicien de lancer leur sort). Les points de vies des joueurs sont représentées au niveau des remparts sous la forme d'une barre. Enfin, chaque joueur dispose d'un pointeur et peut lancer un sort qui peut provoquer des dommages aux ennemis. Le jeu propose deux modes de comptage du score des joueurs. Un score individuel qui représente le nombre d'ennemis détruits par le joueur et un score collectif qui représente le nombre total d'ennemis détruits par les joueurs.

Les utilisateurs peuvent déplacer leur pointeur sur l'écran en le visant avec la WiiMote. Ils peuvent lancer un sort en maintenant enfoncé le bouton B, ce qui consomme de l'énergie. Ils peuvent également recharger leur énergie en maintenant le bouton A enfoncé et en secouant la WiiMote de bas en haut rapidement.

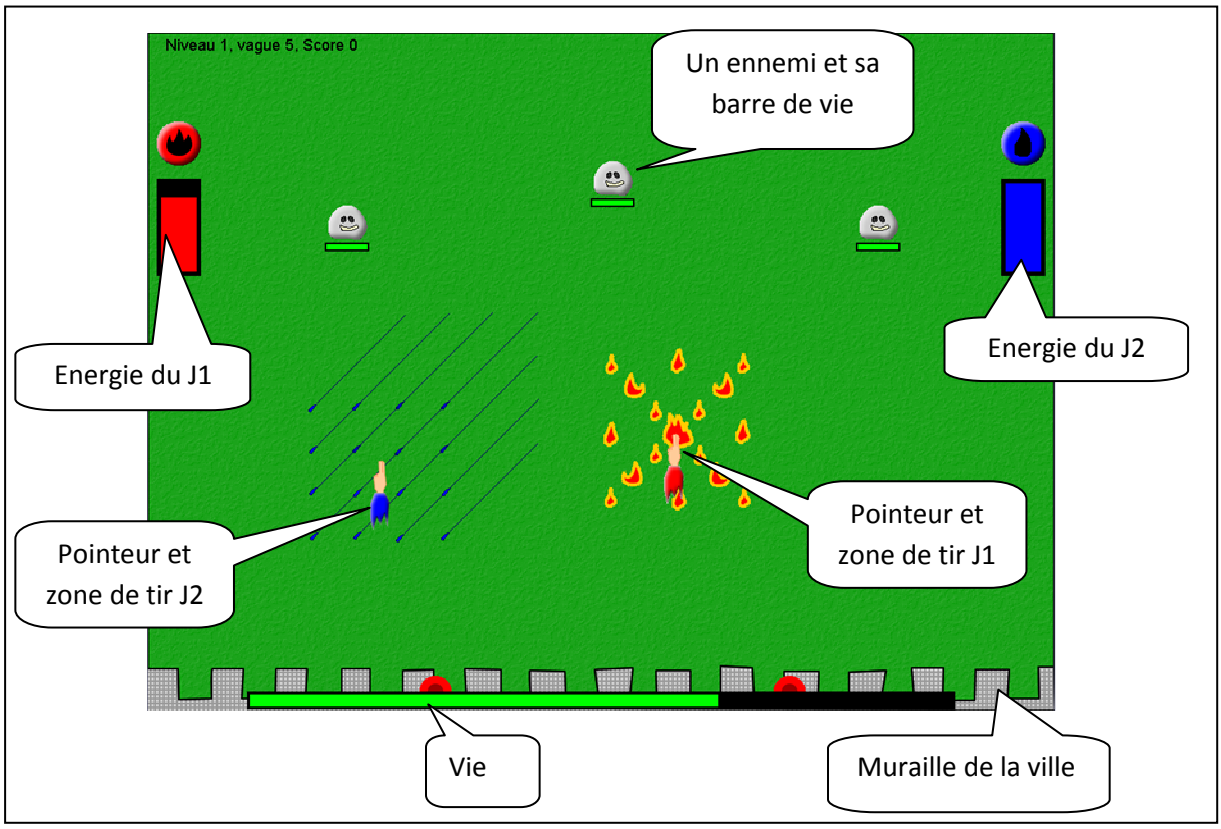


Figure 9 : Interface utilisateur du jeu « les 4 magiciens » (ici en mode à deux joueurs).

Le principe du jeu est de faire arriver des vagues successives d'ennemis sur la ville provenant du haut de l'écran. Chaque vague est constituée soit d'un nombre plus important d'ennemis, ou d'ennemis plus résistants. Le jeu est paramétré pour rendre les dernières vagues impossibles à détruire.

Pour notre expérimentation, nous avons choisi de faire utiliser l'application par seulement deux utilisateurs à la fois (ce qui nous permet d'obtenir plus de résultats qu'en mode à quatre joueurs pour un même nombre de sujets). 24 sujets âgés de 20 à 60 ans ont participé à cette expérimentation, dont 12 hommes et 12 femmes, et 14 travaillant dans le domaine de l'informatique. Nous avons demandé aux sujets de faire une partie du jeu dans chacun des trois modes. L'ordre de passage des modes a été changé pour chaque groupe afin d'éviter l'effet d'apprentissage. Avant de lancer une partie complète dans un mode donné, nous avons fait jouer aux joueurs une petite partie de 1 niveau (10 vagues d'ennemi) pour les familiariser avec le style d'interaction.

A l'issue de l'utilisation de chaque mode et à la fin de l'expérimentation, nous avons questionné les sujets (questionnaire disponible en Annexe 2). Après l'utilisation d'un mode, nous avons demandé aux sujets de noter de 1 (faible) à 10 (fort) la jouabilité du mode de jeu, et les contraintes induites par les actions de l'autre utilisateur. Ensuite, nous avons questionné les sujets sur l'intérêt du jeu en lui-même, en mettant de côté la qualité de l'interface graphique.

Comme le montre la Figure 10, 67% des sujets ont indiqué que le mode collaboratif présentait l'interaction la plus faiblement couplée, comparé aux 15% et 19% pour les deux autres modes. Ensuite, le mode coopératif a été identifié par 60% des sujets comme le mode présentant le plus fort niveau de couplage. Enfin, 56% des participants ont indiqué que le mode individuel (ou exclusif) était le mode de couplage intermédiaire. Par ailleurs, nous avons été surpris par le fait que le mode exclusif suivi de près par le mode coopératif a permis aux joueurs d'obtenir les meilleurs résultats (50% des groupes ont eu leurs meilleurs scores pour le mode coopératif et 50% pour le mode individuel). Enfin, le mode collaboratif a toujours été classé troisième en terme de performance. Pour terminer, la majorité des utilisateurs ont présenté le mode coopératif comme « le plus amusant » et « le plus divertissant » parmi les trois modes. Ainsi, dans le cadre du jeu, les contraintes perçues par les utilisateurs du fait de la présence d'autres utilisateurs sont ressenties positivement.

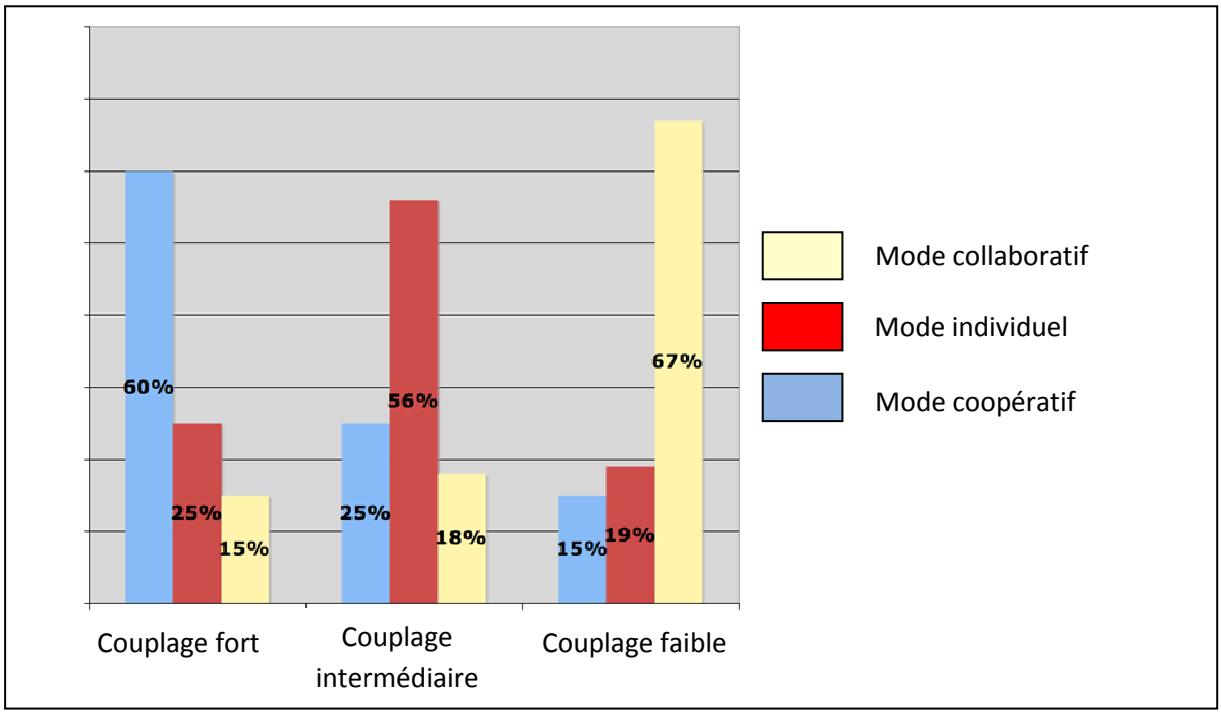


Figure 10 : Perception du couplage en fonction du mode d'utilisation.

Pour conclure, cette expérience nous a permis de montrer que pour une IHM donnée, le couplage de l'interaction au niveau de l'activité est ressenti par les utilisateurs, et peut renforcer ou diminuer l'intérêt d'un système pour les utilisateurs. Ces résultats illustrent la diversité des réponses qu'il est possible d'apporter en terme de couplage des activités, ainsi que la difficulté d'effectuer le bon choix de couplage de l'interaction lors de la conception.

1.6.3 Synthèse

Notre étude expérimentale a permis de conforter nos hypothèses quant à différents niveaux de couplage définis selon les deux axes de l'activité et de l'interaction concrète. Cependant, les résultats obtenus doivent être confirmés par un plus grand nombre de sujets. L'élément le plus saillant de cette étude est la nécessité de considérer la question du couplage de l'interaction aussi bien au niveau des activités, qu'au niveau de l'interaction concrète. En effet, la première expérimentation indique que le niveau de contrainte ressenti par les utilisateurs est différent selon l'interface concrète adoptée, tandis que la deuxième expérimentation met en relief la différence d'intérêt des

utilisateurs pour un système avec une interface concrète similaire et des niveaux de couplage des activités différents.

1.7 Synthèse sur les systèmes multiutilisateurs

Dans cette partie, nous avons introduit le domaine du TCAO et la terminologie de ce domaine. Pour cela, nous avons présenté deux classifications : une classification reposant sur les distances Espace – Temps entre les utilisateurs d'un système et une classification fonctionnelle des collecticiels. Nous avons présenté la définition de concepts clefs du domaine : la collaboration, la coopération, ainsi que les notions de rôle, d'agent, et de groupe. Nous avons ensuite présenté un ensemble de propriétés ergonomiques dédiées à la conception des systèmes multiutilisateurs, comme la conscience de groupe et le couplage de l'interaction. Enfin, pour illustrer la difficulté et l'importance de la prise en compte du couplage de l'interaction lors de la conception des activités mais aussi des interactions concrètes, nous avons présenté une étude expérimentale que nous avons menée sur le couplage et notamment sur la caractérisation du couplage selon deux axes : l'activité et l'interaction concrète.

La partie suivante est consacrée à la présentation du domaine de la multimodalité et en particulier à l'illustration des problèmes spécifiques que pose la conception de l'interaction multimodale.

2 Systèmes Multimodaux

Les systèmes multimodaux et les techniques d'interaction multimodale constituent le sujet d'étude du domaine de la Multimodalité. Un système multimodal est un système proposant aux utilisateurs plusieurs moyens d'interaction (modalités d'interaction) et/ou combinaison de modalités d'interaction.

Le domaine de la Multimodalité s'inscrit dans le domaine de l'IHM. Il consiste à étudier, concevoir mais surtout caractériser une variété de modalités d'interaction et de combinaisons de modalités, dans la perspective d'améliorer l'utilisabilité des systèmes interactifs. Le domaine de la Multimodalité s'intéresse en particulier aux problèmes liés à l'utilisation conjointe de modalités pour une tâche donnée. Nos travaux portant sur la conception de l'interaction, nous focalisons notre présentation du domaine de la Multimodalité sur la conception de l'interaction multimodale. Ainsi nous ne détaillons pas par exemple les nombreux travaux sur la réalisation logicielle des moteurs de fusion pour la multimodalité.

La principale difficulté de conception d'un système multimodal provient de la multiplicité des modalités d'interaction existantes, qui implique aussi une multiplicité exponentielle des combinaisons de modalités envisageables pour un système donné. Lors de la conception d'un système multimodal, le concepteur doit définir les modalités d'interaction qui seront disponibles aux utilisateurs du système et décrire les modalités et/ou combinaisons de modalités qui pourront être utilisées pour chaque tâche. Il convient ensuite de définir comment une modalité d'interaction ou une combinaison de modalités d'interaction est utilisée pour la réalisation d'une tâche donnée. Ainsi, la réalisation d'une tâche peut requérir l'utilisation conjointe de deux modalités, ou laisser le choix à l'utilisateur d'utiliser une modalité parmi plusieurs disponibles.

Apres une présentation de la terminologie du domaine de la Multimodalité, nous présentons les éléments clefs à prendre en considération lors de la conception d'un système multimodal. Enfin, nous illustrons le domaine de la Multimodalité par plusieurs exemples d'applications multimodales.

2.1 Définition d'une modalité et d'un système multimodal

Dans le cadre du domaine de la Multimodalité, cette section présente la terminologie qui est adoptée dans ce mémoire. Nous rappelons d'abord la définition d'une modalité d'interaction, avant de présenter ce qu'est un système multimodal. Nous soulignons ensuite les apports de la multimodalité pour l'interaction homme-machine.

2.1.1 Modalité d'interaction

Nous adoptons pour nos travaux la définition de [Nigay 1996] où, une modalité d'interaction est définie par un couple <D, L> où D désigne un dispositif physique et L désigne un langage d'interaction :

- Un dispositif physique est un élément du système qui acquièrt ou présente des informations et peut être par exemple : un clavier, un microphone, un magnétomètre, un écran, un haut parleur, etc.
- Un langage d'interaction définit une grammaire pour l'usage d'un dispositif physique, comme, le langage pseudo naturel ou la manipulation directe.

Ainsi, une modalité comme la parole peut être décrite par le couple <microphone, langage pseudo naturel>.

2.1.2 Modalité en entrée ou en sortie

Deux types de modalités peuvent être considérés selon le sens de l'interaction entre l'utilisateur et le système. Lorsqu'une modalité permet à l'utilisateur de fournir des informations au système, elle correspond à une modalité d'entrée. A l'inverse, une modalité de sortie est une modalité qui permet au système de présenter des informations à l'utilisateur.

2.1.3 Modalité passive ou active

Parmi les modalités d'entrée, [Turk 2000] distingue deux sous-catégories de modalités : les modalités actives et les modalités passives :

- Une modalité d'entrée est active si elle est utilisée explicitement par l'utilisateur. Un clavier utilisé pour compléter un champ de texte correspond à une modalité active.
- Une modalité d'entrée est dite passive si l'utilisateur ne l'utilise pas de manière explicite. C'est le cas par exemple d'un système de localisation GPS intégré dans un téléphone portable et servant de modalité d'entrée pour le positionnement de l'utilisateur sur une carte.

L'utilisation de modalités passives pour un système interactif est particulièrement intéressante car elle n'augmente pas la charge cognitive des utilisateurs [Oviat 2000]. En effet, les informations produites par ces modalités sont transmises au système sans que l'utilisateur ait besoin d'agir explicitement. Il est cependant préférable que l'utilisateur soit conscient que ces modalités d'interaction passives sont exploitées pour l'interaction et que l'utilisateur puisse les désactiver. C'est

le cas dans les systèmes de messagerie instantanée où l'utilisation de la vidéo est soumise à une acceptation explicite des utilisateurs.

2.1.4 Système multimodal

Un système interactif est multimodal s'il propose au moins deux modalités d'interaction pour un sens donné (entrée du système ou sortie du système). Ainsi, un système est qualifié de multimodal en sortie, respectivement en entrée, s'il dispose d'au moins deux modalités d'interaction en sortie, respectivement en entrée.

2.1.5 WIMP et Post WIMP

L'acronyme WIMP (*Windows, Icons, Menus and Pointing device*) désigne un paradigme d'interaction introduit dans les années 1970 et reposant sur des interfaces graphiques constituées de fenêtres, d'icônes et de menus, et où l'utilisateur interagit avec le système au moyen d'un dispositif de pointage tel que la souris.

Le paradigme d'interaction WIMP restreint les possibilités en termes de modalités d'interaction. Par exemple en entrée, seules les modalités basées sur le clavier et la souris sont considérées. Face à la multitude actuelle des possibilités en termes de modalités (comme la parole, les gestes 2D et 3D, etc.), le terme Post-WIMP [Dam 1997] a été introduit pour désigner les interfaces reposant sur des modalités autres que celles classiques basées sur le clavier et la souris.

Les systèmes Post-WIMP tirent parti des travaux du domaine de la Multimodalité, puisqu'ils définissent un cadre d'utilisation des modalités non-conventionnelles. En revanche, les systèmes Post-WIMP ne sont pas nécessairement multimodaux. Ainsi, le jeu de ballon présenté dans [O'Hara 2009] est un exemple de système Post-WIMP non multimodal. Les utilisateurs sont filmés par une caméra. Le film est augmenté par des ballons rouges projetés sur un écran géant. Les utilisateurs peuvent effectuer un mouvement du bras pour pousser un ballon. L'objectif est de faire fusionner les ballons par collision. Le système est Post-WIMP puisqu'il ne repose pas sur le paradigme WIMP, mais n'est pas multimodal, puisqu'il ne comporte qu'une modalité en entrée (modalité gestuelle) et qu'une modalité en sortie (modalité graphique).

2.1.6 Apport de la Multimodalité

La Multimodalité permet d'envisager l'interaction dans un cadre non figé à un nombre restreint de modalités conventionnelles basées sur la souris, le clavier et l'écran. Dans ce contexte, la Multimodalité aborde l'interaction sous l'angle de la diversité des modalités d'interaction et de leurs combinaisons.

Cette variété permet d'augmenter l'utilisabilité d'un système interactif [Reeves 2004] en améliorant par exemple son adaptabilité aux besoins instantanés des utilisateurs. Ainsi un système proposant plusieurs modalités en entrée pour réaliser une tâche donnée permet aux utilisateurs de choisir la modalité qu'ils souhaitent utiliser en fonction de leurs capacités, de leurs besoins, de leurs préférences et du contexte.

La robustesse d'un système interactif peut également être renforcée par l'utilisation de plusieurs modalités, et ce notamment par l'utilisation conjointe de modalités pour la réalisation d'une tâche donnée. En entrée, il s'agit par exemple de demander à l'utilisateur d'utiliser plusieurs modalités

d'interaction pour une tâche critique. En sortie, elle peut renforcer l'observabilité du système en offrant plus de moyens d'informer l'utilisateur sur l'état du système.

Cependant, comme il est souligné dans [Oviatt 1999] à partir de plusieurs études expérimentales, la Multimodalité n'est pas forcément un apport pour l'interaction. Ainsi des expérimentations [Oviatt 1999] montrent que les utilisateurs d'un système multimodal utilisent de manière récurrente certaines modalités à leur disposition en fonction de leur préférence personnelle (peu de changements après avoir pris en main une modalité), et n'utilisent que rarement plusieurs modalités de façon combinée pour définir des commandes multimodales.

2.2 Espace de conception : élements clefs

La conception d'un système multimodal nécessite de définir pour chaque tâche : la modalité ou la combinaison de modalités qui permet de la réaliser. Pour cela, et face à la multiplicité des solutions de conception, il est nécessaire de disposer d'outils conceptuels pour caractériser une modalité d'interaction, mais également pour caractériser les combinaisons de modalités.

Dans cette section, nous présentons deux modèles conceptuels permettant de caractériser une modalité d'interaction, puis nous présentons les propriétés CARE et TYCOON qui permettent de caractériser des combinaisons de modalités.

2.2.1 Caractérisation d'une modalité

Nous présentons ici deux espaces de caractérisation des modalités d'interaction. La taxonomie de Bernsen [Bernsen 1994] dédiée aux modalités de sortie, et le modèle des modalités [Bouchet 2007] qui identifie un ensemble de propriétés des modalités et de ses composantes : le dispositif physique d'interaction et le langage d'interaction.

2.2.1.1 Taxonomie de Bernsen

Dans [Bernsen 1994], l'auteur propose un espace de caractérisation des modalités en sortie d'un système interactif reposant sur la représentation utilisée par la modalité. Ainsi, une modalité est d'abord considérée selon son média d'expression, qui peut être le graphisme, le son, ou le touché correspondant respectivement aux capacités sensorielles visuelles, auditives et tactiles des utilisateurs. Ensuite, l'auteur identifie quatre propriétés pour caractériser la représentation adoptée par une modalité. Ces propriétés indiquent :

- la nature linguistique ou non de la représentation, c'est-à-dire le fait qu'elle repose ou non sur un langage ;
- la nature dynamique ou statique de la représentation, qui caractérise l'intégration ou non de la dimension temporelle dans la représentation ;
- la nature arbitraire ou non de la représentation, c'est-à-dire le fait qu'elle soit choisie ou non dans le cadre de convention ; et,
- la nature analogue ou non qui indique le fait que la représentation fonctionne par ressemblance avec la réalité ou non.

Avec ces différentes propriétés, il est possible de caractériser une modalité en sortie. Ainsi, l'affichage numérique de l'heure peut être défini par les propriétés {Graphique, Linguistique, Dynamique, Non-Arbitraire, Non-Analogue}.

2.2.1.2 Modèle de modalités

Le modèle de modalités proposé dans [Bouchet 2007] identifie un ensemble de propriétés qui permet de caractériser une modalité d'interaction. Le modèle définit des caractéristiques globales telles que l'adéquation entre le dispositif physique d'interaction et le langage d'interaction, le degré d'interaction, c'est-à-dire la distance (spatiale et temporelle) entre l'objet manipulé et les actions à effectuer selon une modalité, et le type d'interaction proposé qui repose sur des modalités actives ou passives.

Le modèle définit ensuite deux séries de propriétés pour caractériser les éléments du couple que forme une modalité d'interaction : le dispositif et le langage d'interaction.

Les propriétés relatives aux dispositifs physiques sont réparties selon trois critères : celles intrinsèques au dispositif, celles qui sont relatives à leur utilisation, et enfin des propriétés relatives à l'état du dispositif lors de son utilisation. Les caractéristiques intrinsèques d'un dispositif concernent son nom, son poids, ou encore son autonomie et sa précision. Les caractéristiques relatives à l'utilisation du dispositif par un utilisateur recouvrent entre autres le niveau d'expertise requis pour utiliser le dispositif et le mode de communication adopté (gestuel, oral). Enfin, les caractéristiques des dispositifs concernant leur état décrivent leur état de marche ou le niveau de confiance que l'on peut accorder aux données issues du dispositif.

Les propriétés relatives aux langages d'interaction sont classées selon les mêmes catégories. Les caractéristiques intrinsèques à un langage d'interaction incluent le caractère linguistique, arbitraire, ou analogue comme définies par Bernsen [Bernsen 1994], ainsi que les dimensions spatiales ou encore la fréquence des données produites (discrète ou continue). Les caractéristiques d'un langage d'interaction relatives à son utilisation adressent quant à elles uniquement le niveau d'expérience de l'utilisateur.

2.2.2 Caractérisation des formes de multimodalité

Face à la diversité des dispositifs physiques d'interaction disponibles et des langages d'interaction possibles pour ces dispositifs, les espaces de caractérisation des modalités d'interaction illustrent la complexité du choix d'une modalité d'interaction pour une tâche. Cependant, le choix des modalités n'est qu'une des facettes de la conception d'un système multimodal. En effet une deuxième facette de la conception concerne des combinaisons de modalités et la forme que prennent ces combinaisons. Pour illustrer les différentes formes de multimodalité, nous présentons deux ensembles de propriétés permettant de les caractériser : les propriétés CARE et l'espace TYCOON.

2.2.2.1 Propriétés CARE

Les propriétés CARE (Complémentarité, Assignation, Redondance, Equivalence) [Coutaz 1994] caractérisent le lien entre les tâches d'un système et les modalités d'interaction dont il dispose. Les propriétés CARE permettent de spécifier la modalité d'interaction ou la combinaison de modalités utilisées pour réaliser une tâche du système.

La complémentarité de plusieurs modalités pour une tâche indique que l'usage conjoint de ces modalités est nécessaire à la transmission des informations pour la réalisation de la tâche. En entrée, cela signifie que chaque modalité utilisée pour réaliser une tâche fournit une partie de l'information nécessaire à sa réalisation. Concrètement, les informations provenant des différentes modalités d'interaction sont fusionnées pour obtenir la commande finale. Le « met ça là » de Bolt [Bolt 1980] est l'exemple séminal de complémentarité de deux modalités : pour réaliser le déplacement d'un objet, un utilisateur doit désigner un objet puis une destination pour cet objet par un pointage avec le doigt et prononcer la commande « met ça là ». Les deux modalités d'interaction <caméra, pointage> et <microphone, langage pseudo naturel> sont utilisées de manière complémentaire. En sortie, cela signifie que chaque modalité utilisée pour réaliser la tâche va présenter une partie de l'information. Les informations sont décomposées pour être présentées selon les différentes modalités. On parle alors de fission de l'information, en référence à la fusion de l'information réalisée pour des modalités d'entrée complémentaires.

L'assignation d'une modalité à une tâche signifie que l'usage de cette modalité est nécessaire pour réaliser cette tâche, et qu'aucune autre modalité ne peut permettre la réalisation de cette tâche. Subséquemment, cela signifie que l'information transmise par la modalité est nécessaire et suffisante pour la réalisation de la tâche. En entrée d'un système par exemple, le concepteur peut décider que la souris sera le seul moyen de tracer une ligne au sein d'un éditeur de dessin. En sortie, cela revient à considérer qu'un ensemble d'information est présenté par une modalité unique.

La redondance de plusieurs modalités pour une tâche indique d'une part que plusieurs modalités sont nécessaires pour réaliser une tâche, et d'autre part, que chacune d'elles porte l'intégralité de l'information. En entrée d'un système, cela peut consister à demander à un utilisateur d'effectuer la même commande avec deux modalités afin de garantir la cohérence des entrées ou d'augmenter la fiabilité d'un système. Par exemple, pour supprimer un élément sélectionné, le système peut requérir que l'utilisateur prononce « supprimer » et qu'il clique sur un bouton « supprimer ». En sortie, cela peut permettre d'améliorer la prise en compte d'une information par l'utilisateur (propriété d'insistance). Ainsi, la plupart des appareils de navigation GPS informent le conducteur de la direction à prendre au prochain carrefour d'une part graphiquement sur l'écran de l'appareil mais également en diffusant par haut parleur un message oral synthétisé.

Enfin, plusieurs modalités sont considérées comme équivalentes pour une tâche donnée, si chacune d'elles permet de réaliser la tâche. La disponibilité de plusieurs modalités implique qu'un choix est réalisé par l'utilisateur ou le système. En entrée, l'utilisateur pourra choisir une et une seule modalité parmi celles qui lui sont proposées pour réaliser une tâche. Par exemple pour fermer une application, l'utilisateur a souvent le choix de la modalité : il peut soit cliquer avec la souris sur le bouton « fermer », soit enfoncer des touches du clavier (ALT + F4 sous Windows). En sortie, le choix entre plusieurs modalités revient généralement au système, même si ce dernier peut être guidé par le concepteur ou l'utilisateur, notamment par le bais de modalités par défaut. La plupart des téléphones propose ainsi plusieurs manières d'avertir d'un appel : par sonnerie ou par vibration. Le choix entre les deux modalités peut être fait par le système selon le contexte d'usage dans le cas d'un système sensible au contexte, ou par l'utilisateur qui explicitement configure son téléphone et choisit la modalité de sortie à utiliser.

2.2.2.2 Espace TYCOON

L'espace TYCOON (Types de COOpératioNs) [Martin 1999] permet comme les propriétés CARE de caractériser les combinaisons de modalité. L'espace TYCOON identifie six relations que peuvent entretenir deux modalités : le transfert, l'équivalence, la spécialisation, la redondance, la complémentarité, et la concurrence.

Les relations d'équivalence, de spécialisation, de redondance et de complémentarité de l'espace TYCOON correspondent respectivement aux propriétés CARE d'Equivalence, d'Assignation, de Redondance et de Complémentarité. En revanche, les relations de transfert et de concurrence permettent de caractériser des combinaisons de modalités particulières qui ne sont pas couvertes par CARE.

La relation de transfert indique qu'une modalité fournit des informations qui sont utilisées comme entrée d'une autre modalité d'interaction. Une telle combinaison de modalités peut être identifiée par exemple dans le cas d'un système qui permettrait de saisir des commandes en langage pseudo naturel à partir d'un ensemble d'éléments prédéfinis que l'utilisateur sélectionne avec la souris. Ce type de relation de transfert correspond à une modalité définie par le couple (modalité, langage) et non (dispositif, langage). Dans l'exemple ci-dessus la modalité serait : ((souris, manipulation), langage pseudo naturel).

La relation de concurrence indique la possibilité d'utiliser simultanément plusieurs modalités pour réaliser des tâches distinctes. C'est par exemple le cas pour un système de messagerie instantanée et de téléphonie internet comme Skype. En effet, l'usage de la modalité vocale dans le cadre de la communication homme-homme laisse à l'utilisateur la possibilité d'utiliser simultanément la souris et le clavier, y compris dans le cadre de la communication avec la même personne.

2.2.3 Synthèse

Nous avons présenté deux espaces de caractérisation des modalités d'interaction et deux espaces de caractérisation des combinaisons de modalités. La diversité des propriétés introduites par ces espaces de catégorisation souligne la difficulté à laquelle le concepteur d'un système multimodal est confronté. Celle-ci se situe dans le choix des dispositifs physiques d'interaction et des langages d'interaction associés à ces dispositifs, ainsi que dans le choix des combinaisons de modalités disponibles pour réaliser une tâche donnée. Afin d'illustrer la diversité des choix disponibles, nous présentons dans la section suivante des exemples de systèmes multimodaux.

2.3 Exemples de systèmes multimodaux

Nous présentons plusieurs systèmes interactifs multimodaux afin d'illustrer la multiplicité des modalités existantes, en termes de dispositifs physiques, de langages d'interaction, et de combinaisons de ces modalités. Pour cela nous illustrons trois des quatre formes de multimodalité identifiées par les propriétés CARE : Complémentarité, Redondance et Equivalence. L'assignation présente moins d'intérêt à être illustrée car elle correspond au cas le plus classique où une modalité permet de réaliser une tâche. De plus, nous présentons des exemples proposant des modalités très variées, à la fois du point de vue des dispositifs d'interaction que des langages d'interaction.

2.3.1 Complémentarité

A l'origine de l'axe de recherche sur la multimodalité, le paradigme du "met ça là" (*put that there*) [Bolt 1980] repose sur l'usage complémentaire de deux modalités. L'utilisation du système est illustrée à la Figure 11. Il s'agit pour l'utilisateur de désigner du doigt une cible qui peut être une forme géométrique (ici un cercle rose), et d'indiquer une nouvelle position pour cette forme, tout en prononçant la phrase « *put that there* ». L'interaction avec le système repose ainsi sur une combinaison complémentaire d'une modalité décrite par le couple <caméra, geste de désignation>, et d'une modalité décrite par le couple <microphone, langage pseudo naturel>.

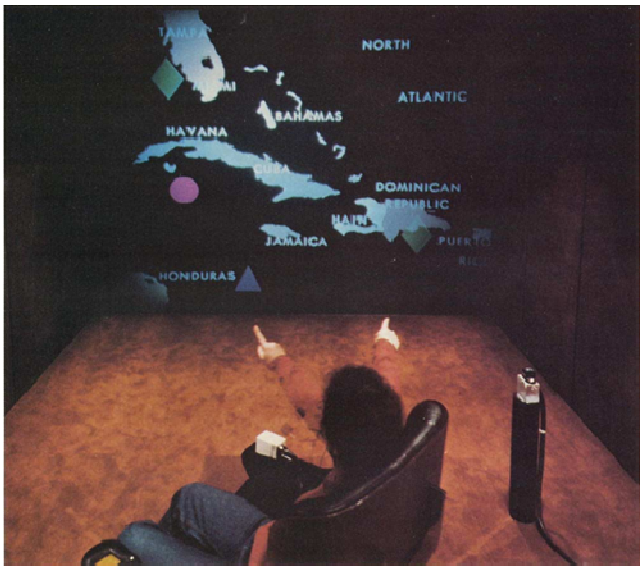


Figure 11 : Met ça là, illustration tirée de [Bolt 1980].

Plusieurs systèmes multimodaux reposent sur cette forme d'interaction (parole + geste), et notamment les systèmes qui reposent sur la manipulation d'une carte [Oviatt 1999] et les systèmes plus récents sur tables interactives. Ainsi dans [Tse 2007], les utilisateurs d'un jeu de stratégie autour d'une table peuvent spécifier des commandes par un usage complémentaire de la parole et d'un geste de désignation sur la table, comme « déplacer un soldat ici ».

Cependant, d'autres modalités d'interaction peuvent être utilisées de façon complémentaire. La Figure 12 issue de [Juras 2008] illustre un mécanisme de zoom reposant sur l'usage complémentaire de deux modalités. La première est composée d'un dispositif de pointage à distance (une Wiimote de Nintendo) et d'un langage de désignation d'un point sur un écran. La deuxième est composée d'un capteur de pression intégré au sein d'un ballon en plastique et d'un langage reposant sur des seuils de pression. Pour zoomer sur un point de l'écran, l'utilisateur désigne ce point avec la WiiMote et presse le ballon en plastique.

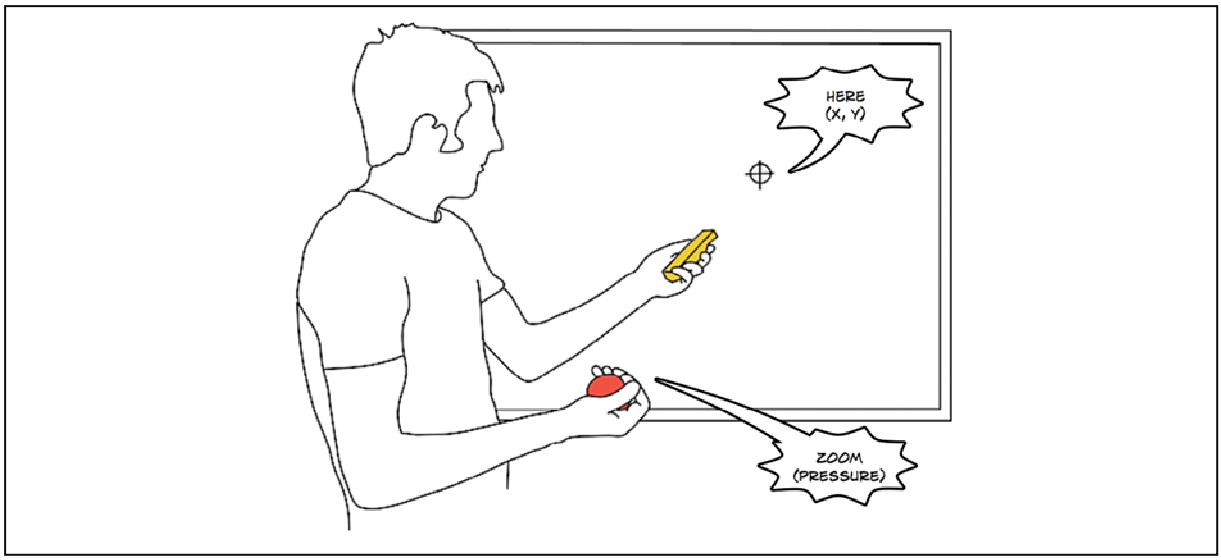


Figure 12 : Utilisation complémentaire de deux modalités : (wiimote, pointage) et (ballon, seuil de pression) [Juras 2008].

2.3.2 Equivalence

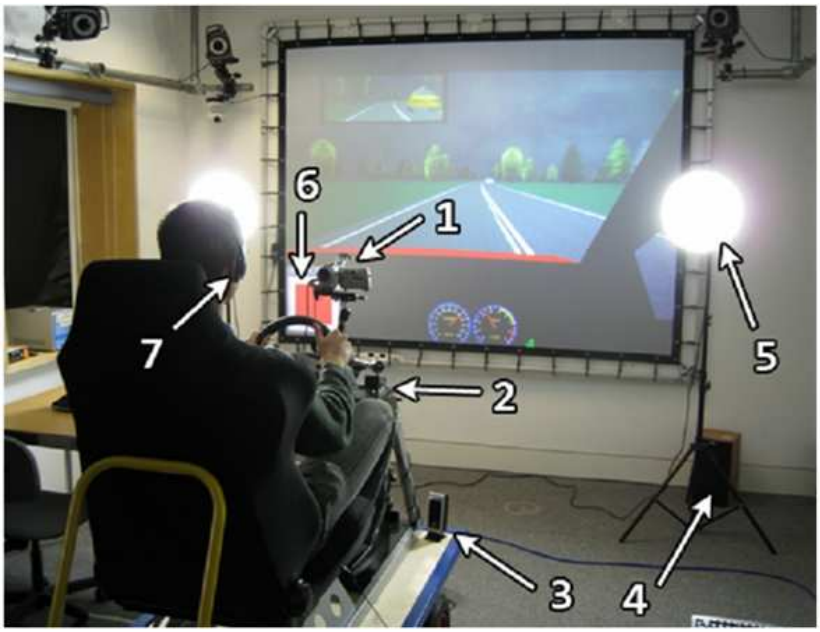
L'équivalence de plusieurs modalités pour une tâche consiste à laisser à l'utilisateur le choix de la modalité qu'il souhaite utiliser pour accomplir la tâche. Ce choix peut être guidé par exemple par son niveau d'expertise du système. Ainsi, de nombreux utilisateurs préfèrent l'usage d'un raccourci clavier tel que CTRL + S pour sauvegarder un document, plutôt que de cliquer avec la souris sur le bouton sauvegarder.

Lorsqu'un système dispose de plusieurs modalités équivalentes, il offre beaucoup de flexibilité à l'utilisateur en lui permettant d'adapter l'utilisation du système au contexte d'usage. Le système de poste de commande militaire présenté dans [Myers 2002] (illustré à la Figure 13) propose ainsi de nombreuses modalités d'interaction pouvant être utilisées de manière équivalente pour certaines tâches. Par exemple, une annotation sur un objet peut être définie au clavier ou par écriture manuscrite. De même, le système permet de naviguer dans le système interne de cartographie aussi bien avec la souris que par manipulation directe sur l'écran. Enfin, toutes les commandes peuvent également être réalisées vocalement. Ainsi selon sa position par rapport à l'écran ou le nombre de personnes dans le poste de commande (niveau de bruit), l'utilisateur choisira une modalité différente.

Figure 13 : Poste de commande militaire, illustration issue de [Myers 2002].

2.3.3 Redondance

L'usage redondant de modalités en sortie est particulièrement utile pour renforcer par exemple l'immersion d'un joueur au sein de l'univers d'un jeu vidéo. Ainsi la vitesse de rotation du moteur, au sein du simulateur de conduite décrit dans [Sezgin 2009] (Figure 14), est présentée à l'utilisateur sous deux formes redondantes. La première forme est visuelle. Il s'agit d'afficher sur un écran un cadrant indiquant le nombre de tours par minute, sous une forme analogue à celle d'un compte tours dans une voiture. La deuxième forme est auditive. Il s'agit de produire un son de moteur à une fréquence plus ou moins importante qui indique le nombre de tours.



Légende

1. Caméra (reconnaissance faciale)
2. Microphone (reconnaissance de la parole)
3. Paire de haut-parleurs
4. Autre paire de haut-parleurs
5. Paire de lampes puissantes
6. Zone de projection de formes
7. Casque (microphone + écouteur)

Figure 14 : Simulateur de conduite, illustration issue de [Sezgin 2009].

En entrée d'un système interactif, la redondance est plus rare, puisqu'il s'agit d'une contrainte imposée à l'utilisateur : il doit utiliser deux modalités pour réaliser une tâche. Les auteurs de [Serrano 2009] proposent un exemple d'utilisation de deux modalités redondantes en entrée pour faire défiler

un diaporama (Figure 15). La première modalité utilisée est constituée d'une manette de jeu vidéo (WiiMote de Nintendo) dont le langage d'interaction repose sur la rotation de la WiiMote. La deuxième modalité est constituée d'une « wiisoft » imaginée par les auteurs qui est en fait un capteur de torsion encapsulé au sein d'une protection de WiiMote, et dont le langage repose sur des seuils de torsion. Pour faire passer le diaporama à la diapositive suivante ou précédente, une interaction redondante est imposée à l'utilisateur. Il s'agit respectivement de faire tourner la WiiMote sur la droite ou la gauche et de tordre la « wiisoft » vers l'avant ou l'arrière.

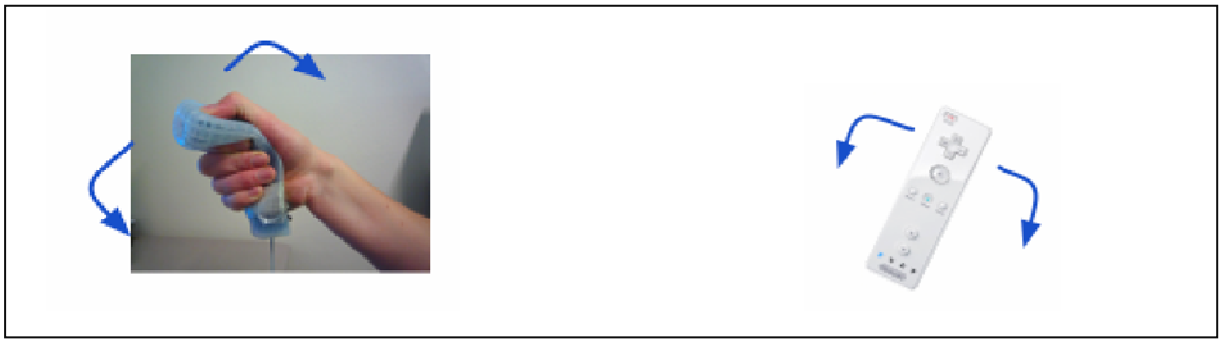


Figure 15 : « Wiisoft » et Wiimote utilisés de manière redondante pour parcourir un diaporama [Serrano 2009].

L'utilisation redondante de la « wiisoft » et de la Wiimote permet d'éviter des changements dans les transparents qui ne soient pas voulus par le présentateur.

Une autre application possible de la redondance en entrée est proposée dans [Kaiser 2006] pour l'apprentissage par le système de nouveaux mots ou abréviations. Le système propose deux modalités : une modalité de reconnaissance de la parole et une autre de reconnaissance de l'écriture manuscrite sur un tableau. L'application proposée permet d'associer le nom d'une personne à une courbe dans un diagramme. Pour cela, l'utilisateur écrit le nom de la personne (Bob) sur le tableau à l'aide d'un feutre, et le relie avec un trait à une courbe. En même temps, il dit « ceci est la courbe de Bob ». Les deux interactions, manuscrite et orale, signifient toutes les deux « ceci est la courbe de bob » et sont utilisées conjointement pour réaliser la tâche : associer le nom Bob à une courbe. Il s'agit bien d'un usage redondant de deux modalités.

2.3.4 Synthèse

Nous avons présenté plusieurs systèmes multimodaux selon les trois principales formes de combinaison de modalités : Complémentarité, Redondance et Equivalence. A travers ces exemples d'applications multimodales, nous avons illustré la diversité des modalités existantes et des combinaisons de modalités possibles.

2.4 Synthèse sur les systèmes multimodaux

Dans cette deuxième partie, nous avons introduit le domaine de la Multimodalité et sa terminologie. Nous avons figé la définition adoptée des termes de modalité et de système multimodal. Nous avons également présenté deux espaces conceptuels pour caractériser des modalités d'interaction ainsi que deux ensembles de propriétés pour caractériser les combinaisons de modalités (les propriétés CARE et l'espace TYCOON). Enfin, nous avons souligné la diversité des modalités d'interaction et des combinaisons de modalités d'interaction en présentant plusieurs systèmes multimodaux existants.

Ayant successivement considéré la conception des systèmes multiutilisateurs (section 1) et des systèmes multimodaux (section 2), nous abordons dans la section suivante la conception des systèmes multiutilisateurs et multimodaux.

3 Systèmes multiutilisateurs et multimodaux

Nous présentons des exemples de systèmes multiutilisateurs et multimodaux, c'est-à-dire, des systèmes qui proposent à plusieurs utilisateurs d'interagir avec des modalités d'interaction variées. Ces exemples nous permettent d'illustrer le vaste espace des possibilités de conception, puisque les questions de conception des systèmes multiutilisateurs et des systèmes multimodaux se conjuguent tout en introduisant de nouveaux défis. L'espace des possibilités illustré, nous présentons et motivons ensuite (section 4) notre sujet de recherche à la croisée de ces deux axes, la conception de systèmes multiutilisateurs et multimodaux.

3.1 Exemples de systèmes multiutilisateurs et multimodaux

Les systèmes multiutilisateurs et multimodaux présentés dans cette section illustrent à la fois les aspects de collaboration et de coopération. Nous commençons par décrire des systèmes collaboratifs pour finir par la présentation de deux systèmes coopératifs. De plus de nombreux travaux récents sont consacrés à l'interaction sur des tables interactives multiutilisateurs (*tabletop*) : aussi plusieurs systèmes multiutilisateurs et multimodaux présentés mettent en jeu des modalités d'interaction sur une ou plusieurs tables interactives.

3.1.1 Environnement collaboratif d'édition musicale

Le prototype d'environnement d'édition musicale présenté dans [Fiebrink 2009] propose une interface collaborative autour d'une table augmentée (Figure 16). Avec ce système, les utilisateurs collaborent à la réalisation d'une œuvre musicale par composition d'échantillons de son (*sample*). Le système propose une interface projetée sur une table qui comprend : un panneau d'édition de la musique (pour positionner les échantillons), ainsi que des interacteurs projetés autour d'un boîtier de commandes. La table autorise plusieurs utilisateurs à interagir simultanément pour contribuer à la réalisation de l'œuvre musicale. Le système est multimodal en entrée dans le sens où il propose d'utiliser le boîtier de commandes augmenté, ainsi que le système de reconnaissance de gestes de la table (réalisé par un système de capteurs infrarouges disposés sous la table). L'interaction multiutilisateur est co-localisée autour d'une table unique contrairement au système intégrant deux tables distantes de [Tuddenham 2009] et présenté dans la suite à la section 3.1.4.

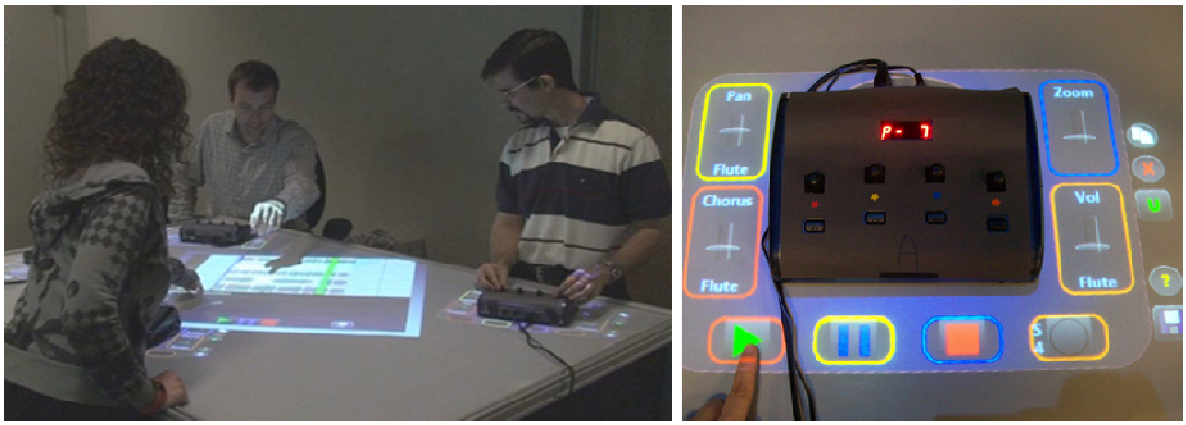


Figure 16 : Environnement collaboratif d'édition musicale, illustration issue de [Fiebrink 2009].

3.1.2 Warcraft 3 collaboratif et multimodal

Warcraft 3 (Blizzard) est un jeu de stratégie en temps réel permettant à plusieurs joueurs de s'affronter ou d'affronter l'ordinateur. Dans Warcraft 3, chaque joueur doit développer un peuple en récoltant des ressources, en construisant des bâtiments et en constituant une armée. Cette dernière peut ensuite servir à attaquer l'ennemi.

Dans l'extension collaborative de Warcraft 3 [Tse 2007, Tse 2008], deux joueurs développent ensemble un même peuple. Cette version se joue autour d'une table augmentée et permet l'usage du geste et de la parole pour interagir.

Les modalités gestuelles et vocales peuvent être utilisées de manière complémentaire selon une forme d'interaction issue du "met ça là" [Bolt 1980] pour déplacer une unité. Ainsi, comme l'illustre la Figure 17, l'utilisateur de gauche désigne un point sur la carte et prononce la phrase « Va ici » (*Move here*), ce qui a pour effet de faire se déplacer l'unité sélectionnée à l'endroit indiqué.

Figure 17 : Utilisation de Warcraft 3 collaboratif et multimodal, illustration issue de [Tse 2007].

Le système propose de plus une utilisation originale de la reconnaissance de geste. Par exemple, un joueur peut délimiter une zone avec les mains pour sélectionner un ensemble d'unités se trouvant dans une zone, puis prononcer la phrase « Nom Unité 1 » (*Label as Unit 1*). Cette commande

multimodale a pour effet de constituer un groupe avec les unités présentes dans la zone délimitée par les mains. Ce groupe est nommé *Unité 1*, et pourra être appelé par ce nom dans la suite du jeu.

3.1.3 Les Sim's collaboratif et multimodal

Les auteurs de [Tse 2007, Tse 2008] proposent également d'étendre le jeu intitulé les Sim's (EA Games) pour être utilisé autour d'une table augmentée. Les Sim's est un jeu de développement et de gestion d'un avatar. Le joueur peut construire une maison pour son avatar, le faire travailler et gérer ses relations sociales avec des personnages animés par l'ordinateur.

Dans la version sur la table augmentée, les deux joueurs collaborent pour gérer un même avatar. Les auteurs proposent plusieurs interactions multimodales semblables à celles pour le jeu Warcraft 3 étendu (section précédent).

En revanche, contrairement à Warcraft 3 étendu, les auteurs proposent une technique d'interaction à la fois multiutilisateur et multimodale. Il s'agit d'ajouter un élément tel qu'un meuble ou un arbre dans la propriété de leur avatar. La technique, illustrée au sein de la Figure 18, consiste pour un utilisateur à désigner un emplacement, puis à prononcer la phrase « créer un arbre » (*Create tree*). Le deuxième utilisateur doit alors prononcer la phrase « ok » (*Okay*) pour confirmer la création de l'arbre. Cette technique d'interaction est particulièrement intéressante car elle exploite deux modalités distinctes (l'une composée <parole+geste>, et l'autre atomique <parole>) utilisées de manière complémentaire par deux utilisateurs différents.

Figure 18 : Les Sim's collaboratif et multimodal, illustration issue de [Tse 2007].

3.1.4 Tables augmentées distantes

Le système présenté dans [Tuddenham 2009] permet à deux utilisateurs distants de travailler de manière collaborative à l'élaboration de plans. Chaque utilisateur travaille sur une table augmentée à l'aide d'un stylet comme l'illustre la Figure 19. Les manipulations d'un utilisateur sont directement répercutées sur la vue de l'autre utilisateur : ceci est vrai pour des tâches d'ajout ou de modification des éléments du plan, mais aussi pour les tâches de navigation. En effet, le déplacement du plan par un utilisateur sur son bureau entraîne un déplacement identique sur le bureau de l'autre utilisateur. Il s'agit donc d'un cas de couplage fort de l'interaction reposant sur une interface WYSIWIS.

L'interface du système est particulièrement intéressante car elle tire parti de la multimodalité pour renforcer la conscience de groupe. Les commandes d'édition du plan ne sont pas multimodales, puisque les utilisateurs n'utilisent que le stylet pour interagir. En revanche, le système utilise une modalité passive pour renforcer la conscience de groupe. Cette modalité est composée d'une caméra et d'un système de reconnaissance de la main et du bras des utilisateurs. Le flux capté est ensuite projeté sur le bureau de l'autre utilisateur sous la forme d'une ombre (rose). Grâce à ce système d'ombre, le deuxième utilisateur peut voir à chaque instant ce que fait le premier utilisateur, et même anticiper sa prochaine action. Le système est ainsi multimodal en entrée puisqu'il propose deux modalités d'interaction en entrée. Les deux modalités (basées sur le dispositif caméra et le dispositif stylet) sont utilisées de manière concurrente au sens de l'espace TYCOON, et l'utilisation de l'une (modalité passive basée sur la caméra) n'entraîne pas de gêne à l'utilisation de l'autre (modalité active basée sur le stylet).

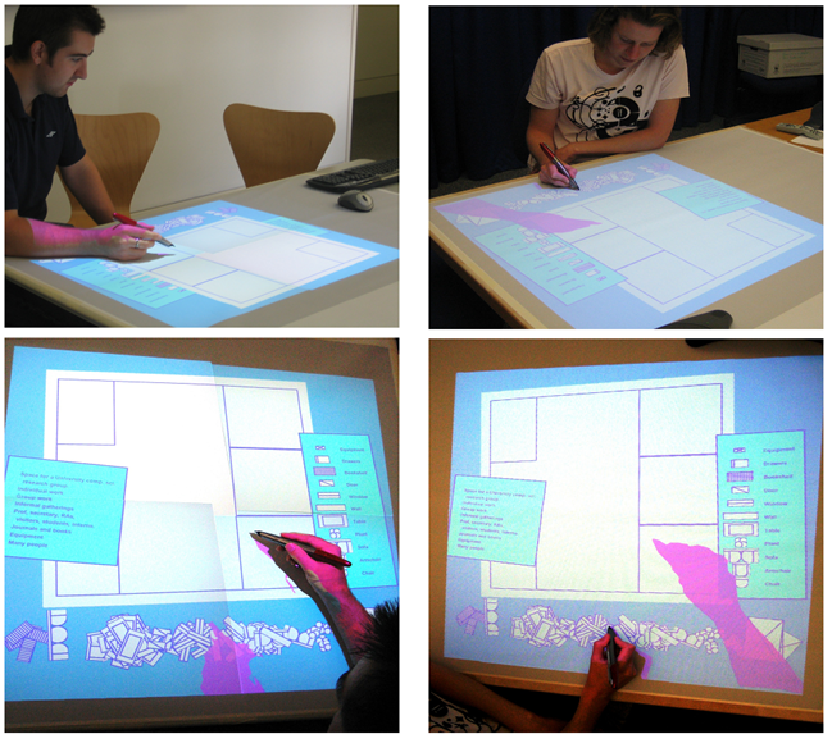


Figure 19 : Interface collaborative pour l'édition de plans, illustration issue de [Tuddenham 2009].

3.1.5 Poste de commandement pour la gestion de situations d'urgence

Contrairement aux systèmes collaboratifs précédents qui impliquent une ou plusieurs tables pour interargir, le poste de commandement collaboratif pour la gestion de situations d'urgence, décrit dans [Rauschert 2002], repose sur un écran vertical et un système de cartographie en trois dimensions. Le système utilise un écran unique partagé par deux utilisateurs comme l'illustre la Figure 20. Les gestes de ces derniers sont captés par deux caméras, et les commandes vocales parole sont captées par des microphones.

Figure 20 : Poste de commandement collaboratif pour la gestion de situations d'urgence, illustration issue de [Rauschert 2002].

Le système propose trois types de fonctionnalités : (1) la navigation dans la carte telle que le déplacement, le zoom, ou le recentrage. (2) la gestion de l'affichage d'informations, telle que l'affichage ou non de calque. (3) des commandes de dessin, telles que tracer un cercle, ou tracer à main levée.

Les interactions comme le zoom en un point et le recentrage sur un point sont réalisées par la combinaison d'une modalité gestuelle avec une modalité vocale. D'autres commandes multimodales sont disponibles au travers de requêtes sur les données de la carte : par exemple, la commande vocale « affiche les services d'urgence dans cette zone », est complétée par un geste de désignation de la zone.

Le système est collaboratif puisque les utilisateurs ont accès aux mêmes fonctionnalités. En effet, ils concourent à la mise en œuvre de plan d'urgence en ayant les mêmes rôles de gestionnaire de plan de secours.

3.1.6 Système de visite réelle et virtuelle

Le système présenté dans [Brown 2005] propose à plusieurs utilisateurs de visiter des lieux, de partager des photos, des commentaires et des notes sur ces lieux, et à des utilisateurs restant à leur domicile d'effectuer une visite virtuelle des lieux aux travers des photos et commentaires laissés par les premiers utilisateurs.

Ce système est coopératif puisque les visiteurs en situation réelle et ceux à distance disposent chacun d'un ensemble de tâches spécifiques (division du travail a priori). De plus le système peut également être caractérisé de collaboratif car plusieurs utilisateurs (visiteurs en situation réelle) concourent à la production des informations qui permettent la visite virtuelle. L'interface du système

est partagée sous la forme d'un WYSIWIS relâché, puisque chaque utilisateur partage les informations de cartographie, mais dispose de sa vue propre sur la cartographie.

L'interface utilisateur illustrée à la Figure 21 est multimodale. Pour la communication entre utilisateurs (communication homme-homme médiatisée), chacun d'entre eux dispose d'un microphone et d'une caméra. Ces deux modalités sont utilisées de manières complémentaires lors de la communication.

Ensuite, le système indique la position des utilisateurs en situation réelle grâce à une modalité passive de géo-localisation (reposant sur le système GPS). L'utilisateur peut également désigner sa position sur la carte directement avec le doigt. Aussi, les deux modalités, l'une passive (GPS) et l'autre active (désignation sur l'écran) sont donc équivalentes pour la tâche de spécification de la position courante.

Figure 21 : Système de visite réelle et virtuelle, illustration issue de [Brown 2005].

3.1.7 Poste de commande de drones

Dans le cadre d'un Projet d'Etude Amont (PEA) de la Direction Générale de l'Armement (DGA) noté FH/PA (Facteurs Humains et Partage d'Autorités), nous avons participé à la réalisation de prototypes de poste de commande de drones. Nous introduisons brièvement ce système, qui est détaillé au Chapitre 5.

Ce système coopératif permet à trois opérateurs de participer à la conduite d'une mission, par exemple une mission d'observation. Les tâches aux cours d'une mission sont réparties entre les utilisateurs selon des rôles métier prédéfinis. Le premier rôle contrôle le drone (opérateur vecteur).

Son objectif est d'amener le drone à survoler l'ensemble des zones qui doivent faire l'objet d'une observation. Le deuxième rôle (opérateur charge utile) consiste à réaliser des observations à l'aide d'outils d'observation présents sur le drone tels qu'une caméra orientable. Enfin, le troisième rôle (opérateur chargé de mission) consiste à faire le lien entre les deux premiers rôles et le monde extérieur, tel que la communication avec la chaîne hiérarchique militaire en transmettant les informations observées et en recevant des ordres pour une modification de la mission en cours.

Sur la base de ces trois rôles, chaque opérateur dispose d'une interface commune et d'une interface spécifique. L'interface commune consiste en un système de cartographie et un système de partage des clichés pris au cours de la mission par l'opérateur charge utile. L'interface commune aux trois opérateurs est déployée sur des ordinateurs (avec une souris, un clavier et un écran) intégrés dans une console de commande. En ce qui concerne les interfaces spécifiques à chaque rôle, l'opérateur vecteur dispose d'une console munie d'un écran de contrôle spécifique indiquant l'orientation, l'altitude, la vitesse et d'autres informations permettant la contrôle du drone. Il dispose également d'un boîtier de commandes comportant plusieurs boutons poussoir et potentiomètres pour modifier les paramètres de vols. Son interface est donc multimodale en entrée. L'utilisateur devant effectuer les observations dispose également d'un boîtier de commandes spécifique à sa tâche d'observation. Celui-ci est muni d'un joystick pour orienter la caméra fixée sur le drone ainsi que de nombreux boutons poussoir pour activer ou désactiver les différents équipements d'observation. Il dispose également d'un écran spécifique affichant la vidéo captée par la caméra embarquée, augmentée par des informations de vols. Son interface est donc également multimodale en entrée. Nous ne détaillons pas l'interface du troisième opérateur qui commande la mission.

Comme chaque opérateur (opérateur vecteur et opérateur charge utile) dispose d'une console spécifique munie d'un écran, leurs interfaces sont multimodales en sortie si nous considérons en plus l'écran de la console partagée. L'opérateur partage son attention visuelle entre l'écran partagé et celui dédié à son rôle métier.

Ce type de système multiutilisateur coopératif et multimodal est actuellement utilisé, notamment par l'Armée Française. Les multiples modalités d'interaction permettent de mener à bien les opérations nécessaires à la réalisation d'une mission, et la présence de plusieurs opérateurs est rendue nécessaire par la diversité des tâches à accomplir, par la complexité de la mission et de l'équipement à contrôler.

3.1.8 Synthèse

La plupart des systèmes multiutilisateurs présentés sont collaboratifs puisque plusieurs utilisateurs jouent le même rôle et ont accès aux mêmes fonctionnalités. Parmi ces systèmes collaboratifs, nous avons présenté plusieurs systèmes reposant sur une table interactive unique et une interaction co-localisée autour de cette table, ou reposant sur plusieurs tables interactives géographiquement distantes. C'est d'ailleurs dans le cas d'une interaction sur des tables interactives distantes que l'intérêt de la multimodalité a été souligné dans le cadre du renforcement de la conscience de groupe [Tuddenham 2009]. Nous avons également présenté deux systèmes coopératifs. L'un deux, le poste de commande de drones est un système coopératif et multimodal utilisé actuellement par l'Armée de Terre.

Parmi les systèmes introduits, les modalités utilisées reposent essentiellement sur le geste et la parole. Il convient de noter que plusieurs systèmes multiutilisateurs intègrent des modalités innovantes, au-delà des interfaces de type WIMP, mais ne sont pas multimodaux. Par exemple, le jeu "les 4 magiciens" décrit à la section 1.6.2, qui permet de faire interagir plusieurs utilisateurs disposant chacun d'une WiiMote autour d'un écran unique.

Le Tableau 3 présente une synthèse des caractéristiques multiutilisateurs et multimodales des systèmes présentés.

Nom du système	Aspects Multiutilisateurs	Aspects Multimodalité (entrée)
Edition musicale	Collaboratif Co-localisé	Modalité tactile Modalité tangible Forme = Assignation
Warcraft 3 étendu	Collaboratif Co-localisé	Modalité parole Modalité tactile Modalité gestuelle Formes = Assignation, Complémentarité
Sim's étendu	Collaboratif Co-localisé	Modalité parole Modalité tactile Formes = Assignation, Complémentarité
Tables augmentées	Collaboratif Distant	Modalité stylet Modalité geste Forme : Concurrence
Poste de commandement	Collaboratif Co-localisé	Modalité parole Modalité gestuelle Formes = Assignation, Complémentarité
Visite réelle et virtuelle	Collaboratif et Coopératif Distant	Modalité parole Modalité visuelle (caméra) Modalité tactile Modalité WIMP Formes : Assignation, Equivalence
Poste de commande de drones	Coopératif Co-localisé	Modalité tactile Modalité tangible Forme : Assignation

Tableau 3 : Synthèse des systèmes multiutilisateurs et multimodaux présentés

3.2 Exemple illustratif : WCCM - Warcraft Collaboratif, Coopératif et Multimodal

Pour illustrer, tout au long du manuscrit, nos travaux qui portent sur les notations pour la spécification de systèmes interactifs multiutilisateurs et multimodaux, nous considérons un exemple de système interactif qui est à la fois collaboratif, coopératif et multimodal. Pour cela nous considérons une extension du système Warcraft 3 collaboratif et multimodal [Tse 2007, Tse 2008] autour d'une table et présenté à la section 3.1.2. Nous étendons ce système pour le rendre à la fois collaboratif, coopératif et multimodal (système noté WCCM pour Warcraft Collaboratif, Coopératif et Multimodal), afin d'obtenir une couverture la plus large des caractéristiques multiutilisateurs et multimodales.

Pour présenter ce système WCCM et les extensions proposées, nous présentons les modifications apportées aux concepts du jeu avant de présenter trois tâches qui illustrent trois types d'activité : la gestion individuelle des paysans, la création collaborative d'unités combattantes et la gestion coopérative des troupes. Chacunes de ces tâches met en œuvre une interaction multimodale.

3.2.1 Concepts de Warcraft étendu

Dans notre version, comparée à Warcraft 3 collaborative et multimodale présenté dans [Tse 2007 ; Tse 2008], nous intégrons la dimension coopérative : pour cela nous définissons deux rôles *sergent* et *chef*. Certaines tâches dépendent alors de ces rôles. Au début d'une partie, les joueurs décident de l'attribution des rôles. Les deux joueurs sont équipés d'un casque muni d'un microphone et interagissent en étant co-localisés autour d'une table interactive.

De plus nous avons simplifié le jeu afin de nous concentrer uniquement sur les aspects pertinents de nos travaux, les caractéristiques multiutilisateurs et multimodales du système illustratif. Ainsi, une partie de Warcraft se joue avec une carte rectangulaire comportant uniquement des *forêts* et des *mines d'or*, contrairement au jeu original qui prévoit de nombreux autres éléments (tels que des rivières ou des bâtiments spéciaux). Les joueurs commencent la partie sur un point de la carte avec quatre unités appelées *paysan* et un bâtiment nommé *hôtel de ville*. Un *paysan* peut être déplacé sur la carte, peut récolter de l'or, couper du bois, et construire un bâtiment qui peut être soit un *baraquement, soit* une *ferme*.

Un *hôtel de ville* permet de créer de nouveaux *paysans*, tandis qu'un *baraquement* permet de créer des unités combattantes : des *fantassins* et des *cavaliers*. Enfin, une *ferme* permet d'augmenter le nombre d'unités (*paysans*, *fantassins* et *cavaliers*) gérées par les joueurs.

3.2.2 Gestion des paysans : tâche individuelle et multimodale

Les *paysans* peuvent être manipulés par n'importe quel joueur. Néanmoins, lorsqu'un joueur sélectionne un paysan, il est le seul à pouvoir lui donner un ordre.

Pour donner un ordre à un paysan :

1. Le joueur doit d'abord sélectionner un paysan en le désignant avec le doigt sur la table. Le paysan est alors sélectionné (retour d'information visuelle). Dans son casque, le joueur entend la phrase « oui, maître » (retour d'information sonore).
2. Le même joueur peut ensuite :
    a. Déplacer le paysan en énonçant « va ici » pendant qu'il désigne un lieu du doigt (complémentarité des modalités vocale et gestuelle).
    b. Lui faire récolter de l'or, en énonçant la commande « récolte de l'or » pendant qu'il désigne une mine d'or du doigt (complémentarité des modalités vocale et gestuelle).
    c. Lui faire couper du bois, en énonçant la commande « coupe du bois » pendant qu'il désigne une forêt du doigt (complémentarité des modalités vocale et gestuelle).
    d. Lui faire construire un bâtiment en énonçant la commande « construis une ferme ici » ou « construis un baraquement ici » pendant qu'il désigne un emplacement du doigt (complémentarité des modalités vocale et gestuelle).

La gestion des *paysans* est collaborative puisque chaque joueur peut y participer en gérant un paysan. En revanche, la gestion d'un paysan donné est une tâche individuelle.

3.2.3 Création d'unités combattantes : tâche collaborative et multimodale

La création d'unités combattantes s'effectue avec les baraquements. L'interaction que nous considérons est inspirée de la création d'arbres dans la version du jeu « Les Sim's » proposée dans [Tse 2007] (interaction présentée à la section 3.1.3 de ce chapitre). La tâche requiert l'intervention des deux utilisateurs indépendamment de leur rôle.

La création d'une nouvelle unité combattante suit trois étapes :

1. Pour commencer, un des deux joueurs sélectionne un baraquement en le désignant du doigt. Le bâtiment apparaît alors comme sélectionné, et un son spécifique est joué dans les casques des deux joueurs (retour d'information visuelle et sonore).
2. Le même joueur peut alors choisir un type d'unité à créer. Pour cela, il énonce la commande « crée un fantassin » pour créer un fantassin, ou « crée un cavalier » pour créer un cavalier.
3. Enfin, le deuxième joueur doit confirmer la création de l'unité en énonçant la commande « ok ».

N'importe quel joueur peut être à l'origine de la création d'une unité. En revanche, seul le joueur qui sélectionne le «baraquement » peut décider de l'unité à construire, et il est impératif qu'un autre joueur confirme la création de l'unité. Il s'agit donc d'une interaction collaborative puisque les utilisateurs ne sont pas identifiés en fonction de leur rôle. De plus pour réaliser la tâche, l'interaction est multimodale (parole et geste) et fait intervenir plusieurs utilisateurs.

3.2.4 Gestion des troupes : tâche coopérative et multimodale

La troisième tâche est la gestion des troupes (*cavaliers* et *fantassins)*. Le jeu Warcraft permet de gérer chaque soldat individuellement ou de gérer plusieurs soldats regroupés au sein d'un *groupe*. Nous adoptons une vision coopérative de cette gestion des troupes (soldats isolés et groupes) en répartissant les responsabilités entre le sergent et le chef :

- Le *sergent* a la responsabilité de gérer les soldats isolés (qui ne font pas partie d'un *groupe*) et a la possibilité de constituer de nouveaux *groupes*.
- Le *chef* a la responsabilité de gérer les *groupes* de soldats.

Un joueur jouant le rôle de *sergent* peut sélectionner un soldat en le désignant du doigt. De la même manière, un joueur jouant le rôle de *chef* peut sélectionner un *groupe* en sélectionnant l'un des soldats qui le constitue. Le chef peut également sélectionner un *groupe* en énonçant le nom d'un *groupe*. Après sélection, un soldat isolé ou un groupe peut être déplacé de la même manière qu'un paysan, c'est-à-dire en énonçant la commande « aller ici » tout en désignant un lieu du doigt sur la table.

Le joueur jouant le rôle de *sergent* a la possibilité de constituer de nouveaux *groupes*. Pour cela, nous reprenons l'interaction proposée dans [Tse 2007], qui consiste à délimiter une zone avec les mains (comme le montre la Figure 17) puis à énoncer un nom de *groupe*. Tous les soldats présents dans la zone délimitée par les deux mains sont alors rassemblés au sein d'un *groupe* portant le nom indiqué. Les *groupes* ainsi créés peuvent alors être gérés par le joueur jouant le rôle de *chef*.

3.2.5 Synthèse

Le système WCCM présenté propose des interactions collaboratives, coopératives et multimodales. En cela il constitue un bon exemple illustratif de nos travaux dédiés aux caractéristiques multiutilisateurs et multimodales de l'interaction. Le système WCCM est utilisé lors de l'analyse des notations de spécification existantes (Chapitre 2) afin de permettre une comparaison fine des pouvoirs d'expression des notations mais aussi pour illustrer nos contributions (Chapitre 3 et Chapitre 4), ainsi que leurs évaluations (Chapitre 5).

Néanmoins il convient de noter que WCCM implique une co-localisation des utilisateurs autour d'une table interactive. Le système WCCM ne permet donc pas de traiter de la répartition dans l'espace des utilisateurs. Au regard des systèmes présentés à la section 3.1, une extension pour deux utilisateurs distants géographiquement peut être envisagée [Tuddenham 2009], en utilisant deux tables et une projection de l'ombre de la main des utilisateurs comme illustré à la Figure 19.

De plus, le système WCCM est de type synchrone puisque les utilisateurs sont présents autour de la table pendant la durée de la partie. L'aspect asynchrone et synchrone-asynchrone de l'interaction n'est donc pas couvert par cet exemple.

4 Sujet et motivations de l'étude

Ce chapitre présente les concepts clefs des domaines du TCAO et de la Multimodalité et met en évidence la richesse des possibilités pour la conception de systèmes multiutilisateurs et multimodaux.

Pour les systèmes multiutilisateurs, la conception de l'interface partagée entre les utilisateurs doit reposer sur des critères ergonomiques propres aux systèmes multiutilisateurs tels que le couplage de l'interaction et la conscience de groupe. Nous avons étudié en détail et de façon expérimentale les différents niveaux de couplage de l'interaction. De plus l'interface partagée doit aussi fournir des moyens d'assurer la conscience de groupe en prenant en compte la proximité ou l'éloignement des utilisateurs.

Pour les systèmes multimodaux, la conception consiste à associer des modalités (dispositifs physiques et langages d'interaction) et des combinaisons de modalités à chaque tâche du système, tout en visant la cohérence globale de l'interaction concrète. En considérant les propriétés CARE (section 2.2.2.1 de ce chapitre) pour l'interaction multimodale en entrée, un ensemble de modalités équivalentes peut être associé à une tâche pour rendre le système multimodal plus flexible en offrant des choix à l'utilisateur (contrairement à l'assignation d'une modalité simple ou composée à une tâche). La redondance de plusieurs modalités pour une tâche donnée permet d'augmenter la robustesse du système, tandis que la complémentarité augmente l'efficacité et le caractère naturel de l'interaction en tirant partie des caractéristiques de chaque modalité mise en œuvre.

Pour concevoir des systèmes multiutilisateurs et multimodaux, les choix de conception liés aux systèmes multiutilisateurs et aux systèmes multimodaux se conjuguent. L'espace des possibilités de conception devient donc très vaste. Par exemple, comme présenté à la section 3.1.4, l'usage concurrent de deux modalités peut être exploité pour renforcer la conscience de groupe. La

conjugaison des caractéristiques multiutilisateurs et multimodales d'un système implique aussi des nouveaux problèmes de conception, tels que la disponibilité des modalités d'interaction et leur répartition entre les utilisateurs, ou encore l'usage concurrent de modalités ou de combinaisons de modalités pour la réalisation de tâches communes ou individuelles dans le cadre de travail de groupe.

Face à ces constats, notre travail se focalise sur cette articulation entre interaction multiutilisateur et interaction multimodale. Ce travail est novateur car, dans les deux domaines respectifs que sont la TCAO et la Multimodalité, le sujet est peu abordé et donne souvent lieu à des réalisations ad-hoc sans résultats de conception généraux ni outils pour accompagner la conception. Comme première étape de travail à l'intersection de deux domaines, notre travail concerne les phases amont du cycle de vie logiciel et se focalise uniquement sur la spécification des solutions de conception d'interaction multiutilisateur et multimodale. L'étape suivante serait alors d'étudier quels sont les besoins pour le développement logiciel de tes systèmes multiutilisateurs et multimodaux.

Pour nos travaux dédiés à la spécification de solutions de conception d'interaction multiutilisateur et multimodale nous avons adopté la démarche de travail suivante.

La première étape consiste à étudier les notations existantes pour décrire la conception de systèmes multiutilisateurs et de systèmes multimodaux. Pour cette étude nous considérons aussi les systèmes de réalité augmentée car ils intègrent souvent plusieurs modalités d'interaction et sont mono et multi-utilisateurs comme le soulignent les travaux de thèse de P. Renevier [Renevier 2004]. Pour permettre une comparaison fine des pouvoirs d'expression des notations existantes, nous avons appliqué les notations à la description d'un même système, notre système WCCM (Warcraft Collaboratif, Coopératif et Multimodal). Cette étude détaillée des notations existantes nous permet de mettre en évidence les éléments saillants de chaque notation, d'identifier les similitudes et différences des concepts véhiculés par chaque notation mais aussi d'identifier les limites. Cette étude de l'existant fait l'objet du Chapitre 2.

En nous reposant sur les conclusions de l'étude de l'existant, nous proposons une nouvelle notation pour la spécification de systèmes multiutilisateurs et multimodaux : la notation COMM. COMM consiste en une extension de notations existantes présentées au Chapitre 2, en introduisant des nouveaux concepts tels que les rôles interactifs et les tâches modales. Pour présenter la notation COMM au Chapitre 3, nous justifions les extensions apportées aux notations existantes en explicitant les limites identifiées, nous définissons ces extensions et nous les illustrons toujours avec le système WCCM. Utiliser le même exemple de système que pour l 'étude des notations existantes permet de souligner l'apport de la notation COMM en terme de concision et pouvoir d'expression par rapport aux notations existantes.

La troisième étape de nos travaux consiste à outiller la notation proposée au Chapitre 3. Pour cela nous avons développé un éditeur de spécifications COMM sous la forme d'une application internet riche. La réalisation d'un éditeur constitue une première phase d'outillage de notre notation COMM. D'autres outils peuvent faciliter l'usage de la notation, tels que qu'un outil d'analyse des spécifications produites ou encore des outils de génération de code. Cependant ces autres outils requièrent d'abord une version informatique des spécifications que nous visons à produire avec notre éditeur de spécifications COMM. Nous présentons l'éditeur e-COMM développé, ses fonctionnalités et son interface graphique au Chapitre 4.

Enfin, une étape incontournable est d'évaluer les contributions, que ce soit de façon conceptuelle et/ou expérimentale. L'évaluation concerne donc à la fois la notation COMM et l'éditeur e-COMM. Nous avons consigné dans le Chapitre 5 l'ensemble des évaluations menées tout au long de ce travail. Valider une notation de conception n'est pas une tâche facile, surtout dans le temps imparti d'une thèse. Nous avons abordé ce problème de validation selon trois facettes complémentaires : (1) montrer que la notation permet de décrire efficacement des systèmes multiutilisateurs et multimodaux au regard des notations existantes (2) montrer que la notation peut être utilisée par d'autres concepteurs que les membres de l'équipe IIHM du LIG (3) montrer que la notation passe à l'échelle et peut effectivement être utilisée dans un projet conséquent, au-delà d'une étude de cas comme WCCM.

Chapitre 2 : Etat de l'art des notations

Ce chapitre présente un état de l'art des notations existantes pour la spécification de systèmes interactifs multiutilisateurs et multimodaux. Nous mettons en exergue les caractéristiques des notations dans le but de leur analyse comparative au regard de notre objectif : spécifier l'interaction multiutilisateur et multimodale.

Ce chapitre s'articule en trois parties :

- La première partie présente une cartographie des caractéristiques des notations, ainsi qu'une grille d'analyse comparative des notations reposant sur ces caractéristiques.
- La deuxième partie présente des notations identifiées dans la littérature. Dans cette partie, nous visons à être le plus exhaustif possible en termes type de notations existantes. Nous étendons ici notre comparaison de 5 notations [Jourde 2008a] [Jourde 2008b]. Les caractéristiques de chaque notation sont précisées et leur usage est illustré avec l'application WCCM (Warcraft Collaboratif, Coopératif et Multimodal) présentée au Chapitre 1. L'analyse des notations existantes est guidée par la grille définie dans la première partie.
- La troisième partie présente une synthèse de ces notations sous la forme de tableaux reprenant les différentes caractéristiques de notre grille d'analyse.

Remarque : Dans ce chapitre, et dans un souci de commodité linguistique, les termes « modélisation », « spécification » et « description » sont utilisés sans distinction afin de signifier l'activité qui consiste à « représenter certains aspects d'un système dans une forme donnée ».

1 Caractéristiques des notations

Pour guider notre analyse des notations de spécification existantes, nous identifions des caractéristiques. Nous avons organisé ces caractéristiques selon quatre catégories :

- une catégorie de caractéristiques générales,
- une catégorie de caractéristiques spécifiques à l'interaction homme-machine,
- une catégorie couvrant les aspects liés à la dimension multiutilisateur, et
- une catégorie couvrant les aspects liés à la dimension multimodale.

Les quatre premières sections de cette partie sont consacrées aux catégories de caractéristiques. La cinquième section présente une synthèse de ces caractéristiques, sous la forme d'une grille d'analyse.

1.1 Caractéristiques générales

Cette section présente plusieurs caractéristiques générales, applicables à toutes les notations. Nous qualifions ces caractéristiques de générales car elles visent à identifier le périmètre d'une notation sans décrire ses capacités à spécifier l'interaction homme-machine.

1.1.1 Domaine d'origine

Une première caractéristique d'une notation est son domaine d'origine. En effet, une notation permet de décrire un ou plusieurs aspects d'un système selon des concepts et des relations qui sont propres à son domaine. Aussi, identifier le domaine d'origine d'une notation, c'est identifier les concepts sur lesquels elle s'appuie.

Les notations permettant de modéliser l'interaction homme-machine proviennent de plusieurs domaines : psychologie cognitive, ethnographie, ingénierie logicielle, TCAO. Ainsi par exemple, la notation K-MAD (*Kernel of Model for Activity Description*) [Lucquiaud 2005, Baron 2006] est issue du domaine de la psychologie cognitive et vise à décrire les tâches et les objets ayant une influence sur l'activité de l'utilisateur. La notation CTT (*Concurs Task Tree*) [Paterno 1997] provient quant à elle du domaine de l'interaction homme-machine, et propose de décrire l'interaction de manière plus proche du système que K-MAD. Enfin, une notation comme GTA (*Groupware Task Analysis*) [Veer 2000] provient du domaine de la TCAO, et est enrichie des apports de domaines comme l'ethnographie dans sa méthodologie et la psychologie cognitive pour la forme de ses modèles et certains de ces concepts.

Nous utilisons cette caractéristique pour classer les différentes notations selon leur domaine d'origine.

1.1.2 Objectif

Une notation peut viser différents objectifs de modélisation. La notation CTT [Paterno 1997] par exemple cherche à répondre à plusieurs objectifs : permettre l'analyse de la tâche, et spécifier des solutions d'interactions, ainsi que servir de support à l'évaluation des modèles produits. Une notation telle que CUA (*Collaborative Usability Analysis*) [Pinelle 2003] propose de réaliser l'analyse de la tâche en s'appuyant sur des mécanismes de collaboration[5], puis d'effectuer une évaluation prédictive de l'utilisabilité sur les modèles produits. La notation GTA [Veer 2000] quant à elle repose sur une méthode de travail complète et couvre l'analyse des besoins et la spécification de solutions. La méthode inclut la description de modèles de tâches hiérarchiques pour l'interaction abstraite et la description d'interactions telles que des clics souris pour l'interaction concrète. La spécification de l'interaction concrète est faite avec la notation NUAN (*New User Action Notation*) [Venema 1999]. En termes de cycle de vie logiciel, ces notations sont utilisées pour réaliser l'analyse de la tâche et pour spécifier une solution d'interaction conçue. Les modèles produits peuvent ensuite servir à la génération de code ou de prototype, être embarqués à l'exécution, ou être le support pour des tests ou des évaluations.

Pour cette étude des notations existantes, nous analysons le positionnement des notations selon les grandes étapes du cycle de vie logiciel couverte : analyse des besoins, analyse de la tâche, spécification, conception générale, conception détaillée, et enfin support à la génération de code ou prototype. Nous analysons également la couverture logicielle des notations en termes de composants logiciels du modèle ARCH [Bass 1992] : noyau fonctionnel, adaptateur du noyau fonctionnel, contrôleur de dialogue, interaction abstraite et interaction concrète.

[5] Mécanismes de collaboration : ensemble de primitives d'action de groupe définies par les auteurs de [Pinelle 2003], tel qu'émettre un message oral ou obtenir une ressource.

1.1.3 Nombre et types de vues

Des notations proposent plusieurs vues pour décrire un système interactif. Chaque vue est associée à un ensemble de concepts. Certains concepts peuvent être identifiés au sein de plusieurs vues et être représentés de manière différente dans chacune des vues. Le nombre de représentations et leur type indiquent à gros grain les concepts pris en charge par une notation.

Ainsi, la notation CTT [Paterno 1997] propose une vue unique visant à décrire le modèle de tâche d'un système interactif : il s'agit d'un arbre de tâches. Les concepts secondaires tels que les rôles et les objets utilisés sont décrits au sein de la description de chaque tâche. Lorsque l'on décrit un système coopératif avec CTTE [Mori 2002], plusieurs vues sont définies : un arbre de tâches pour décrire la coopération et un arbre de tâches individuelles par rôle dans l'application. Néanmoins, ces différentes vues forment un seul et unique point de vue sur l'interaction : une description des tâches.

Au contraire, la notation MABTA [Lim 2004] propose quatre vues d'une application interactive : un modèle des rôles et des utilisateurs, un diagramme de flot de travail, un modèle de tâches et enfin des maquettes de l'interface graphique.

Enfin, une notation comme GTA [Veer 2000] est elle-même une combinaison de plusieurs notations et permet la description de diagrammes de séquences, de modèles de dialogue NUAN [Venema 1999] et d'arbres de tâches. Ces différentes notations sont autant de vues complémentaires sur un système interactif.

1.1.4 Forme des représentations

Comme nous l'avons souligné précédemment, une notation peut proposer plusieurs vues pour spécifier un système interactif. Chaque vue peut représenter des concepts (parfois identiques) sous une forme différente. Aussi nous nous focalisons ici sur les différentes formes de représentation des vues véhiculées par une notation.

Par exemple le modèle des rôles et des utilisateurs de la notation MABTA [Lim 2004] est représenté sous la forme d'un arbre, tandis qu'un modèle similaire dans la notation TOUCHE [Penichet 2006] est représenté par un diagramme objet relationnel. En effet, la première notation s'intéresse strictement à la modélisation des relations d'héritage et d'instance entre rôles et utilisateurs, tandis que la deuxième s'intéresse à décrire des tâches de coopération entre ces rôles. Certaines notations proposent plusieurs vues ayant une représentation identique, à l'instar des arbres de tâches coopératives et individuelles de CTT [Paterno 1997, Mori 2002]. Pour d'autres notations, telles que la notation CUA [Pinelle 2003], chaque vue adopte une forme différente. Ainsi, les scénarios de CUA sont décrits sous une forme textuelle (à la façon d'une pièce de théâtre). Les modèles de flot de travail sont décrits sous la forme d'un diagramme non hiérarchique reliant plusieurs tâches. Enfin le modèle de tâches adopte une forme arborescente. Enfin, la notation NUAN [Venema 1999] propose de décrire sous la forme d'un tableau des séquences d'actions utilisateur et système. La disposition des actions dans le tableau indique les relations entre ces actions.

Dans le cadre de notre analyse, nous détaillons la forme de chaque représentation proposée par une notation : textuelle, tabulaire, dessin, diagramme. Pour cette dernière, nous précisons la forme du diagramme comme : non hiérarchique, hiérarchique par imbrication ou arborescent.

1.1.5 Outils logiciels

Un point important pour les notations de spécification concerne la disponibilité d'outils logiciels pour accompagner la notation. Les outils logiciels sont utiles car ils structurent le processus de conception et aident le concepteur à comprendre les concepts qu'il/elle manipule [Veer 2000]. L'importance d'outils accompagnant les notations a également été soulignée par [Mori 2002] et [Molina 2009] comme vecteur à l'utilisation et à la diffusion d'une notation. Enfin les auteurs de GTA [Veer 2000] insistent sur la diffusion publique des outils accompagnant les notations de spécifications.

Un outil accompagnant une notation peut être de multiple nature. Nous identifions trois grandes classes d'outils : des éditeurs de spécification, des assistants d'analyse des spécifications produites et des générateurs de code.

Ces éditeurs peuvent être complétés par des outils d'aide à l'analyse des modélisations produites. C'est le cas de l'outil CTTE qui permet l'édition d'arbre de tâches selon la notation CTT [Mori 2002], et qui permet aussi d'animer les spécifications. De même l'éditeur EUTERPE [Veer 2000] qui permet la spécification selon la notation GTA (arbre de tâches, NUAN) est étendu pour couvrir toute la méthode de conception DUTCH et inclut par exemple l'analyse semi-automatique des modèles produits. L'utilisateur de l'outil peut ainsi vérifier des assertions sur sa spécification, telles que la présence d'associations d'au moins une tâche par rôle décrit, qui détecte ainsi la présence de rôles non-utilisés dans le modèle.

Enfin, certain éditeurs de spécification permettent également de générer du code ou des prototypes à partir de spécifications. L'outil de spécification de diagrammes UML StarUML [StarUML] permet par exemple de générer des classes décrites au sein de diagrammes de classes UML. L'outil EUTERPE associé à GTA permet quant à lui de générer la spécification au format HTML à partir des diagrammes produits. Chaque tâche y est par exemple décrite sous la forme d'un tableau contenant l'ensemble des propriétés décrites dans l'outil. Cette présentation est plus utilisable lors du développement, puisqu'elle n'impose pas la manipulation de l'outil par les développeurs, mais seulement la consultation des pages web générées.

1.2 Caractéristiques liées à l'interaction

Cette section rassemble les caractéristiques des notations qui concernent la spécification de l'interaction. Notre attention est ici portée sur la capacité des notations à décrire l'interaction et non sur les dimensions aux cas du multiutilisateur et de la multimodalité qui font l'objet des sections 1.3 et 1.4. Toutefois, ces dimensions seront ici abordées pour souligner l'intérêt de certaines caractéristiques.

1.2.1 Description structurée de l'interaction

La plupart des notations disposent de vues pour décrire l'enchainement logique ou temporel des interactions.

Le modèle de tâches est l'approche la plus courante et peut prendre de nombreuses formes comme nous l'avons évoqué en 1.1.4 de ce chapitre. La forme la plus courante est un arbre de tâches. Celle-ci offre une structure hiérarchique qui consiste à décrire une tâche comme étant une composition de sous-tâches, du plus abstrait au plus concret. Les feuilles de l'arbre sont alors des tâches

élémentaires. Par exemple, les arbres de tâches de la notation CTT entrent dans ce cadre [Paterno 1997]. La granularité des tâches élémentaires dépend d'une part des objectifs de la notation, et d'autre part des préoccupations du concepteur qui réalise les arbres. Ainsi par exemple, pour la notation CTT, une tâche élémentaire est une tâche qui n'est pas décomposée en sous-tâches. En revanche pour la notation GTA [Veer 2000], les tâches élémentaires de l'arbre sont des tâches concrètes qui sont détaillées par la suite avec la notation NUAN [Venema 1999].

Une autre forme consiste à décrire un flot d'activités dont la description introduit une logique temporelle forte. La notation MABTA [Lim 2004] intègre un diagramme de flot de travail dans ses représentations, qui prend la forme d'un ensemble de boîtes reliées par des flèches. La notation Orchestra [David 2006] propose quant à elle de décrire le flot de travail sous la forme de mesures d'une portée musicale. Chaque mesure permet de décrire une tâche donnée. Dans les deux cas précédents, les flots de travail sont non hiérarchiques, c'est-à-dire que les tâches représentées ne sont pas décomposées en unités de travail plus fines. Néanmoins, les diagrammes de flot de travail peuvent être hiérarchiques, c'est le cas par exemple de la notation CIAN [Molina 2006]. CIAN propose en effet de décrire une décomposition en sous-tâches sous la forme d'un nouveau flot de travail. Ce nouveau flot de travail est représenté directement comme contenu de la tâche décomposée.

Les notations précédemment introduites pour décrire l'interaction permettent essentiellement de décrire l'interaction à haut niveau d'abstraction. Une notation comme NUAN [Venema 1999] prend en charge la description de l'interaction concrète sous la forme d'une séquence d'actions utilisateur et/ou système.

Enfin, UML permet de décrire l'interaction au travers de son diagramme d'activité à l'instar du diagramme de flot de travail de CIAN [Molina 2006]. En outre, UML permet de modéliser le dialogue concret de l'interface en décrivant les interacteurs qui la composent sous la forme de diagrammes d'état-transition.

Quel que soit la forme que prend la décomposition de l'interaction, elle se révèle utile pour décrire des activités coopératives, car les sous-tâches sont souvent assignées à un rôle utilisateur. C'est le cas notamment dans la notation CTT [Paterno 1997] où chaque tâche élémentaire de l'arbre de tâches coopératives peut être associée à un rôle [Mori 2002].

Dans la suite, et par commodité de langage, la notion de tâche est utilisée pour faire référence aux éléments composant l'interaction. Elle recouvre ainsi les activités pouvant être décrites dans un diagramme d'activité, les tâches d'un flot de travail, les tâches d'un modèle de tâches, et les actions utilisateur et réactions du système.

1.2.2 Relations de composition

Une description structurée de l'interaction repose sur l'usage de relations de composition afin de décrire l'enchaînement logique ou temporel des tâches. Les relations les plus courantes sont la séquence, l'alternative et la concurrence.

Les notations CTT [Paterno 1997], K-MAD [Lucquiaud 2005, Baron 2006] et CUA [Pinelle 2003] introduisent des opérateurs logiques et temporels pour décrire les relations entre tâches. Le nombre et la précision de ces opérateurs varient selon les notations. Ainsi, K-MAD propose quatre opérateurs

: la séquence, l'alternative, le parallélisme et l'absence d'ordre. La notation CTT utilise des opérateurs similaires (respectivement nommés activation, choix, parallélisme, et ordre indéfini) mais propose des d'opérateurs complémentaires pour signifier l'entrelacement entre deux tâches, la désactivation d'une tâche par une autre, ou le fait qu'une tâche suspende une autre tâche.

Pour les diagrammes d'activités et diagrammes de flot de travail, les relations temporelles prennent la forme de liens décrivant la relation. Ainsi une flèche entre deux activités d'un diagramme d'activités UML décrit que l'activité à l'origine de la flèche doit être réalisée avant de réaliser l'activité à l'extrémité de la flèche.

Dans le cas d'une notation comme NUAN, c'est la position des éléments textuels dans le tableau qui définit l'ordre des actions. Ainsi, une séquence d'actions est décrite implicitement dans le tableau par un saut de ligne. Le parallélisme est quant à lui décrit explicitement par un opérateur (||).

Pour notre analyse, la capacité d'une notation à décrire les relations entre tâches est détaillée en fonction d'une part de la manière de leur mise en œuvre parmi : {implicite, opérateur entre tâche, lien entre tâche}, et d'autre part la précision des relations représentées : {abstraite, concrète} et le nombre des relations proposées.

1.2.3 Objet du domaine

Le concept d'objet est présent dans la plupart des notations de description de l'interaction. Ces objets, parfois appelés objet de la tâche [Mori 2002], sont souvent des objets métiers, c'est-à-dire des objets correspondant aux concepts du métier que prend en charge l'application. Ils apparaissent souvent dans le nom des tâches lors de la description de l'interaction. Par exemple la tâche qui consiste à « Créer un fantassin » dans WCCM (décrit au Chapitre 1) utilise le concept de « fantassin ». Outre des objets métiers, une tâche peut être liée à des objets créés pour faciliter l'interaction. Ceci correspond aux cas de séparations sémantiques expliqués dans [Coutaz 2001]. Par exemple, dans WCCM, l'objet « Lasso » est manipulé pour sélectionner un ensemble d'icônes, et notamment un ensemble d'unités.

Les objets peuvent être décrits de manière isolée au sein de diagrammes de classe UML ou de diagramme RML [Treatteberg 2002], mais peuvent être également décrits au sein de la description d'une tâche abstraite comme dans la notation CIAN [Molina 2006]. Enfin, ils peuvent être associés implicitement à une tâche comme dans la notation MABTA [Venema 1999], ou être associés explicitement à une tâche comme dans les notations CTT [Paterno 1997, Mori 2002] et K-MAD [Lucquiaud 2005, Baron 2006].

Dans le cadre de systèmes multiutilisateurs, les objets manipulés par plusieurs utilisateurs lors de l'utilisation de l'application peuvent être partagés entre les utilisateurs. La notation UML-G [Rubart 2004] prend en charge la description du partage grâce à l'utilisation de prototypes pour UML. Nous décrivons plus précisément cet aspect dans à la section 2.1.1 de ce chapitre.

1.2.4 Contexte d'exécution d'une tâche

Plusieurs notations s'intéressent à représenter explicitement le contexte d'exécution d'une tâche. Le contexte peut exprimer la localisation des utilisateurs (co-localisés ou distants), comme cela est suggéré par les auteurs de la notation Orchestra [David 2006].

Le contexte peut aussi décrire les plateformes informatiques utilisées pour réaliser une tâche à l'instar de la notation CTT [Mori 2002]. Outre la plateforme informatique nécessaire à la réalisation d'une tâche, le contexte peut aussi inclure une description de l'environnement physique. Par exemple la notation K-MAD [Lucquiaud 2005, Baron 2006] propose ainsi de décrire les dispositifs physiques à disposition des utilisateurs. Enfin utilisateurs, plateforme informatique et physique peuvent être décrits conjointement au sein de diagrammes d'interaction avec la notation ASUR [Dubois 2002a].

De nombreux paramètres peuvent être pris en compte lors de la description du contexte physique, tels que la luminosité et le bruit ambiant. C'est pourquoi les notations existantes proposent souvent de décrire ces contextes d'utilisation de manière textuelle puis de les associer à la réalisation d'une ou plusieurs tâches comme dans la notation Orchestra [David 2006]. C'est également l'approche que proposent les auteurs de la notation CUA [Pinelle 2003], puisqu'ils préconisent de décrire des scénarios (contexte, utilisateur, rôle) auquels est associé un arbre de tâche.

Dans le cadre de notre analyse, nous étudions la capacité d'une notation à décrire le contexte d'exécution d'une tâche selon les éléments explicitement prévus par la notation comme les dispositifs physiques, l'environnement physique, ou encore la localisation des utilisateurs.

1.2.5 Actions du système

Certaines notations qui permettent l'analyse des besoins se focalisent sur les actions que peuvent réaliser les utilisateurs. D'autres, qui visent à fournir des spécifications plus proches de l'implémentation, cherchent à décrire également les actions du système. Celles-ci peuvent être de deux ordres : il peut s'agir d'une part d'actions initiées par le système, on parle alors de proaction, d'autre part de réactions du système à des sollicitations des utilisateurs, on parle alors de rétroaction. La majorité des notations existantes proposent un moyen de décrire les actions du système. La notation CTT [Paterno 1997] par exemple utilise le concept de tâche système qui permet de décrire aussi bien les tâches de proaction que de rétroaction. De plus, CTT permet de décrire textuellement les effets de chaque tâche sur les objets du domaine. Ces deux possibilités sont également présentes dans la notation K-MAD [Lucquiaud 2005, Baron 2006] : en effet bien que la notation K-MAD soit centrée sur les actions des utilisateurs, elle vise aussi à décrire des systèmes complets existants. Les actions du système ayant des conséquences sur les utilisateurs peuvent donc y être décrites. Plusieurs notations reposant sur CTT telles que CIAN [Molina 2006] ou Touche [Penichet 2006] prennent également en charge la description d'actions du système. D'autres notations telles que MABTA [Lim 2004] se focalisent sur la description des actions des utilisateurs et ne proposent pas de moyen particulier pour décrire les actions du système (tâche de rétroaction).

La notation NUAN [Venema 1999] permet de décomposer les réactions du système en deux catégories qui sont exprimées dans deux colonnes distinctes : les réactions immédiates à l'interaction de l'utilisateur (par exemple, le déplacement continu d'une forme lors d'un glisser-déposer), et les réactions du système suite à la réalisation d'une tâche (par exemple, l'insertion de l'objet déposé dans une liste). Cette distinction est explicitée dans le modèle pipeline [Nigay 94] [Coutaz 94] qui articule les retours d'information (rétroactions lexicales, syntaxiques et sémantiques) et les effets des actions utilisateurs (réaction du système).

La conscience des autres et de leurs actions, qui est un aspect important des systèmes interactifs multiutilisateurs est lié à la représentation des actions du système. Elle est traitée dans la section suivante consacrée à la dimension multiutilisateur.

1.3 Caractéristiques liées à la dimension multiutilisateur

La dimension multiutilisateur d'un système interactif introduit plusieurs éléments qui doivent être considérés lors de la conception. Cette section aborde dans l'ordre : la modélisation de la coopération et de la collaboration entre les utilisateurs, la modélisation des objets partagés entre les utilisateurs et enfin, la répartition spatiale et temporelle des utilisateurs.

1.3.1 Coopération et collaboration

Nous avons présenté au Chapitre 1 section 1.3 la distinction entre activité coopérative et collaborative. Ainsi, une activité coopérative implique une répartition du travail a priori entre les participants, tandis que l'activité collaborative implique une répartition dynamique du travail. Ainsi, la répartition du travail entre les participants ne peut être décrite lors de la conception dans le cas d'activités collaboratives tandis qu'elle peut l'être pour des activités coopératives.

Les notations CTT [Paterno 1999], K-MAD [Lucquiaud 2005, Baron 2006], MABTA [Lim 2004] et GTA [Veer 2000] entre autres, permettent de décrire les activités coopératives. Dans le cas de la notation CTT, l'activité coopérative est représentée par la description conjointe d'un arbre de tâches coopératives avec plusieurs arbres de tâches individuelles. D'autres notations telles que CUA [Pinelle 2003] et CIAN [Molina 2006] introduisent le concept de tâche collaborative pour modéliser les activités collaboratives.

La spécification de l'interaction multiutilisateur qu'elle soit coopérative ou collaborative est un point central de notre étude. Nous portons donc un intérêt particulier à la capacité des notations à exprimer la coopération et la collaboration. Les deux sections suivantes décrivent en détail les concepts utilisés pour décrire l'une et l'autre.

1.3.2 Coopération

La coopération implique une division du travail entre des participants ayant des responsabilités données. Pour exprimer la responsabilité de la réalisation d'une partie du travail, les notations font intervenir le concept de rôle. Un rôle est ainsi défini comme un ensemble de responsabilités qui peuvent être endossées par un utilisateur. Un utilisateur jouant un rôle est qualifié d'acteur.

Ainsi, la notation CTT [Paterno 1997, Mori 2002] permet d'associer des rôles aux tâches d'un arbre de tâches coopératives, et un arbre de tâches individuelles est associé à chaque rôle. La notation K-MAD [Lucquiaud 2005, Baron 2006] quant à elle propose de décrire séparément les utilisateurs et le rôle qu'ils jouent, puis d'associer les utilisateurs aux tâches de l'arbre. D'autres notations, tels que TOUCHE [Penichet 2006], CIAN [Molina 2006] et TaskMODL [Treatteberg 2002] proposent de décrire d'une part les rôles, les acteurs et leurs relations dans une représentation appelée sociogramme (ou diagramme organisationnel), puis d'associer ces rôles au sein des modèles de tâches. La notation TOUCHE introduit également dans son sociogramme la possibilité de décrire des tâches coopératives qui relient deux rôles. Enfin, la notation MABTA [Lim 2004] offre la possibilité de diviser son modèle de tâches en colonnes, où chaque colonne correspond à un rôle.

Certaines des notations, telles que la notation GTA [Veer 2000], considèrent qu'un rôle n'est pas nécessairement joué par un utilisateur. C'est pourquoi GTA définit la notion d'Agent qui peut être un utilisateur ou un agent logiciel et peut jouer un rôle. La notation CIAN [Molina 2006] reprend le concept d'agent dans la description de son sociogramme, mais le limite à la notion d'agent logiciel. Ainsi, le concept d'acteur dans la notation CIAN concerne uniquement des utilisateurs.

1.3.3 Collaboration

Actuellement, peu de notations permettent la modélisation de la collaboration entre des utilisateurs même si le terme de tâche collaboratives ext explicitement utilisé. Par exemple la notation CTML (*Collaborative Task Modeling Language*) [Wurdel 2008a] propose de décrire la collaboration de tuples formés d'un ensemble de rôles, d'un ensemble d'acteurs, d'un ensemble de tâches collaboratives (un ensemble pour chaque rôle) et d'un ensemble d'objets du domaine. Les tuples sont utilisés pour simuler l'utilisation de l'application au sein de parcours d'arbres. Dans ce cadre, la définition est conforme à celle de la collaboration puisque elle se réfère à l'exécution des tâches, c'est-à-dire aux cas où les éléments du tuple sont alors connus. Néanmoins elle ne correspond pas à la définition de la collaboration si on se place dans le cadre de la description à priori de l'interaction, ce qui est l'objet de notre démarche. De ce point de vue, CTML ne permet pas de décrire la collaboration.

Nous notons néanmoins deux notations : CIAN [Molina 2006] et CUA [Pinelle 2003] qui explicitent de façon assez similaire la collaboration.

La notation CIAN [Molina 2006] utilise le concept de tâche collaborative en complément du concept de tâche coopérative. Pour CIAN, une tâche collaborative est une tâche abstraite (elle-même décomposée en sous-tâches) à laquelle sont associés un ensemble de rôles participant à la collaboration. La décomposition en sous-tâches ne fait apparaître aucun rôle conformément à la définition.

La notation CUA [Pinelle 2003] propose également d'utiliser le concept de tâche collaborative, selon une approche similaire à celle de CIAN. Néanmoins, CUA introduit en plus un ensemble de mécanismes de collaboration, qui référencent des primitives de collaboration possibles entre des utilisateurs. Ces mécanismes de collaboration peuvent être associés aux feuilles d'un arbre de tâches CUA, et décrivent les possibilités de collaboration entre les utilisateurs.

1.3.4 Objets partagés

Nous avons évoqué plus haut que le concept d'objet de la tâche est présent dans la plupart des notations. Dans le cadre d'applications multiutilisateur, de nombreux objets sont partagés entre les utilisateurs. Ces objets peuvent être soit utilisés directement par les utilisateurs lors de l'interaction, soit servir de support à la présentation d'informations permettant d'assurer la conscience des autres et de leurs actions.

La notation CIAN [Molina 2006] propose d'associer à une tâche coopérative ou collaborative donnée, un ensemble d'objets qui sont partagés entre les utilisateurs. Il s'agit ainsi de décrire des classes d'objets partagés mais également des instances. Pour chaque instance, il est possible de décrire si l'objet est créé par la tâche, ou s'il fait l'objet de modifications lors de la réalisation de la tâche. Pour compléter cette description, CIAN propose de décrire la dimension du partage pour chacun de ces

objets vis-à-vis de l'interface. Ainsi, un objet peut être présenté et manipulé en parallèle sur des espaces de travail à destination du groupe, présenté et manipulé en parallèle sur des espaces de travail individuels ou être partagé mais n'être modifiable qu'à tour de rôle.

La notation TaskMODL [Treatteberg 2002] permet également de décrire les objets partagés qui sont utilisés lors de la réalisation des tâches. Comme pour CIAN, il est possible de décrire des classes d'objets et des instances. La notation propose également de modéliser le lien d'utilisation d'instances d'objets entre deux tâches de l'arbre. Ainsi lorsque qu'un objet est créé suite à la réalisation d'une tâche et associé à la réalisation d'une autre tâche de l'arbre, un lien peut être décrit et porter mention de l'instance d'objet qui est manipulée par les deux tâches. Notons que la notation TaskMODL propose également de décrire le modèle de dialogue des interacteurs qui composent l'interface, sous la forme de diagrammes d'état-transition hiérarchiques. Ces descriptions complémentaires peuvent permettre de décrire le changement d'état d'un objet lors de la réalisation d'une tâche.

La notation NUAN [Venema 1999] permet quant à elle de décrire lors de séquences d'action-réaction les modifications d'état des objets partagés aussi bien pour l'utilisateur qui les manipule que pour les autres utilisateurs.

Enfin, la notation UML-G [Rubart 2004] se focalise sur la modélisation des objets partagés, et propose de décrire la politique de contrôle d'accès, de modification et de distribution sur les objets. Il est ainsi possible de décrire qu'un utilisateur peut poser un verrou sur un objet afin de le modifier. Il est également possible d'indiquer si les modifications d'un objet par un utilisateur entraînent une notification de ces modifications aux autres utilisateurs. Ceci est utile pour décrire l'observabilité et les moyens mis en œuvre pour la conscience de groupe. Par ailleurs, la distribution d'un objet entre des utilisateurs peut être décrite comme étant centralisée ou répliquée.

Pour notre analyse des notations existantes, nous considérons la capacité d'une notation à décrire les objets qui sont partagés entre les utilisateurs. Nous précisons aussi les extensions proposées par les notations pour décrire plus finement ce partage d'objets, comme les aspects liés à l'observabilité.

1.3.5 Répartitions spatiale et temporelle des utilisateurs

Un système coopératif ou collaboratif peut être utilisé de manière synchrone ou asynchrone, et de manière co-localisée ou distante par les utilisateurs. Selon les notations, il est possible de spécifier ou non la localisation des utilisateurs et l'aspect synchrone de l'interaction.

Une notation comme Orchestra [David 2006] permet de décrire la synchronisation temporelle sous la forme de situations d'utilisation. En effet, une situation regroupe des utilisateurs dans un espace de temps donné, ce qui exprime l'aspect synchrone. La description des situations d'Orchestra fait également intervenir la notion de contexte d'utilisation qui fixe, entre autre, le lieu de l'interaction pour les utilisateurs. C'est dans ce cadre qu'il est possible d'exprimer la co-localisation des utilisateurs ou la distance entre les utilisateurs.

Dans les autres notations, l'aspect de synchronisation entre les utilisateurs est implicitement décrit par le concepteur sans que des artéfacts particuliers ne soient proposés. Ainsi des tâches CTT [Mori 2002], CIAN [Molina 2006] ou TaskMODL [Treatteberg 2002] peuvent être liées par une dimension synchrone forte. Pour cela, le concepteur peut décrire au sein d'un même arbre de tâches l'ensemble

des interactions se situant dans un même lieu, ou se déroulant au sein d'une période de temps donnée. De la même manière, les notations CIAN [Molina 2006] et TOUCHE [Penichet 2006] permettent de décrire des groupes dans leurs sociogrammes qui peuvent être utilisés pour décrire le caractère synchrone ou la localisation. Ainsi, un groupe nommé « participant à une réunion » indique une dimension de synchronisation, et un groupe nommé « utilisateurs d'une table tactile » indique une dimension spatiale.

1.4 Caractéristiques liées à la dimension multimodale

Après la dimension multiutilisateur, la dimension multimodale des notations constitue le deuxième axe central de notre analyse et l'objet de cette section. Cette section concerne la capacité des notations à décrire la multimodalité en nous référant aux définitions adoptées au Chapitre 1 section 2.1.

Au sein des sous-sections suivantes, nous observons la capacité des notations à décrire l'usage de modalités d'interaction en termes de dispositifs physiques d'interaction et de langages d'interaction. Nous abordons également la capacité à décrire l'association d'une ou plusieurs modalités d'interaction aux tâches.

1.4.1 Modalité d'interaction

Quelques notations permettent de spécifier l'usage de modalités d'interaction. Parmi celles-ci, le framework DynaMo-AID [Clerkx 2008] réutilise la notation CTT [Paterno 1997] et lui ajoute la possibilité d'associer aux tâches feuilles d'un arbre de tâches une modalité ou une combinaison de modalités d'interaction. Cela permet d'indiquer pour chaque tâche la (les) modalité(s) utilisée(s), mais également lorsqu'il y en a plusieurs, la nature de la combinaison sous la forme d'opérateur reposant sur les propriétés CARE [Coutaz 1994].

Les autres notations identifiées ne répondent pas spécifiquement à la question de l'expression de la multimodalité. En revanche, plusieurs permettent de décrire des dispositifs d'interaction.

1.4.2 Dispositif d'interaction

Plusieurs notations permettent de spécifier les dispositifs d'interactions. La représentation d'un dispositif varie d'une notation à l'autre.

Pour certaines, telles que CTT [Paterno 1997], cela se limite à décrire la plateforme utilisée telles qu'un PDA, un téléphone ou un ordinateur de bureau. La plateforme choisie indique implicitement les dispositifs disponibles : un clavier, une souris et un écran pour un ordinateur de bureau.

D'autres notations telles que TaskMODL [Treatteberg 2002], K-MAD [Lucquiaud 2005, Baron 2006] ou NUAN [Venema 1999] vont plus loin et permettent de décrire explicitement les dispositifs d'interaction utilisés pour chaque tâche.

Avec les notations UML-G [Rubart 2004] et TaskMODL, il est également possible de décrire, de manière complémentaire, les dispositifs physiques utilisés par des diagrammes d'état-transition.

Les notations dédiées à la spécification de systèmes de réalité augmentée telles qu'IRVO [Chalon 2005] et ASUR [Dubois 2002a] proposent de modéliser au sein de diagrammes d'interaction des dispositifs, des transducteurs (qui correspondent au langage d'interaction) et leurs relations.

Pour les besoins de notre analyse, nous retenons d'une part la capacité des notations à exprimer les dispositifs utilisés et d'autre part la capacité à modéliser le lien entre les dispositifs et les tâches que propose un système interactif.

1.4.3 Langage d'interaction

En nous reposant sur la définition d'une modalité [Nigay 1996], nous considérons qu'un langage d'interaction identifie les manières d'utiliser un dispositif physique en termes d'actions élémentaires.

Les notations d'arbres de tâches proposent de décrire les dispositifs utilisés pour réaliser une tâche mais ne permettent pas de préciser la manière d'utiliser le dispositif. Ainsi si le dispositif est un bouton, il doit être possible de décrire avec la notation que l'utilisateur doit « appuyer sur ce bouton » pour réaliser la tâche. Aucune notation d'arbres de tâches ne prévoit explicitement de spécifier les actions possibles sur les dispositifs. Cependant, il est possible de décrire une action sur un dispositif sous la forme d'une sous-tâche élémentaire. Ainsi, pour l'exemple du dispositif bouton, une tâche élémentaire pourrait correspondre à l'appui sur le bouton.

A l'inverse des notations d'arbres de tâches, la notation NUAN [Venema 1999] permet de décrire finement les actions qu'effectue l'utilisateur sur les dispositifs, bien que cela soit limité à la souris et au clavier. Ainsi l'usage de la souris peut être décrit aux travers de primitives tels que CLICK, HOLD ou MOVE.

Les notations IRVO [Chalon 2005] et ASUR [Dubois 2002a] proposent quant à elles de décrire des transducteurs qui peuvent être des capteurs (entrée) ou des effecteurs (sortie). Ceux-ci ne décrivent pas en détail les opérations réalisables via un dispositif, mais servent à expliciter les liens d'interaction entre mondes réels et virtuels.

Plusieurs notations proposent de décrire le langage d'interaction en sortie pour les interfaces visuelles sous la forme de maquettes graphiques. C'est le cas de la notation MABTA, de la notation TaskMODL [Treatteberg 2002] ou encore de la notation SpieLan [Bergh 2006]. Cette dernière utilise les CAN (*Canonicals Abstracts Prototypes*) [Constantine 2003] pour indiquer le comportement interactif de ces maquettes.

Pour notre analyse des notations existantes, nous étudions la capacité des notations à décrire la manière d'utiliser un dispositif d'interaction. Nous analysons ainsi la capacité à décrire des primitives d'utilisation ou des tâches très concrètes qui correspondent à l'usage de dispositifs.

1.4.4 Combinaison de modalités

La combinaison de modalité d'interaction pour la réalisation d'une tâche donnée peut prendre plusieurs formes que nous pouvons décrire avec les propriétés CARE (définies au Chapitre 1, section 2.2.2.1). De plus, la combinaison de modalités (complémentaire et redondante de CARE) impliquent des contraintes temporelles dans l'usage des modalités.

Nous constatons que les quatre formes de relation entre modalités définies par CARE sont traduisibles avec les notations d'arbre de tâches. Ainsi en considérant des tâches concrètes (correspondant à l'usage d'un dispositif), il est possible d'exprimer la complémentarité et la redondance de CARE par un opérateur ET, et l'équivalence par un opérateur OU (alternative).

Enfin, la notation NUAN [Venema 1999] permet de décrire d'une part des combinaisons d'entrées clavier/souris que ce soit en redondance, équivalence ou complémentarité, et d'autre part des combinaisons de sortie. Mais la description des interactions est limitée au poste de travail et au triplet (clavier, souris, écran).

1.5 Synthèse des caractéristiques

Dans les sections précédentes, nous avons exposé les principales caractéristiques des notations existantes. Nous avons organisé ces caractéristiques selon quatre catégories :

- une catégorie de caractéristiques générales,
- une catégorie de caractéristiques spécifiques à l'Interaction Homme-Machine,
- une catégorie couvrant les aspects liés à la dimension multiutilisateur, et,
- une catégorie couvrant les aspects liés à la dimension multimodale.

Le Tableau 4 récapitule les caractéristiques de chacune de ces quatre catégories.

Au sein de la deuxième partie de ce chapitre, les notations existantes sont passées en revue à la lumière de ces caractéristiques. Chaque notation est décrite en détail et illustrée avec le cas d'étude WCCM décrit au Chapitre 1. Puis, plusieurs tableaux de synthèse récapitulent les caractéristiques des notations existantes.

1 - Caractéristiques générales			
	Domaine		Le domaine d'origine de la notation : psychologie cognitive, ethnographie, ingénierie logicielle, TCAO…
	Objectif		Les ou les objectifs de modélisation visés par la notation
	Nombre		Le nombre de modèles proposés
	Type		Le type de représentations par exemple, un modèle de tâche, un diagramme organisationnel, ou encore, une machine à état
	Forme		La forme des représentations proposées, tels que graphique, tabulaire ou textuelle
	Outils		L'existence et la disponibilité d'outils basés sur la notation qui peuvent être des éditeurs, des outils d'analyse, ou encore des générateurs de code
2 - Spécification de l'interaction			
	Description structurée de l'interaction		La structure des descriptions de l'interaction : arbre de tâches, diagramme d'activité, flot de travail…
	Relation de composition		Les différentes relations qu'il est possible d'exprimer entre les tâches, telles que la séquence ou l'alternative.
	Objet du domaine		Les objets qui sont liés à la réalisation d'une tâche
	Contexte		Le contexte de la réalisation d'un ensemble de tâches
	Action du système		Les actions du système, que ce soit à son initiative, ou en réponse à des sollicitations des utilisateurs
3 - Dimension multiutilisateur			
	Coopération		L'interaction coopérative entre les utilisateurs
	Collaboration		L'interaction collaborative entre les utilisateurs
	Object partagés		Le partage des objets entres les utilisateurs
	Répartition spatiale		La répartition spatiale des utilisateurs
	Répartition temporelle		La répartition temporelle des utilisateurs
4 - Dimension multimodale			
	Modalité		L'utilisation de modalités d'interaction
		Dispositif physique	Les dispositifs physiques mis en jeu dans l'interaction
		Langage d'interaction	La manière dont les dispositifs physiques sont utilisés
	Combinaison		Les combinaisons de modalités d'interaction

Tableau 4 : Caractéristiques des notations : notre grille d'analyse

2 Revue des notations existantes

Pour notre étude comparative des notations existantes, les notations sont organisées selon leur domaine d'origine (Figure 22) du plus générique au plus spécialisé. Ainsi, la première section est dédiée aux notations issues du domaine du génie logiciel. Puis, la deuxième section se focalise sur les notations introduites dans le domaine de l'Interaction Homme-Machine. Ensuite, nous abordons les notations de trois sous-domaines du domaine de l'interaction homme machine que sont : le travail coopératif assisté par ordinateur, la multimodalité, et la réalité augmentée.

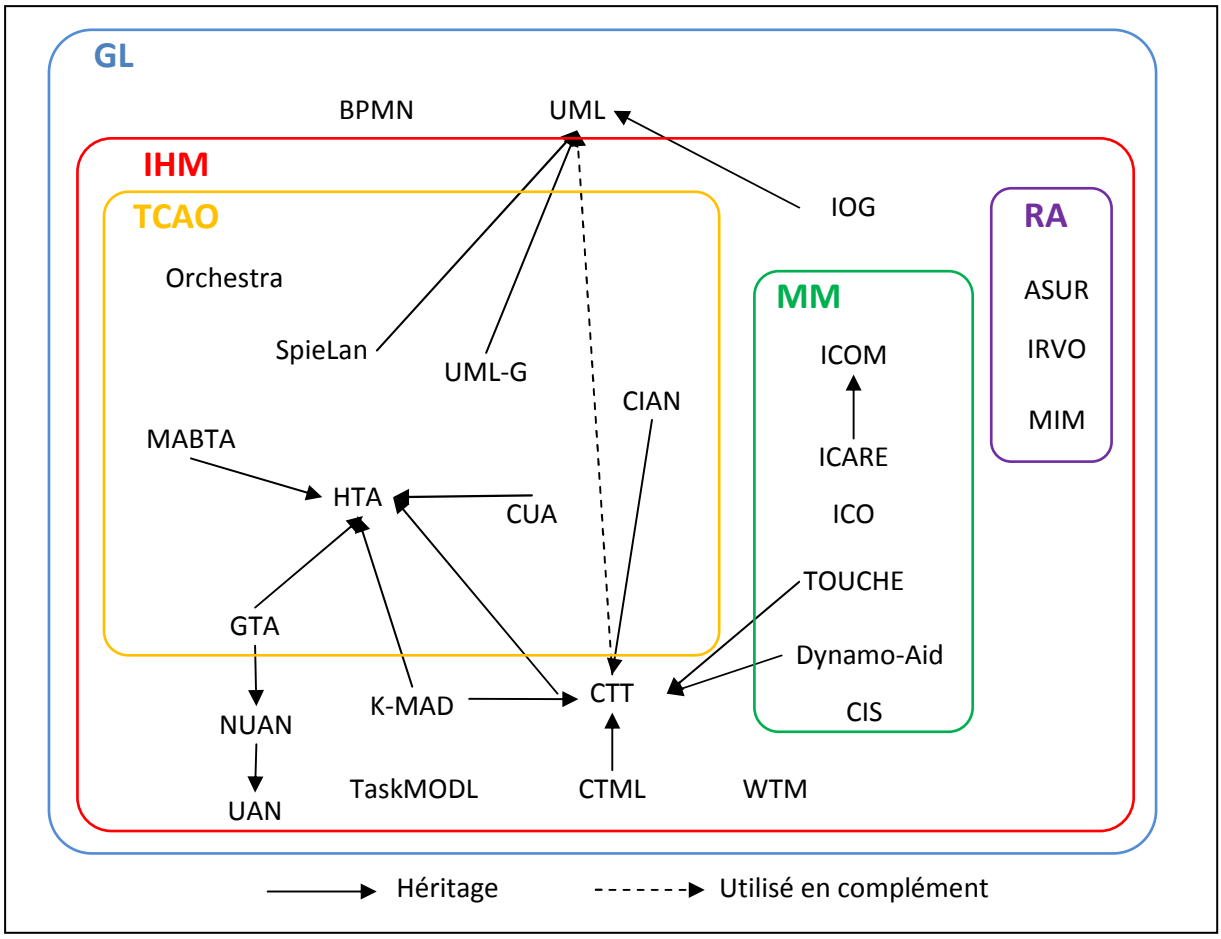


Figure 22 : Notations de spécification organisées selon leur domaine.

Les notations sont présentées au regard des caractéristiques présentées dans section 1.5 de ce chapitre. De plus, nous illustrons les notations par la modélisation d'un cas d'étude unique : l'application WCCM (Warcraft Collaboratif, Coopératif et Multimodal). Il constitue un référentiel commun d'utilisation des notations.

2.1 Domaine : Génie Logiciel

Dans cette section, nous présentons les notations UML et BPMN qui sont deux standards pour la modélisation logicielle issus du domaine du Génie Logiciel. Celles-ci prennent en compte plusieurs aspects de la modélisation de systèmes multiutilisateurs.

2.1.1 UML

UML (*Unified Modeling Language*) [UML] est un langage de modélisation graphique à base de pictogrammes qui permet la spécification d'un système informatique. UML est un standard défini par OMG (Object Management Group) qui est une association ayant pour but de promouvoir le modèle objet. UML est un outil de modélisation et non une méthode de conception. En effet, l'utilisation des différents diagrammes que propose UML est laissée libre.

Les diagrammes UML peuvent servir lors de plusieurs étapes du cycle de vie logiciel : analyse des besoins, spécification, conception architecturale et détaillée. En outre, les modélisations produites au

sein d'outils tels que [StarUML] peuvent servir de support à la génération de code d'application selon plusieurs langages de programmation tels que C++ ou Java.

Les diagrammes UML orientés vers la spécification du système couvrent l'ensemble les composants du modèle ARCH [Bass 1992] à l'exception de l'interaction concrète.

UML permet de décrire un système selon plusieurs vues. Chaque vue vise à décrire un ensemble de préoccupations particulières. Ainsi, le standard UML 2 définit 13 types de diagrammes. Les différents diagrammes peuvent être répartis selon deux catégories : ceux pour modéliser la structure d'une application, et ceux pour modéliser son comportement.

Les diagrammes permettant de modéliser la structure d'une application sont :

- Le *diagramme de classe*, qui permet de décrire les concepts utilisés dans les autres diagrammes, ainsi que les relations existant entre ces concepts
- Le *diagramme de paquetage*, qui permet de diviser les modélisations en paquetage et qui décrit les relations entre ces paquetages.
- Le *diagramme d'objet*, qui permet de décrire des instances des classes décrites dans un diagramme de classe.
- Le *diagramme de composant* qui permet de décrire des composants logiciels de haut niveau ainsi que les relations entre ces composants.
- Le *diagramme de structure composite* qui permet de décrire le comportement interne d'un élément, et ses relations avec les autres.
- Et le *diagramme de déploiement* qui permet de décrire la répartition des composants logiciels sur une architecture matérielle.

Les diagrammes permettant de modéliser le comportement d'une application sont :

- Le *diagramme de cas d'utilisation* qui définit les différents scénarios d'utilisation d'une application.
- Le *diagramme d'activité* qui permet de décrire le comportement de l'application à haut niveau d'abstraction.
- Le *diagramme d'état-transition* qui permet de décrire de manière précise le comportement de certains éléments.
- Le *diagramme de séquence* qui permet de décrire de manière précise un enchaînement d'actions en se focalisant sur la dimension temporelle.
- Le *diagramme de communication* qui permet également de décrire un enchaînement d'actions en se focalisant par contre sur les relations entre objets.
- Le *diagramme de temps* qui permet de décrire des changements d'état ou de valeur au cours du temps, et de décrire des contraintes temporelles.
- Et enfin, le *diagramme d'interaction globale* qui rassemble différents diagrammes pour décrire à très haut niveau d'abstraction l'enchaînement des différentes activités.

Notre objet d'étude est la spécification de l'interaction homme-machine multiutilisateur et multimodale, aussi, nous revenons en détail sur un sous ensemble des diagrammes UML pertinent pour l'interaction. Ainsi, nous ne considérons pas les diagrammes de composant, de déploiement, de

paquetage, de temps, et de structure composite qui nous apparaissent comme éloignés de notre sujet d'étude.

Le premier diagramme que nous considérons est celui des cas d'utilisation. Celui-ci permet d'exprimer les possibilités d'interaction que propose un système à ses utilisateurs. Ce type de diagramme est utile lors de la définition des besoins utilisateurs, car il permet d'identifier les fonctionnalités accessibles à chaque catégorie d'utilisateur identifié par un rôle.

La Figure 23 contient deux diagrammes de cas d'utilisation pour l'application WCCM. Le diagramme de cas d'utilisation de la Figure 23 (a) décrit les cas d'utilisation pour un rôle *Utilisateur*. Ce rôle est joué par tous les utilisateurs de WCCM. Le premier cas d'utilisation correspond à la création d'un paysan et requiert un utilisateur comme l'indique la relation entre le rôle *Utilisateur* et le cas d'utilisation *Créer un paysan*. D'autres cas d'utilisation correspondent à la construction d'une ferme ou d'un baraquement. Enfin, les cas d'utilisation *Créer un fantassin* et *Créer un cavalier* sont particulièrement intéressants, car ils requièrent la présence de plusieurs utilisateurs. Nous avons représenté cela par la multiplicité 2 entre le rôle utilisateur et les cas d'utilisation *Créer un fantassin* et *Créer un cavalier*. Le diagramme de cas d'utilisation de la Figure 23 (b) décrit quant à lui les cas d'utilisation pour les rôles *Sergent* et *Chef*. Un utilisateur jouant le rôle de *Sergent* peut ainsi *Déplacer un soldat* et *Former un groupe* tandis qu'un utilisateur jouant le rôle de *Chef* peut *Déplacer un groupe*. Enfin, chaque cas d'utilisation peut ensuite être décrit textuellement par son nom, son objectif, ses acteurs principaux et secondaires, des pré-conditions, des scénarii ou des post-conditions.

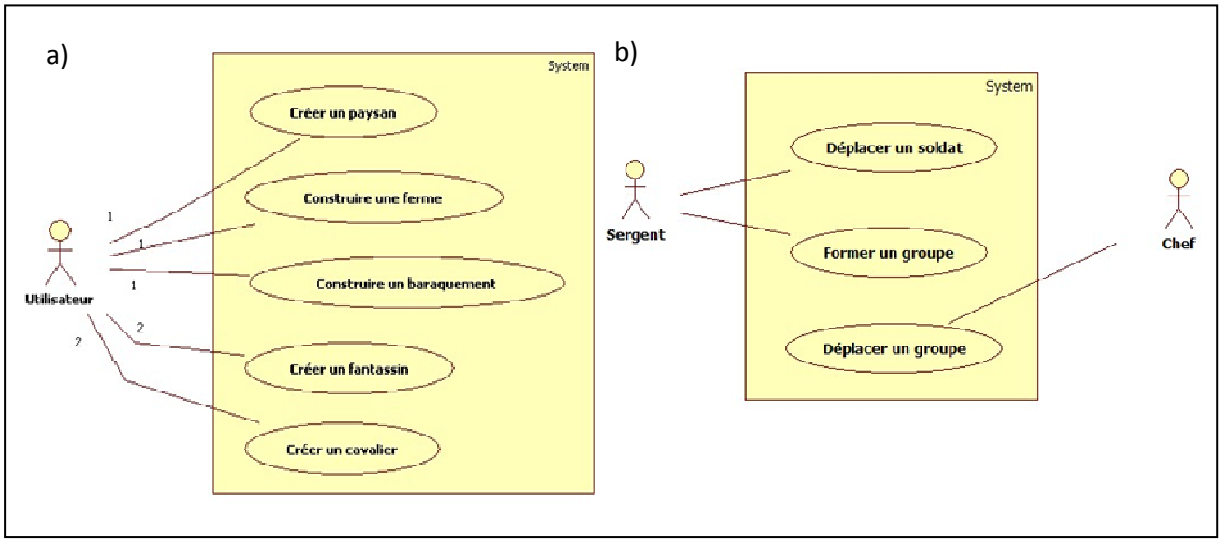


Figure 23 : Diagramme de cas d'utilisation pour les activités (a) collaboratives, (b) coopératives de WCCM.

Un diagramme particulièrement utile pour affiner un cas d'utilisation est celui de séquence. Celui-ci permet en effet de décrire un scénario d'exécution pour des acteurs donnés. La Figure 24 illustre ainsi un cas nominal d'exécution du cas d'utilisation *Créer un fantassin* introduit à la Figure 23. Nous décrivons ainsi que l'utilisateur 1 jouant le rôle d'*Utilisateur* commence par sélectionner un baraquement, ce qui conduit le système à lui présenter le choix des soldats que peut produire ce bâtiment : fantassin ou cavalier. Le même utilisateur choisit le fantassin. Le système demande alors au deuxième utilisateur s'il est d'accord pour produire le fantassin. Ce dernier confirme. Le système déclenche alors la création du fantassin.

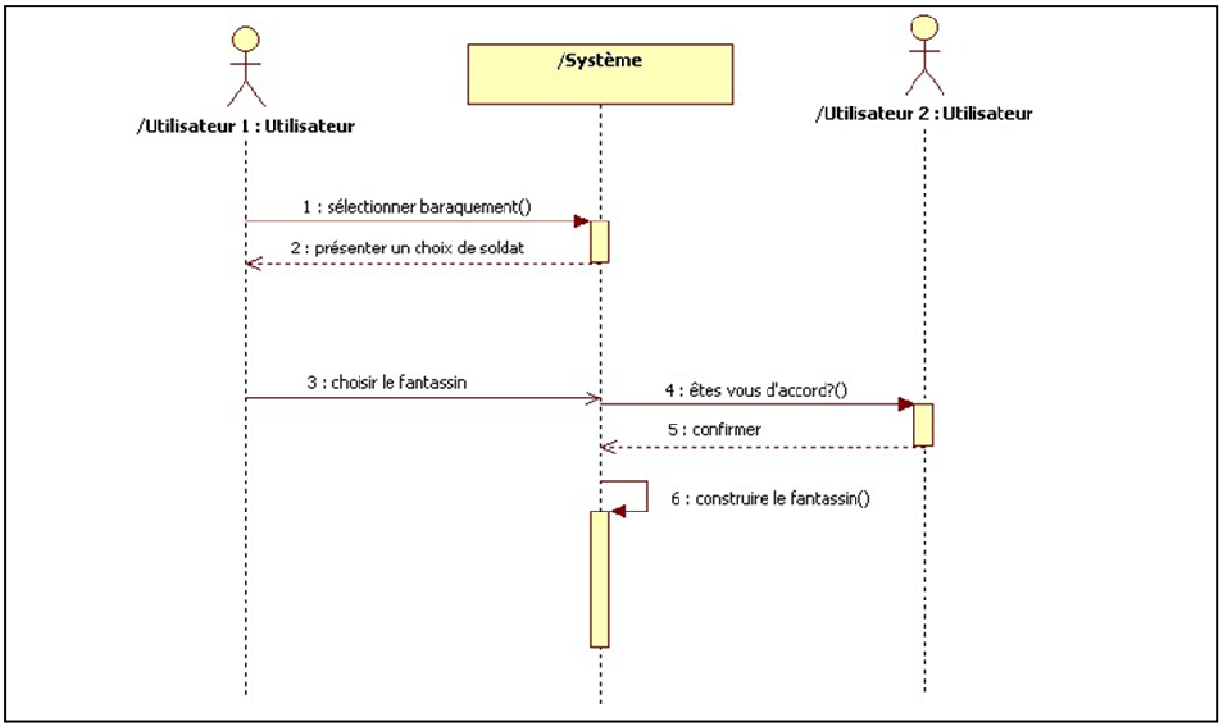


Figure 24 : Diagramme de séquence pour la création d'un fantassin.

Un nouveau type de diagramme a été introduit avec la définition d'UML 2. Il s'agit du diagramme de communication. Le diagramme de communication permet de décrire des interactions entre les utilisateurs et les composants du système à l'instar du diagramme de séquence. Toutefois, alors qu'un diagramme de séquence insiste sur l'aspect temporel avec une ligne de temps explicite, un diagramme de communication met l'accent sur les relations entre les objets, l'aspect temporel passant sur un second plan. Ceci est particulièrement visible si l'on observe comparativement les diagrammes de la Figure 24 et de la Figure 25 qui décrivent tout les deux la création d'un fantassin. A la Figure 25, les relations entre objets sont décrites explicitement par un lien qui porte un message. Notons que le diagramme de communication définit néanmoins un ordre sur les relations, ordre décrit grâce aux étiquettes numérotées que portent les messages. Ainsi, le premier utilisateur sélectionne un baraquement (1), qui en retour affiche un choix d'unité (2). L'utilisateur peut alors choisir de créer un fantassin (3). Le baraquement demande ensuite l'accord du deuxième utilisateur pour la création du fantassin (4). Ce dernier acquiesce (5). Enfin, le baraquement crée un objet fantassin (6).

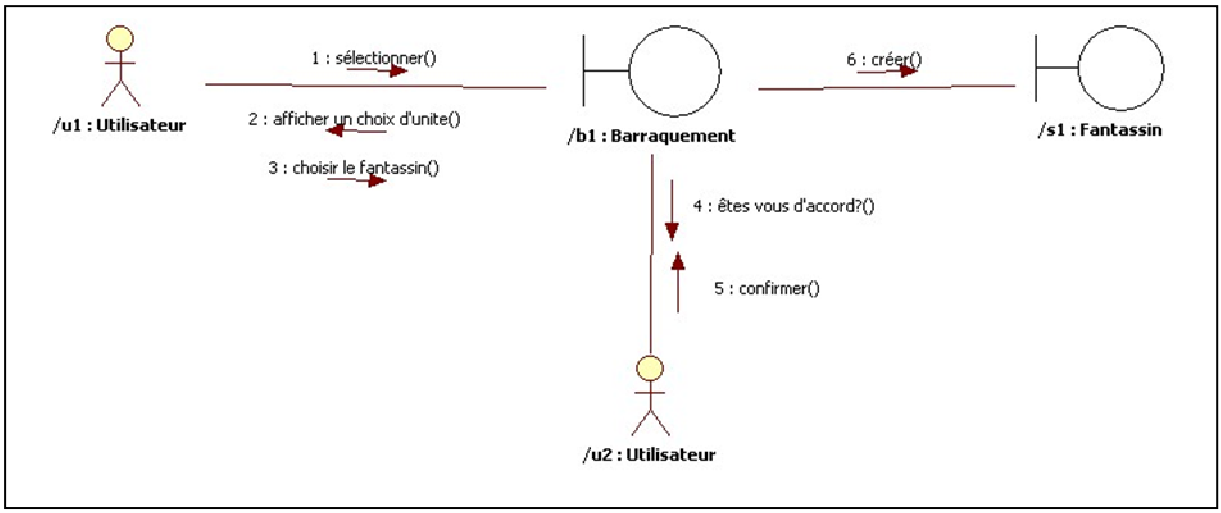


Figure 25 : Diagramme de communication pour la création d'un fantassin.

Tandis que le diagramme de séquence ou de communication sont utiles pour décrire un cas d'utilisation donné, le diagramme d'activité peut servir à décrire un ensemble de chemins d'utilisation possibles. La Figure 26 illustre ainsi un diagramme d'activité concernant la gestion d'un baraquement, qui inclut donc le cas nominal de création d'un fantassin. Pour gérer un baraquement, un utilisateur doit en premier lieu *Sélectionner ce baraquement*. Une fois qu'un baraquement est sélectionné, l'utilisateur peut, s'il dispose des ressources suffisantes, *Créer un fantassin* ou *Créer un cavalier*. Il peut également *Désélectionner le baraquement*, ce qui met fin à sa gestion. A la suite de l'action de création d'un fantassin, un autre utilisateur peut alors choisir de *Confirmer* la création du fantassin, ou au contraire de l'*Annuler*. Dans le premier cas, cela conduit le système à réaliser l'action de *Produire un fantassin* tandis que l'annulation conduit directement à la fin de l'activité de gestion. La création d'un cavalier est réalisée de la même manière que celle d'un fantassin.

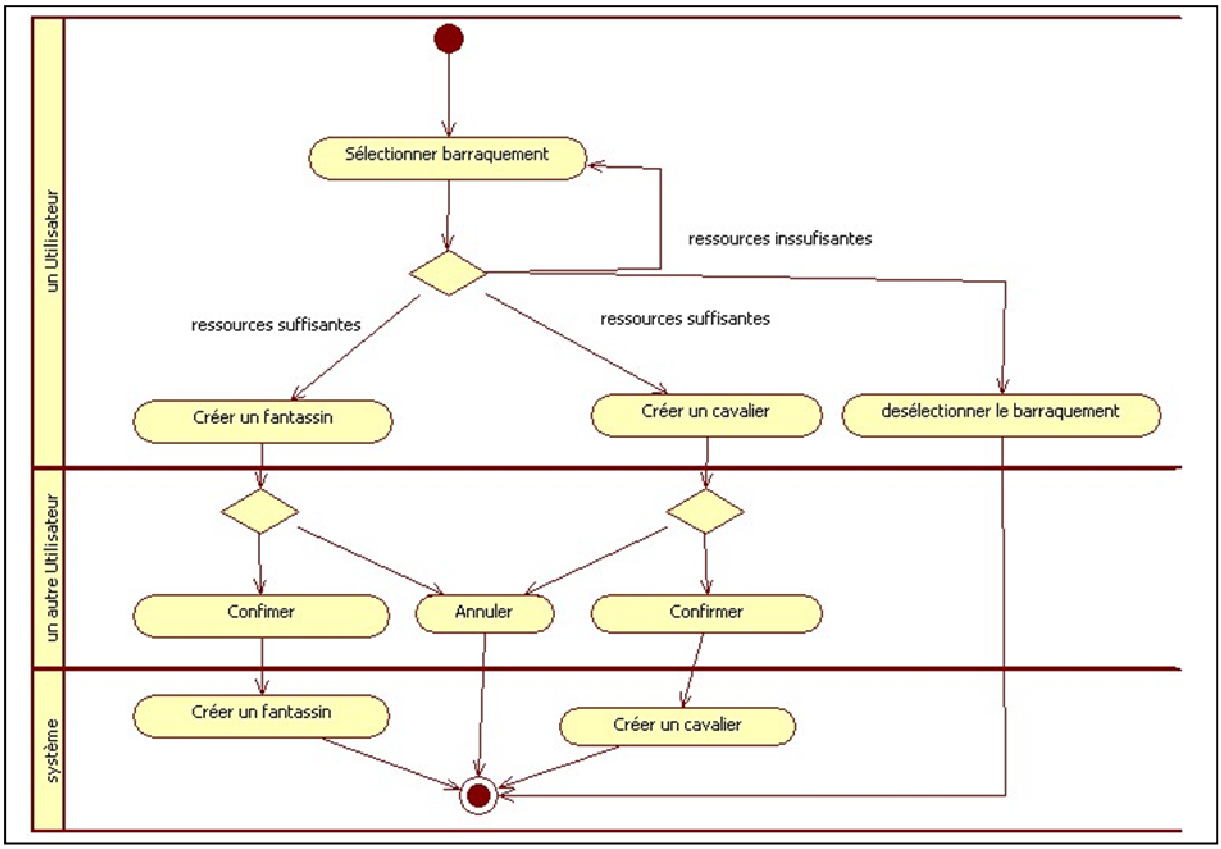


Figure 26 : Diagramme d'activité pour la gestion d'un baraquement.

Les trois diagrammes précédents permettent de décrire l'interaction des utilisateurs avec le système. Le diagramme de classe permet quant à lui de décrire les concepts du domaine de l'application, et les relations existant entre ces concepts. La Figure 27 décrit ainsi les concepts de WCCM. Elle introduit le concept d'*Unité* qui se spécialise en *Humain* ou *Bâtiment*. La relation de spécialisation est représentée par la flèche vide. Les unités appartiennent à un *Camp*, ce qui est représenté par la classe *Camp*, ainsi que la relation portant un losange entre les concepts d'*Unité* et de *Camp*. Cette relation indique qu'un *Camp* est relié à un ensemble d'unités. Le concept d'*Humain* se spécialise lui-même en concepts de *Soldat* et de *Paysan*. Les soldats peuvent être soit des *Cavaliers*, soit des *Fantassins*, et peuvent être rassemblés sous la forme d'un *Groupe*. Le diagramme décrit aussi les trois types de bâtiments qui spécialise la classe *Bâtiment* : l'*hôtel de ville*, le *Baraquement*, et la *Ferme*. Des attributs peuvent être spécifiés pour chaque concept à l'instar de l'attribut *Nom* pour un *Groupe*, ou *Couleur* pour un *Camp*. De manière similaire, des méthodes peuvent être spécifiées tels que la méthode *CreerUnPaysan* pour la classe *Hôtel de ville* ou encore la méthode *Frapper* pour la classe *Soldat*.

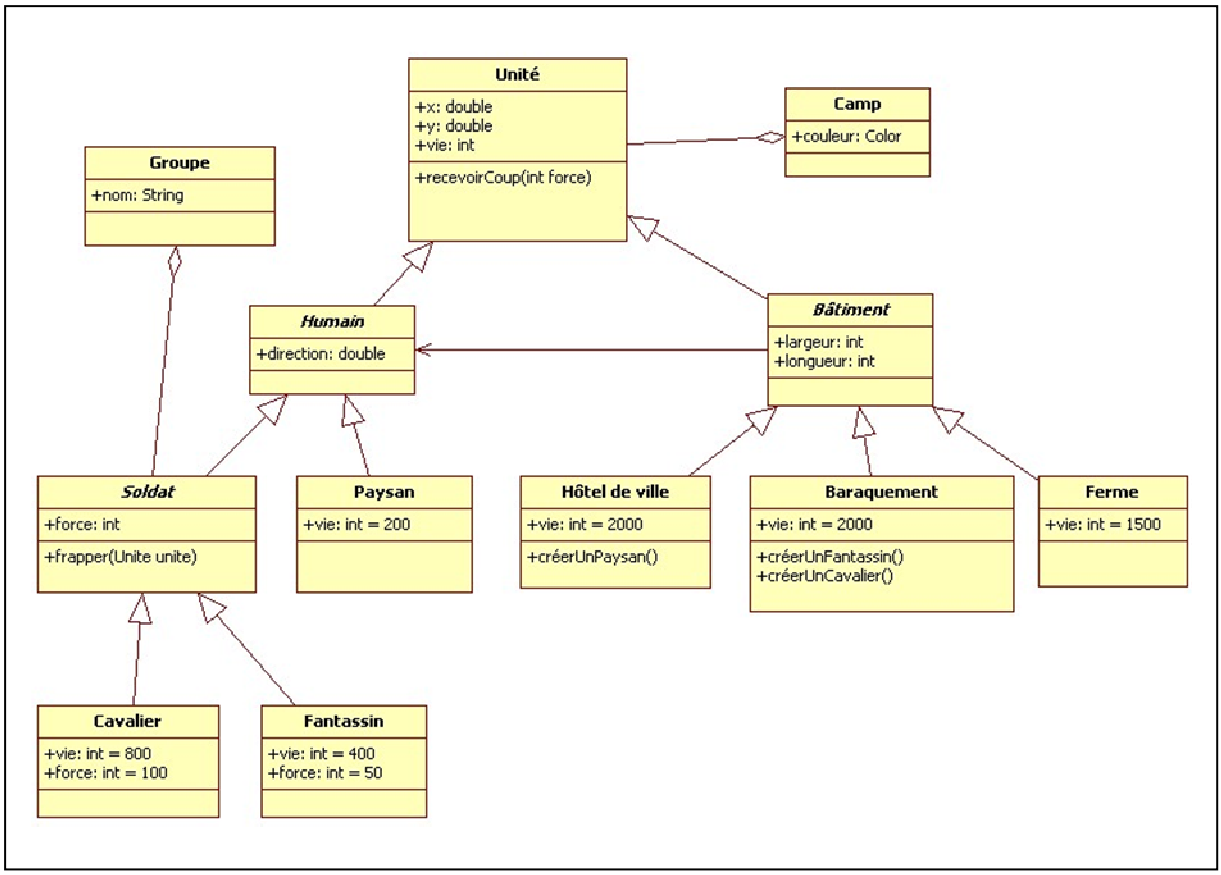


Figure 27 : Diagramme de classe UML.

Lorsque les concepts du domaine d'une application sont établis comme c'est le cas pour WCCM au sein de la Figure 27, il devient possible de décrire des diagrammes d'objets. Ce type de diagramme permet de décrire un état donné du système sous la forme d'un ensemble d'objets, instances des classes décrites au sein d'un diagramme de classe. Un état de l'application WCCM est ainsi décrit à laFigure 28. Elle présente deux objets *C1* et *C2* instances de la classe *Camp* et dont l'attribut *Couleur* est respectivement rempli par "rouge" et "bleu". Chaque camp est lié à un ensemble d'unités. Pour le camp rouge, il s'agit de deux objets, instances de la classe *Fantassin* : *F1* et *F2*, ainsi qu'un objet *Hôtel de ville* : *H1* et un objet *paysan* : *P1*. Pour le camp bleu, il s'agit d'un *Hôtel de ville* : *H2* et d'un *Paysan* : *p2*. Un groupe de soldats est également représenté par une instance de la classe *Groupe* : g1. Le nom de ce groupe est défini par l'attribut *Nom* de l'instance qui a la valeur "les vaillants".

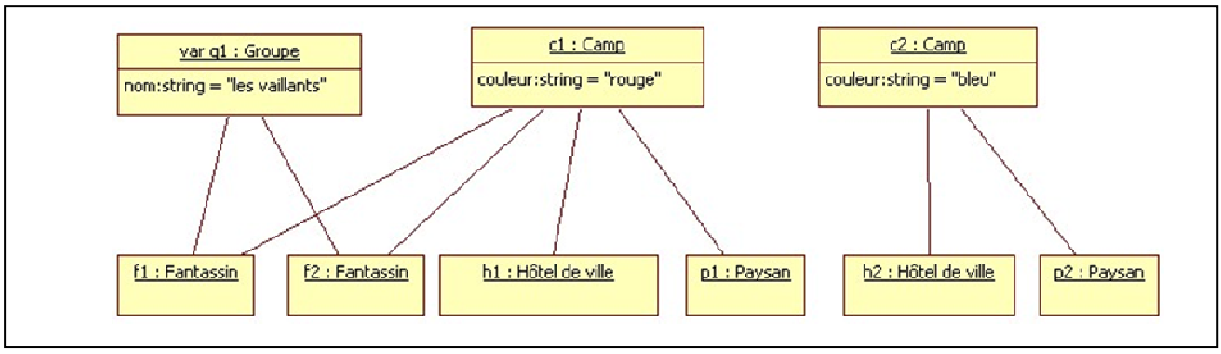


Figure 28 : Diagramme d'objet UML.

Sur la base des classes décrites dans le diagramme de classe, nous pouvons également décrire finement le comportement des instances de ces objets au sein du système grâce à des diagrammes d'état-transition. Ainsi, la Figure 29 illustre le comportement des objets interactifs de la classe humain. Dans ce diagramme d'état-transition, nous avons spécifié deux états qui correspondent à l'état sélectionné ou non d'un objet de type *Humain*. L'objet commence dans l'état *Désélectionné* à la suite de sa création. Ceci est décrit d'une part par un état initial représenté par un cercle plein, et d'autre part par une transition *Humain créé* en direction de l'état *Désélectionné.* Nous avons représenté plusieurs transitions sur ce diagramme correspondant aux différentes manipulations pouvant engendrer un changement d'état de l'objet. Ainsi, lorsque l'*Humain* est à l'état *Désélectionné* et qu'il est sélectionné, son état devient *Sélectionné*. Depuis ce deuxième état, l'objet peut être *Déplacé* ce qui ne le fait pas pour autant changer d'état. Quel que soit son état, l'objet peut être détruit, ce que représentent les deux transitions *Humain tué* vers un état final.

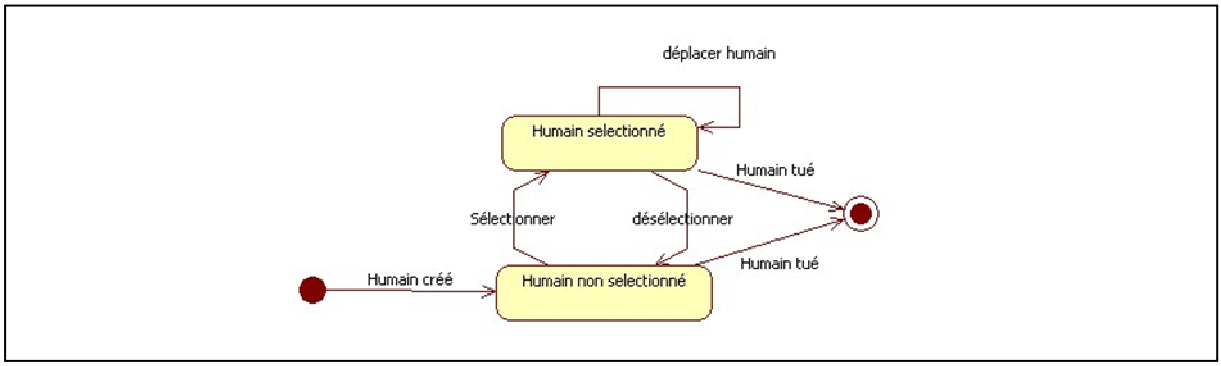


Figure 29 : Diagramme d'état-transition UML.

Le dernier diagramme qu'UML propose a été introduit avec UML 2. Il s'agit du diagramme global d'interaction. Il s'agit d'une forme particulière du diagramme d'activité dans lequel chaque nœud est lui-même un diagramme d'activité. L'intérêt de ce type de représentation est de monter en niveau d'abstraction pour décrire l'interaction, d'où le nom de ce type de diagramme. Les relations entre les nœuds sont les mêmes que pour un diagramme d'activité. La Figure 30 illustre un tel diagramme pour l'application WCCM. Elle décrit les quatre activités qui peuvent être réalisées : la gestion d'un baraquement qui est décrit à la Figure 24, la gestion d'un hôtel de ville qui permet la création de paysans, la gestion des soldats qui permet de les déplacer, de former des groupes et de déplacer ces groupes, ainsi que la gestion des paysans, qui permet de les déplacer, de récolter des ressources et de construire des bâtiments.

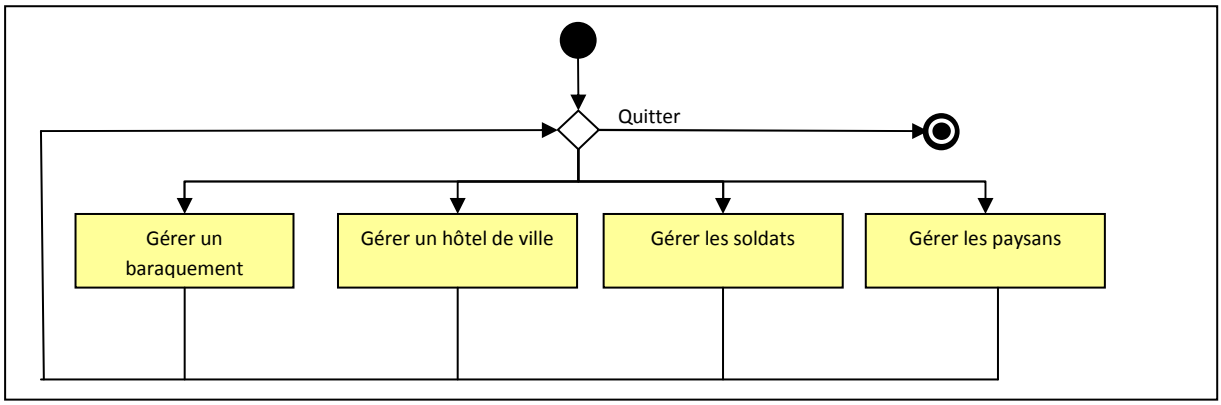


Figure 30 : Diagramme d'interaction de WCCM.

Les différents diagrammes UML permettent de décrire plusieurs aspects d'une application interactive comme l'illustrent les modélisations de l'application WCCM. Le comportement d'une application peut être décrit au sein d'un diagramme global d'interaction, et de diagrammes de cas d'utilisation. Le premier peut être détaillé par plusieurs diagrammes d'activités, tandis que les cas nominaux d'utilisation peuvent être décrits par des diagrammes de séquence ou de communication. En ce qui concerne l'interaction utilisateur sur un objet interactif en particulier, elle peut être décrite grâce à un diagramme d'état-transition.

La modélisation des objets du domaine sous forme de diagrammes de classe permet de fixer les concepts d'une application, qui peuvent ensuite être utilisés au sein des autres diagrammes, et notamment le diagramme d'objet qui permet de décrire un état du système en termes d'instances de classe. Les diagrammes que nous n'avons pas décrits ici traitent de l'aspect architectural et détaillé de l'application, et permettent par exemple pour le diagramme de composant de définir les composants logiciels à réaliser. L'allocation de ces composants logiciels au matériel peut ensuite être décrite au sein de diagrammes de déploiement.

Le formalisme UML n'est pas prévu pour décrire l'interaction concrète, ni l'interaction multimodale. Le point de vue adopté par le formalisme est celui du système : l'organisation et la description de la fonctionnalité priment sur la description de l'interaction concrète avec l'utilisateur. Il est cependant possible d'adapter un diagramme de classe pour décrire les dispositifs physiques disponibles, ou encore d'utiliser un type énuméré pour les primitives d'interaction en relation avec un dispositif.

De nombreuses notations que nous présentons dans la suite de ce mémoire se réfèrent à UML et se définissent comme complémentaires à UML. Ainsi, les auteurs de CTT [Paterno 1997] annoncent que la notation d'arbre de tâches CTT est complémentaire [Mori 2002] aux diagrammes de classe UML. De plus, les auteurs de CIAN [Parades 2008] intègrent CTT et UML au sein d'une méthode de conception centrée utilisateur, notée : CIAM (*Collaborative Interactive Applications Methodology*). Enfin, les auteurs de la notation Orchestra [David 2006] indiquent explicitement la complémentarité de leur notation avec les notations CTT et UML.

Conclusion sur UML

Les 13 diagrammes UML offrent un fort pouvoir d'expression pour spécifier un système et chaque diagramme UML pris isolément est simple. Néanmoins, maitriser la spécification globale en UML d'un système est difficile et nécessite de l'expérience. Heureusement, sa qualité de standard fait que de nombreux outils informatiques sont disponibles. Ils permettent d'éditer des spécifications, d'aider à la gestion de projet ou encore de générer du code à partir de spécifications produites.

UML est donc un outil à prendre en compte pour la spécification de collecticiels multimodaux, mais ne peut être utilisé seul, malgré les différents types de diagrammes disponibles. Il convient de se tourner vers des notations plus spécifiques aux approches de conception centrée utilisateur et vers des notations permettant d'exprimer l'interaction concrète.

2.1.2 UML-G

UML-G [Rubart 2002] [Rubart 2004] propose d'adapter le formalisme UML pour la conception de collecticiels par le biais de plusieurs stéréotypes. Un stéréotype est un mécanisme d'extension prévu au sein de la notation UML, qui permet d'enrichir les concepts initiaux de la notation par des

propriétés supplémentaires. Le premier est le stéréotype <<*Shared*>>. Son utilisation pour un objet ou une relation signifie que ce dernier peut être partagé lors d'une session collaborative. Un objet partageable possède des propriétés spécifiques : *{lockable}*, *{observable}*, *{distribution}, {access-controllable}, et {time-persistence}*. La propriété booléenne *{lockable}* permet d'indiquer qu'un objet est verrouillé par le premier utilisateur qui le manipule. Cette propriété peut être complétée par la propriété *{access-controllable}* qui permet d'indiquer qui peut avoir accès à l'objet. La propriété booléenne *{observable}* indique que l'objet est observable par tous. La propriété *{distribution}* indique l'état de distribution d'un objet. Elle peut prendre plusieurs valeurs telles que *asymmetric*, *semireplicated*, ou *replicated*. Enfin, la propriété *{time-persistence}* décrit la persistance temporelle des données, et peut être fixée comme *absolue* ou *relative*. Ensuite, UML-G propose les stéréotypes complémentaires : <<*SharedRole*>>, <<*SharedActor*>> et <<*SharedActivities*>> qui héritent du stéréotype <<*Shared*>> ; ils définissent les concepts de rôle, d'acteur et d'activité collaborative.

L'utilisation de ces stéréotypes au sein d'un diagramme de classe est illustrée à la Figure 31. Celle-ci correspond au diagramme de classe de la Figure 27 étendu par les stéréotypes <<*SharedRole*>> et <<*Shared*>>. Il décrit deux rôles participant à l'activité collaborative : le rôle de *Chef* et le rôle de *Sergent* qui sont marqués par le stéréotype <<*SharedRole*>>. Les classes *Unité* et *Groupe* sont marqués par le stéréotype <<*Shared*>> afin d'indiquer que les instances de ces classes sont partagées entre les utilisateurs. La propriété *{observable}* est spécifiée comme vraie pour ces deux classes afin d'indiquer que les deux utilisateurs peuvent voir les instances de ces classes. Enfin, les relations entre les rôles de *Chef* et de *Sergent* et les classes indiquent la capacité de ces rôles à manipuler ces objets. Ainsi, la relation entre le rôle de *Chef* et la classe *Groupe* indique qu'un Chef peut manipuler plusieurs groupes.

Bien que les stéréotypes puissent être utilisés au sein de tous les diagrammes UML, nous n'illustrons pas ici leur utilisation, pour éviter de dupliquer les spécifications UML de la section 2.1.1 de ce chapitre. Toutefois une utilisation intéressante du stéréotype <<*Shared*>> peut être de marquer les transitions d'un diagramme d'état-transition tel que celui de la Figure 29. Dans ce cas, la propriété *{observable}* à vrai ou faux permet d'indiquer l'observabilité des changements d'états pour les utilisateurs. De plus, pour les diagrammes d'activité, l'utilisation du stéréotype <<*SharedActivities*>> peut être pertinente pour différencier les activités individuelles des activités collaboratives. Ainsi, pour le diagramme d'activité de la Figure 26, l'ensemble des activités peuvent être marquées du stéréotype <<*SharedActivities*>>, puisque ce diagramme décrit l'activité collaborative de création d'un soldat.

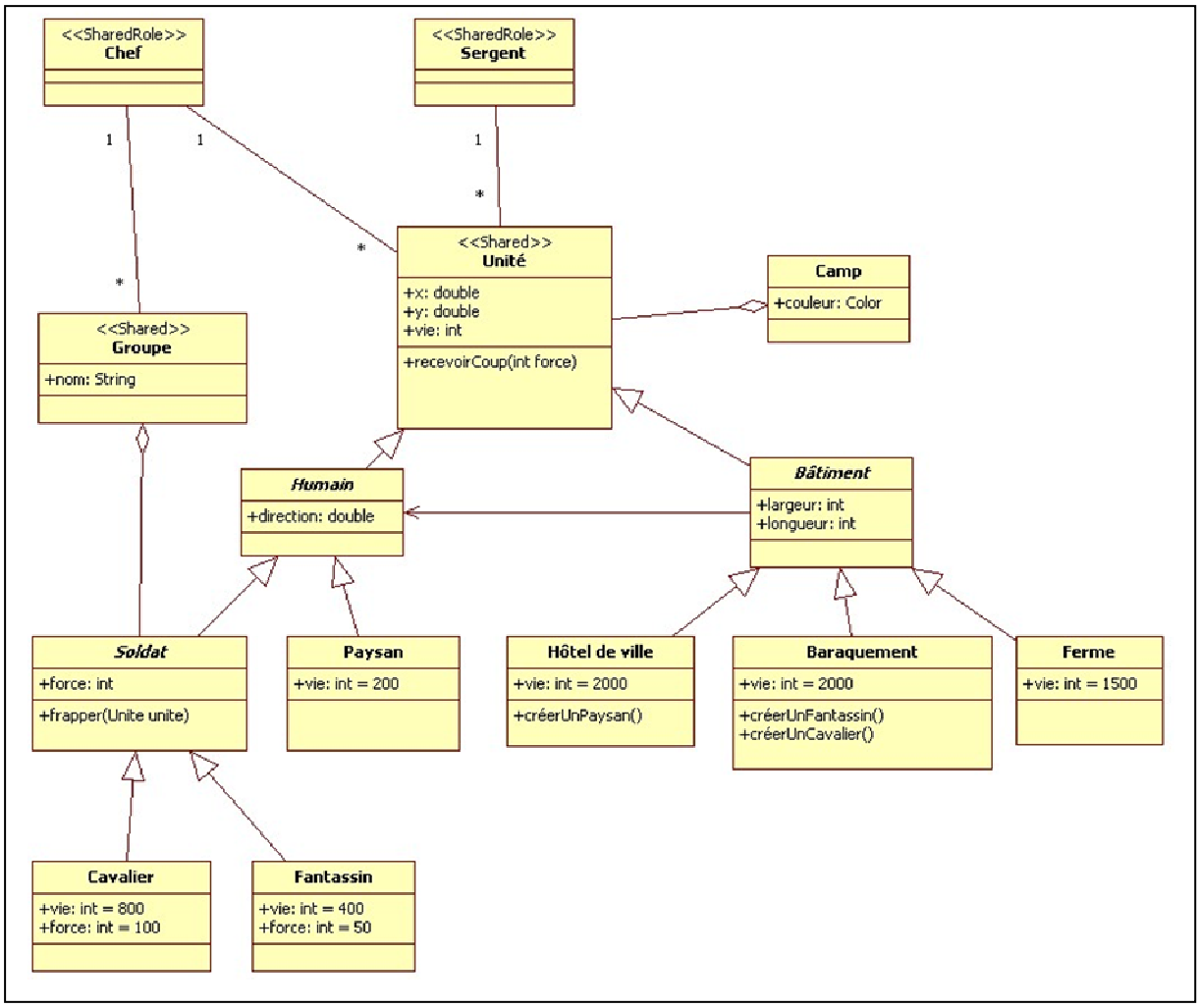


Figure 31 : Diagramme de classe UML-G.

Conclusion sur UML-G

L'extension UML-G pour le formalisme UML explicite des propriétés spécifiques aux systèmes multiutilisateurs. Les prototypes introduits peuvent s'appliquer au sein de tous les diagrammes UML ce qui permet de distinguer les éléments liés à des activités individuelles de ceux liés aux activités collaboratives. La propriété d'observabilité est tout particulièrement intéressante puisqu'elle permet de spécifier finement par exemple un WYSIWIG relâché, c'est-à-dire une observabilité partielle des changements d'état d'objets partagés. UML-G n'apporte en revanche rien de plus en ce qui concerne la dimension multimodale de l'interaction.

2.1.3 BPMN

BPMN (*Business Process Modeling Notation*) [BPMN] est un standard pour la description des processus métier dont la version 1.0 a été publiquement établie en 2004. La notation BPMN vise à décrire de manière compréhensible par différents acteurs les processus métier, afin de faciliter la réalisation de fonctionnalités logicielles. BPMN permet de décrire graphiquement des processus métier sous la forme de flots de données. Un modèle de processus métier est défini par des objets graphiques qui représentent les activités, reliés par un flot de contrôle qui définit l'ordre de

réalisation de ces activités. BPMN couvre principalement le contrôleur de dialogue du modèle ARCH [Bass 1992].

La notation est assez proche des diagrammes d'activités UML. Chaque activité peut être décomposée en sous activités de manière similaire à UML. Toutefois, BPMN offre des possibilités supplémentaires pour la décoration des activités et la description d'événements entre les activités. Ainsi, une activité BPMN peut être définie comme pouvant être répétée de manière séquentielle ou être réalisée plusieurs fois en parallèle. Les événements entre activités peuvent être des messages, des erreurs, une annulation, ou le déclenchement d'une minuterie (*timer*).

La notation peut être utilisée pour décrire des activités individuelles ou des activités coopératives. Pour ces dernières, à l'image des diagrammes d'activités UML, la distinction entre les utilisateurs est réalisée sur la base de rôle métier. Aussi, pour décrire l'activité collaborative de création d'un soldat de WCCM, la Figure 32 inclut deux pseudo-rôles : *un Utilisateur* et *L'autre utilisateur*. Ces pseudo-rôles nous permettent de décrire, sous une forme coopérative, une collaboration, qui n'est pas initialement prévue dans la notation. L'interaction s'effectue en trois étapes. Tout d'abord, un utilisateur peut *Sélectionner un baraquement*. Ensuite, il peut *Sélectionner un type de soldat*. Cette deuxième sélection est réalisée par une commande vocale, c'est pourquoi, une activité *Entendre le type de soldat* est définie pour *L'autre utilisateur*. Un message *Type de soldat* est précisé entre les deux activités pour dénoter le lien entre elles. *L'autre utilisateur* peut alors réaliser la troisième étape : *Confirmer la création*. Cette confirmation étant également réalisée oralement, une activité *Entendre la confirmation* est définie pour le premier utilisateur. Cet utilisateur peut également ne pas *Confirmer la création*, ce qui l'annule. L'annulation est spécifiée par le cercle contenant une croix à la Figure 32.

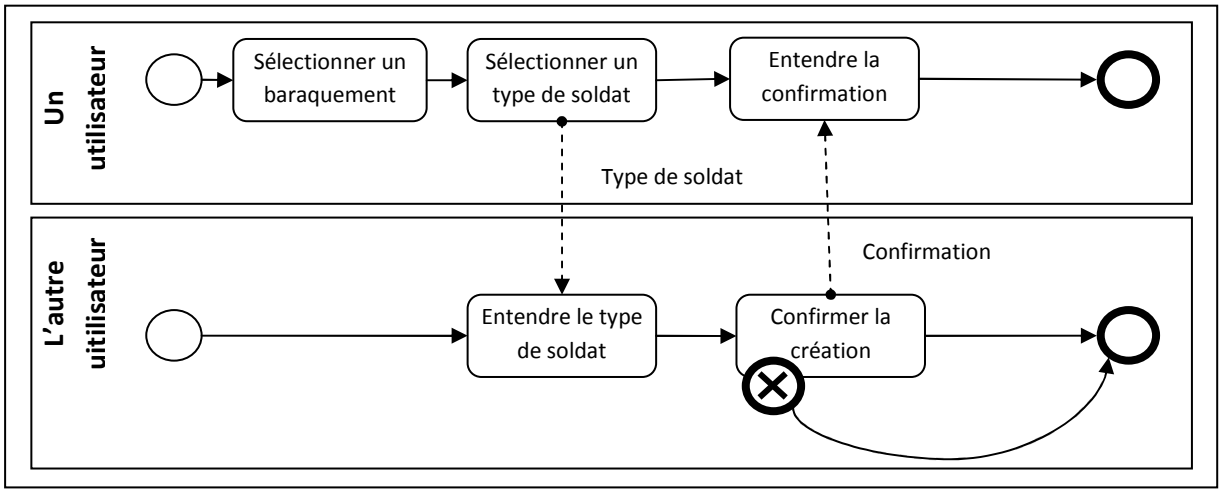


Figure 32 : Activité collaborative de création d'un soldat avec BPMN.

Conclusion sur BPMP

BPMN prend en charge la description de processus métier, sous une forme proche du diagramme d'activité d'UML. Nous avons montré qu'il est possible de décrire des activités coopératives en séparant et coordonnant les activités par rôles métier. La possibilité d'exprimer des messages entres activités y est pertinente puisqu'elle permet de préciser les relations de coopération entres les

utilisateurs. Enfin, BPMN se focalise sur la description du métier, ne se préoccupant pas de l'interaction concrète, et encore moins de l'interaction multimodale.

2.1.4 Synthèse

UML et BPMN sont deux standards utilisés pour le développement logiciel qui adoptent toute deux un point de vue système. De par son approche de modélisation des processus métier, BPMN se focalise sur les étapes d'analyse des besoins et de spécification du cycle de vie logiciel, tandis qu'UML a un fort pouvoir d'expression et concerne aussi les étapes de conception architecturale et de conception détaillée. L'extension UML-G précise la dimension multiutilisateur, sans pour autant étendre les capacités d'expression d'UML, mais en forçant le concepteur à se poser des questions supplémentaires, telles que le partage des objets, leur répartition et leur observation par les utilisateurs, des caractéristiques importantes des collecticiels (Chapitre 1).

2.2 Domaine : Interaction Homme-Machine

Cette section est consacrée aux notations issues du domaine de l'Interaction Homme-Machine. Ces notations de spécification d'applications interactives incluent des concepts variées, issus de la Psychologie Cognitive, de l'Ergonomie, et du Génie Logiciel.

Bien que certaines notations prennent en compte la dimension multiutilisateur, toutes sont initialement prévues pour décrire l'interaction homme-machine mono-utilisateur. Ces notations permettent de réaliser des descriptions de haut niveau d'abstraction telles que des modèles de tâches pour CTT ou K-MAD, ou de décrire finement l'interaction concrète comme pour UAN ou IOG.

2.2.1 CTT

CTT (*Concurrent Task Tree*) est une notation d'arbre de tâches, destinée à l'origine pour l'analyse des besoins et la spécification de l'interaction de systèmes mono-utilisateur [Paterno 1997]. La notation a ensuite été étendue et outillée pour prendre en compte les activités coopératives [Mori 2002]. CTT couvre les composants de contrôleur de dialogue et d'interaction abstraite du modèle ARCH [Bass 1992].

Les activités coopératives sont décrites en utilisant plusieurs arbres de tâches : un arbre pour décrire les activités coopératives et un arbre de tâches individuelles par rôle identifié. Un arbre de tâches coopératives contient des tâches coopératives et des tâches individuelles de haut niveau d'abstraction. Ces dernières sont détaillées au sein des arbres de tâches individuelles jusqu'au niveau des tâches concrètes. Pour WCCM, nous avons considéré un arbre de tâches coopératives (Figure 33) et deux arbres de tâches individuelles pour les rôles de S*ergent* (Figure 34 A) et de *Chef* (Figure 34 B). Au sein de ces arbres, nous avons détaillé uniquement la tâche coopérative de gestion des soldats.

La notation CTT introduit cinq types de tâches : les tâches individuelles, les tâches abstraites qui doivent être décomposées en sous-tâches, les tâches système, les tâches mentales et les tâches coopératives. Ces dernières sont des tâches abstraites dont la décomposition fait apparaître des tâches individuelles associées à plusieurs rôles. Les relations entre les tâches sont définies par des opérateurs hérités de LOTOS. Par exemple, dans les Figure 33 et Figure 34, nous utilisons les opérateurs suivants : la concurrence (|||), l'activation (>>), l'alternative ([]) et un opérateur unaire pour l'itération (*).

Les tâches coopératives ne doivent pas apparaître au sein des arbres de tâches individuelles. Aussi, nous avons maintenu la cohérence entre l'arbre de tâches coopératives et les arbres de tâches individuelles, en utilisant des tâches abstraites en lieu et place des tâches coopératives, comme l'illustre la tâche *Gérer les soldats* des Figure 33 et Figure 34.

La notation CTT ne prend pas en charge la collaboration entre les utilisateurs, aussi, les tâches collaboratives de gestion des bâtiments et de paysans ne peuvent être décrites précisément, c'est pourquoi, nous ne les avons pas détaillées ici.

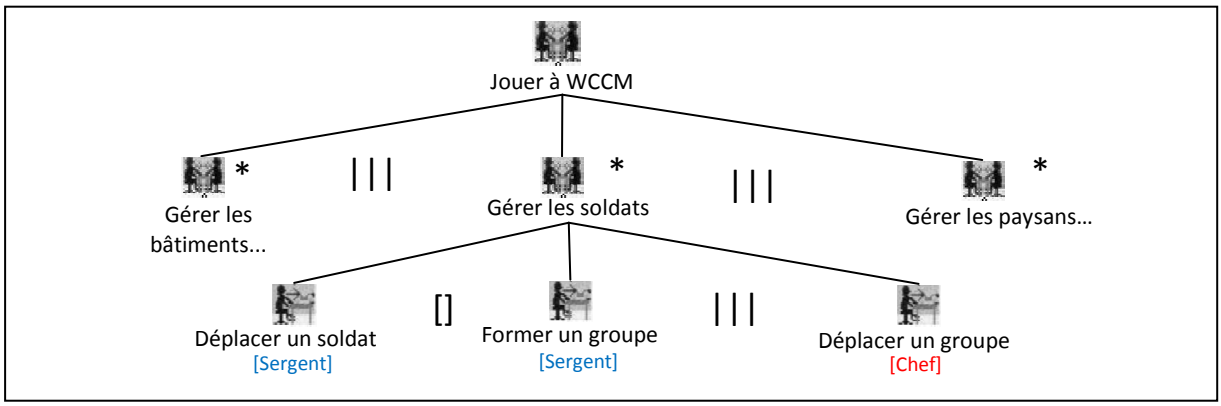


Figure 33 : Arbre de tâches cooperatives CTT.

L'intérêt de la distinction entre arbre de tâches coopératives et arbre de tâches individuelles réside également dans le niveau utilisé pour décrire ces deux types d'arbres : tandis que l'arbre de tâches coopératives doit décrire les tâches à un haut niveau d'abstraction, c'est-à-dire sans décomposer les tâches individuelles qu'il contient, les arbres de tâches individuelles peuvent en revanche aller jusqu'au niveau des tâches concrètes. C'est le cas notamment pour la tâche *Former un groupe*, qui n'est pas détaillée dans l'arbre de tâches coopératives de la Figure 33, et que nous avons décomposée en deux sous-tâches séquentielles au sein de la Figure 34 A : une tâche *Sélectionner des soldats* et une tâche *Indiquer le nom du groupe*. De la même manière, à la Figure 34 B la tâche *Déplacer un groupe* associé au rôle de *Chef* est composée de trois sous-tâches : *Sélectionner un groupe*, *Donner l'ordre de mouvement* et *Indiquer la destination*.

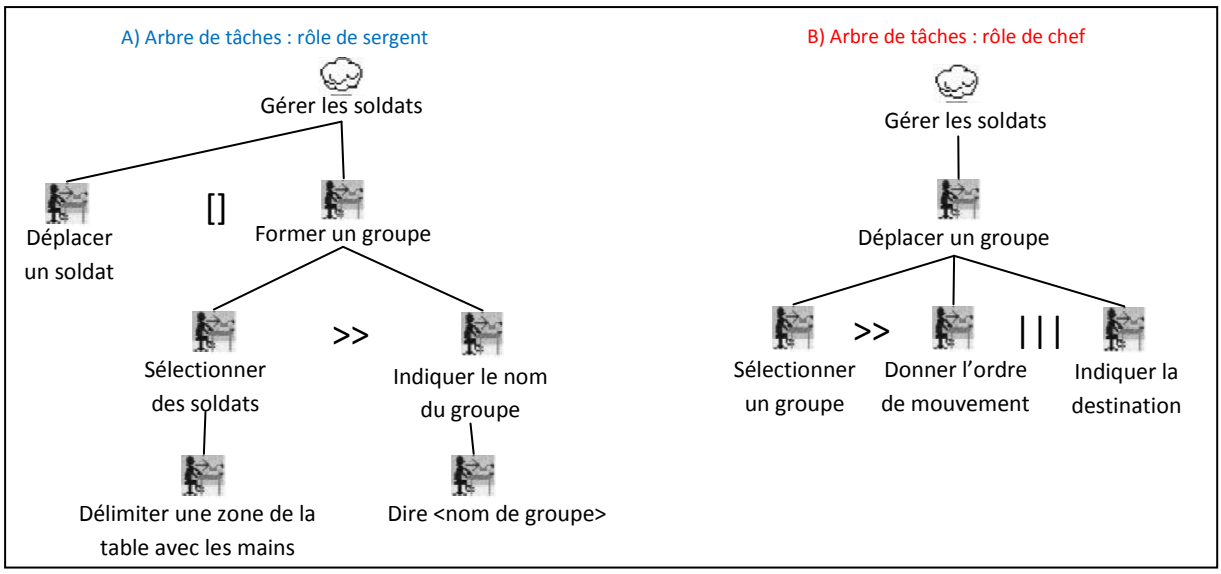


Figure 34 : Arbres de tâches individuelles CTT pour le Sergent (A) et pour le Chef (B).

La notation CTT n'offre pas de moyen spécifique pour décrire l'interaction concrète, en termes de dispositifs ou de modalités d'interaction. Cependant, il est possible de la définir partiellement par le nom d'une tâche, par exemple : *Délimiter une zone de la table avec les mains*, ou *Dire <nom de groupe>,* qui indique respectivement et explicitement l'usage de la table augmentée et d'un microphone, ainsi que d'une modalité gestuelle et de commandes vocales.

Conclusion sur CTT

CTT permet de décrire de manière précise l'interaction abstraite entre les rôles utilisateur et le système interactif du point de vue des utilisateurs. Son objectif est de décrire le modèle de tâches d'un collecticiel, et donc l'ensemble des interactions possibles entre les utilisateurs et le système. Pour cela, CTT est parfaitement adapté grâce à ces différents types de tâches et la décomposition logique hiérarchique précisée par les nombreux opérateurs temporels. Le point de vue centrée utilisateur de CTT vient compléter le point de vue centrée système qu'adopte une notation tel qu'UML. Enfin, et comme nous l'avons explicité et illustré, l'interaction concrète et l'interaction multimodale ne sont pas traitées par la notation CTT, même si la description de dispositifs est possible au sein du nom des tâches.

2.2.2 K-MAD

K-MAD (*Kernel of Model for Activity Description*) repose sur le modèle N-MDA (Noyau du modèle de description de l'activité) [Lucquiaud 2005] et a été élaboré par la comparaison et l'extraction de caractéristiques communes entre plusieurs modèles de tâches existants : Diane+ [Tarby 2001], MAD [Scapin 1989], CTT [Paterno 1997] et GTA [Veer 2000], et leurs outils respectifs (Tamot, IMAD, CTTE et Euterpe). Ses concepts de bases pour la modélisation des tâches sont : les utilisateurs, les rôles, les objets abstraits (et concrets = instances) et les événements. K-MAD couvre ainsi l'interaction abstraite et le contrôleur de dialogue du modèle ARCH [Bass 1992]. K-MAD est également un formalisme qui permet de décrire l'ensemble de ces aspects. Le formalisme repose principalement sur cinq facettes :

1. Un modèle des tâches,
2. Un modèle des objets abstraits,
3. Un modèle des objets concrets,
4. Un modèle des rôles et des acteurs,
5. Un modèle d'événement.

Le formalisme K-MAD dispose d'un outil d'édition K-MADe (*K-MAD environment*)[Baron 2006] qui prend en charge chacune de ces facettes.

Pour la description du modèle des tâches, K-MAD propose de distinguer quatre types de tâches : abstraite, utilisateur, système et interactive. A l'instar de la notation CTT, les tâches abstraites sont des tâches qui sont décomposées en sous-tâches des autres types.

La structure d'un arbre de tâches K-MAD est proche de celle d'un arbre de tâches CTT. Ainsi l'arbre de tâches K-MAD de la Figure 35 est semblable à l'arbre de tâches coopératives de CTT de la Figure 33. Cependant, contrairement à CTT, la notation K-MAD impose que la composition de sous-tâches

s'effectue avec un opérateur unique. Ainsi, la tâche *Gérer les soldats*, qui était décomposée avec CTT en trois sous-tâches reliées par un opérateur d'alternative et un opérateur de concurrence, est spécifiée en K-MAD avec une tâche abstraite supplémentaire, que nous avons appelée également *Gérer les unités*, et qui regroupe les tâches *Déplacer un soldat* et *Former un groupe*.

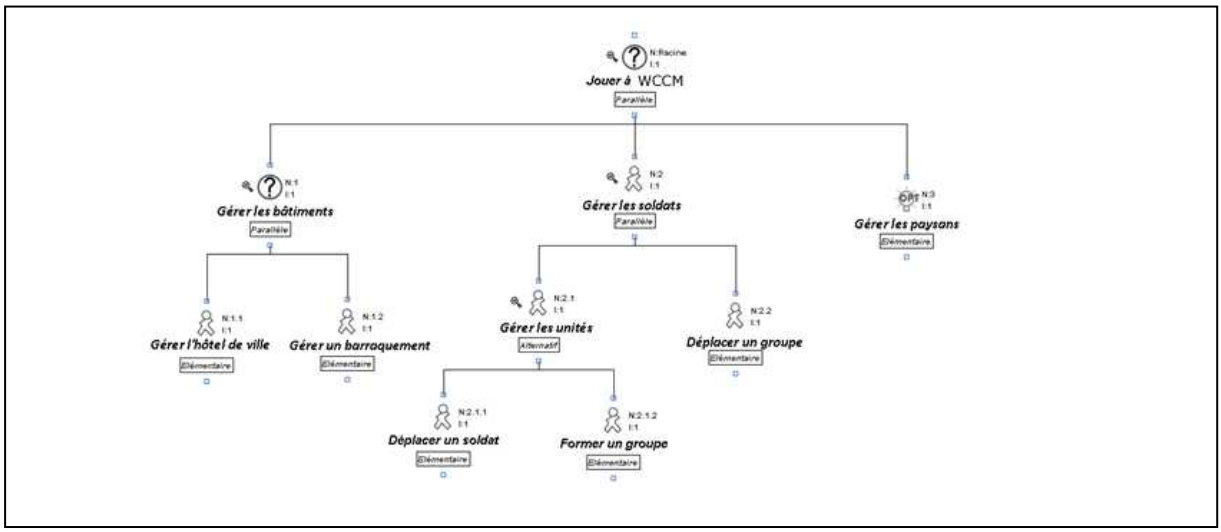


Figure 35 : Arbre de tâches K-MAD de WCCM.

Le modèle de tâches K-MAD ne fait pas apparaître explicitement les rôles et les acteurs qui réalisent les tâches. Ceci est dû au choix de décrire les aspects complémentaires à l'arbre de tâches au sein d'autres représentations. En effet, pour chaque tâche élémentaire représentée, K-MAD préconise d'écrire une fiche descriptive comportant : les acteurs et les rôles associés à la tâche, le but de la tâche, les pré-conditions et les post-conditions de la tâche, ainsi que les objets abstraits et concrets manipulés. L'ensemble de ces concepts est ensuite décrit au sein d'une représentation spécifique en marge de l'arbre de tâches. C'est le cas notamment des rôles et des acteurs qui sont modélisés au sein d'un tableau.

L'outil K-MADe est un éditeur dédié à cette notation. Aussi, la représentation des objets se fait sous forme tabulaire. Néanmoins, sans employer cet outil, il est possible de modéliser les objets sous la forme de diagrammes de classe UML, qui ont la même sémantique que les tableaux de K-MADe.

K-MAD propose également de décrire des événements déclencheurs pour des tâches. Ceux-ci peuvent activer une tâche, et peuvent être définis lors de la conception. Ils peuvent prendre comme support la description des objets ou des tâches.

Conclusion sur K-MAD

La méthode K-MAD est intéressante car elle propose de décrire le lien entre les tâches et les concepts tels que les rôles et les objets (abstraits et concrets). Elle permet également de décrire de manière précise la décomposition d'une activité coopérative ou individuelle au travers de l'arbre de tâches. Toutefois les concepts liés aux tâches ne sont pas explicitement visibles au sein de l'arbre de tâches. Au travers des objets concrets qui peuvent être modélisés avec K-MAD, nous avons la possibilité de décrire des dispositifs physiques, afin de faire le lien entre ces dispositifs d'interaction et les tâches, et donc les rôles. Ce formalisme nous permet donc d'exprimer pour chaque rôle les dispositifs d'interaction en entrée et en sortie correspondant. En revanche, K-MAD ne propose pas

de moyen pour décrire la multimodalité, comme l'usage complémentaire ou redondant (propriétés CARE). En effet, les opérateurs logiques et temporels proposés sont de trop haut niveau d'abstraction pour décrire finement, par exemple, la complémentarité de deux modalités d'interaction, dans une fenêtre temporelle, pour une tâche donnée.

2.2.3 TaskMODL

La notation TaskMODL est l'une des notations proposées par Treatteberg [Treatteberg 2002]. Chaque notation permet la représentation d'un aspect d'un système multiutilisateur. Les auteurs proposent ainsi trois types de diagramme :

- RML (*Referent Modelling Language*), similaire aux diagrammes de classe UML, qui permet de décrire les concepts du domaine, les instances, les ensembles et les relations entre ceux-ci ;
- TaskMODL (*Task MODeL*) qui est dédié à la description du modèle de tâches ;
- DiaMODL (*Dialog MODel*) qui permet de décrire de manière détaillée le dialogue concret au niveau des interacteurs.

DiaMODL a fait l'objet de plusieurs publications, notamment dans le cadre d'une méthode [Traetteberg 2008b] visant à décrire le modèle de tâches à l'aide de DiaMODL et du langage de description de processus métier BPMN (*Business Process Modeling Notation*) [BPMN]. La notation DiaMODL dispose d'un outil logiciel de spécification contrairement à RML et TaskMODL [Traetteberg 2008a].

Ces notations sont combinées pour décrire plusieurs aspects des systèmes multiutilisateurs selon quatre étapes de modélisation :

1. Les données du domaine (RML),
2. Les rôles et leurs relations (RML),
3. Les tâches utilisateur (TaskMODL + RML),
4. L'interaction concrète en termes de
    a. comportement (DiaMODL)
    b. et de présentation (DiaMODL + RML + maquettes).

Nous illustrons les différents aspects traités par les notations RML, TaskMODL et DiaMODL en appliquant les étapes de description dans l'ordre préconisé (1, 2, 3, 4) sur l'application interactive WCCM.

La première étape consiste à modéliser les concepts de l'application WCCM à l'aide de RML, comme le montre la Figure 36. La structure de ce diagramme est la même que celle du diagramme de classe UML de la Figure 27. Cependant, lors des relations d'héritages décrites un symbole $\otimes$ indique une exclusion entre les sous-concepts. Par exemple à la Figure 36un *Bâtiment* ne peut être également un *Humain*. D'autre part, le symbole $\oplus$ indique une agrégation. Ce dernier est d'ailleurs utilisé pour la relation entre *Soldat* et *Groupe* à la Figure 36.

A partir des concepts, et à l'instar du diagramme d'objet UML, il est possible de représenter des instances de ces concepts. RML fournit un moyen d'intégrer ces deux types de représentation, comme le montre l'élément *Hôtel_de_ville_1* de la Figure 36.

RML permet également de décrire des diagrammes d'état-transition. Comme ils sont semblables à ceux qui peuvent être décrits en UML, nous ne revenons pas sur ce type de description.

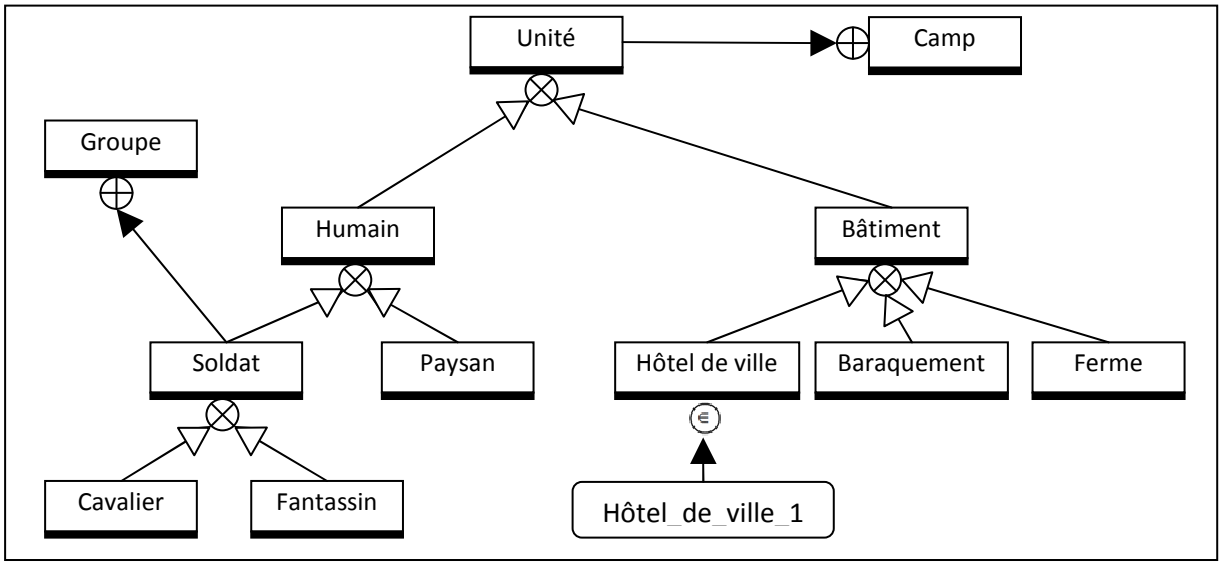


Figure 36 : Modèle des concepts de WCCM avec RML.

La deuxième étape de modélisation consiste à décrire les rôles, les acteurs et les relations qui les unissent. Cette modélisation est décrite avec la notation RML, comme l'illustre la Figure 37. Elle décrit deux rôles : *Sergent* et *Chef* joué par deux utilisateurs : *Utilisateur 1* et *Utilisateur 2*. Les deux rôles sont la spécialisation du rôle d'*Utilisateur*.

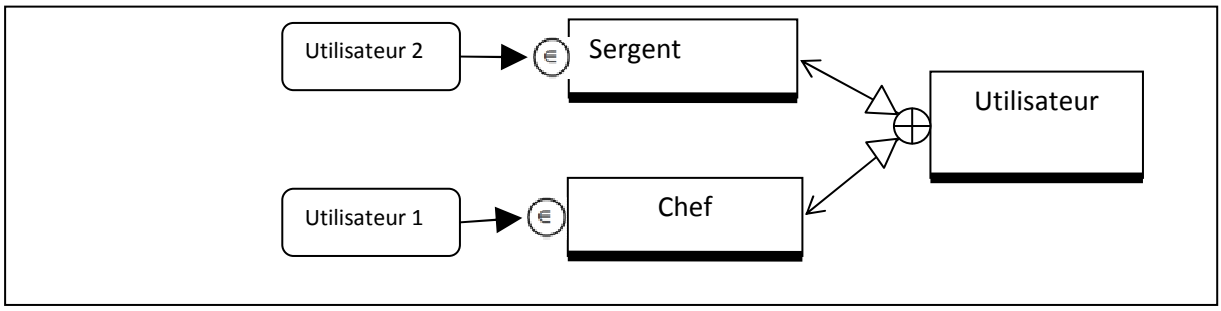


Figure 37 : Modèle des rôles et utilisateurs de WCCM avec RML.

La troisième étape focalise sur la description des tâches utilisateur à l'aide de la notation TaskMODL sous la forme d'une structure arborescente. Chaque tâche est représentée par une boîte dans laquelle figurent :

1. Le nom de la tâche,
2. la description textuelle de la tâche,
3. le ou les symboles RML qui représentent le ou les acteurs, rôles, ou groupes qui réalisent la tâche,
4. le diagramme RML décrivant les objets manipulés par la tâche.

Pour une meilleure lisibilité, la Figure 38 ne comporte que le nom de la tâche et les rôles associés. La composition de plusieurs sous-tâches est décrite par un unique opérateur, parmi séquence, parallélisme, ordre quelconque et choix. Des pré-conditions et post-conditions peuvent être ajoutées

à la représentation sous la forme de diagrammes RML incluant éventuellement des diagrammes d'état.

L'arbre de tâches TaskMODL de l'application WCCM est représenté à la Figure 38. Celui-ci est quasiment identique à l'arbre de tâches K-MAD de la Figure 35. La différence se situe principalement dans la description des pré- et post-conditions directement au sein d'arbre de tâches. Par exemple, la pré-condition *A* pour la tâche *Gérer les troupes* explicite la nécessité de disposer d'une instance d'un *Soldat*. La post-condition *B* à la tâche *Former un groupe* indique que la tâche produit une instance de *Groupe*. Avec cette notation, il est possible d'avoir une même condition qui soit à la fois pré- et post-condition d'une tâche. C'est le cas de la pré-condition B qui est également une post-condition *B*. Ces conditions sont exprimées au sein du même schéma à l'aide de la notation RML. Dans notre cas, nous les avons décrites juste en dessous pour des raisons de place.

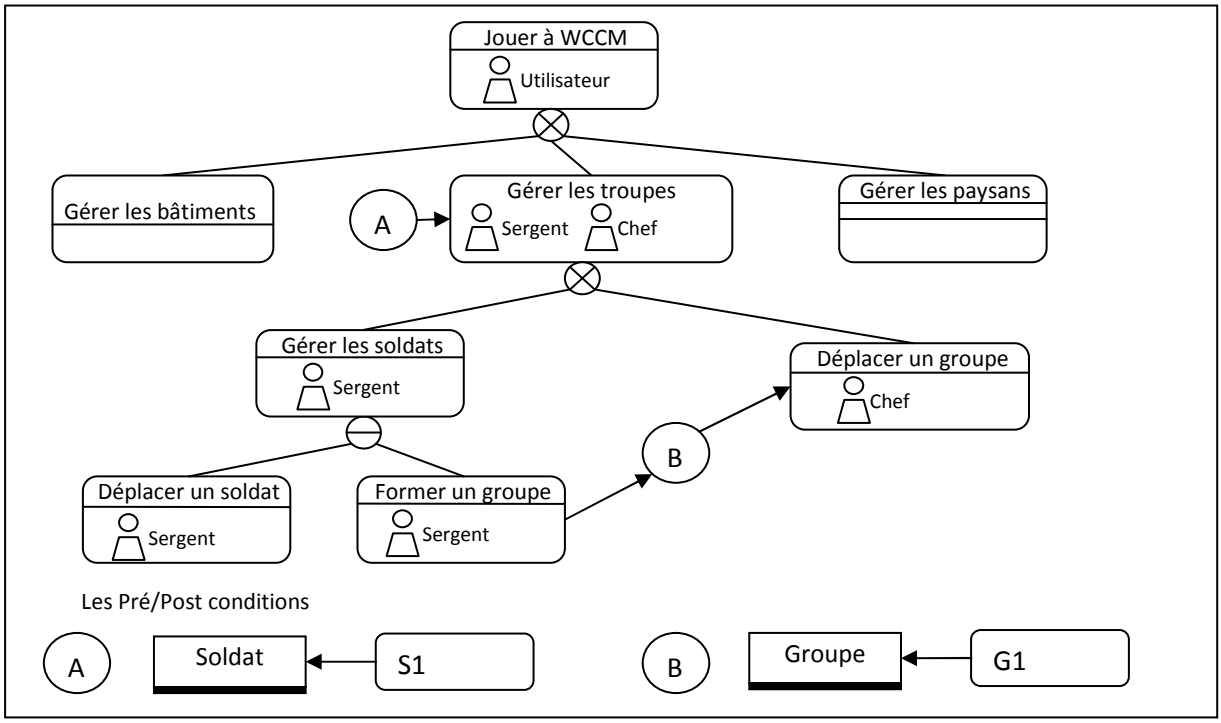


Figure 38 : Modèle de tâche TaskMODL de WCCM.

La dernière phase de conception consiste à décrire l'interaction concrète en termes de présentation et de comportement.

La présentation est décrite par une maquette telle que celle de la Figure 39. Celle-ci inclut les trois zones de jeu : la carte sur laquelle évoluent les unités, la mini-carte qui fournit une vue d'ensemble et un panneau d'information contextuelle. Le panneau principal est la carte, c'est pourquoi c'est le seul qui est détaillé ici. Celui-ci contient des éléments constituant l'interface graphique de l'application : les unités humaines de types fantassin et paysans, les bâtiments de type ferme, hôtel de ville et baraquement, ainsi que les éléments de ressources tel qu'une mine d'or ou des forêts.

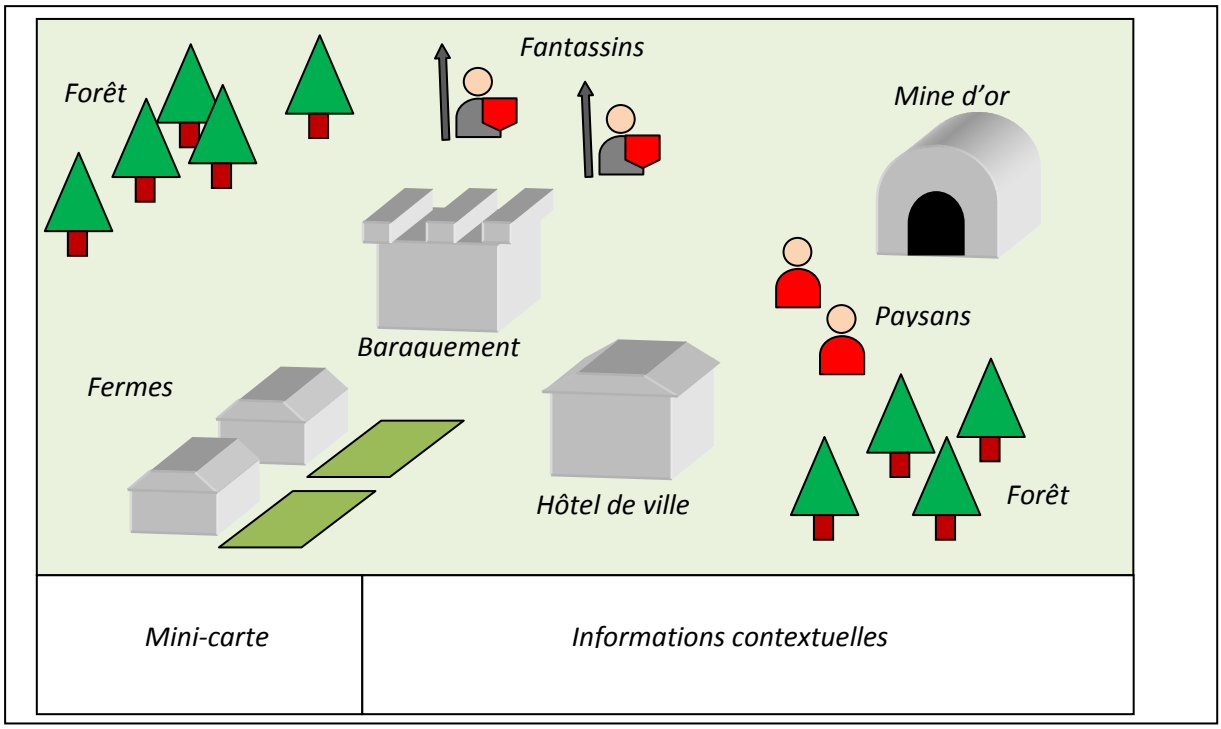


Figure 39 : Modèle de l'interface graphique de WCCM.

Pour décrire le comportement de l'interface concrète, plusieurs possibilités sont offertes par la notation. La première possibilité consiste à décrire les interacteurs qui composent l'interface utilisateur au moyen de diagrammes d'état-transition. L'utilisation de diagramme d'état-transition est illustrée à la Figure 29, lors de la modélisation avec UML, aussi nous n'illustrons pas à nouveau cette approche.

La seconde méthode consiste à utiliser la notation DiaMODL. Celle-ci permet de décrire les interacteurs au sein d'une représentation de flot de données. La Figure 40 illustre l'utilisation de DiaMODL pour la tâche de déplacement d'un *Fantassin*. Un utilisateur jouant le rôle de *Sergent* peut déplacer un *Fantassin* sur la *carte* grâce à l'utilisation conjointe du microphone et de la table augmentée. La prise en compte de l'ordre de marche est indiquée par un message diffusé dans le casque du joueur, puis, le soldat se met en marche, ce qui se traduit par un déplacement du fantassin sur la carte, observable sur la table augmentée.

Toutefois, la description de la Figure 40 est partielle et il convient de noter que, par souci de clarté les éléments suivants ne sont pas décrits :

- Une modélisation du rôle de *Chef*. En effet, ce dernier voit les effets des ordres de déplacement sur les soldats.
- Une modélisation du fantassin sous forme d'un diagramme d'état-transition, afin de rendre perceptible le changement d'état de celui-ci et l'impact de ces changements d'état sur les dispositifs de sortie.

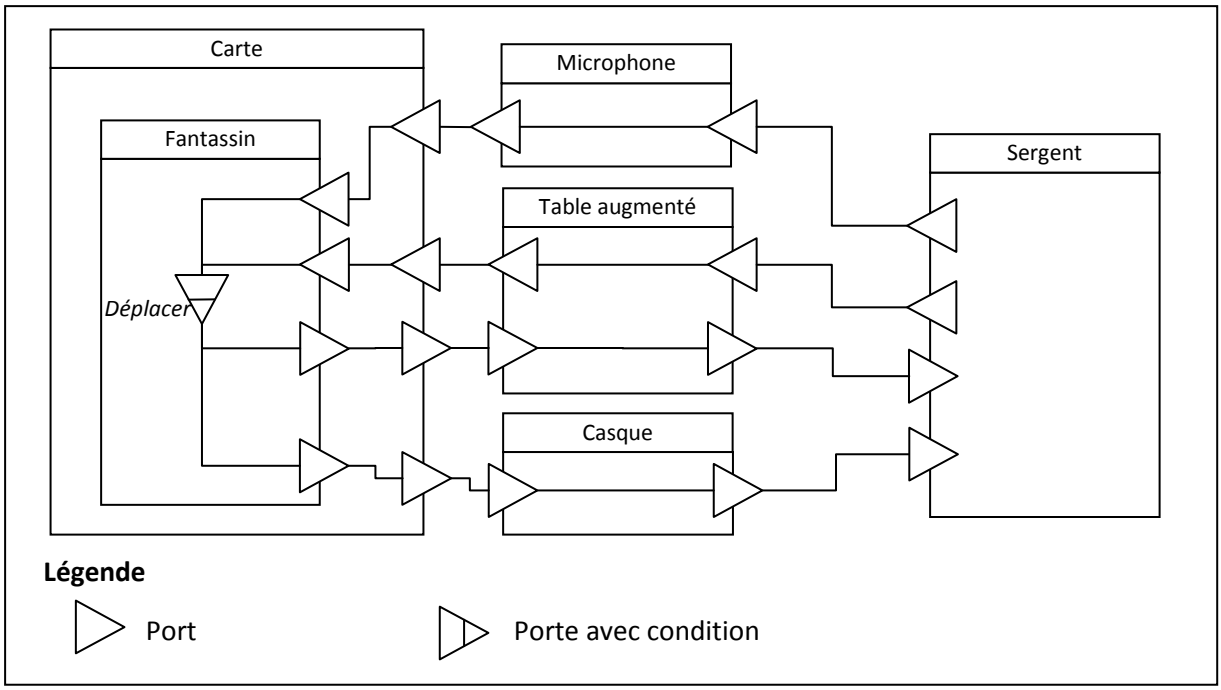


Figure 40 : Modélisation du dialogue avec DiaMODL entre un utilisateur et le système pour le déplacement d'un fantassin dans WCCM.

Conclusion sur TaskMODL/RML/DiaMODL

La combinaison des notations TaskMODL/RML/DiaMODL permet la description de plusieurs aspects des systèmes multiutilisateurs :

1. Les concepts du domaine (objets du domaine et dispositifs physiques, ou éléments du système),
2. les relations entre les rôles, les groupes et les acteurs à l'aide d'un sociogramme,
3. un modèle de tâches hiérarchique muni de pré- et post-conditions,
4. une description de l'interaction concrète en termes de présentation (maquette) et de comportement (diagramme d'état et modélisation du dialogue avec les interacteurs).

Les descriptions que permettent la combinaison des trois notations couvrent l'ensemble des composants du modèle ARCH, du noyau fonctionnel à l'interaction concrète.

Ces notations sont intéressantes car elles offrent de nombreux moyens pour décrire plusieurs facettes qui composent un système interactif multiutilisateur en capturant à la fois les aspects statiques et dynamiques de l'interaction et de l'activité de groupe. Les notations TaskMODL, RML, et DiaMODL peuvent s'intégrer les unes aux autres au sein d'une représentation unique. Ainsi, par exemple il est possible de décrire des diagrammes DiaMODL et RML au sein d'un arbre de tâches pour exprimer des pré-conditions et des post-conditions. Si l'intégration des notations est en soit intéressante, l'usage d'un diagramme pour décrire une pré-condition n'est pas toujours adapté du fait de la taille des diagrammes résultants. Notons aussi, que parmi les trois notations, seule la notation DiaMODL est supportée par un outil d'édition.

Enfin, le cas de la multimodalité n'est pas explicitement pris en charge. Toutefois, de la même manière que pour la notation CTT, il est possible d'exprimer l'usage d'une modalité au sein du nom des tâches.

2.2.4 CTML

La notation CTML (*Collaborative Task Modeling Language*) est intégrée dans la plate-forme de spécification d'environnement collaboratif que proposent les auteurs de [Wurdel 2008a, Wurdel 2008b, Wurdel 2008c]. Cette plate-forme inclut un outil logiciel de modélisation et de simulation de spécification CTML. L'approche proposée consiste à intégrer les descriptions du domaine reposant sur UML, et les descriptions de l'interaction reposant sur CTT.

Précisons pour commencer que le terme collaboratif utilisé par les auteurs correspondant au terme coopératif comme défini au Chapitre 1 section 1.3. Aussi, et par soucis de clarté, nous utilisons le terme coopération dans la suite de cette section pour décrire les possibilités offertes par la notation CTML.

La notation CTML définit deux types de modèles que le concepteur doit décrire :

- Un modèle du domaine qui décrit les concepts de l'application, dont les rôles utilisateurs, ainsi que les relations entre ces concepts,
- un modèle de tâches coopératives reposant sur les rôles décrits dans le modèle du domaine.

Pour cela, les auteurs proposent de réutiliser deux notations existantes plutôt qu'une nouvelle notation. Il s'agit d'utiliser un diagramme de classes pour modéliser le domaine et d'utiliser la notation CTT étendue pour modéliser les tâches, c'est-à-dire : un arbre de tâches coopératives, et un arbre de tâches individuelles pour chaque rôle.

L'utilisation d'un diagramme de classe UML pour modéliser le domaine a été traitée dans la section consacrée à la notation UML. Toutefois, les auteurs préconisent de décrire les rôles utilisateurs au sein du diagramme de classe. Pour cela, ils décrivent une classe *Rôle* dont les autres rôles utilisateur héritent. Les relations entres les concepts de rôles et les concepts de l'application sont décrites avec les relations fournies pour le diagramme de classe UML.

Pour la description du modèle de tâches, les auteurs de [Sinnig 2007] introduisent plusieurs extensions pour la notation CTT. Premièrement, deux nouveaux opérateurs unaires sont ajoutés : (#) qui dénote la capacité de réaliser une tâche donnée plusieurs fois en parallèle et (stop) qui dénote la terminaison de la tâche mère. Deuxièmement, l'opérateur binaire d'alternative ([]) est étendu afin de signifier que le choix est déterministe ([]D) ou non déterministe ([]N) du point de vue de l'utilisateur. Un choix est déterministe pour l'utilisateur si celui-ci peut effectuer ce choix. Au contraire un choix non déterministe pour l'utilisateur est un choix effectué par le système. La Figure 41 reprend l'arbre de tâches coopératives CTT de la Figure 33, et l'adapte selon la notation CTML ce qui le rend plus précis. Nous utilisons l'opérateur itération concurrente (#) pour la tâche coopérative *Gestion des troupes*. Celle-ci peut en effet être menée de front, à la fois par le *Sergent* et par le *Chef*. Nous utilisons l'ancien opérateur d'itération séquentielle (*) sur les sous-tâches individuelles *Gérer les soldats* et *Déplacer un groupe*. Pour la tâche *Déplacer un groupe*, cela signifie qu'un seul groupe peut être déplacé à la fois. La distinction entre itération séquentielle et itération parallèle nous apparaît comme très utile pour décrire des interactions simultanées de plusieurs utilisateurs. Nous utilisons

dans l'arbre de tâches coopératives l'opérateur de choix déterministe ([]D) entre les tâches *Déplacer un soldat* et *Former un groupe* pour indiquer que le choix revient à l'utilisateur.

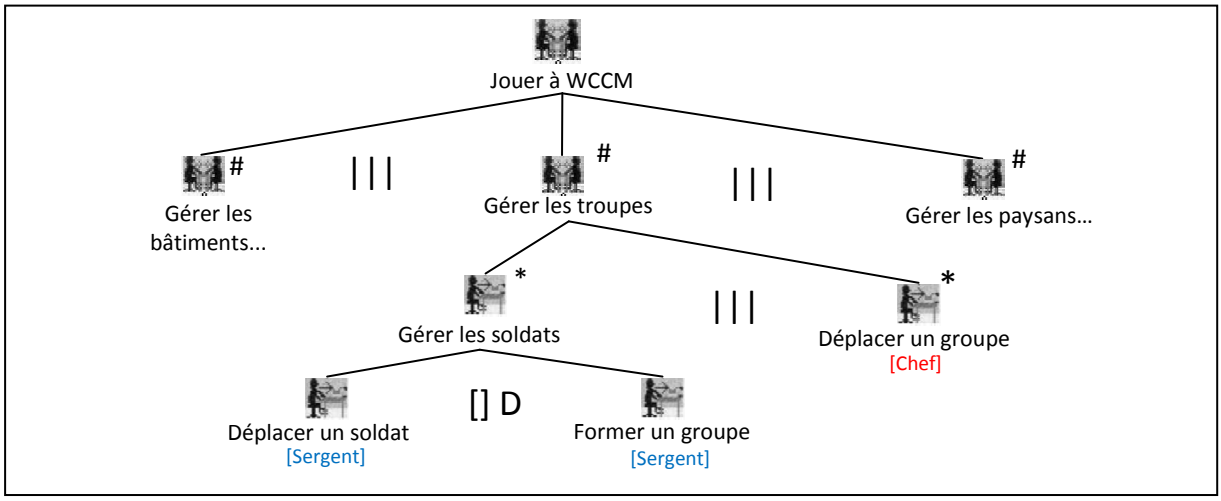


Figure 41 : Arbre de tâches coopératives de l'application WCCM avec CTML.

Nous ne décrivons pas ici les arbres de tâches individuelles avec CTML, car ils sont proches de ceux décrits avec la notation CTT présentée à la section 2.2.1 de ce chapitre. En effet, l'intérêt des extensions est souligné au travers de la modélisation de l'arbre de tâches coopératives.

Un point qui n'est pas visible sur les modélisations est la possibilité d'exprimer des pré-conditions et des effets des tâches à partir de la description des concepts du domaine. Ainsi, il est possible d'exprimer que l'existence d'un *Groupe* est nécessaire pour la tâche *Déplacer un groupe* de la Figure 41, sous la forme de *Groupe.oneInstance* si une classe *Groupe* est décrite dans le modèle du domaine.

Conclusion sur CTML

La notation CTML repose sur l'intégration des notations UML et CTT dont l'intérêt et les limites ont été soulignés dans les sections 2.1.1 et 2.2.1 de ce chapitre. L'enrichissement des opérateurs CTT apporté par CTML permet de décrire avec plus de précision les relations entre les tâches dans un contexte multiutilisateur. De plus, CTML permet de décrire les relations entre concepts et tâches en termes des objets du domaine. Enfin, l'outil de spécification et de simulation permet de réaliser l'analyse à priori des spécifications produites.

2.2.5 Web Task Model

WebTaskModel [Bomsdorf 2007] est une approche de spécification de processus métier dédiée aux applications web et reposant sur l'utilisation d'un modèle de tâches hiérarchique similaire aux notations CTT ou K-MAD. L'outil *WTM Simulator* permet de simuler les modèles de tâches produits avec la notation WebTaskModel [Bomsdorf 2009] grâce à une transformation préalable des modèles de tâches en machines à états. La simulation sert à identifier au plus tôt des éventuels défauts d'utilisabilité.

Les applications web sont la plupart du temps des collecticiels, comme les ERP dans les entreprises ou les réseaux sociaux pour les particuliers. Il nous a donc semblé important de présenter cette

notation qui s'attache à les décrire spécifiquement. Cependant, l'application que nous avons choisie pour illustrer les différentes notations n'est pas une application web. Aussi, nous illustrons cette notation à la Figure 42 en considérant une application d'agence de voyage issue de [Bomsdorf 2007].

La notation WebTaskModel permet de décrire l'interaction d'une application web au travers d'une représentation arborescente. WebTaskModel comporte 4 types de tâches présents dans la notation CTT : les tâches coopératives, les tâches individuelles, les tâches systèmes et les tâches abstraites. A l'instar de la notation K-MAD, les opérateurs entre tâches sont situés directement au niveau de la décomposition. WebTaskModel introduit un opérateur de séquence particulier : *SeqB* utilisé pour la tâche *Payment*. Cet opérateur indique la capacité pour l'utilisateur de revenir en arrière. Les retours aux pages précédentes sont classiques dans les applications web, mais l'opérateur peut également être intéressant pour d'autres types applications. En effet, un operateur de ce type peut s'imaginer pour signifier qu'un utilisateur annule une tâche, ou pour indiquer la capacité pour l'utilisateur de corriger une erreur. Le dernier point saillant de cette notation est sa capacité à décrire la répétition d'une tâche sous la forme d'une multiplicité comme c'est le cas pour les diagrammes de classes d'UML. La multiplicité d'une tâche est comprise entre une borne inférieure et supérieure. Ainsi par exemple, le *0-1* de la tâche *Alter shipping data* dénote le caractère optionnel de cette tâche puisqu'elle peut être réalisée entre zéro et une fois.

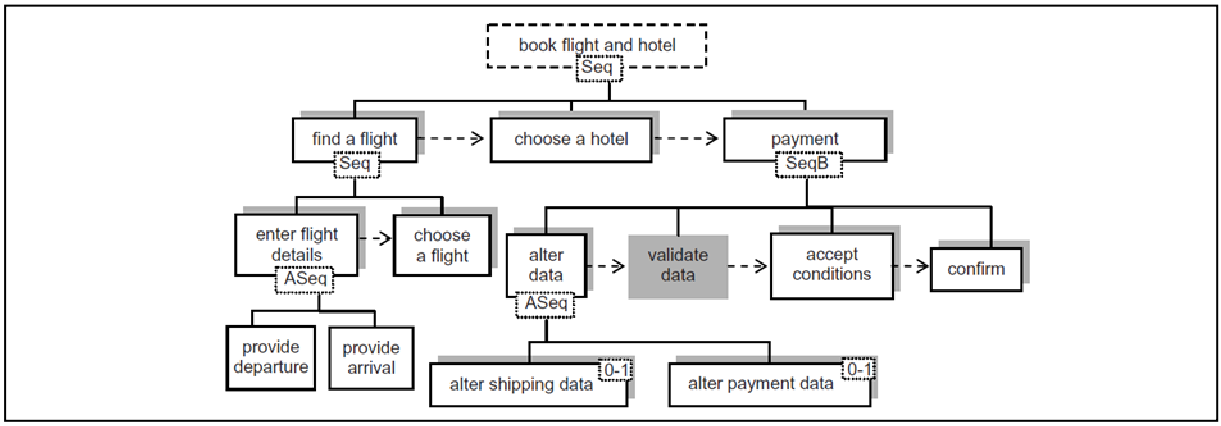


Figure 42 : Description du site web d'une agence de voyage avec WebTaskModel [Bomsdorf 2007].

Enfin, la notation permet de décrire les objets, rôles ou contextes pour une tâche donnée. Ces descriptions peuvent être faites sous la forme de diagrammes de classe et d'objet UML, et d'annotations pour les rôles.

Conclusion sur WebTaskModel

WebTaskModel est un outil adapté à la modélisation d'applications web. Au sein d'arbre de tâches proches de ceux de CTT ou K-MAD, la dimension multiutilisateur est traitée par l'utilisation de rôles utilisateurs associés aux tâches. L'opérateur de séquence avec retour est intéressant car il peut être envisagé dans d'autres types d'applications, pour indiquer la capacité du système à permettre la correction des erreurs par l'utilisateur. En ce qui concerne l'interaction concrète et multimodale, le prédicat d'une telle notation est l'exécution au sein d'un navigateur web qui restreint donc les capacités d'interaction. Aussi, les modalités d'interaction ne sont pas décrites avec WebTaskModel.

2.2.6 SpIeLan

La notation SpIeLan (*Staged Participatory multimedia events Interface LANguage)* est dédiée à la modélisation d'événements multimédia participatifs [Bergh 2006, Bergh 2007]. Pour cela, elle propose de décomposer ces événements participatifs en scène, et de décrire chaque scène individuellement. Pour chaque scène, les rôles des utilisateurs sont décrits ainsi que les actions qu'elles/ils peuvent entreprendre. Pour cela, SpIeLan utilise trois modèles différents : un modèle de scénario, un modèle de présentation, et un modèle de comportement. Par le biais de ces trois représentations, SpIeLan propose de décrire principalement l'interface utilisateur concrète, en termes de présentation et de comportement, et le dialogue homme-machine en termes de changements d'état de l'interface.

La modélisation de la présentation de l'interface utilisateur concrète passe par la description de maquettes. Au sein de celles-ci, il s'agit d'une part de définir des zones d'affichage qui servent de repère pour l'implémentation, mais également de déterminer les types d'interaction possibles qui sont accessibles aux utilisateurs par le biais de cette interface. Pour ce deuxième point, SpIeLan intègre les CAN (*Canonical Abstract Prototype*) [Constantine 2003] et en propose une extension [Bergh 2006]. Ces prototypes sont des représentations d'interacteurs couramment utilisés pour les interfaces homme-machine WIMP. Il s'agit de symboles qui peuvent être combinés pour modéliser les interacteurs, leur contenu, et les actions utilisateur qu'ils permettent.

La Figure 43 illustre l'utilisation des *Canonical Abstract Prototypes* dans le cadre de la modélisation de l'interface concrète de l'application WCCM. L'interface graphique décrite correspond à un état de l'application où la vue des joueurs est centrée sur une zone comprenant quelques unités, dont des soldats et un baraquement. Le baraquement est sélectionné par le *Chef*. Un groupe constitué de plusieurs soldats est également visible.

La partie haute de la figure représente l'interface graphique de la table interactive, qui comporte une vue sur la carte (zone en haut), une mini-carte (zone en bas à gauche), et un panneau contextuel (zone en bas à droite). Sur la carte, les éléments numérotés de 1 à 5 représentent des soldats. L'élément de numéro 6 correspond au groupe constitué par les soldats 3, 4 et 5. L'élément de numéro 7 est le baraquement. Le symbole associé aux éléments de 1 à 7 indique que ces éléments peuvent être sélectionnés et déplacés.

Dans le panneau contextuel est également représenté l'élément numéro 7, qui représente le fait que le bâtiment est sélectionné. Ce panneau inclut les éléments 8 et 9 qui correspondent à l'icône des types d'unité que le baraquement peut produire (*Fantassin* ou *Cavalier*). Le symbole indique qu'il s'agit d'une icône de choix dans une liste, bien que le choix soit réalisé par l'intermédiaire d'une commande vocale.

Dans la partie inférieure de la Figure 43, sont décrits les rôles utilisateurs : *Sergent* et *Chef*. Une liste de CAN munis d'une étiquette numérotée est associée à chaque rôle. Cette liste de CAN représente le lien entre les éléments de l'interface et les rôles. Ainsi, les éléments 7, 8 et 9 correspondant au baraquement et aux icônes de type de soldat à produire sont associés au rôle de *Chef* afin de décrire le fait que c'est le *Chef* qui a sélectionné le baraquement.

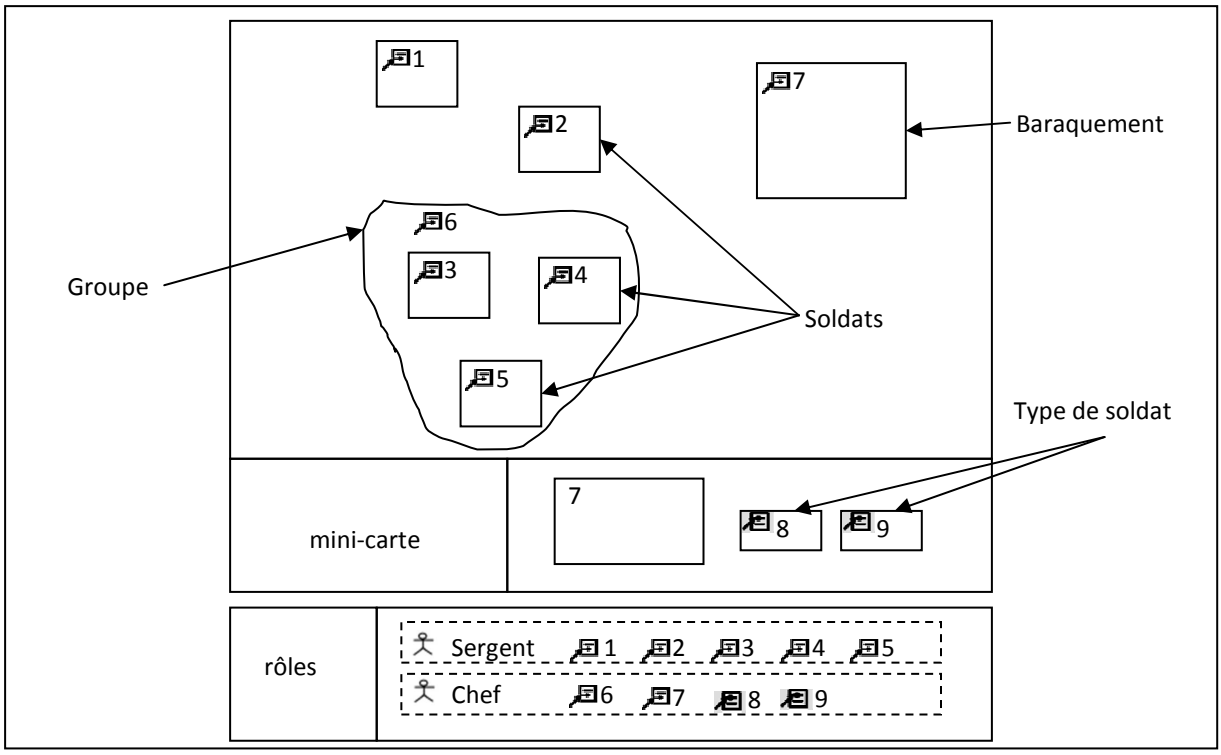


Figure 43 : Modèle SpIeLan de l'interface de l'application WCCM (avec les *Canonical Abstract Prototypes*).

A partir de la description de l'interface, SpIeLan propose de décrire les interactions que peuvent réaliser les utilisateurs de manière plus détaillée. Ces descriptions sont représentées sous la forme d'un tableau tel que celui de la Figure 44. Il comporte trois parties principales : la première décrit le nom de la tâche : déplacer un groupe ; la deuxième décrit les données manipulées (en entrée et en sortie de la tâche) ; la troisième partie décrit les événements système que génère l'utilisateur au cours de la tâche (découpé en trois niveaux : Début, Fin, et Sortie de la tâche). Pour les données manipulées, nous avons représenté en entrée la présence du groupe 6, et le fait que le groupe se déplace. Les événements en entrée sont : au début, le fait de prononcer le nom du groupe pour le sélectionner, puis une combinaison du pointage du doigt et de l'assignement de l'ordre de déplacement. En sortie, un son est produit pour signifier la prise en compte de l'ordre. Prévu pour décrire l'interaction WIMP, les CAN doivent être étendus pour prendre en charge l'usage de nouvelles modalités d'interaction telles que la parole ou le geste.

Déplacer un groupe			
Donnée	Entrée		Sortie
	6		6 se déplace
Événements	Début	Fin	Fin
	Prononcer le nom du groupe	Pointer du doigt + Prononcer "Aller ici"	Son : "oui, seigneurs"

Figure 44 : Description de la tâche déplacer un groupe avec SpieLan.

Une fois que l'ensemble des interactions avec le système sont décrites, SpIeLan propose de faire le lien entre ces différentes interactions au sein d'un diagramme d'activité UML. Ce type de diagramme

étant décrit à la section 2.1.1 de ce chapitre, et la notation SpIeLan n'y apportant pas de modification, nous ne revenons pas sur ce type de modélisation de SpIeLan.

Conclusion sur SpIeLan

La notation SpIeLan s'intéresse principalement à la modélisation de l'interface utilisateur concrète. Ceci est rendu possible grâce à la description de l'interface par le biais de maquettes graphiques en s'appuyant sur une version étendue des *Canonical Abstract Prototype* et sur une description tabulaire des tâches utilisateurs en termes d'opérations élémentaires réalisées par l'utilisateur sur l'interface. L'utilisation des diagrammes d'activité UML permet de mettre en perspective ces interactions fines sur le système en les replaçant dans le cadre de l'utilisation complète du système. La maquette munie des *Canonical Abstract Prototype* est particulièrement intéressante dans le cadre des systèmes multiutilisateurs puisque : (1) elle permet de décrire la répartition des objets sur une ou plusieurs surfaces d'interaction ; (2) elle permet de décrire l'association de tâches et d'objets à un utilisateur. En revanche, SpIeLan ne prend pas en compte la dimension multimodale de l'interaction notamment car les *Canonical Abstract Prototype* sont dédiés aux interactions classiques WIMP (souris, clavier).

2.2.7 UAN

UAN (User Action Notation) [Hartson 1990] permet de décrire l'interaction concrète sous la forme de tableaux. Cette notation a été redéfinie pour devenir NUAN (New User Action Notation) [Venema 1999] puis intégrée au sein de la notation GTA [Veer 2000] qui est présentée à la section 2.3.1 de ce chapitre.

Pour chaque tâche d'un système interactif, UAN permet de décrire la ou les actions utilisateurs tels que des clics souris, les réactions du système, tels que l'affichage d'une icône sélectionnée, et des changements d'état du système tels que la sélection d'un fichier. La syntaxe de la notation UAN étant très proche de celle de NUAN, nous reportons le lecteur à la section 2.3.1 dédiée à NUAN.

2.2.8 IOG

Les diagrammes d'état IOG (*Interactions Object Graphs*) [Carr 1994] s'intéressent à décrire le dialogue concret entre l'utilisateur et le système. La notation repose sur les diagrammes d'état-transition comme en UML, ainsi que sur la syntaxe d'UAN [Hartson 1990]. IOG introduit deux nouveaux types d'état. Le premier type d'état permet de décrire l'état d'un objet, par exemple, les coordonnées d'un soldat sur la carte. Le deuxième type d'état permet d'inclure au sein d'un *State Sharts* des représentations de l'interface graphique. La Figure 45 reprend le diagramme d'état-transition UML de la Figure 29 concernant l'état d'un soldat. Celui-ci peut avoir deux états : *Sélectionné* et *Non sélectionné*. Avec IOG, nous représentons ces deux états avec l'icône d'un soldat non sélectionné à gauche, et sélectionné à droite (un cercle vert entoure le soldat), plutôt que textuellement.

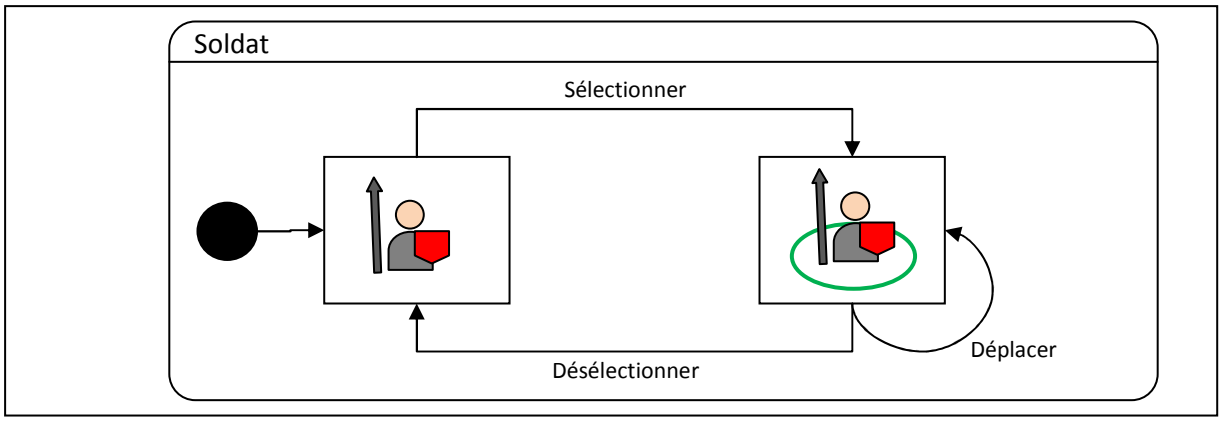


Figure 45 : Modélisation du comportement d'un soldat avec IOG.

2.2.9 Synthèse

Les notations traitées dans cette section sont issues du domaine de l'Interaction Homme Machine.

Parmi ces notations, CTT, K-MAD, TaskMODL et WebTaskModel utilisent un arbre de tâches pour décrire l'interaction à haut niveau d'abstraction jusqu'aux tâches élémentaires. Ce type de représentation est particulièrement intéressant car elle structure la modélisation. En effet, contrairement à un diagramme d'activité UML qui impose d'avoir clairement identifié les tâches élémentaires, un arbre de tâches permet de décrire d'abord l'interaction à haut niveau d'abstraction, telle que la coopération, puis par décompositions successives d'aboutir à des tâches élémentaires.

Les tâches élémentaires de ces arbres de tâches représentent des primitives d'actions sur les concepts métier du domaine. Ces actions peuvent être décrites en détail au sein de représentations complémentaires telles que DiaMODL ou UAN. En revanche, ces deux notations ne prennent pas en compte des modalités d'interaction telles que l'usage de la parole ou des gestes et sont dédiées uniquement à l'interaction de type WIMP.

Bien que la plupart des notations considérées ne soient pas spécialement conçues pour décrire des applications multiutilisateurs, plusieurs d'entre elles proposent de décrire ce type d'activité au travers d'arbre de tâches coopératives (CTT, K-MAD et TaskMODL), en s'appuyant sur les concepts de rôle et d'acteur. La question de la collaboration n'est en revanche traitée par aucune des notations présentées.

Pour conclure, les notations présentées dans cette section permettent de décrire l'interaction abstraite mono-utilisateur, et pour certaines l'interaction coopérative. En revanche, l'interaction concrète est décrite au mieux sous l'angle limitant du paradigme WIMP. La dimension multimodale de l'interaction n'est pas traitée spécifiquement par aucune notation bien que certaine (comme K-MAD ou CTT) permettent d'exprimer l'usage de plateformes ou de dispositifs physiques.

2.3 Domaine : Travail Coopératif Assisté par Ordinateur

Cette section est consacrée aux notations élaborées dans le cadre du sous-domaine de l'Interaction Homme-Machine qu'est le Travail Coopératif Assisté par Ordinateur. Les concepts récurrents de

cette section sont donc les tâches coopératives, les tâches collaboratives, les groupes, les rôles ou les acteurs.

Nous abordons dans l'ordre les notations GTA, MABTA, CIAN, Touche, CUA, qui s'intègrent au sein d'une méthode de conception et préconisent l'usage de plusieurs modèles. Enfin, nous décrivons la notation Orchestra, qui s'intéresse spécifiquement à la description de situations d'interaction multiutilisateur.

2.3.1 GTA

GTA (*Groupware Task Analysis*) [Veer 2000] est une notation centrée, comme son nom l'indique, sur l'analyse de la tâche. L'ambition de GTA est de servir à la fois de support à l'analyse des besoins afin de structurer les données collectées sur les activités et de support à la conception. Afin de modéliser les activités de groupe, GTA s'appuie sur quatre types de représentation :

1. Une représentation structurée des activités sous forme d'un arbre de tâches,
2. une représentation de la dynamique de l'activité sous forme de diagramme d'activité,
3. une représentation des objets ou artefacts sous forme d'un diagramme de classe,
4. une représentation de l'environnement physique et culturel (principalement du point de vue des usages).

Enfin, GTA repose sur un modèle conceptuel assez simple puisqu'il se limite à six concepts : tâche, rôle, objet, agent, évènement et but. Les descriptions que permettent GTA couvrent l'ensemble des composants du modèle ARCH.

La Figure 46 illustre les rôles et agents de WCCM. Nous considérons les rôles de *Sergent* et *Chef*, des spécialisations du rôle *Utilisateur*, ainsi que le rôle de *Système.* Nous définissons trois agents : Utilisateur *1, Utilisateur 2*, et *Système* qui endossent respectivement les rôles de *Sergent*, de *Chef* et *Système.* La Figure 46 ne fait apparaître aucun lien entre les agents et les rôles. En effet, ces liens ne sont pas décrits visuellement par la notation. À cela s'ajoute un modèle des artefacts, que nous n'avons pas représenté ici, qui se réduit aux objets : Table augmentée (table + caméra + vidéo projecteur), casque (microphone + écouteurs).

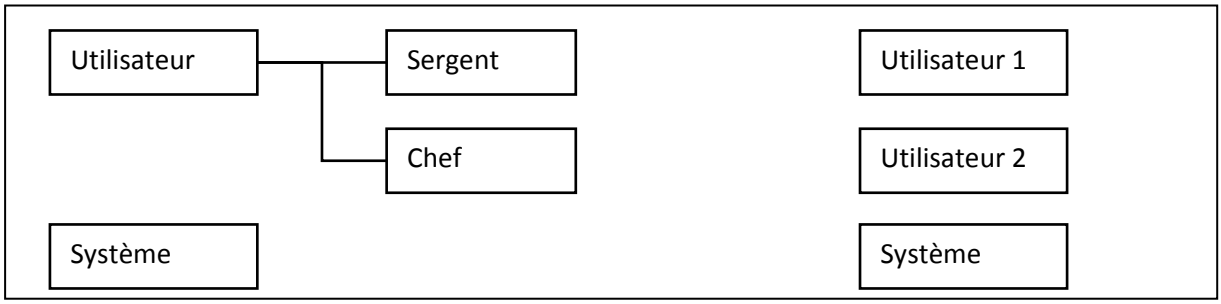


Figure 46 : Description des rôles et agents avec GTA.

La Figure 47 illustre l'arbre de tâches de l'application WCCM selon la notation GTA. Les trois principales sous-tâches de l'application consistent à *Gérer les bâtiments*, *Gérer les soldats* et *Gérer les paysans*. La structure de cet arbre est très proche de celui décrit avec la notation K-MAD à la Figure 35. Notamment car la composition de plusieurs tâches s'effectuent, dans les deux cas, au travers d'un unique opérateur.

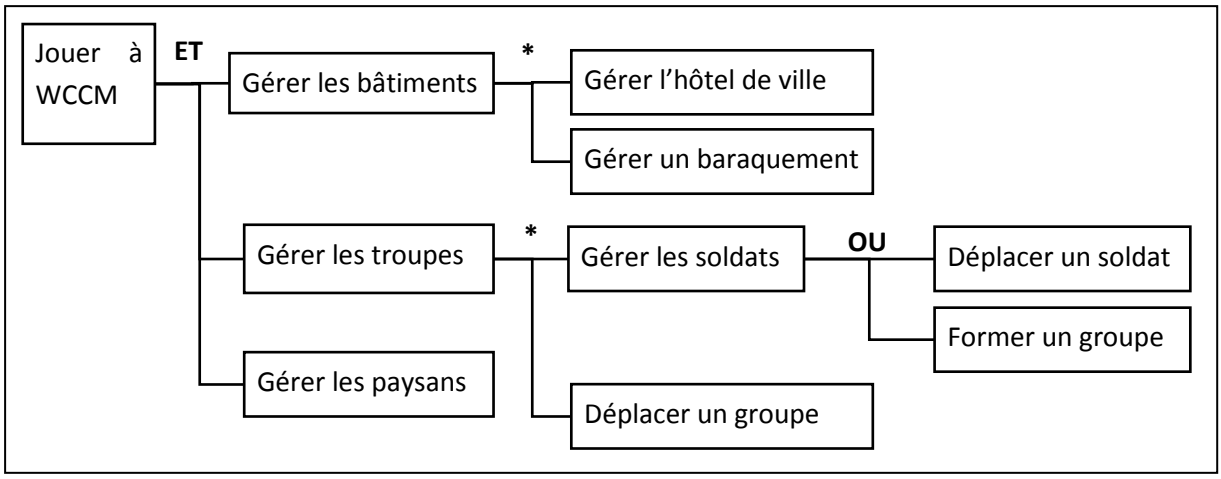


Figure 47 : Modèle de tâche GTA de WCCM.

A l'instar d'autres notations d'arbre de tâches, chaque nœud de l'arbre est décoré. Ces décorations concernent les objets manipulés, tels qu'un bâtiment, les rôles associés à la tâche comme le rôle *Sergent*, les pré- et post-conditions à l'exécution d'une tâche ainsi que les tâches déclenchées. Ce dernier aspect permet en fait de décrire la dynamique de l'activité de groupe puisque c'est à partir de ces éléments de la décoration que l'on peut en déduire les différents scénarios sous la forme d'un diagramme d'activité comme le montre la Figure 48. Cette représentation indique la répartition des tâches entre les utilisateurs et le système.

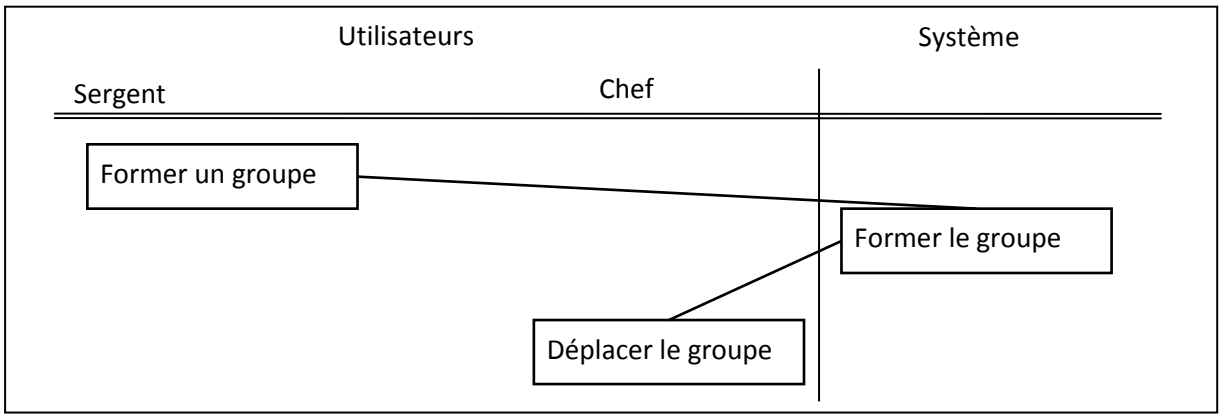


Figure 48 : Diagramme d'activité GTA.

Enfin, GTA s'appuie sur NUAN (*New User Action Notation*) pour spécifier les tâches concrètes. Le Tableau 5 montre la modélisation de la tâche *Former un groupe* introduite dans l'arbre de tâches de la Figure 47. Pour cette spécification, nous avons dû introduire des nouvelles primitives qui correspondent à l'usage de la parole et de gestes. Ainsi, le sergent qui souhaite former un groupe doit délimiter une zone avec les mains, ce qui affiche la zone délimitée sur la table et sélectionne tous les soldats qui s'y trouvent. Ensuite, il doit donner oralement le nom d'un groupe ce qui affiche le groupe formé sur la table, joue le son "Oui ! Mon seigneur" et crée le groupe. Les primitives originales de NUAN concernent uniquement l'usage de la souris et du clavier. Nous avons donc utilisé au sein de la Tableau 5 nos propres primitives pour décrire l'usage de la parole et du geste.

About : Former un groupe

User actions	Interface actions	Computation
DELIMIT ZONE (<hands>)	DISPLAY ZONE (table)	SELECT Soldat IN ZONE
SAY (« a group name »)	DISPLAY GROUP (table) PLAYSONG «yes sir»	CREATE GROUP

Tableau 5 : Modélisation de la tâche de formation d'un groupe avec NUAN

La représentation des activités individuelles ou de groupe est proposée sous la forme classique d'un arbre de tâches. Cependant, comme nous l'avons souligné précédemment, la notation manque de souplesse car un seul opérateur est applicable pour la décomposition d'une tâche en un ensemble de sous-tâches. Ceci conduit à introduire des sous-tâches articulatoires comme la tâche *Gérer les soldats*. De plus, la méthode ne préconise aucun opérateur de décomposition et aucun moyen pour préciser l'ordonnancement des tâches. Il est néanmoins possible d'exprimer un tel entrelacement au prix d'une inflation du nombre de tâches et donc d'un effort de conception. A l'opposé, la décoration de chaque nœud de l'arbre est précise et permet notamment de spécifier les aspects dynamiques des activités grâce à la notion de tâche déclenchée, ainsi que de spécifier le ou les rôles associés et les artefacts manipulés.

Cet arbre de tâches peut être complété par une description des tâches élémentaires (feuilles de l'arbre) par une description de la tâche concrète à l'aide du formalisme NUAN. Nous avons toutefois été confrontés à plusieurs difficultés pour décrire une tâche concrète comme la formation d'un groupe. En effet, NUAN se focalise sur des interfaces WIMP qui donc ne prend pas en compte les nouvelles générations de dispositifs, comme une surface interactive. Nous avons ainsi ajouté de nouvelles primitives pour exprimer l'usage du geste et de la parole : *DELIMIT ZONE (<hands>)* et *SAY(<group name>)*.

Enfin, concernant les rôles, il existe une incertitude sur la coexistence de plusieurs rôles au sein d'une même tâche : est-ce que les sous-tâches héritent ou non des rôles associés à une tâche parent ? Nous avons fait l'hypothèse qu'il n'y a pas d'héritage. De plus, nous avons été amenés à introduire le rôle *Système* afin de rendre explicite son intervention pour la création d'un groupe, comme le montre le Tableau 5.

Conclusion sur l'utilisation de GTA

GTA est une notation satisfaisante pour décrire des systèmes coopératifs dont l'interaction est de type WIMP. En appliquant GTA à notre étude de cas, nous avons aussi identifié une incertitude sur la notation elle-même quant à la composition des rôles.

2.3.2 MABTA

La notation MABTA (*Multiple Aspect Based Task Analysis*) [Lim 2004] est dédiée à l'analyse des besoins. Elle s'appuie sur un ensemble de concepts proche de l'ontologie de GTA et reprend les concepts de rôle, de tâche, de but, et d'objet. À l'aide de ces concepts, les auteurs identifient plusieurs aspects des systèmes collaboratifs qui conduisent à l'utilisation de quatre modèles :

1. Le premier est un modèle hiérarchique des rôles et des utilisateurs intervenant dans une application collaborative ;
2. Le deuxième est un modèle de tâche de groupe, qui décrit la collaboration et la communication entre les acteurs d'un collecticiel ;
3. Le troisième est un modèle hiérarchique de tâches, qui permet d'affiner les tâches du premier modèle, ainsi que d'introduire des tâches non collaboratives ;
4. Enfin, le quatrième modèle propose de représenter l'interface graphique abstraite du système.

La Figure 49 décrit le modèle des rôles et des utilisateurs de l'application WCCM que nous avons produit lors la première phase de modélisation MABTA. Chaque utilisateur est identifié par une étiquette préfixée par la lettre U et un numéro d'utilisateur (ce qui donne la forme : Un). Des relations peuvent ensuite être ajoutées. MABTA propose une relation *est Hiérarchiquement supérieur à*, soit entre deux rôles, soit entre deux utilisateurs, et une relation *Joue le rôle de* qui permet d'associer un utilisateur à un rôle précis. Pour ces deux relations, la disposition verticale des rectangles est importante, car c'est le rectangle le plus haut qui est soit *Hiérarchiquement supérieur à* soit *Joué* par le rectangle le plus bas. La Figure 49 illustre les rôles que nous avons identifiés : le rôle de *Chef* qui est joué par l'*Utilisateur 1* et le rôle de *Sergent* qui est joué par l'*Utilisateur 2*. Nous ajoutons un rôle nommé *Utilisateur* dont les rôles de *Chef* et de *Sergent* sont la spécialisation.

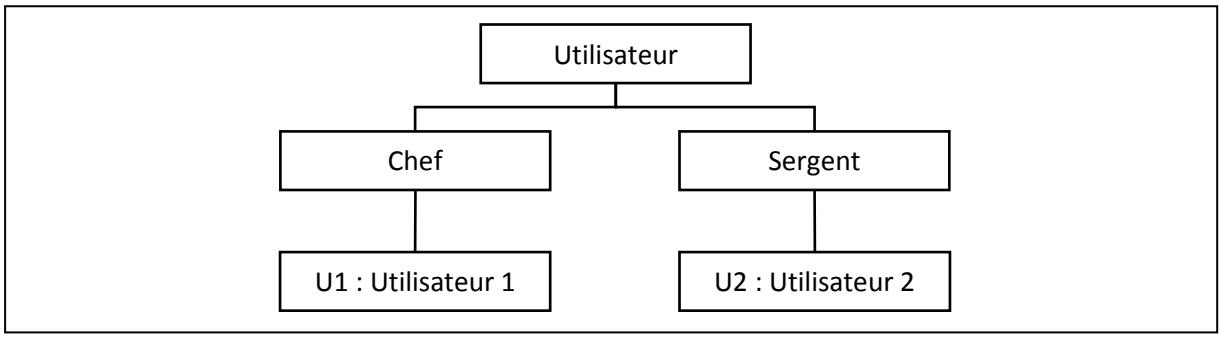


Figure 49 : Modèles de rôles et des utilisateurs MABTA.

La seconde phase de modélisation MABTA consiste à décrire la collaboration entre les utilisateurs à l'aide d'un modèle de tâches de groupe. Comme l'illustre la Figure 50, toute tâche collaborative est typée et son type est indiqué par une lettre : *C* pour une tâche de coordination, *S* pour une tâche mono-utilisateur, et *D* pour une tâche de prise de décision. Pour chaque utilisateur du système, une colonne est utilisée pour décrire les tâches qu'il peut effectuer. Les tâches collaboratives sont reliées entre elles au moyen d'une relation d'*Influence,* signifiant que la réalisation d'une tâche a une influence sur la réalisation d'une autre tâche.

Pour l'application WCCM, la création d'un fantassin s'effectue en deux étapes illustrées à la Figure 50, tandis que le *Chef* peut donner son accord à la création en *Acquiesçant*. Le sergent peut également *Former un groupe* tandis que le *Chef* peut *Déplacer un groupe* créé. Nous utilisons la relation *Influence* pour signifier que la formation d'un groupe engendre la possibilité de le déplacer.

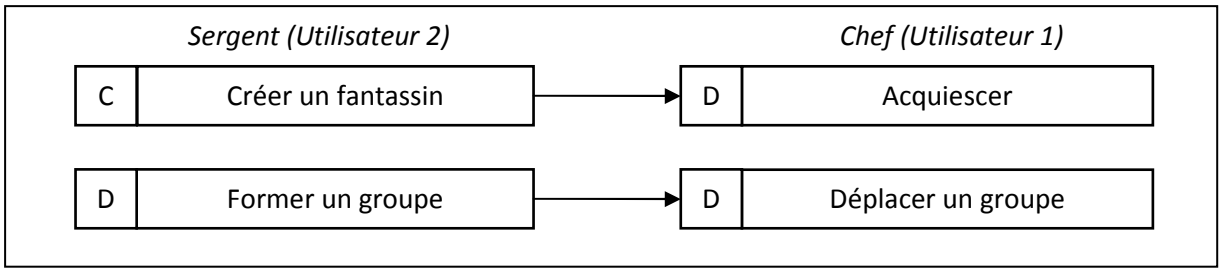


Figure 50 : Modèle de tâche de groupe MABTA.

À la suite de la description du modèle de tâches de groupe, MABTA propose d'affiner ces tâches par le biais d'un modèle hiérarchique de tâche. Celui-ci permet également d'ajouter des tâches mono-utilisateur qui n'ont pas d'influence sur la collaboration. Nous représentons une partie du modèle de tâches de notre cas d'étude au sein d'une description où la répartition des les tâches entre les deux utilisateurs est exprimée par une séparation en deux colonnes. Le modèle hiérarchique de tâches permet d'exprimer une planification entre les sous-tâches. Cette planification s'exprime au moyen du label *Plan*, suivi d'un ordre de réalisation des sous-tâches.

Ainsi, nous décomposons la tâche *Créer un fantassin* en deux sous tâches successives : *Sélectionner un baraquement* puis *Créer un fantassin*. La séquence est matérialisée par un *Plan : 1 -> 2*. De manière similaire, nous avons décomposé la tâche *Acquiescer* en deux sous-tâches *Entendre le type d'unité créé* et *Donner son accord*. Il convient de noter que nous avons déplacé la relation d'influence entre la tâche *Créer un fantassin* et la tâche *Acquiescer*, qui est présente dans le modèle de tâches de groupe (Figure 50). Elle relie maintenant la sous-tâche *Créer un fantassin* et la tâche *Acquiescer*. Ceci exprime le fait que c'est à la suite de l'exécution de la demande de création d'un fantassin que l'autre utilisateur peut donner son accord à la création. Nous avons décomposé de manière assez similaire les tâches *Former un groupe* et *Déplacer le groupe*. Toutefois, le plan de la tâche *Déplacer le groupe* comporte une séquence suivie d'une concurrence entre les tâches *Donner l'ordre de se déplacer* et *Indiquer une destination*.

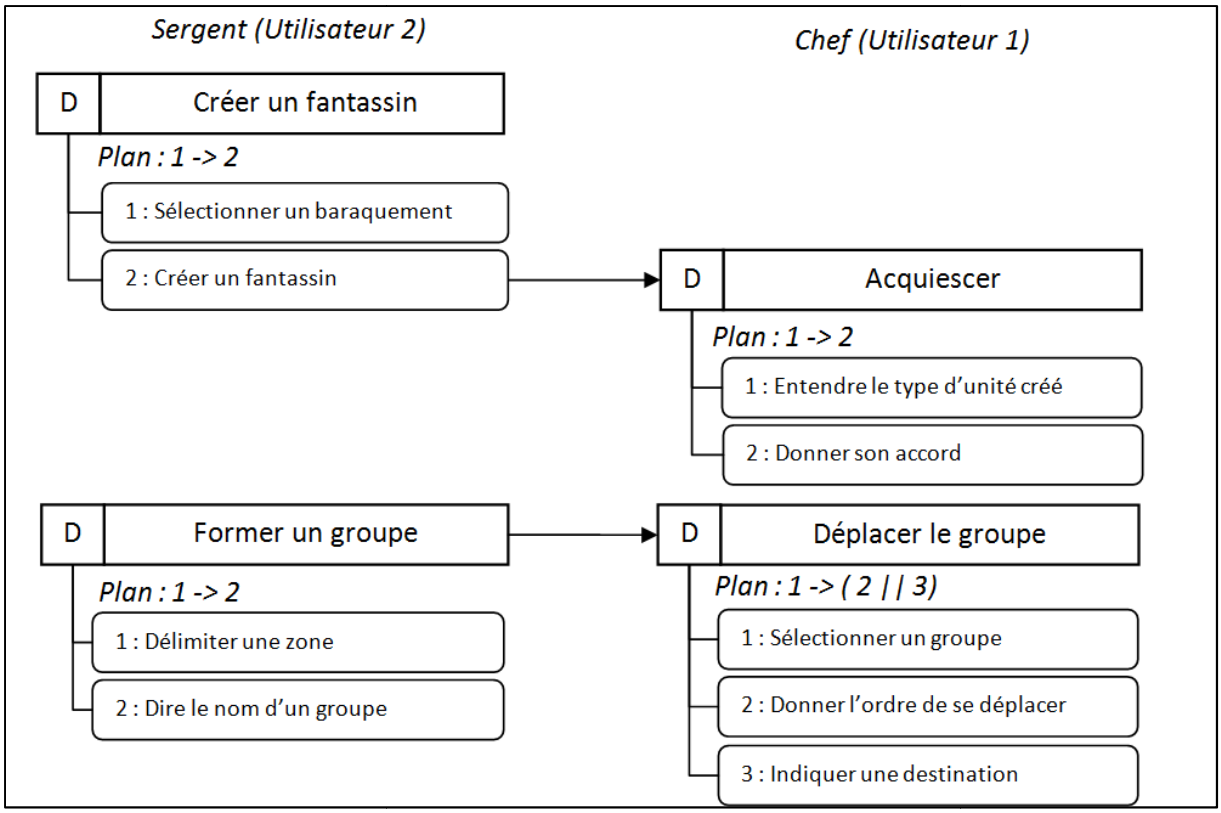


Figure 51 : Décomposition hiérarchique en sous-tâches individuelles.

Enfin, les auteurs de MABTA préconisent de réaliser un modèle de l'interface graphique sous la forme de maquettes. Une telle maquette est illustrée à la Figure 39 dans la section consacrée à la notation TaskMODL.

Conclusion sur MABTA

Pour conclure, MABTA permet au travers de ses quatre représentations de modéliser de manière complémentaire les objets graphiques manipulables (modèle de l'interface graphique), les tâches de bas niveaux qui manipulent ces objets (modèle hiérarchique de tâches), les tâches de haut niveau qui décrivent la collaboration entre les utilisateurs (modèle de tâches de groupe) et les différents utilisateurs et rôles intervenant dans l'utilisation d'un système collaboratif (modèle des rôles et utilisateurs).

Malgré cette complémentarité, les différentes représentations de MABTA ne permettent pas de modéliser précisément le comportement d'un système collaboratif. En effet, chaque représentation souffre d'un manque de précision dans les moyens d'expression (relation *influence*) ou d'un manque de moyens d'expression (pas de représentation des modalités d'entrée ou de sortie). Rappelons que MABTA est initialement destinée à l'analyse des besoins, ce qui explique pour une large part l'imprécision pour la spécification d'un système. En effet, une solution de conception produite à partir d'une analyse de besoins réalisée avec MABTA peut ensuite être décrite au moyen d'une autre notation.

2.3.3 CIAN

La notation CIAN (*Collaborative Interactive Applications Notation*) est proposée dans le cadre d'une méthode de conception appelée CIAM (M pour *Methodology*) [Molina 2006, Molina 2007]. Cette méthode s'appuie sur la description de plusieurs modèles complémentaires élaborés dans l'ordre suivant :

1. Description d'un sociogramme
2. Description des activités de groupe par :
    a. un modèle d'interaction de groupe (nommé modèle d'Inter – Action)
    b. un modèle de responsabilité
3. Description fine des activités par :
    a. un modèle d'interaction individuelle
    b. un modèle de tâche reposant sur CTT [Paterno 1997, Mori 2002]

L'utilisation de cette méthode et des éléments de notation associés sont décrits en détail pour deux applications : AULA une application collaborative et ubiquitaire d'aide à l'apprentissage de langues [Parades 2008], et un outil de gestion de projets [Molina 2008].

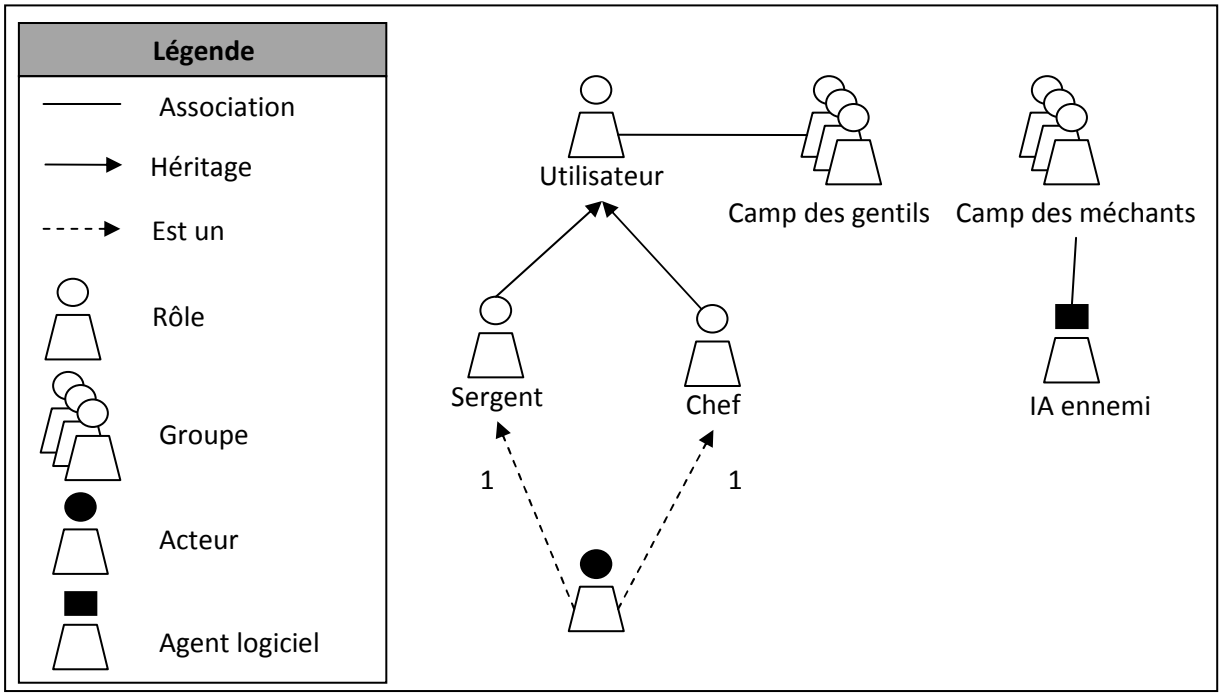


Figure 52 : Sociogramme CIAN de l'application WCCM.

La spécification de l'application WCCM avec la méthode CIAM commence par la description d'un sociogramme. Un sociogramme CIAN décrit les concepts de rôle, de groupe, d'acteur et d'agent logiciel, ainsi que les relations entre ces différents concepts, comme illustrés à Figure 52 pour le cas de WCCM. Cette description repose sur la notion d'héritage, d'association et d'instanciation (légende de la Figure 52). A la Figure 52, les deux rôles principaux de WCCM sont décrits : *Sergent* et *Chef* qui héritent tous deux d'un rôle *utilisateur*. Nous avons également représenté deux groupes : le *Camp des gentils* qui comprend les *utilisateurs* et le *Camp des ennemis* qui est joué par un agent logiciel

appelée *IA ennemi* (pour Intelligence Artificielle). Enfin, nous associons deux acteurs respectivement un *Chef* et un *Sergent*, comme cela est indiqué par les deux relations *Est un* entre l'icône d'acteur et les rôles *Sergent* et *Chef*.

La deuxième étape de CIAM consiste à décrire le diagramme d'Inter-Action. Celui-ci structure le flot de travail et permet de décrire les tâches à haut niveau d'abstraction ainsi que les relations logiques et temporelles entre ces tâches. Un diagramme d'Inter-Action possède un point d'entrée et un point de sortie ; les symboles employés pour les représenter sont ceux des diagrammes d'état-transition d'UML.

Sur un principe identique aux machines à états hiérarchiques, une tâche peut être décrite en détail par un sous-ensemble de sous-tâches. Visuellement, ces sous-tâches sont imbriquées au sein d'une tâche de plus haut niveau. La Figure 53 illustre le diagramme d'Inter-Action pour l'application WCCM. Nous décrivons une tâche globale : *Jouer à* WCCM qui est décomposée en trois sous-tâches :

- La tâche individuelle : *Gérer les paysans* ( ),
- La tâche coopérative : *Gérer les troupes* ( ),
- Et la tâche collaborative : *Gérer les bâtiments* ( ).

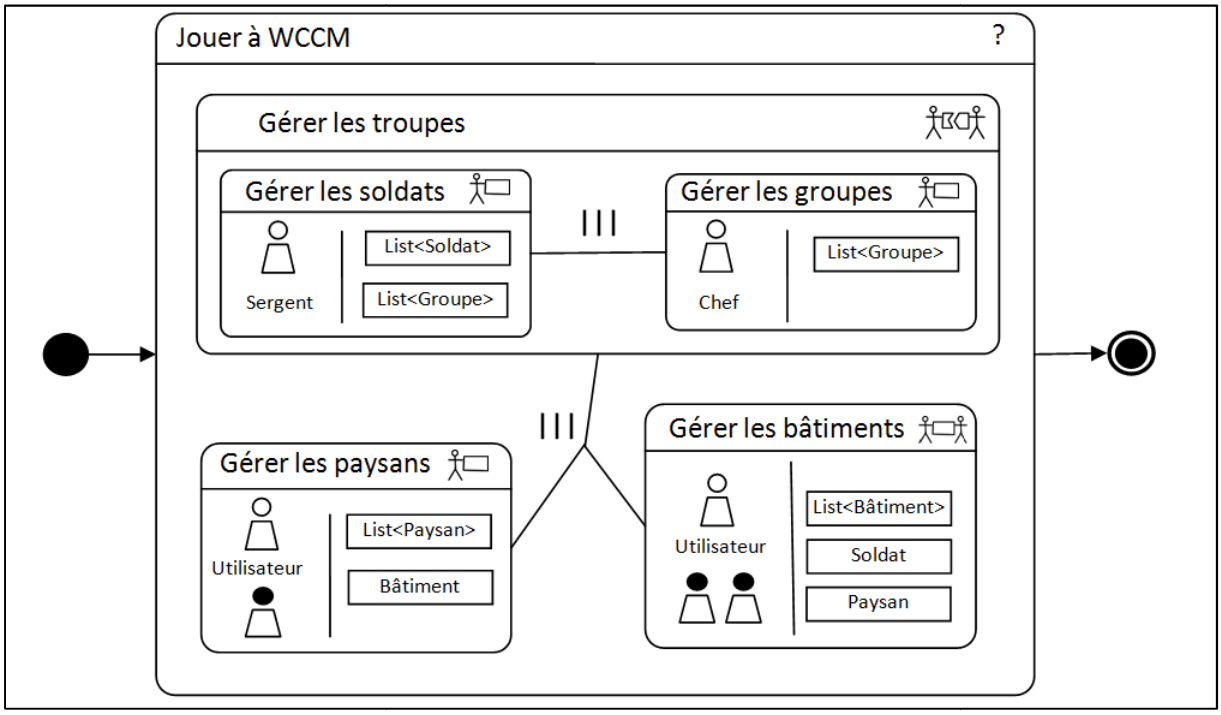


Figure 53 : Diagramme d'Inter-Action CIAN de l'application WCCM.

Les relations entre tâches sont exprimées avec des opérateurs de la notation CTT. Ainsi, les sous-tâches *Gérer les soldats* et *Gérer les groupes* sont par exemple reliées par l'opérateur de concurrence (|||) de la notation CTT.

La description du diagramme d'Inter-Action permet d'identifier l'association des tâches aux différents rôles. La tâche coopérative *Gérer les troupes* est elle-même décomposée en deux sous-tâches individuelles, *Gérer les soldats* et *Gérer les groupes*, qui sont respectivement associées aux rôles de *Sergent* et de *Chef*.

Afin de compléter cette représentation, CIAM propose de décrire un modèle de responsabilité sous la forme d'un tableau. Celui-ci permet de préciser pour chaque tâche : son type, les objets manipulés, ainsi que les pré-requis en termes de tâches et de données. La Tableau 6 présente le modèle de responsabilité pour les trois tâches de WCCM. Nous notons que cette représentation est redondante par rapport au modèle d'Inter-Action, et qu'un éditeur adéquat pour le modèle d'Inter-Action permettrait de générer automatiquement le modèle de responsabilité.

Responsabilité	Type de tâche	Objet du Domaine	Pré-requis	
			Tâche	Données
Gérer les troupes		Soldat, Groupe		Soldat
Gérer les soldats		Soldat, Groupe		Soldat
Gérer les groupes		Groupe		Groupe

Tableau 6 : Modèle de responsabilité CIAN.

La dernière étape de la méthode CIAM consiste en la conception détaillée des tâches et des données du domaine. Pour cela, il convient de considérer chaque tâche de haut niveau d'abstraction décrite au sein du modèle d'Inter-Action pour les détailler. La forme générale de cette description détaillée consiste en trois diagrammes par tâche de haut niveau d'abstraction :

- Le premier correspond à la description des rôles intervenant dans la réalisation de la tâche, ainsi que la relation d'utilisation entre ces rôles et les objets du domaine manipulés pour cette tâche ;
- Le second consiste à décrire les objets du domaine à l'aide d'un diagramme de classe UML ;
- Le troisième consiste à décrire de manière précise les sous-tâches de la tâche de haut niveau d'abstraction.

Cette dernière partie peut prendre des formes différentes selon qu'il s'agit de détailler une tâche individuelle, collaborative, ou coopérative. Les auteurs de CIAN propose d'utiliser un arbre de tâches CTT pour décrire les activités individuelles ou collaboratives, tandis qu'ils proposent d'utiliser un modèle d'Inter-Action pour décrire les activités coopératives. Dans ce dernier cas, chaque tâche du modèle d'Inter-Action doit faire l'objet d'une description détaillée.

Nous montrons à la Figure 54 la description détaillée de la tâche *Gérer les bâtiments*. Nous retrouvons le nom de la tâche dans l'encadré ainsi que le symbole graphique qui indique qu'il s'agit d'une tâche collaborative. Pour chaque tâche, il convient de spécifier :

- les participants impliqués dans la réalisation de la tâche (deux acteurs jouant un rôle d'Utilisateur à la Figure 54).
- les objets du domaine décrits à l'aide d'un diagramme de classe UML (une liste de bâtiments qui peuvent être soit un Baraquement soit un Hôtel de ville à la Figure 54). Le symbole indique que ces objets sont observables par les deux utilisateurs. Il est possible de définir si ces objets sont privés, partagés ou en accès restreint (notion d'espace) ;
- l'interaction sous la forme d'un arbre de tâches CTT.

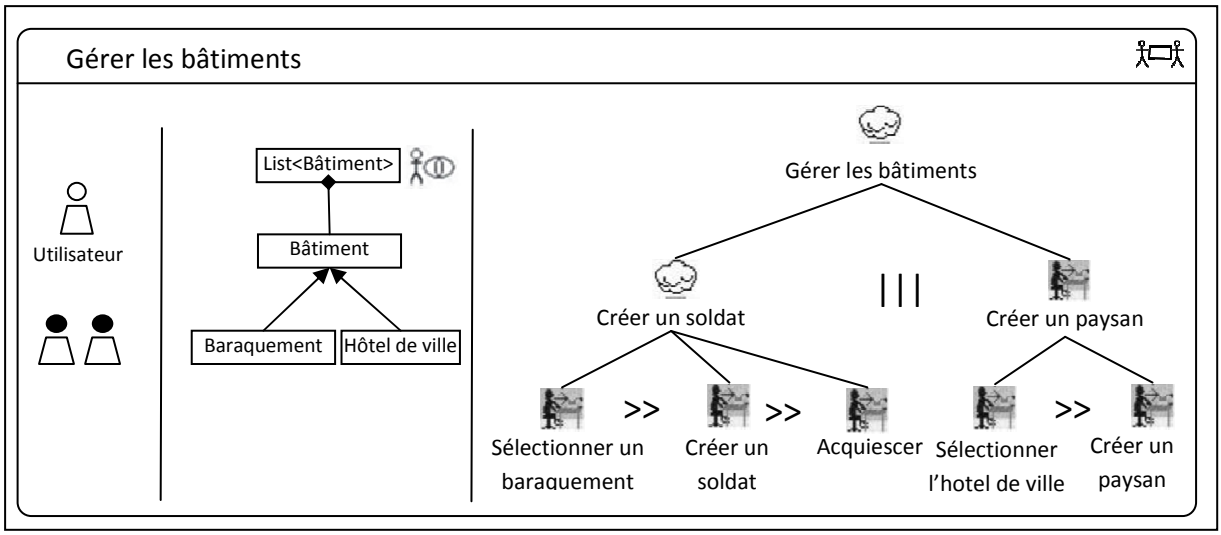


Figure 54 : Description détaillée de la tâche collaborative de gestion des bâtiments.

Conclusion sur CIAN

La méthode CIAM propose une approche itérative de conception par affinements successifs pour modéliser l'interaction de système coopératifs et/ou collaboratifs. En partant de la définition des rôles, des groupes et des agents, elle permet de décrire dans un premier temps les tâches de haut niveau d'abstraction au sein d'un diagramme de flot de travail (diagramme d'Inter-Action) qui sont ensuite affinées plusieurs fois pour décrire *in-fine* les tâches individuelles et collaboratives élémentaires. Cette méthode et la notation CIAN qui l'accompagne, permet également de décrire la relation entre les tâches concrètes et les objets du domaine. Ce type de représentation synthétique est pratique pour représenter différents aspects d'une partie d'un système complexe. En revanche, aucune vue d'ensemble sur les données du domaine n'est proposée. Toutefois, cela peut être réalisé par le bais d'un diagramme de classe d'UML complémentaire regroupant les objets décrits dans les différentes représentations détaillées et en ajoutant ceux qui ne sont pas directement liés à l'interaction. Enfin, la notion d'espace de travail pour la représentation des objets utilisés par les tâches (espace privé, public ou d'accès restreint) permet d'indiquer, sans préciser le dispositif concret d'interaction utilisé, quel est le degré de partage des objets du domaine.

Du point de vue de l'interaction concrète, CIAN n'apporte rien de plus par rapport à CTT et permet uniquement de décrire le comportement de l'interaction (action utilisateur et/ou réaction du système). En effet, aucune modélisation complémentaire n'est prévue pour représenter l'interface utilisateur concrète. Enfin, la notation CIAN, comme CTT, ne permet pas de décrire l'usage de modalité d'interaction ou de combinaison de modalités.

2.3.4 TOUCHE

Les auteurs de [Penichet 2006] proposent une approche de modélisation des systèmes collaboratifs pour permettre une meilleure compréhension de l'organisation des différents utilisateurs, des collaborations entre ces utilisateurs et des tâches qu'ils réalisent individuellement. Cette approche a été étendue dans [Penichet 2009a] et [Penichet 2009b] pour devenir la méthode de conception de l'interaction homme machine multiutilisateur nommée TOUCHE (*Task-Oriented and User-Centred Process Model for Developing Interfaces for Human–Computer–Human Environments*). Plus

récemment, un outil logiciel (*TOUCHE Case Tool*) a été réalisé en support à cette méthode de conception [Penichet 2010].

A l'instar d'UML, elle propose une approche de description selon deux types de modèles : les modèles qui exposent la structure de l'application et ceux qui exposent le comportement de l'application. La description de la structure de l'application s'effectue d'une part au sein de diagrammes de classe UML et d'autre part au travers de diagrammes de structure organisationnelle (OSD). Enfin, la description du comportement est fournie par un diagramme de collaboration (CD) et de diagrammes de tâches (TD). Pour ces derniers, les auteurs proposent d'utiliser la notation CTT, c'est-à-dire de décrire un arbre de tâches coopératifs et plusieurs arbres de tâches individuelles (un par rôle).

L'utilisation du diagramme de classe d'UML est détaillée et illustrée à la section 2.1.1 de ce chapitre. Son utilisation dans le cadre la méthode TOUCHE ne diffère pas de ce que propose le formalisme UML.

La Figure 55 présente le diagramme organisationnel de l'application WCCM. Cellui-ci comporte quatre types de concepts parmi ceux proposés par la notation : les utilisateurs, les rôles, les groupes et les agents logiciels. Nous avons exprimé que deux utilisateurs nommés *Joueur 1* et *Joueur 2* étaient requis. Ces utilisateurs jouent respectivement les rôles de *Sergent* et de *Chef*. Nous avons également exprimé que le rôle de *Chef* était hiérarchiquement supérieur à celui de *Sergent*. L'agrégation des deux utilisateurs forme le groupe *Camp des gentils*, tandis qu'une agrégation d'agents logiciels *IA ennemi* forme le groupe *Camp des méchants*.

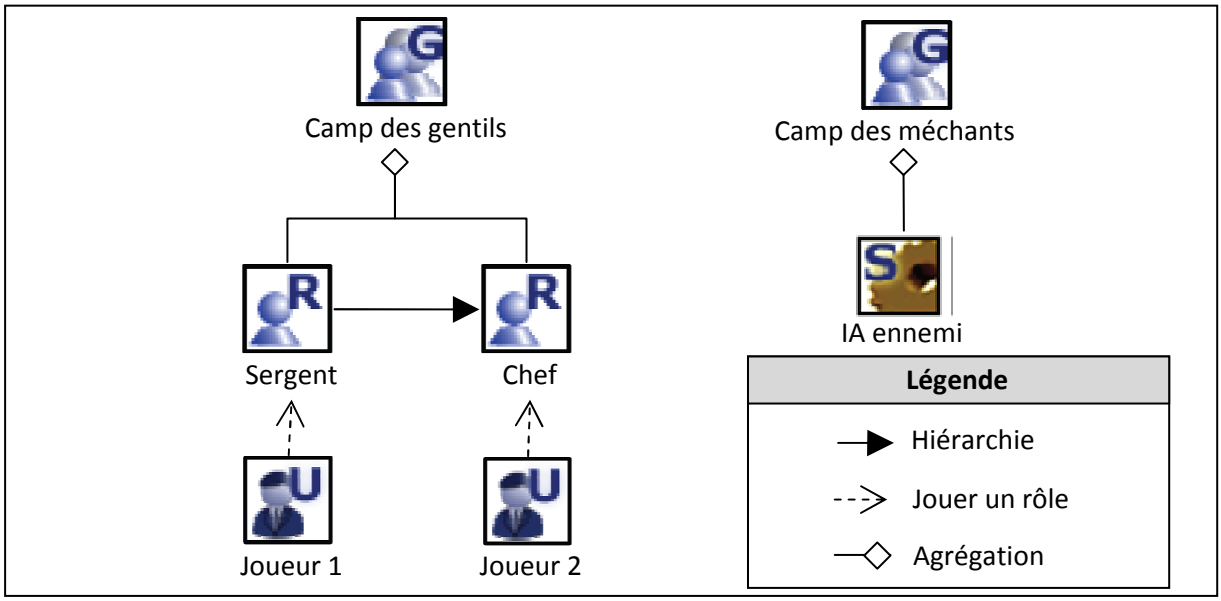


Figure 55 : Diagramme de structure organisationnelle.

A partir du diagramme organisationnel, il est possible de décrire les relations de coopération existant entre les rôles identifiés. Pour cela, il faut reprendre les éléments du diagramme organisationnel précédent, et ajouter les relations de coopération existantes entre les rôles, les acteurs et les groupes. Une tâche coopérative fait intervenir plusieurs intervenants. La tâche coopérative est décomposée en deux sous-tâches (une pour chaque intervenant). Chacune de ces tâches est décrite par une flèche munie d'une étiquette.

La Figure 56 décrit une partie du diagramme de collaboration pour la tâche coopérative de gestion des troupes. Une interaction est décrite entre le rôle de *Sergent* et le rôle de *Chef*, il s'agit de *Former un groupe.* Une deuxième interaction est décrite uniquement concernant le rôle de *Chef* : *Déplacer un groupe*. La position de l'origine de la flèche est très importante puisqu'elle indique ici que c'est le rôle de S*ergent* qui effectue la formation du groupe.

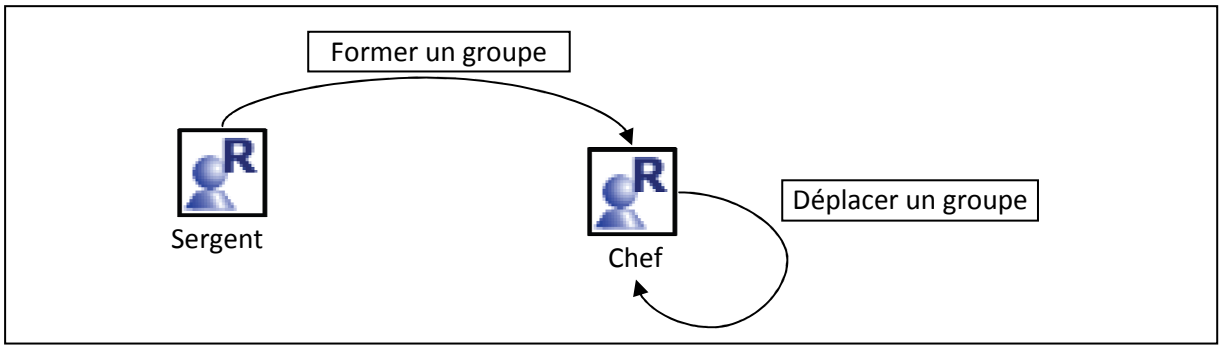


Figure 56 : Diagramme de collaboration la tâche de gestion des troupes.

Enfin, la méthode de conception préconise de décrire un modèle de tâches (TD) coopératives de l'application. Pour cela, il propose d'utiliser la notation CTT, et de décrire un arbre coopératif reprenant les tâches coopératives identifiées au sein du diagramme de collaboration, puis de décrire les arbres de tâches individuelles correspondant aux différents intervenants. Ce type de description est décrit à la section 2.2.1 de ce chapitre consacrée à la notation CTT.

Conclusion sur TOUCHE

La méthode de conception TOUCHE se concentre sur la modélisation des intervenants au sein de situations collaboratives. Elle permet d'expliciter les relations existantes entre ces intervenants qui sont répartis en trois catégories : Acteur, Groupe et Rôle. Cela permet d'exprimer la collaboration du point de vue des utilisateurs du système. Le diagramme organisationnel est très proche du sociogramme de CIAN, aussi bien en termes de concepts qu'en termes de représentation, mais il est plus pauvre en ce qui concerne la description des rôles, de par l'absence de la relation d'héritage. En revanche, le diagramme de collaboration permet d'expliciter de manière synthétique le réseau de collaboration entre les intervenants d'une situation, ce que les autres notations présentées jusqu'ici proposent de décrire au sein d'un modèle de tâches coopératives. Enfin, la reprise de CTT n'apporte pas de concept supplémentaire pour la conception de système multiutilisateur. Pour conclure, la dimension la plus intéressante de TOUCHE au regard des notations précédentes est incarnée par le diagramme de collaboration qui permet de représenter de manière synthétique la collaboration entre les intervenants. Ce diagramme semble être un bon préalable à la modélisation des tâches, et ainsi servir de référence lors de la structuration du modèle de tâches de l'application, par exemple avec CTT ou K-MAD.

2.3.5 CUA

CUA (*Collaboration Usability Analysis*) [Pinelle 2003] est une méthode d'analyse centrée sur les problèmes d'utilisabilité dans le domaine du travail collaboratif. Cette méthode sert à analyser et à évaluer l'interaction collaborative mais elle est également utilisable en tant que méthode de conception des systèmes interactifs multiutilisateurs. Les auteurs de CUA ont par la suite élaboré une

version de la méthode d'analyse spécifique aux applications autour de table interactive nommée T-CUA (*Tabletops-CUA*) [Pinelle 2007]. Cependant, T-CUA se focalise sur l'analyse de l'utilisabilité tandis que CUA offre une méthode et des outils de conception. L'approche proposée est itérative et offre un moyen pour passer du scénario d'activité à l'expression de l'interface utilisateur concrète. Les principales phases sont :

1. la description d'un diagramme de flot de travail qui s'appuie sur un ensemble de scénarios faisant intervenir l'ensemble des acteurs identifiés ;
2. la description précise de chaque scénario selon un canevas fourni par CUA comportant la description de l'activité du scénario, la description des utilisateurs, le résultat attendu, et le contexte du scénario ;
3. la description d'un modèle de tâches multiutilisateur.

Le diagramme de flot de travail CUA décrit une série de scénarios réalisés par les différents rôles. La Figure 57 illustre plusieurs scénarios pour l'application WCCM autour des rôles de *Sergent* et de *Chef*. Le premier scénario consiste à *Créer un soldat*. Ce scénario est multiutilisateur puisqu'il est associé à la fois aux deux rôles de *Sergent* et de *Chef*. Il est également collaboratif, ce qui est indiqué à la Figure 57 par le fait que les deux rectangles *Créer un soldat* sont au même niveau. Le scénario *Déplacer un soldat* est quant à lui individuel et réservé au rôle de *Sergent*. Enfin, les deux derniers scénarios décrivent la coopération entre les utilisateurs. En effet, le scénario *Former un groupe* est réservé au *Sergent*, tandis que le scénario *Déplacer un groupe* est réservé au *Chef*. Le lien entre les deux scénarios et le décalage vertical indiquent l'enchaînement temporel.

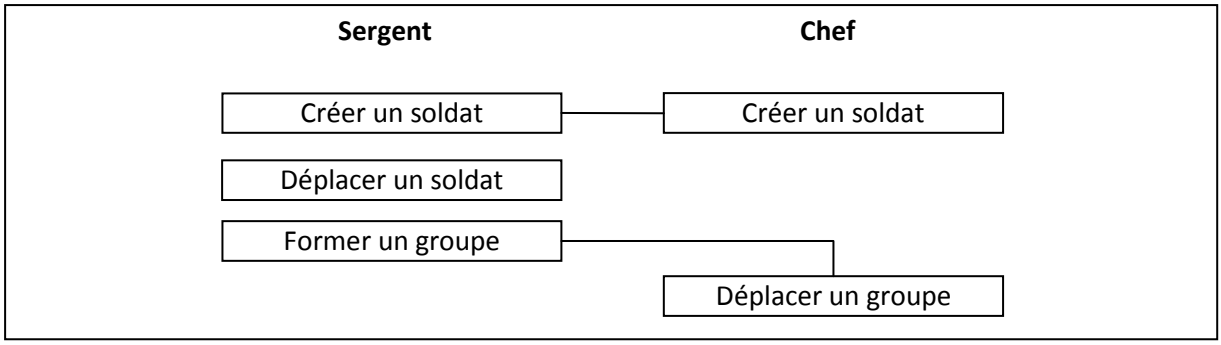


Figure 57 : Diagramme de flot de travail CUA.

Les différents scénarios décrits au sein de la Figure 57 sont ensuite détaillés textuellement afin d'identifier, le détail de l'activité, les utilisateurs, le but du scénario, ainsi que son contexte. La Figure 58 illustre une telle description pour le scénario collaboratif de création d'un soldat.

Scénario : créer un soldat

Description de l'activité : Les utilisateurs U1 et U2 jouent ensemble une partie de WCCM. Durant la partie, l'un des deux utilisateurs décide de créer un fantassin (un des deux types de soldats disponibles). Il commence par sélectionner le baraquement, puis énonce la commande « créer un fantassin ». L'ayant entendu, le deuxième utilisateur est d'accord avec le choix de son équipier et prononce « ok ». La création du fantassin commence alors dans le baraquement.

Description de l'utilisateur U1 : U1 est l'un des deux joueurs, il joue le rôle de chef.

Description de l'utilisateur U2 : U2 est l'autre joueur, il joue le rôle de sergent.

But : U1 et U2 créent un fantassin pour augmenter la puissance de leur armée.

Contexte du scénario : Dans la salle où se situe la table, U1 et U2 font une partie de WCCM contre l'ordinateur. Ce dernier devenant assez agressif, les joueurs cherchent à accroître la puissance de leurs troupes.

Figure 58 : Description du scénario collaboratif « créer un soldat » selon CUA.

L'étape suivante consiste à décrire le modèle de tâches pour chacun des scénarios. Le modèle présenté à la Figure 59 traduit le scénario présenté à la Figure 58. De façon similaire à MABTA, les tâches sont organisées en colonne, avec un acteur associé à chaque colonne. Un encadré décrit une tâche abstraite. Celle-ci est détaillée en termes de tâches collaboratives (notée CTI pour Instance de Tâche Collaborative) et en termes de tâches individuelles (notée ITI pour Instance de Tâche Individuelle). La notation inclut également un moyen pour exprimer des relations entre tâches mais uniquement pour les tâches abstraites. Au sein de la Figure 59, la tâche C*réer un soldat* est décrite par trois tâches successives : *Sélectionner un baraquement*, *Choisir un type d'unité*, et *Confirmer*. Enfin, nous avons décoré la tâche *Sélectionner un baraquement* avec le symbole "+" pour indiquer qu'elle peut être répétée plusieurs fois.

Notons que le modèle de tâches de la Figure 59 ne couvre pas tous les cas du scénario décrit dans la Figure 58. En effet, dans ce dernier, il est précisé que l'un ou l'autre des utilisateurs peut être à l'initiative de la création de l'unité, tandis que l'autre utilisateur donne son accord à la création. Pour décrire cette symétrie dans un modèle de tâches organisé selon des colonnes indiquant les rôles, il convient alors de reproduire le même modèle, mais en inversant les entêtes de colonnes.

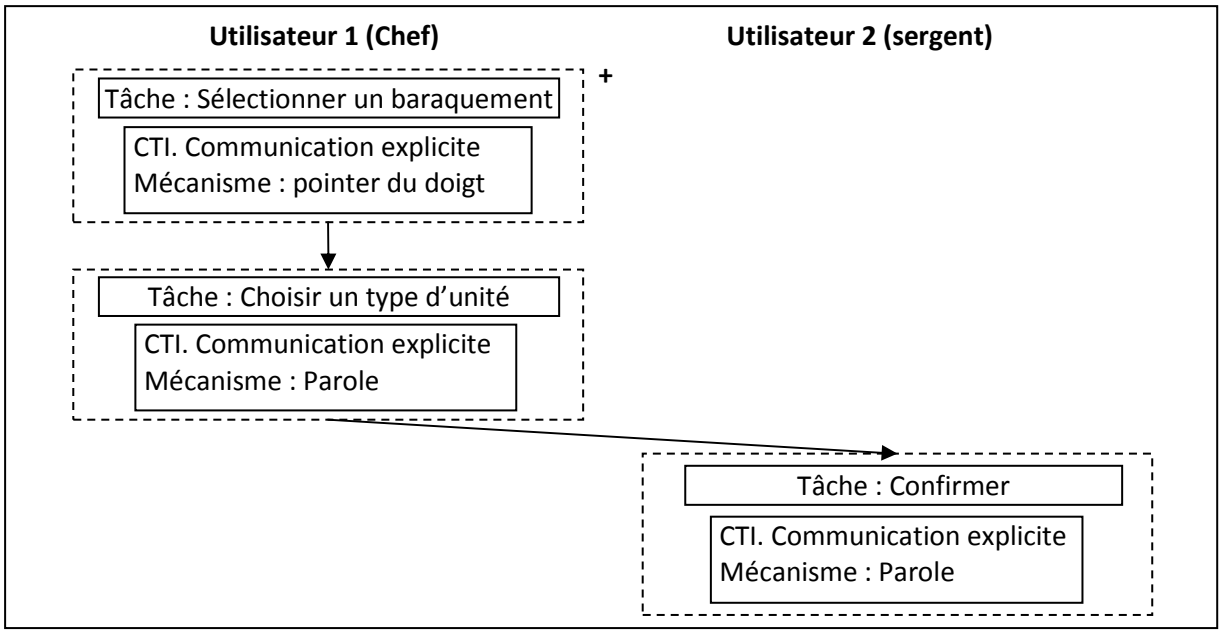


Figure 59 : Modèle de tâches CUA pour le scénario « créer un soldat ».

CUA propose également de décrire les mécanismes utilisés par les utilisateurs pour collaborer. Pour cela, les auteurs de CUA proposent un ensemble de mécanismes de collaboration où sont distinguer les mécanismes de communication et les mécanismes de coordination. Les mécanismes de communication désignent les comportements de communication explicite tels que les messages parlés, écrits ou gestuels. Dans la catégorie des mécanismes de coordination sont inclus l'accès partagé (aux outils ou aux objets) ou encore les transferts. Ainsi les mécanismes de coordination associés à la catégorie des accès partagés désignent l'obtention ou la réservation d'une ressource, ou encore la protection de son travail. Ces différents mécanismes permettent de préciser en quoi consistent les CTI utilisées dans le modèle de tâches CUA. Ils permettent donc d'indiquer les mécanismes de collaboration accessibles aux utilisateurs d'un scénario donné. Ces mécanismes pourraient être réutilisés au sein d'autres notations de modèle de tâches telles que CTT ou K-MAD. Dans le modèle de tâche CUA de la Figure 59, nous avons par exemple décrit trois mécanismes de collaboration, un pour chaque tâche. Pour la tâche *Sélectionner un baraquement*, il s'agit d'une communication explicite : *Pointer du doigt* le baraquement. Enfin, pour les tâches *Choisir un type d'unité* et *Confirmer*, il s'agit d'un mécanisme de communication explicite : la parole.

Conclusion sur CUA

L'approche de conception CUA propose de décrire l'interaction abstraite selon trois niveaux de modélisation : flot de travail, scénario, puis arbre de tâches. L'apport de CUA par rapport aux autres notations réside surtout dans les mécanismes de collaboration pour décrire les tâches collaboratives. Ces mécanismes permettent de décrire finement les interactions collaboratives entre les différents acteurs. Ces mécanismes ont été clairement identifiés et classés par les auteurs de CUA et sont exploitables en complément des autres notations. En revanche, l'interaction concrète et multimodale n'est pas abordée dans la notation.

2.3.6 Orchestra

Orchestra est une notation qui propose de décrire le flot de travail collaboratif et individuel sous la forme d'une partition musicale [David 2006], où chaque ligne d'une portée correspond à un concept des collecticiels : les rôles utilisateur, les activités des utilisateurs, les processus en termes d'états et de transitions, les artéfacts utilisés lors des activités et le contexte de la situation collaborative qui inclut les plateformes utilisées. Chaque mesure décrite sur la portée correspond à une situation de travail et à une tâche collaborative ou individuelle donnée. Au sein de chaque mesure des notes peuvent être décrites. En fonction de la ligne sur laquelle elles sont disposées, ces notes correspondent à un rôle, une activité, un état d'un processus, un artéfact ou un élément du contexte.

La Figure 60 illustre la modélisation de la gestion des soldats avec la notation Orchestra. La première situation correspond à la tâche de déplacement d'un soldat par l'utilisateur 2 qui joue le rôle de sergent. Dans cette même situation, l'utilisateur 1 qui joue le rôle de Chef observe. Dans la deuxième situation, nous avons représenté la tâche de formation d'un groupe par le sergent. Enfin, la troisième situation décrit la tâche de déplacement d'un groupe par le chef.

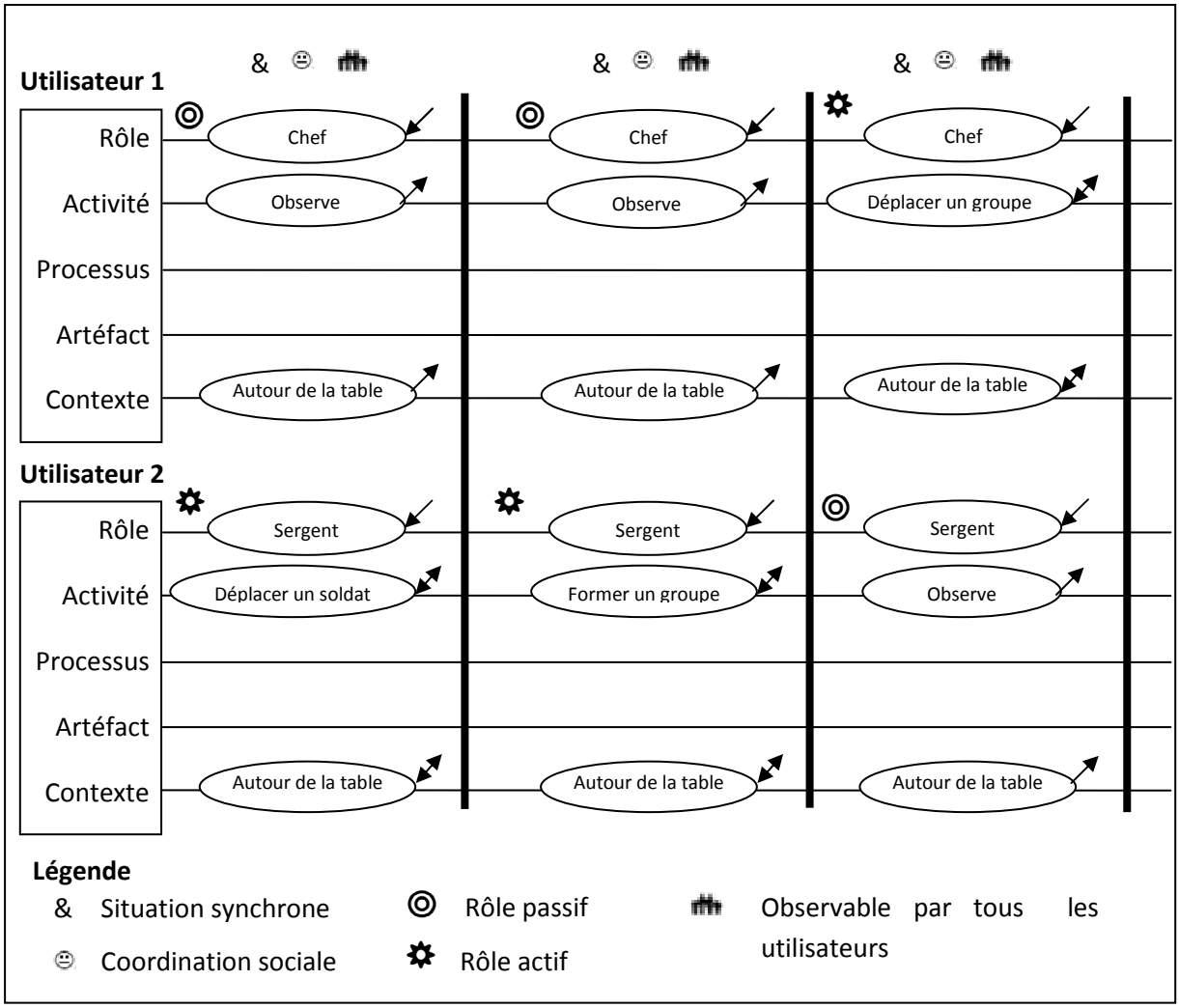


Figure 60 : Gestion des soldats modélisée avec Orchestra.

Afin de représenter la participation des éléments de la situation, Orchestra identifie trois types de participations effectuées par les acteurs de la situation : *l'action* (↓), *l'observation* (↑), ou *l'édition*

(↕). Ces flèches peuvent être doublées pour un rôle pour indiquer qu'il s'agit du rôle principal de la situation. Ainsi, dans la première situation, l'activité *Déplacer un soldat* correspond à une *édition* pour l'*Utilisateur 2*, tandis que le contexte *Autour de la table* est marqué comme *Observé* pour l'*Utilisateur 1*.

Orchestra propose de compléter la description de la participation des acteurs, en indiquant si un rôle est *actif* (✲) ou *passif* (◎). Nous pouvons ainsi décrire que le rôle joué par l'*Utilisateurs 1* est *Passif* pendant les deux premières situations et que l'utilisateur 2 joue un rôle *actif* pendant ces mêmes situations. A l'inverse, l'*Utilisateur 1* joue un rôle *Actif* pendant la troisième situation, tandis que l'*Utilisateur 2* y joue un rôle *passif*.

Orchestra permet de décorer les situations afin de préciser les aspects de synchronisation (*Synchrone* ou *Asynchrone*), de conscience de groupe (nulle, pour certains acteurs, pour tous les acteurs) et la manière de se coordonner (coordination sociale ou assurée par l'ordinateur). Ainsi nous avons décoré les trois situations avec un *&* indiquant qu'elle se joue de manière synchrone. Nous précisons avec que les activités de l'utilisateur 2 sont observables par tous, lors des trois situations, puisqu'il s'agit de manipulations réalisées sur le la table augmentée ou énoncées à voix haute.

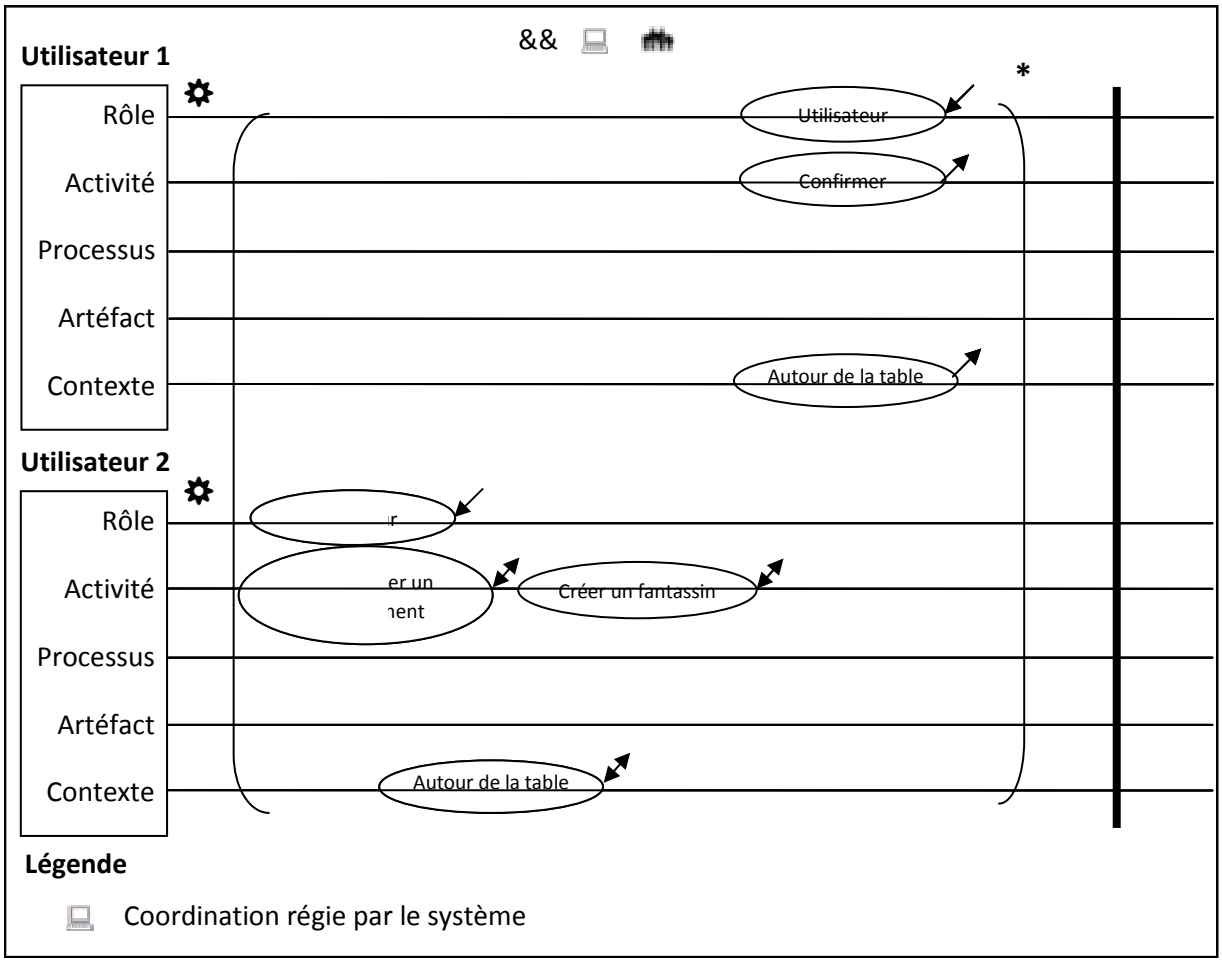


Figure 61 : Création d'un fantassin modélisée avec Orchestra.

La Figure 61 illustre la situation d'interaction collaborative qui consiste à créer un fantassin. Un utilisateur sélectionne un baraquement, puis lance la création d'un fantassin. Le deuxième utilisateur doit confirmer pour que la création soit effective. Cette interaction peut être déclenchée par

n'importe lequel des deux utilisateurs, or Orchestra utilise une mesure par utilisateur, ce qui nous contraint à décrire la situation deux fois (une situation à l'initiative d'un des deux utilisateurs). Ce problème est similaire à celui identifié pour le modèle des tâches CUA (Figure 59). A la Figure 61, nous ne montrons qu'une seule de des deux situations, celle à l'initiative du l'*Utilisateur 2*. Cette situation collaborative est fortement synchrone, ce que nous indiquons par le symbole (&&). Toutes les manipulations sont observables par les deux participants ce qui est indiqué par le symbole ( ). De plus, la collaboration est régie selon les règles du jeu ( ), c'est-à-dire par le système et la reconnaissance des utilisateurs.

Une extension d'Orchestra a été proposée [David 2007] afin de décrire la répétition des situations, les sauts ou encore la durée des situations collaboratives. Pour la répétition, Orchestra propose quatre options :

- Indiquer qu'une situation est répétée un nombre déterminé de fois ;
- Préciser qu'une situation peut être répétée un nombre indéterminé de fois avec l'opérateur (*). Nous avons illustré la répétition d'une situation à la Figure 61;
- Préciser une condition d'arrêt pour une situation selon une condition logique ;
- Préciser une condition d'arrêt pour une situation selon une condition temporelle.

Orchestra propose aussi de décrire des sauts entre une situation et une autre. Enfin, concernant la durée des situations collaboratives, il peut être utile de préciser s'il s'agit d'une collaboration d'une durée importante ou ponctuelle telle que la situation de la Figure 61.

Dans [David 2007], la possibilité de décrire des patrons d'interaction (*pattern*) est introduite. Des exemples de patrons incluent la situation de question-réponse ou de validation sous la forme d'une combinaison des symboles d'annotation de situation. L'utilisation de patrons sur un cas d'étude complet est illustrée dans [David 2009].

La modélisation des situations collaboratives avec Orchestra nous aider à identifier plusieurs situations distinctes correspondant aux différentes activités de WCCM. Ce découpage incite à préciser les rôles, les artéfacts et le contexte liés à chacune des activités, ainsi que leurs relations avec les acteurs. La modélisation des rôles endossés par les acteurs au cours du temps s'exprime par l'intermédiaire de la succession de situations.

La conscience de groupe pour chaque situation est très simple à décrire par l'intermédiaire de décorations de situation. Ces décorations permettent d'expliciter l'observabilité des actions et des artéfacts aux membres du groupe. Néanmoins, dans le cas de la conscience partielle destinée à certains acteurs, il n'est pas possible de préciser quels sont les acteurs concernés. La modélisation de la conscience de groupe est renforcée par l'intermédiaire des décorations des différents éléments de la portée musicale et permet de répondre ainsi aux questions : *Qui* fait *quoi* ? Sur *quoi* ? Observable par *qui* ?

Conclusion sur Orchestra

Pour conclure, Orchestra offre un support intéressant à la fois pour la modélisation du flot de travail, et pour la description de scénarios d'usage. Dans ce cadre, elle permet de mettre en relief la dynamique des rôles endossés par les utilisateurs au cours du temps et de leurs activités. Elle permet également d'établir clairement les tâches collaboratives, coopératives ou individuelles par le biais

des différentes portées (une par utilisateur). La description des concepts complémentaires que sont les artéfacts et le contexte des situations permet de préciser l'environnement de chaque situation. Cette notation nous semble donc pertinente pour l'analyse de la tâche à haut niveau d'abstraction, mais plus difficile à employer dans le cas d'une phase de spécification détaillée. En ce sens, les descriptions Orchestra peuvent être efficaces pour définir un ensemble de situations de travail dont les tâches utilisateurs restent à un haut niveau d'abstraction. Un arbre de tâches détaillé peut ensuite être proposé pour chaque tâche de haut niveau d'abstraction par exemple avec CTT.

2.3.7 Synthèse

Les notations abordées dans cette section prennent en charge la spécification de l'interaction multiutilisateur, qu'elle soit coopérative ou collaborative.

Toutes les notations traitent l'aspect coopératif de l'interaction au travers des concepts de rôles, d'acteurs et d'utilisateurs. Cela repose sur la description d'un modèle de tâches coopératives sous la forme d'un arbre de tâches (GTA, CUA) ou d'un flot de travail (CIAN, MABTA, Orchestra). De plus, les notations GTA, CIAN et MABTA proposent un modèle spécifique pour décrire les relations entre les groupes, rôles et acteurs d'une application multiutilisateur.

Par contraste, l'aspect collaboratif de l'interaction n'est abordé que par deux notations seulement : CIAN et CUA qui introduisent le concept de tâche collaborative. Ce concept permet de décrire un arbre de tâches dont les tâches peuvent être réalisées indifféremment par des utilisateurs jouant un ensemble de rôles métier. Cependant, les sous-tâches d'une tâche collaborative ne peuvent être associées à aucun rôle ce qui induit un manque de précision, et rend difficile la description de tâches telles que *Créer un soldat* de notre cas d'étude.

D'autres aspects liés aux systèmes multiutilisateurs sont également mis en exergue par les différentes notations. Ainsi, l'observabilité des actions des autres ou encore le partage d'objets entre les utilisateurs sont abordés et intégrés aux notations sous la forme de décorations.

De plus, les notations de cette section traitent peu de la dimension concrète de l'interaction et pas du tout de la dimension multimodale. En effet, les notations d'arbre de tâches GTA, CUA, CIAN, MABTA, et CIAN permettent de décomposer l'interaction jusqu'au niveau des tâches élémentaires décrivant des opérations de base sur les concepts du domaine. Seul NUAN intégré à la notation GTA permet de décrire l'interaction concrète en termes d'actions sur les dispositifs tels qu'une souris ou un clavier. En revanche, NUAN ne permet pas la description de l'usage d'une modalité vocale ou gestuelle.

En synthèse, la description de l'interaction abstraite multiutilisateur proposée par les notations existantes satisfait nos objectifs en ce qui concerne la dimension coopérative. En revanche, l'interaction collaborative telle qu'elle est décrite manque de précision. Enfin, la dimension concrète et multimodale de l'interaction n'est pas traitée dans les notations présentées à l'exception de NUAN qui se limite à l'interaction au clavier et à la souris.

2.4 Domaine : Multimodalité

Les notations traitant explicitement la dimension multiutilisateur de l'interaction étant présentées, nous décrivons dans cette section les notations consacrées à la dimension multimodale de l'interaction.

Nous présentons d'abord la notion de configuration d'entrée ICOM, puis la notation de réseaux de Petri ICO et la notation de configuration d'assemblage de composants ICARE. Nous décrivons enfin la plateforme Dynamo-AID et la notation CIS. Bien que la plateforme Dynamo-AID soit issue de travaux sur l'informatique pervasive, l'incrément proposé au modèle de tâches dans la description des modalités d'interaction nous a incité à présenter Dynamo-Aid dans cette section.

2.4.1 ICOM

La notation ICOM (Input Configuration Model) [Dragicevic 2001, Dragicevic 2004, Navarre 2006] permet de décrire les entrées du système sous la forme d'un diagramme d'un flot de données composé de modules interconnectés. ICOM permet de décrire l'interaction concrète Post-WIMP et multimodale pour la réalisation de tâches données d'un système interactif et ne se préoccupe pas de l'aspect multiutilisateur. La notation ICOM est opérationnalisée au sein d'un outil de prototypage rapide, l'environnement ICON, qui permet de spécifier des diagrammes selon la notation ICOM mais également de les exécuter.

Pour décrire une configuration d'entrées, la notation ICOM utilise des entités de base appelées *dispositif*. Les entités disposent d'entrées et de sorties qui sont définies sous la forme de *ports* d'entrée et de sortie. Ces ports sont définis par un type de base tel que le type booléen ou le type entier, ou sont composés pour former un type complexe. Les dispositifs peuvent être de trois types : des dispositifs d'interaction tels qu'une souris, des dispositifs d'application qui correspondent aux tâches élémentaires proposées par le système et les dispositifs d'adaptation qui permettent de transformer des informations provenant des dispositifs d'interaction dans une forme compréhensible par les dispositifs d'application. Une configuration d'entrées est ainsi définie par un ensemble de dispositifs interconnectés.

La Figure 62 illustre la configuration d'entrées pour l'application WCCM. Pour interagir avec le système, l'utilisateur est filmé par une caméra définie comme dispositif d'interaction et sa voix est captée par un microphone. La vidéo est transformée par un dispositif d'adaptation qui effectue une reconnaissance de gestes et qui identifie ainsi les coordonnées du doigt de l'utilisateur sur la table. De manière similaire, un dispositif d'adaptation effectue la reconnaissance vocale. Enfin, le dispositif d'application WCCM prend en entrée le point correspondant aux coordonnées du doigt de l'utilisateur sur la table, ainsi qu'une chaîne de caractères (*String*) qui correspond aux commandes vocales.

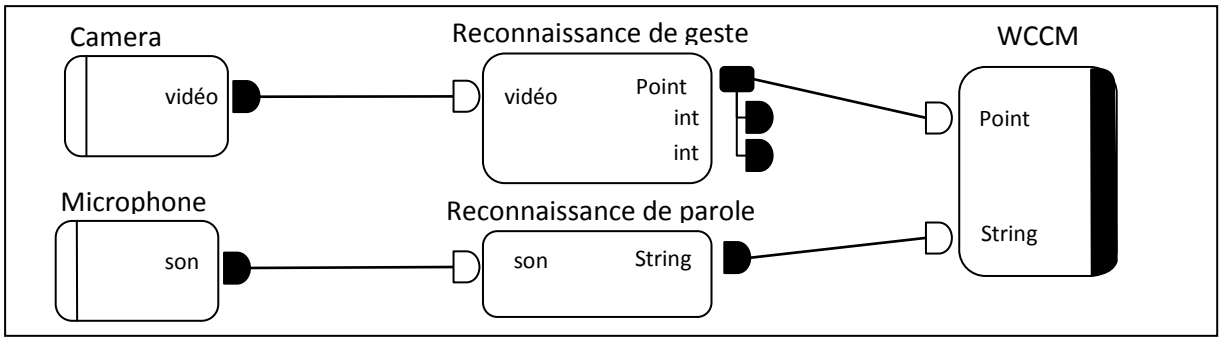


Figure 62 : Configuration d'entrées (ICOM) pour WCCM.

Ce type de modélisation permet de définir les modalités d'interaction mises en jeu pour l'interaction en entrée, en termes de dispositifs physiques d'interaction tels que la caméra et le microphone, mais également en termes de transformations sous la forme des dispositifs adaptateurs. En revanche, le contenu des boîtes, c'est-à-dire le comportement des dispositifs, n'est pas décrit explicitement. Il en découle que le comportement interactif global de l'application ne peut être décrit.

Enfin, la Figure 62 décrit la configuration en entrée pour un utilisateur seulement. La notation ICOM ne prévoit en effet pas de moyen de décrire la présence de plusieurs utilisateurs. Toutefois, la configuration d'entrée décrite ici est également valable pour le deuxième utilisateur puisqu'il interagit avec le système aux travers des mêmes modalités d'interaction.

Conclusion sur ICOM

La notation ICOM propose de décrire l'interaction concrète multimodale sous la forme d'un flot de données entre les dispositifs physiques d'interaction et les tâches. ICOM est plus orienté vers le prototypage de l'interaction multimodale à l'aide de l'outil ICON que vers la description de l'interaction. En effet, l'interconnexion d'entités disposant d'entrées et de sorties ne permet pas de décrire en détail le comportement interactif attendu puisque le comportement interne des entités n'est pas décrit. Enfin, ICOM ne considère pas les interactions multiutilisateurs.

2.4.2 ICO

ICO (*Interactive Cooperative Objects*) est un langage de description de l'interaction dont les bases ont été établies dans [Palanque 1990]. Le principe est de décrire l'interaction par un réseau de Petri. La notation ICO a fait l'objet de plusieurs évolutions pour intégrer l'aspect multimodal [Navarre 2006] et permet de décrire des interfaces multimodales WIMP et Post WIMP [Navarre 2009].

Trois types d'éléments peuvent être décrits au sein d'un diagramme ICO :

- Les places correspondent à l'état de variable de l'application et sont représentées sous la forme d'une ellipse munie d'une étiquette.
- Les transitions correspondent à des actions et sont représentées par un rectangle muni d'une étiquette
- Enfin, les arcs représentés par des flèches connectent les transitions et les places.

L'interaction est ainsi représentée par un ensemble de *places* et de *transitions* reliées entre elles par des arcs. Un dernier élément vient compléter les diagrammes ICO : il s'agit des *jetons* qui peuvent

être disposés sur les places. La configuration des jetons sur les places représente un état de l'interface utilisateur, et renseigne sur les actions que l'utilisateur peut accomplir. Les jetons peuvent être déplacés sur le graphe, de place en place, en suivant des règles strictes. Ces règles de déplacement permettent de décrire l'évolution de l'état de l'interface.

La Figure 63 illustre la modélisation de l'interaction qui concerne la tâche collaborative de création d'un soldat au sein d'un baraquement. La partie gauche de la figure décrit un premier réseau de Petri possédant deux places : *Baraquement sélectionné*, et *Baraquement non sélectionné*. La place *Baraquement non sélectionné* dispose d'un jeton ce qui indique l'état initial du baraquement. Dans cet état, le baraquement peut être sélectionné, ce qui est indiqué par la transition *Sélectionner* et les arcs entre les deux places. La partie droite de la figure décrit un deuxième réseau de Petri composé des trois places. La place initiale étant : *baraquement inactif*. La transition (1) correspond à la demande de création d'un soldat. Celle-ci est réalisable si le baraquement est préalablement sélectionné. Pour représenter cette condition, un arc de test (2) est décrit entre la place *Baraquement sélectionné* et la transition *Créer soldat*. Il indique que la transition *Créer soldat* ne peut être franchie que si un jeton est présent sur la place *Baraquement sélectionné*. Enfin, la création d'un soldat doit être confirmée, ce qui est représenté par la place *Attente de confirmation création soldat* et la transition : *Confirmer*. Le baraquement redevient inactif lorsque le soldat est créé. Ceci est représenté par la transition *Soldat créé* entre la place *Création soldat* et la place *Baraquement inactif*.

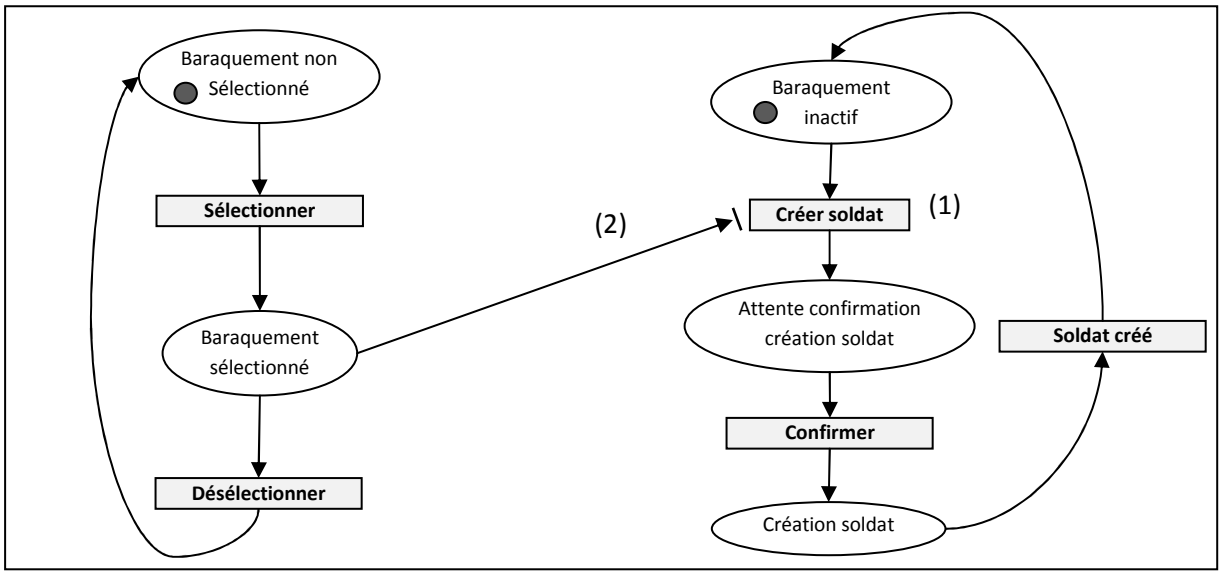


Figure 63 : La création d'un soldat dans un baraquement selon la notation ICO.

L'interaction décrite à la Figure 63 correspond à une tâche collaborative qui met obligatoirement en jeu deux utilisateurs distincts, puisque l'utilisateur qui confirme la création d'un soldat doit être différent de celui qui la demande. La notation ICO est prévue pour représenter des interactions mono-utilisateur, c'est pourquoi la figure ne comporte pas de mention explicitant la présence de deux utilisateurs.

La notation ICO permet de décrire l'interaction concrète telle que l'usage d'une modalité d'interaction. La Figure 64 modélise la reconnaissance de gestes sur la table augmentée de WCCM. Les gestes reconnus par la table sont : le pointage, c'est-à-dire toucher un point de la table ; et le glisser-déposer, c'est-à-dire faire glisser son doigt sur la table d'un point à un autre. Pour expliciter

ces comportements, ICO permet de décrire des événements générés lors du passage d'une transition. Ainsi, la transition *Relever son doigt* (1) génère un événement "toucher" lorsqu'elle est franchie. De la même manière, les transitions *Déplacer son doigt* (2), *Déplacer son doigt* (3) et *Relever son doigt* (4) génèrent chacune un événement lorsqu'elles sont franchies. Ces événements correspondent aux trois actions primitives d'un glisser-déposer : "prendre", "glisser" et "déposer".

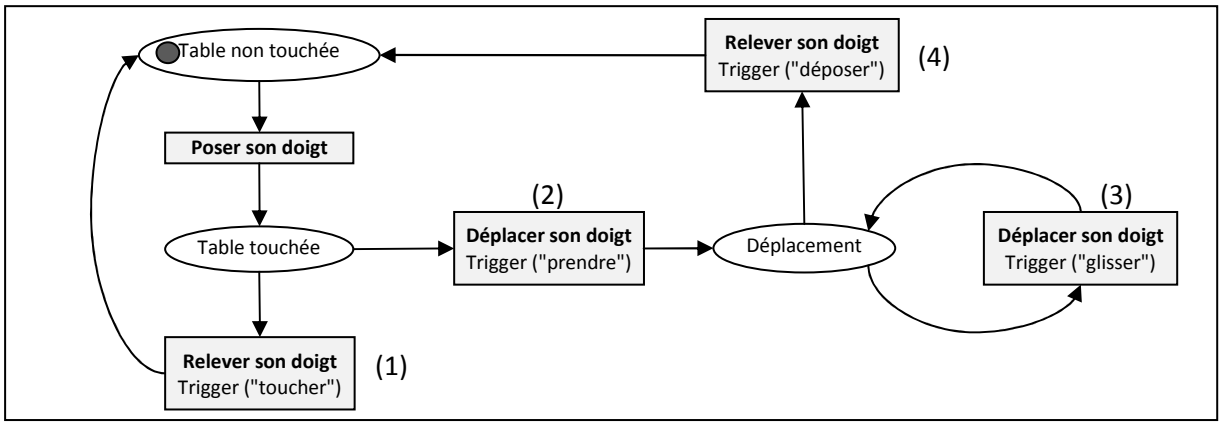


Figure 64 : Modélisation de l'interaction d'un utilisateur avec la table de WCCM selon la notation ICO.

Conclusion sur ICO

ICO propose de décrire l'interaction d'un utilisateur avec le système sous la forme d'un réseau de Petri. A l'instar des diagrammes d'état-transition, ce type de représentation permet de décrire très précisément le comportement du système. Le dialogue utilisateur-système peut être décrit aussi bien à haut niveau d'abstraction en termes de tâches (Figure 63), qu'à bas niveau d'abstraction en termes d'utilisation de modalités d'interaction (Figure 64). Enfin, la notation ICO se limite au système interactif mono-utilisateur, mais des extensions sont possibles telles que l'usage de jetons colorés pour représenter des utilisateurs différents.

2.4.3 ICARE

A l'instar de la notation ICOM, ICARE (Interaction Complémentarité Assignation Redondance Equivalence) [Bouchet 2006] permet de décrire des assemblages de composants au sein d'un diagramme de flot de données pour décrire l'interaction multimodale en entrée d'un système interactif. ICARE repose sur la définition d'une modalité comme un couple <dispositif physique, langage d'interaction>, et propose ainsi de décrire une modalité sous la forme de deux composants : un composant dispositif et un composant langage d'interaction. Pour décrire des combinaisons de modalités d'interactions en entrée, ICARE propose des composants de combinaison : la complémentarité, la redondance, et la redondance/équivalence, qui reposent sur les propriétés CARE (section 2.2.2.1 du Chapitre 1).

Avec ICARE, pour chaque tâche d'une application interactive, une description sous forme d'un assemblage de composants est produite. Celle-ci permet d'exprimer la modalité ou combinaison de modalités utilisée pour réaliser la tâche. Le diagramme de la Figure 65 décrit la tâche multimodale *Déplacer un soldat* de l'application WCCM. Pour réaliser cette tâche, un utilisateur doit énoncer la commande « aller ici » pendant qu'il pointe du doigt un emplacement de la carte. Le diagramme identifie deux dispositifs physiques : la *Caméra*, et le *Microphone*. Pour chacun de ces dispositifs, un

composant langage d'interaction est décrit, respectivement *Reconnaissance de geste* et *Reconnaissance de parole*. Les flèches entre les composants indiquent le sens de la communication entre les composants. Ainsi, le composant *Caméra* émet des données vers le composant de R*econnaissance de geste*. L'assemblage de ces deux composants forme une modalité gestuelle. De la même manière, les composants *Microphone* et *Reconnaissance de parole* forment une modalité vocale. Un composant de C*omplémentarité* est représenté sous la forme d'un rectangle au bord vert muni de plusieurs ports d'entrée. Les liens entre la modalité gestuelle, la modalité vocale et le composant de *complémentarité* indiquent que les deux modalités sont complémentaires pour la réalisation de la tâche *déplacer un soldat*.

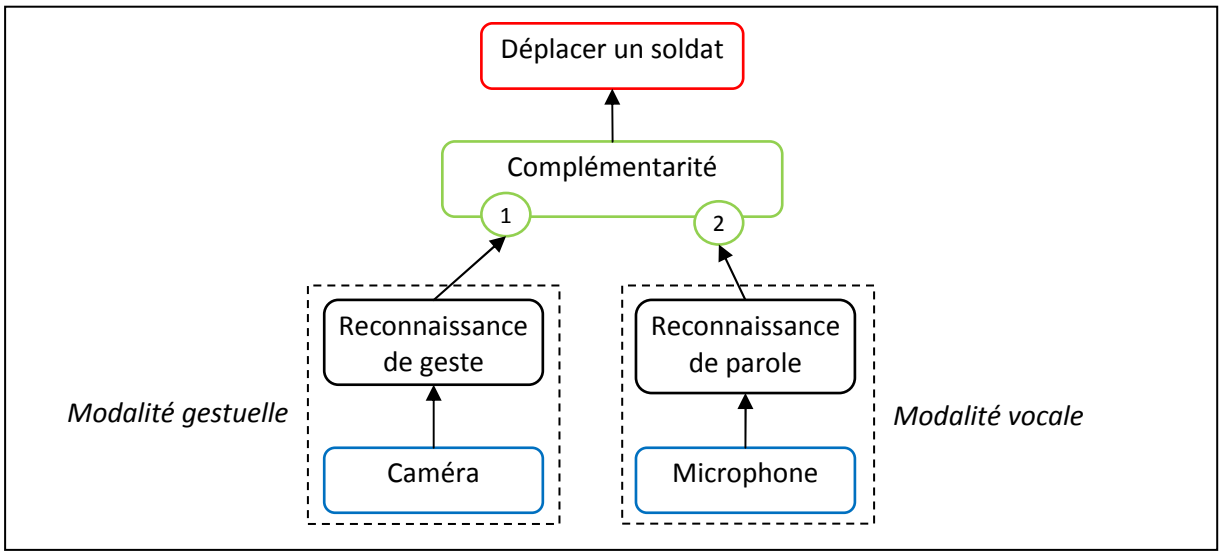


Figure 65 : Assemblage de composants ICARE pour la tâche multimodale de déplacement d'un soldat.

Au sein de l'outil de prototypage ICARE, les diagrammes font apparaître des composants qui sont décrits par de nombreuses propriétés telles que la précision, la stabilité ou encore le mode de communication. Les assemblages produits peuvent alors être exécutés, et les combinaisons de modalités testées, afin de valider l'utilisation de ces modalités pour une tâche donnée.

ICARE possède plusieurs limites : Premièrement, seules des combinaisons de modalités d'interaction en entrée d'un système peuvent être spécifiées et testées. En effet, ICARE ne prévoit pas la possibilité d'exprimer l'interaction multimodale en sortie, ce qui rend également l'expression de rétroaction du système impossible. Deuxièmement, l'interaction multiutilisateur n'est pas considérée. Troisièmement, comme la notation ICOM, le comportement réel d'un assemblage n'est pas décrit explicitement. En effet, ce comportement dépend du comportement des composants qui n'est pas explicité dans les diagrammes. Ainsi, la Figure 65 indique l'usage d'une modalité gestuelle, mais elle ne renseigne pas sur le geste qui est reconnu. De même, la complémentarité des deux modalités est indiquée mais l'assemblage ne fournit pas d'indication sur l'ordre d'utilisation des modalités. Enfin, les composants sont reliés entre eux par des ports qui ne sont pas décrits explicitement sur les diagrammes.

Le modèle ICARE à été étendue au sein du modèle OpenInterface [Serrano 2010]. Ce dernier à donné lieu à une plateforme logicielle pour le développement de l'interaction multimodale en entrée qui a été développée par un consortium européen OpenInterface [OpenInterface]. Certaines limites d'ICARE vis-à-vis de nos objectifs de spécification de l'interaction restent présentes dans

OpenInterface, notamment, l'impossibilité de décrire le comportement interne des composants. En revanche, les ports d'entrée et de sorties des composants OpenInterface sont explicites.

Conclusion sur ICARE

ICARE est une approche adaptée pour décrire l'interaction multimodale en entrée car les diagrammes font apparaître explicitement les dispositifs physiques, les langages d'interaction qui forment des modalités, ainsi que les combinaisons de modalités au travers de l'utilisation des propriétés CARE. En revanche, ICARE se limite à l'interaction multimodale mono-utilisateur en entrée d'un système interactif. Enfin, le comportement d'un diagramme reste de très haut niveau d'abstraction si aucune description complémentaire n'est apportée pour décrire les composants.

2.4.4 Dynamo-AID

Issue de travaux sur l'informatique pervasive, DynaMo-AID [Clerckx 2005] est une méthode de conception orientée prototypage pour les applications sensibles au contexte. L'adaptation au contexte peut s'effectuer par la répartition de l'interface homme machine sur plusieurs supports et par l'utilisation de modalités et de combinaisons de modalités. En effet, l'extension de DynaMo-AID [Clerckx 2008] ajoute la possibilité d'associer des combinaisons de modalités explicitement basées sur les propriétés CARE (section 2.2.2.1 du Chapitre 1).

Dans le cadre de DynaMo-AID, plusieurs notations sont utilisées pour modéliser l'interaction : un modèle de tâches, un modèle de dialogue, un modèle de contexte et un modèle de présentation. Tous ces modèles sont intégrés au sein d'un modèle de l'interface qui explicite leur interrelation. Ce dernier modèle définit le prototype d'une application qui peut ainsi être exécuté.

Pour décrire le modèle de tâche, les auteurs utilisent la notation CTT étendue. La principale extension est la possibilité de décrire, pour une tâche donnée, une modalité ou une combinaison de modalités pour la réaliser. Pour cela, les propriétés CARE sont utilisées. La Figure 66 reprend la tâche *Déplacer un soldat* décrite au sein de l'arbre de tâches individuelles CTT de la Figure 34 pour le rôle de S*ergent*. Pour déplacer un soldat, le *Sergent* doit d'abord le sélectionner, ce qui est indiqué par la tâche *Sélectionner un soldat* en le pointant du doigt sur la table. Pour décrire ceci, une étiquette *A[gestuelle]* est associée à la tâche *Sélectionner un soldat*. Le *[gestuelle]* indique la modalité utilisée. Le *A* correspond à la propriété CARE d'Assignation. A[gestuelle] indique donc que la modalité gestuelle est Assignée à la tâche S*électionner un soldat*, ce qui signifie que seule cette modalité peut être utilisée pour réaliser la tâche. Lorsqu'un soldat est sélectionné, le sergent peut le déplacer. Pour cela, il énonce la commande « aller ici », pendant qu'il désigne un point de la table avec son doigt. La modalité gestuelle et la modalité vocale sont utilisées de manière complémentaire pour réaliser cette tâche. Ceci est indiqué par l'étiquette *C[gestuelle, vocale]* attachée à la tâche *Déplacer le soldat*. Le *C* fait référence à la propriété CARE de Complémentarité.

Une autre extension au modèle de tâches est réalisée pour décrire des points de décision entre plusieurs contextes d'utilisation sous la forme de tâches de décision. Une tâche de décision est liée à un contexte donné qui est décrit au sein du modèle de contexte. Plusieurs sous-tâches peuvent être exprimées en lien avec ce contexte d'utilisation.

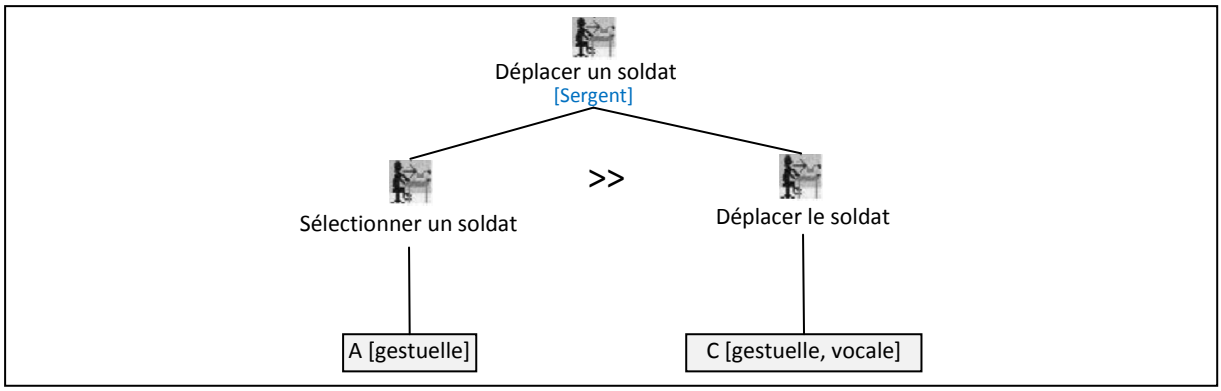


Figure 66 : Interaction multimodale pour déplacer un groupe modélisée avec Dynamo-AID.

Le modèle de dialogue avec Dynamo-AID est décrit sous la forme d'un diagramme d'état-transition. Chaque état décrit un état de l'interface utilisateur, tandis que les transitions entre états sont décrites en termes de réalisation de tâches du modèle de tâches. Cela permet d'exprimer le lien entre le modèle de tâches et modèle de dialogue. Les diagrammes d'état-transition sont décrits en détails à la section 2.1.1 de ce chapitre consacrée à UML.

Le modèle de contexte est décrit graphiquement par des agrégations d'informations et leurs associations à une tâche de décision comme l'illustre la Figure 67. Dans le cas de l'application WCCM, il y a un unique contexte d'utilisation : *Jouer à WCCM*. Le contexte est défini par les objets ou des informations sur le contexte notés CCO (*Concrete Context Object*). La Figure 67 en contient quatre : une caméra, un microphone, des écouteurs et la table. Ces CCO sont regroupés ensembles pour définir une entité abstraite de contexte notée ACO (*Abstract Context Object*). Un ACO est ensuite lié à la tâche *Jouer à WCCM* par le biais d'une tâche abstraite *Utiliser la table augmentée*.

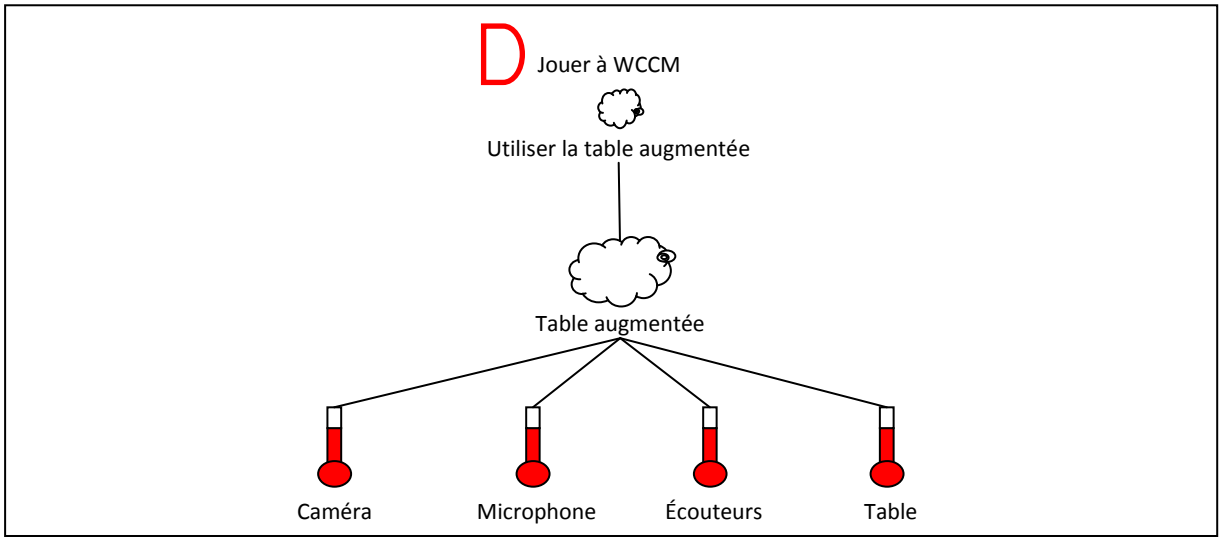


Figure 67 : Contexte d'utilisation de la table augmentée de WCCM selon Dynamo-AID.

Enfin, le modèle de présentation permet de décrire les composants de l'interface abstraite de manière hiérarchique. Ces composants sont ensuite liés aux tâches du modèle de tâches.

Les différents modèles proposés peuvent être décrits grâce au support d'un outil logiciel. Celui-ci permet également d'exporter les descriptions au format XML.

Conclusion sur Dynamo-AID

La méthode de développement Dynamo-AID permet d'exprimer le modèle de tâches d'une application en tenant compte du contexte d'exécution. La méthode introduit une extension du modèle de tâches CTT pour décrire l'usage de modalités ou de combinaisons de modalités d'interaction pour la réalisation d'une tâche donnée. La description de combinaisons repose sur les propriétés CARE, mais ne permet pas l'expression de relations temporelles précises, telles que le la relation "durant" de Allen [Allen 1983]. D'autre part, les descriptions de modalités indiquent avec quoi interagit l'utilisateur (dispositif), mais pas comment (langage). Enfin, Dynamo-AID n'est pas prévu explicitement pour les systèmes multiutilisateurs, même si CTT permet de décrire des interactions coopératives.

2.4.5 CIS

La notation CIS (*Complexity of Interaction Sequences*) [Appert 2007] prend en charge la description de techniques d'interaction dans une forme proche d'un diagramme d'état-transition. CIS permet de décrire des états de l'interface utilisateur, ainsi que les actions utilisateur possibles pour ces états. Plusieurs actions utilisateur forment ainsi des commandes qui correspondent aux tâches élémentaires d'un arbre de tâches CTT ou K-MAD, telles que la tâche *Déplacer un soldat* identifiée avec plusieurs modèles de tâches. CIS propose de décrire et d'évaluer les techniques décrites, ce qui permet au concepteur d'explorer différentes techniques d'interaction pour une tâche donnée.

A l'instar de NUAN, cette notation permet de représenter l'interaction concrète sous la forme de séquences d'actions utilisateur pour la réalisation d'une tâche donnée. Elle présente le même inconvénient, c'est-à-dire que les éléments du langage sont décrits en termes de primitives d'interaction au moyen de dispositifs physiques donnés, en l'occurrence, la souris. La Figure 68 illustre la technique d'interaction de la tâche *Déplacer un soldat* à l'aide du geste et de la parole. Ces modalités d'interaction vocales et gestuelles ne sont pas prévues dans la notation, aussi, nous introduisons les primitives *Toucher* pour la modalité gestuelle, et *Dire « aller ici »* pour la modalité vocale.

Au sein de la Figure 68, une condition doit être remplie pour réaliser la tâche. Celle-ci est décrite au dessus de l'état initial (représenté par un cercle gris) : il s'agit qu'un soldat soit présent sur la carte. La transition portant le numéro (1) correspond à une action d'acquisition consistant pour l'utilisateur à amener son doigt au-dessus du soldat. L'action de validation (2) consiste pour l'utilisateur à toucher le soldat. Toucher le soldat le rend sélectionné ce qui est indiqué par *S.sélectionné := on* dans le cadre gris de la transition (2). Une pré-condition est décrite dans le cadre gris de la transition (1) : *S.selectionné == on*. Celle-ci indique que si le soldat (S) est sélectionné, alors l'action de le sélectionner devient optionnelle. Les transitions (1) et (2) amènent l'interface dans un état où le soldat peut recevoir un ordre de déplacement. Pour donner un ordre de déplacement, l'utilisateur doit dire « aller ici » pendant qu'il touche du doigt un emplacement sur la carte. La notation CIS permet de décrire des actions parallèles en reliant deux transitions par un trait gris épais. Les transitions (3a) et (3b) sont ainsi reliées pour décrire le fait qu'elles doivent être réalisées en parallèle, même si une description plus fine serait souhaitable pour décrire qu'une action s'effectue *pendant* une autre. La partie (3a) décrit d'une part une transition d'acquisition correspondant à une action de l'utilisateur d'amener son doigt au dessus de la carte, puis à une action de validation qui

consiste à *toucher* la carte. La partie (3b) quant à elle décrit une action de validation consistant à *dire « aller ici »*.

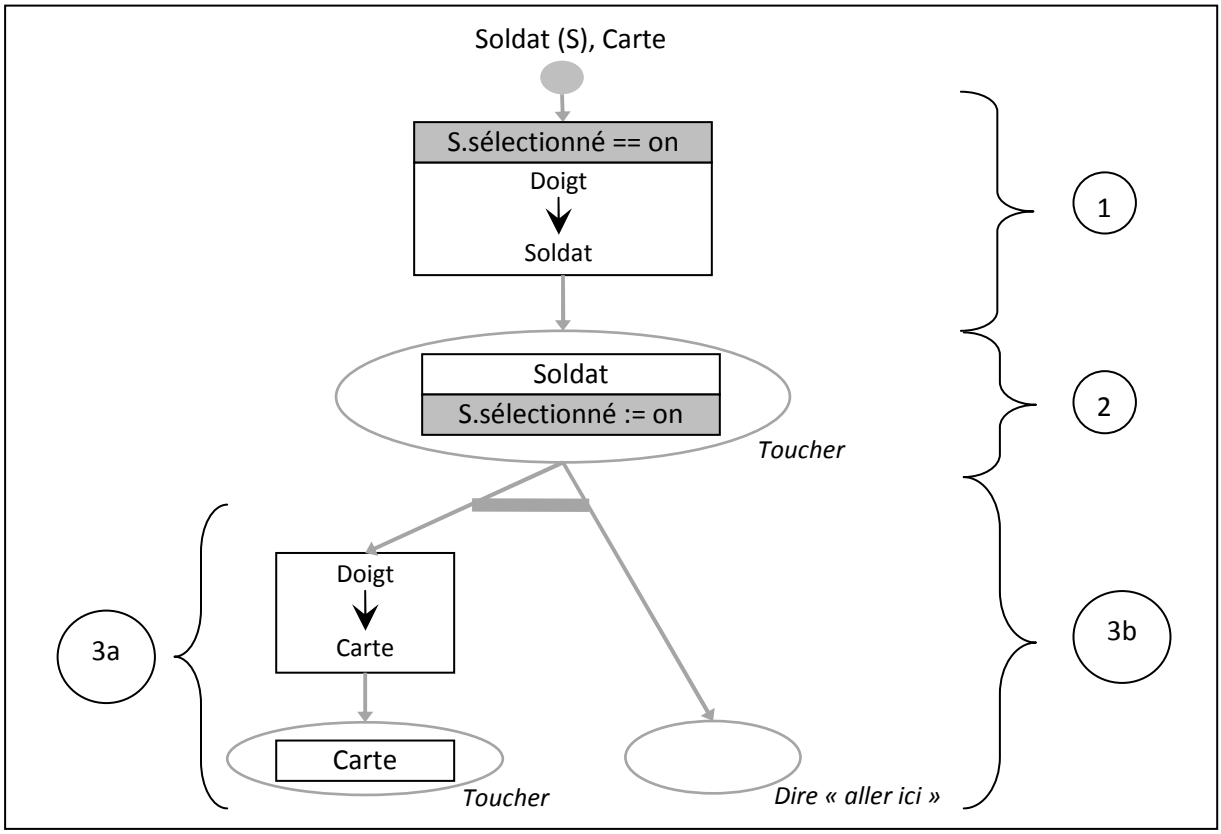


Figure 68 : Déplacer un soldat avec CIS.

Conclusion sur CIS

La notation CIS permettant la description de séquences d'interaction, et même d'interactions parallèles, est un outil intéressant pour décrire l'interaction concrète de systèmes interactifs. Au travers de l'utilisation de cette notation, l'usage de modalités autres que celles initialement envisagées par la notation, notamment le geste et la parole, permet d'identifier son intérêt pour décrire l'interaction concrète multimodale. La multimodalité doit être par ailleurs plus précise en ce qui concerne le parallélisme. Enfin, la notation CIS n'est pas prévue pour décrire l'interaction concrète multiutilisateur et ne bénéficie pas d'un outil logiciel pour réaliser les spécifications.

2.4.6 Synthèse

Nous avons présenté dans cette section plusieurs notations permettant d'exprimer l'interaction multimodale. Parmi elles, les notations ICOM et ICARE visent le prototypage rapide d'applications multimodales, et reposent sur l'usage de composants logiciels préexistants. Le concept de base de ces notations est de distinguer des composants de dispositifs physiques d'interaction, des composants de langage d'interaction ou d'adaptation, et des composants correspondant aux tâches d'une application interactive. L'assemblage de composants décrit alors QUELLE modalité d'interaction ou QUELLE combinaison de modalités est utilisée pour réaliser une tâche donnée. Cependant, l'absence de description des composants induit qu'un assemblage ICARE ou qu'une

configuration d'entrées ICOM ne décrit pas COMMENT sont utilisés les dispositifs. Enfin, les deux approches se limitent à l'entrée du système interactif, c'est-à-dire de l'utilisateur vers le système.

A l'instar de la notation ICARE, la plateforme Dynamo-AID propose de décrire pour chaque tâche d'un système QUELLE est la modalité ou QUELLE combinaison de modalités est utilisée pour la réaliser. L'association d'une modalité ou d'une combinaison de modalités repose sur l'usage des propriétés CARE. Néanmoins, les deux notations ne permettent pas de décrire précisément les relations temporelles entre plusieurs modalités.

La notation ICO permet aussi bien la description de l'interaction à haut niveau d'abstraction que la description fine de l'usage d'une modalité. Cependant, la notation ne fournit aucune aide ou guide pour aboutir à une modélisation de ces aspects.

Contrairement aux autres notations, CIS propose de décrire l'usage de modalités d'interaction pour réaliser une tâche donnée en termes d'actions élémentaires. Cependant, les relations temporelles sont décrites à haut niveau d'abstraction par des opérateurs de parallélisme et de séquence. De plus, comme NUAN, cette notation doit être étendue pour prendre en compte des modalités d'interaction telles que la parole ou le geste.

Pour conclure, les différentes notations permettent d'exprimer l'usage d'une modalité ou d'une combinaison de modalités pour une tâche donnée. Les descriptions sont cependant trop abstraites pour décrire COMMENT les modalités sont utilisées. De plus, les relations temporelles disponibles pour exprimer les combinaisons de modalités ne sont pas précises. Aussi, sur la base de ce que proposent ces notations, des évolutions doivent être envisagées pour décrire COMMENT les modalités sont utilisées et dans quel ordre.

2.5 Domaine : Réalité Augmentée

Cette section regroupe plusieurs notations élaborées dans le cadre du domaine de la réalité augmentée, sous-domaine de l'interaction homme machine. La réalité augmentée met en jeu de nouveaux dispositifs d'interaction. Aussi les modèles pour la réalité augmentée décrivent souvent ces dispositifs et comment ils sont utilisés. Ce sous-domaine de l'interaction homme-machine semble donc pertinent à étudier dans le cadre de notre étude pour la description de l'interaction concrète.

Dans cette section nous décrivons d'abord la notation ASUR, puis IRVO et enfin le modèle d'interaction mixte.

2.5.1 ASUR

ASUR (*Adaptor, System, User, Real object*) [Dubois 2002a, Dubois 2002b] est une notation graphique pour spécifier les interactions utilisateur au sein de systèmes de réalité augmentée. La notation ASUR propose de décrire un système interactif sous la forme d'un ensemble constitué de quatre types d'entités, qui sont appelés des composants. Les composants *S* représentent le système. Les composants *U* correspondent aux utilisateurs du système. Les composants *R* sont des objets réels impliqués dans la réalisation de tâches en tant qu'outils (*Rtool*) ou en tant qu'objets de la tâche (Robject). Enfin, les composants *A* sont des adaptateurs pour les entrées (*Ain*) ou pour les sorties (Aout). Ces adaptateurs font le lien entre les trois autres types de composants. ASUR permet de décrire les relations entre les composants de deux manières. Une flèche simple entre deux

composants décrit un échange de données orienté selon le sens de la flèche, tandis qu'une flèche en pointillé indique que l'entité à la source de la flèche est représentée par l'entité à la destination de la flèche. Enfin, un trait doublé entre deux entités indique un lien physique entre deux composants.

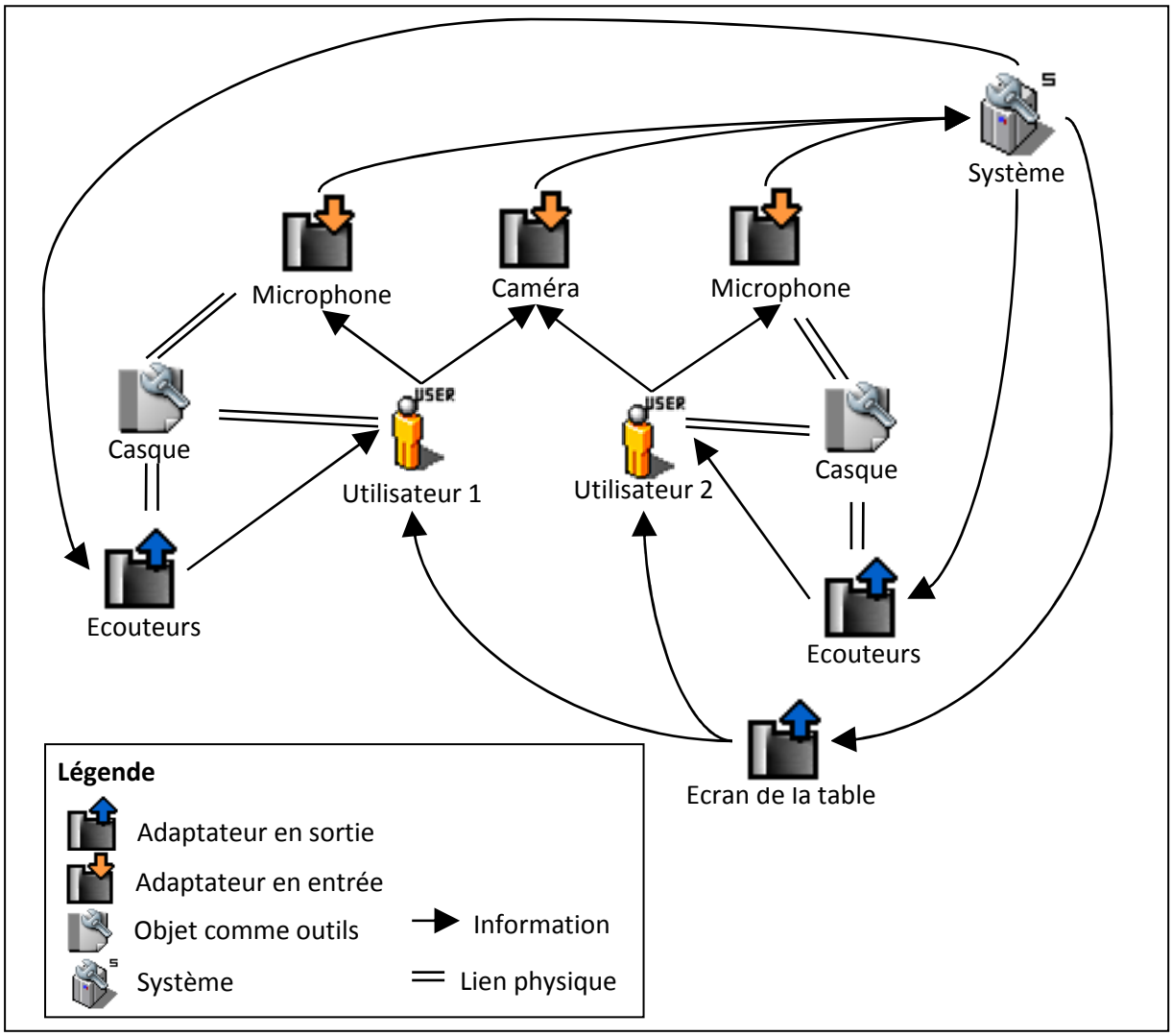


Figure 69 : Description ASUR de l'interaction autour de la table de WCCM.

Bien qu'ASUR soit prévu pour des systèmes de réalité augmentée, nous l'utilisons pour spécifier l'application WCCM. Il s'agit en effet d'une des rares notations permettant de décrire les dispositifs physiques d'interaction et leurs relations avec les utilisateurs. La Figure 69 illustre la modélisation de l'interaction autour de la table pour l'application WCCM avec la notation ASUR. L'outil d'édition et d'analyse de spécification ASUR, GuideMe [Viala 2004], introduit des icônes représentant les différents composants ASUR qui sont utilisées à la Figure 69. Deux *Utilisateurs* y sont représentés, la représentation de rôle n'étant pas possible. Chaque *Utilisateur* porte un *Casque*, ce qui est représenté par le lien physique entre les *Utilisateurs* et les *Casques*. Chaque *Casque* est lui-même lié à un *Microphone* et des *Ecouteurs*. En entrée du système, les gestes des utilisateurs sont captés par une *Caméra*, et les commandes vocales des utilisateurs sont captées par les M*icrophones* des *Casques*. En sortie du système, l'*Ecran de la table* affiche l'état du jeu tandis que les *Ecouteurs* diffusent de la musique et des sons. Un intérêt de cette représentation est d'expliciter les dispositifs partagés par les utilisateurs : à la Figure 69, l'*Ecran de la table* (adaptateur de sortie) et la caméra qui

capte les mouvements sont partagés par les utilisateurs. A l'opposé, chaque utilisateur dispose de son propre casque.

La Figure 69 décrit l'interaction globale entre les utilisateurs et le système. ASUR permet également de décrire des situations particulières d'interaction. La Figure 70 illustre par exemple l'interaction d'un utilisateur avec un bâtiment *Hôtel de ville* pour la création d'un *Paysan*. Elle comporte un utilisateur et les adaptateurs déjà présents à la Figure 69. Elle comporte également deux entités virtuelles : *Paysan* et *Hôtel de ville*. Un tel diagramme permet de modéliser les interactions multimodales en entrée et en sortie du système. Ainsi, l'interaction de l'utilisateur avec le bâtiment *Hôtel de ville* est réalisée au travers d'une combinaison de modalités d'entrée : le geste et la parole captés respectivement par la caméra et le microphone. La création du paysan correspond quant à elles une combinaison de modalités de sortie. Le paysan émet le son « je suis prêt » tandis qu'il devient visible sur l'écran de la table.

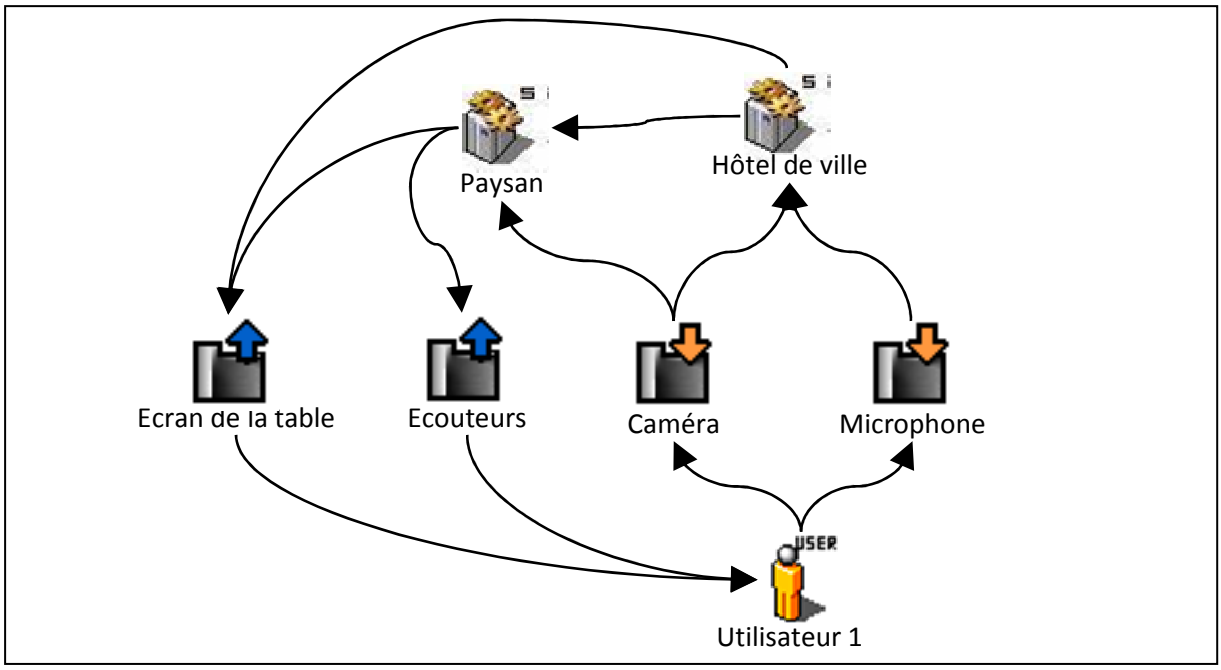


Figure 70 : Description ASUR de l'interaction avec un bâtiment Hôtel de ville pour la création d'un Paysan dans WCCM.

La notation ASUR peut servir à préciser la description d'une ou plusieurs tâches particulière d'un sytème en association avec d'autres notations. Comme l'association GTA et NUAN (décrite à la section 2.3.1), dans [Charfi 2009], ASUR est employé conjointement avec la notation d'arbre de tâches K-MAD [Lucquiaud 2005, Baron 2006] (présentée à la section 2.2.2 de ce chapitre). Cette association entre les deux notations repose sur des règles de construction, et sur une frontière qui délimite les élements devant être décrits avec K-MAD et ceux qui doivent être décrits avec ASUR. Ainsi par exemple, la première règle indique qu'un raffinement d'une tâche K-MAD avec la notation ASUR implique que la tâche ASUR soit de type Interactive.

Conclusion sur ASUR

Bien qu'initialement prévue pour décrire l'interaction dans les systèmes de réalité augmentée, la notation ASUR présente des intérêts pour la modélisation de systèmes multiutilisateurs et multimodaux. Pour ce qui concerne l'aspect multiutilisateur, l'intérêt principal réside dans la

possibilité de décrire le partage ou l'absence de partage des entités entre les utilisateurs. En ce qui concerne l'aspect multimodal, la notation ASUR permet de décrire les relations entre un utilisateur et un élément du système au travers de combinaisons de modalités d'interaction. Enfin, la notation ASUR reste très abstraite pour ce qui est de la description précise des relations entre les entités. En ce sens, elle peut servir à préciser le cadre général de l'interaction pour des tâches particulières dans le cadre d'une utilisation conjointe avec d'autres notations, telles qu'un modèle de tâches.

2.5.2 IRVO

La notation IRVO (*Interacting with Real and Virtual Objects*) [Chalon 2005] permet de représenter l'interaction entre des utilisateurs et un système de réalité mixte en représentant explicitement les objets réels et virtuels impliqués dans l'interaction, ainsi que leurs relations.

Un diagramme IRVO repose sur des frontières représentées par des lignes. Une ligne horizontale en pointillé représente ainsi la frontière entre le monde réel et le monde virtuel tandis que des lignes pleines verticales explicitent la frontière entre plusieurs lieux.

Au sein des diagrammes, la notation IRVO permet de représenter plusieurs catégories d'entités :

- Les utilisateurs (*U*) dont on peut décrire les canaux de communication : visuel (*V*), auditif (*A*), haptique (*KH*), olfactif (*S*), et gustatif (*T*).
- Les objets qui sont perceptibles ou manipulables par les utilisateurs. Ces objets sont soit des objets du domaine (*O*) qui sont impliqués dans la réalisation des tâches, soit des outils (*T*) qui sont des objets intermédiaires qui aident les utilisateurs à réaliser les tâches.
- Le modèle interne (*M*) de l'application qui représente l'application sans l'interaction concrète.
- Enfin, la transmission d'informations entre le monde réel et virtuel est décrite par des entités appelées transducteur. Les transducteurs qui correspondent à une transmission d'informations du monde réel au monde virtuel sont appelés capteurs (*S*), tandis que ceux qui correspondent à une transmission d'informations du monde virtuel au monde réel sont notés effecteurs *(E*).

Les relations (flèches) représentent les échanges d'informations entre les entités. Les flèches touchent deux entités du même monde ou peuvent traverser un transducteur pour relier une entité réelle et une entité virtuelle. Quand une relation concerne un utilisateur, elle correspond à une action lorsque la flèche part de l'utilisateur, et une perception lorsque la flèche arrive sur l'utilisateur. Un canal de communication peut être spécifié pour une relation. Enfin, une flèche en pointillé indique une relation de faible importance vis-à-vis de la tâche.

IRVO étant dédié aux systèmes de réalité mixte, il est donc possible de décrire des objets mixtes, ceux-ci étant décrits par des boîtes en pointillé disposées à la frontière entre le monde réel et virtuel, et englobant un objet réel et un objet virtuel.

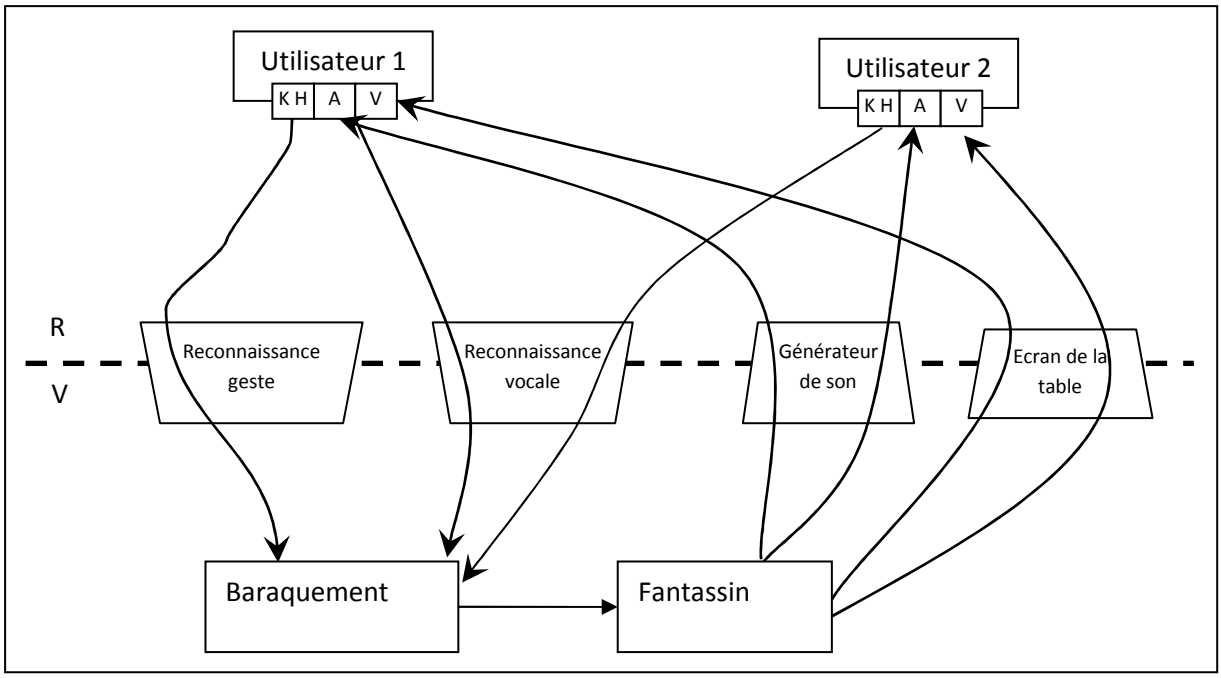


Figure 71 : Modèle IRVO pour la tâche de création d'un fantassin dans WCCM.

La Figure 71 illustre un modèle IRVO pour la tâche de création d'un fantassin. Comme la notation ASUR, IRVO permet de décrire les différents utilisateurs présents pour la réalisation de cette tâche collaborative. Dans le monde virtuel, sont représentés un objet *Baraquement* et un objet *Fantassin*.

Les trois étapes de l'interaction multimodale et collaborative permettant de créer un fantassin sont représentées par trois flèches. La première étape consiste à sélectionner le baraquement, ce qui est décrit par la flèche entre l'*Utilisateur 1* et le *Baraquement* en passant par le transducteur de *Reconnaissance de geste*. La deuxième étape consiste à prononcer à voix haute la création du fantassin, ce qui est décrit par la flèche partant de l'*Utilisateur 1* et rejoignant le *Baraquement* en passant cette fois par la *Reconnaissance vocale*. Enfin, la troisième étape consiste pour le deuxième utilisateur à dire « ok », ce qui est indiqué par la flèche partant de l'*Utilisateur 2* vers le *Baraquement* en passant également par la *Reconnaissance vocale*. Les trois flèches sont associées aux canaux de communication des utilisateurs, respectivement haptique et auditif. Lorsque le fantassin est créé, une combinaison de modalités est utilisée pour avertir les utilisateurs. Il s'agit d'une part de l'affichage du fantassin sur la table et d'autre part de la diffusion d'un son tel que « fantassin, au rapport ». Ceci est figuré par les flèches partant de l'objet virtuel *Fantassin* et rejoignant respectivement les canaux de communication visuel et auditif des deux utilisateurs.

Conclusion sur IRVO

A l'instar de la notation ASUR, la notation IRVO permet de représenter des situations d'interaction multiutilisateur et multimodale. Cependant pour notre étude, la notation IRVO semble plus précise sur deux points. D'une part, la division verticale entre le monde virtuel et monde réel permet d'expliciter l'appartenance de chaque objet de manière plus évidente que des icônes. D'autre part, la spécification du canal de communication de l'utilisateur rend explicite la multimodalité dans le cas où elle fait intervenir plusieurs canaux de communication différents. Enfin, IRVO, comme ASUR ne suffit pas à décrire seule l'interaction, mais peut être utilisé en complément de modèle de tâches par exemple.

2.5.3 Modèle d'interaction mixte

Le modèle d'interaction mixte [Coutrix 2009] pour décrire l'interaction dans le cadre de systèmes de réalité mixte repose sur le concept d'objet mixte, c'est-à-dire d'objet qui possède des propriétés physiques et numériques. Nous adoptons l'acronyme MIM pour le Modèle d'Interaction Mixte.

Un objet mixte est défini comme la réunion d'un objet physique et d'un objet numérique au sein d'une même entité. Cet objet mixte possède ainsi des propriétés physiques de l'objet physique et des propriétés numériques de l'objet numérique. La définition du lien entre ces deux types de propriétés repose sur la définition de la multimodalité comme un couple <Dispositif physique, Langage d'interaction>.

Un diagramme d'interaction mixte peut être utilisé pour décrire l'interaction d'un utilisateur avec un ou plusieurs objets mixtes. La Figure 72 illustre un tel diagramme pour l'interaction avec un soldat de l'application WCCM. Celle-ci intègre un utilisateur et trois objets mixtes. Le premier est la table augmentée. Les doigts de l'utilisateur sont filmés lorsqu'il la touche, ce qui fournit une position au système. Le deuxième correspond à un ordre. Dans le monde physique, il s'agit d'un son émis par l'utilisateur lorsqu'il parle. Ce son est capté par un microphone et analysé pour identifier des ordres tels que celui de se déplacer. Le troisième objet consiste en un soldat, qui possède une position, une orientation et une vitesse, et dont une représentation est projetée sur la table. Le lien entre les objets servant d'outils et l'objet soldat est une relation d'animation qui dépend des ordres de déplacement donnés.

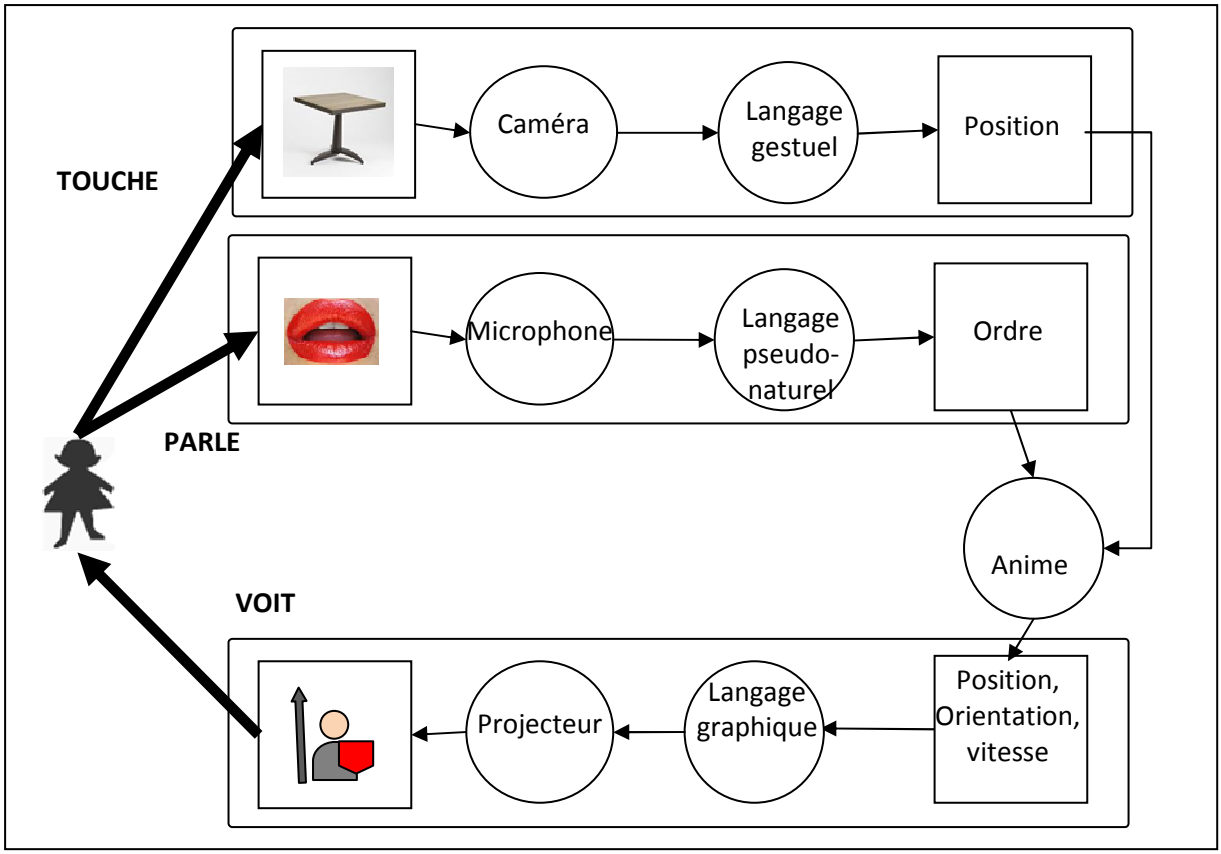


Figure 72 : Diagramme d'interaction mixte (MIM) pour l'interaction avec un soldat dans WCCM.

A l'instar de la notation ICARE qui repose également sur la définition d'une modalité comme un couple <dispositif, langage>, le modèle d'interaction mixte ne permet pas de décrire précisément le langage d'interaction. Aussi, le comportement interactif du système reste très difficile à expliciter. Ceci est également vrai pour la dimension temporelle de l'interaction multimodale, qui ne peut être exprimée avec le modèle d'interaction mixte.

Le modèle dispose d'un outil de prototypage (OP, pour *Object Prototyping*) sous la forme d'une boîte à outils. Celui-ci permet de réaliser des prototypes sur la base des diagrammes d'interaction décrits selon le modèle d'interaction mixte. Cet outil nécessite de programmer en Qt le diagramme selon le modèle d'interaction mixte, contrairement à ICON, ICARE ou OpenInterface qui incluent un éditeur graphique.

Conclusion sur le modèle d'interaction mixte

Le modèle d'interaction mixte est prévu pour décrire l'interaction pour des systèmes de réalité mixte, même s'il permet également de décrire l'interaction d'un système multimodal tel que WCCM. Le point d'intérêt vis-à-vis de nos objectifs est qu'il traite la multimodalité en sortie de la même manière que la multimodalité en entrée. Ce type de représentation est en revanche très peu précis pour la spécification, et comme ICARE, il décrit QUELLE modalité est utilisée, mais pas COMMENT.

2.5.4 Synthèse

Les trois notations ASUR, IRVO, et le modèle d'interaction mixte permettent de décrire à haut niveau d'abstraction les relations entre des utilisateurs et des objets réels et virtuels, dans le cadre de la réalisation d'une tâche. Ces notations permettent ainsi de décrire les dispositifs physiques disponibles, ainsi que les associations utilisateurs – dispositifs.

Ces notations sont en revanche de très haut niveau d'abstraction, et n'explicitent pas les relations temporelles entres les entités représentés. Ce type de représentation doit donc être complété par d'autres descriptions comme un modèle de tâches.

Les notations de modèle de tâches sont en effet un complément tout à fait adapté pour décrire l'interaction dans le cadre des entités décrites.

La dimension multiutilisateur n'est pas traitée explicitement par ces notations. Il est certes possible de décrire plusieurs utilisateurs, mais pas les rôles des utilisateurs qui sont à la base de la description de la coopération. De plus, l'absence d'expression de relations temporelles empêche de décrire les activités communes entre les utilisateurs.

La description de modalités d'interaction est prise en charge explicitement par le modèle d'interaction mixte sous la forme <dispositif, langage d'interaction>, et par les notations IRVO et ASUR sous la forme de dispositifs associés à des adaptateurs. Cependant, et comme pour les notations dédiées à la multimodalité, les langages d'interaction ne sont pas explicités, ce qui rend la description d'une modalité incomplète. Néanmoins, un point intéressant de ces notations est d'expliciter le lien entre les modalités en entrée et les modalités en sortie.

Pour conclure, les notations du domaine de la Réalité Augmentée proposent une représentation de l'interaction sous la forme de compositions des objets (réels et virtuels) et des utilisateurs qui y prennent part. L'absence de relations temporelles nuit cependant à la précision de la description

produite aussi bien pour ce qui concerne la dimension multimodale que pour la dimension multiutilisateur.

3 Synthèse globale

Dans ce chapitre, nous avons présenté un ensemble de caractéristiques des notations de spécification de l'interaction multimodale et/ou multiutilisateur, puis nous avons passé en revue les notations existantes sur la base de ces caractéristiques tout en illustrant leur utilisation sur une même application. Dans les sections suivantes, nous proposons une synthèse des notations selon les quatre catégories de caractéristiques présentées à la section 1.5 de ce chapitre. La cinquième section présente nos conclusions générales motivant les travaux présentés au chapitre suivant.

3.1 Synthèse des caractéristiques générales

Le Tableau 7 récapitule les caractéristiques générales des notations dans l'ordre où elles ont été présentées. Ce tableau possède 8 colonnes.

La première colonne indique le domaine des notations par un acronyme (GL = Génie Logiciel, IHM = Interaction Homme-Machine, TCAO = Travail Coopératif Assisté par Ordinateur, PC = Psychologie Cognitive, MM = Multimodalité, RA = Réalité Augmentée). Un symbole ">" entre deux domaines indique que celui de droite est sous-domaine de celui de gauche.

La deuxième colonne indique le nom ou un acronyme de la notation.

La troisième colonne décrit l'objectif de la notation et est décomposée en deux sous-colonnes. La première sous-colonne indique l'objectif en terme des étapes du cycle de vie logiciel couverte (A = Analyse des besoins, S = Spécification, CG = Conception Générale, CD = Conception Détaillée), de prise en charge de l'analyse de la tâche (T), ou de capacité à réaliser des prototypes (P). La deuxième sous-colonne indique la couverture de la notation des composants du modèle ARCH avec un pictogramme reprenant la forme d'une arche avec à gauche le noyau fonctionnel, au-dessus le contrôleur de dialogue et à droite l'interaction. Les éléments couverts par la notation sont en blanc.

Les quatrième et cinquième colonnes concernent respectivement le nombre de représentations différentes (#R) et les types de représentations (TR) que propose la notation. Les types de représentations sont définis parmi Graphique (G), Dessin (D), Textuelle (Te) et Tabulaire (Ta).

Les sixième et septième colonnes ne font pas partie des caractéristiques générales que nous avons définies en section 1.1 de ce chapitre, mais indiquent à gros grain si la notation prend en charge les dimensions multiutilisateur (MU) et multimodale (MM).

Enfin, la huitième colonne décrit l'existence d'outils logiciels associés à la notation. Cette colonne est divisée en quatre sous-colonnes qui indiquent : la présence d'un éditeur de spécification selon la notation (E), la capacité de l'outil à générer du code ou des prototypes exécutables à partir des spécifications (G), la capacité de l'outil à aider l'analyse des spécifications produites (A), et enfin la disponibilité publique de l'outil logiciel (*).

Domaine	Notation	Objectif	#R	TR	MU	MM	Outils

								E	G	A	*
GL	UML	A, S, CG, CD		13	G, Te	oui	non	X	X	X	X
GL	UML-G	A, S, CG, CD		13	G, Te	oui	non	X	X	X	X
GL	BPMN	A, S		1	G	oui	non	X	X	X	X
IHM	CTT	A, S, T		1	G	oui	non	X	X	X	X
IHM - PC	K-MAD	A, T		4	G, Ta	oui	non	X	X	X	X
IHM	TaskMODL	A, S, CG, CD, T		3+	G, D	oui	non	-	-	-	-
IHM	CTML	S, T		1	G	oui	non	X	-	X	X
IHM	WTM	S, T		1	G	oui	non	X	X	X	-
IHM	SpieLan	S, CD		2	G, D	oui	non	X	X	X	-
IHM	UAN	S		1	Ta	non	oui	-	-	-	-
IHM	IOG	S		1	G	non	non	-	-	-	-
IHM>TCAO	GTA	A, S, CG, CD, T		4	G, Ta, Te	oui	non	X	X	X	X
IHM>TCAO	MABTA	A, T		4	G, D	oui	non	-	-	-	-
IHM>TCAO	CIAN	A, S, CG, CD, T		5	G, Ta	oui	non	-	-	-	-
IHM>TCAO	TOUCHE	A, S, CG, CD, T		4	G, Ta, Te	oui	non	X	-	X	X
IHM>TCAO	CUA	A, T		3	G, Te	oui	non	-	-	-	-
IHM>TCAO	Orchestra	S, T		1	G	oui	non	-	-	-	-
IHM>MM	ICOM	A, P		1	G	non	oui	X	X	-	X
IHM>MM	ICO	A, P		1	G	non	oui	X	-	X	X
IHM>MM	ICARE	A, P		1	G	non	oui	X	X	-	X
IHM>MM	Dynamo-AID	S, T, P		4	G	oui	oui	X	-	-	-
IHM>MM	CIS	A, S		1	G	non	oui	-	-	-	-
IHM>RA	Azur	A, S		1	G	non	oui	X	-	X	X
IHM>RA	IRVO	A, S		1	G	non	oui	-	-	-	-
IHM>RA	MIM	A, P		1	G	non	oui	-	-	-	-

Tableau 7 : Synthèse des caractéristiques générales des notations existantes.

La plupart des notations présentées sont proposées pour répondre à un objectif précis. Ainsi dans le cas de la notation WTM, il s'agit de permettre l'analyse de la tâche pour des applications web, tandis qu'ICARE constitue un outil de prototypage rapide pour des applications multimodales. Dans le tableau de synthèse, ces notations peuvent être identifiées rapidement car ce sont celles qui possèdent une ou deux lettres dans la colonne Objectif. Les autres notations quant à elles concernent plusieurs étapes du cycle de vie logiciel, et s'intègrent dans la cadre d'une méthode de conception propre à la notation. C'est le cas des notations UML, TaskMODL ou encore de CIAN. Ces notations s'identifient par la présence de nombreuses lettres dans la colonne Objectif.

De plus lorsque les notations s'intègrent au sein d'une méthode de conception, elles adoptent souvent plusieurs points de vue sur le système à décrire. Chaque vue est spécifiée au sein d'une représentation spécifique. C'est pourquoi, ces notations proposent généralement plusieurs représentations. C'est le cas des notations UML, CIAN, TOUCHE, TaskMODL et GTA. Les notations

adoptant un point de vue particulier sur le système proposent généralement une seule représentation. La notation Orchestra propose ainsi une représentation unique.

La plupart des notations adoptent des représentations sous forme de graphiques pour décrire l'interaction. Il peut s'agir d'arbres de tâches (CTT, K-MAD), de diagrammes d'activité (UML, GTA) ou de diagrammes de flot de travail (CUA, CIAN), ou de flot de données (ICARE, ICOM). Certaines notations se différencient et adoptent une représentation tabulaire (UAN, NUAN), de dessin (MABTA, GTA) ou simplement textuelle (CUA).

Parmi les notations présentées, à l'exception d'UAN, aucune de celles provenant des domaines du Génie Logiciel et de l'Interaction Homme-Machine et de la TCAO ne prennent en charge la description de l'interaction concrète et de la multimodalité. De plus UAN est limité à l'interaction au clavier et à la souris. La prise en compte de l'interaction concrète et de la multimodalité est effective dans les notations provenant du domaine de la Multimodalité, mais également du domaine de la Réalité Augmentée.

La dimension multiutilisateur de l'interaction est prise en considération par les notations provenant des domaines du Génie Logiciel, de l'Interaction Homme-Machine (sauf IOG et UAN) et bien évidement que par les notations provenant du domaine de la TCAO. Parmi les notations du domaine de la multimodalité et de la réalité augmentée, seul la plateforme Dynamo-Aid prend en considération la dimension multiutilisateur coopérative en s'appuyant sur la notation d'arbre de tâches CTT.

Enfin, 15 des 25 notations présentées disposent d'un outil d'édition de spécification. Parmi ces outils, certains proposent des fonctionnalités complémentaires. Pour 10 des 15 notations munies d'éditeur, il s'agit de générer du code applicatif (StarUML) ou de la documentation (EUTERPE). Pour des notations telles que CTT, TOUCHE ou K-MAD, il s'agit également d'outils d'aide à l'analyse des spécifications produites. Enfin, la plupart des outils sont disponibles publiquement et constituent un vecteur important de diffusion de ces notations.

3.2 Synthèse pour la spécification de l'interaction

Le Tableau 8 constitue une synthèse des caractéristiques des notations pour la description de l'interaction. Il possède 7 colonnes.

La première colonne précise le nom ou l'acronyme de la notation.

La deuxième colonne (Structure) indique la capacité des notations à décrire de manière structurée l'interaction à haut niveau d'abstraction. Ceci peut être effectué par exemple par une décomposition de tâches en sous-tâches au sein d'un arbre de tâches. Pour indiquer la manière dont la notation structure la description de l'interaction, cette colonne est divisée en six sous-colonnes, chacune correspond à une structure particulière : arbre de tâches (AT), flot de travail (FT), de flot de contrôle (FC) tel qu'un diagramme d'activité UML, flot de données (FD) tel qu'un diagramme ICARE, diagramme d'état-transition (ET) ou équivalent, et enfin tableau (Ta). Les descriptions textuelles ne sont pas considérées dans cette colonne.

La troisième colonne (Composition) indique la capacité des notations à décrire la composition logique et temporelle des tâches. Cette colonne est divisée en deux sous-colonnes. La première

indique la possibilité de décrire des relations de base (B) que sont l'alternative, la séquence, le parallélisme et l'absence d'ordre. La deuxième sous-colonne indique l'existence d'opérateurs plus avancés (A), tels que l'opérateur suspendre-reprendre de CTT.

La quatrième colonne (Objet) précise si la notation prend en charge la description des objets de la tâche et l'association de ces objets aux tâches. Cette colonne est divisée en trois sous-colonnes. La première (S) indique si la notation prévoit la description des concepts du domaine au sein d'une représentation spécifique. La deuxième (L) indique si la notation permet de décrire le lien entre les concepts du domaine et les tâches. Enfin la troisième colonne (I) indique si ce lien est décrit conjointement au diagramme représentant l'interaction. Pour ces sous-colonnes, un (X) indique une prise en compte explicite par la notation et un (p) signifie que c'est possible bien que cela ne soit pas explicitement prévu par la notation. Enfin, un (-) indique l'absence de prise en compte.

La cinquième colonne (Contexte) indique la capacité à décrire le contexte de la réalisation d'une tâche ou d'un ensemble de tâches. Comme pour la colonne précédente, la sous-colonne (S) indique si la notation prévoit la description du contexte. La sous-colonne (L) indique le lien entre les contextes et les tâches, enfin la sous colonne (I) indique si ce lien est décrit au sein du diagramme représentant l'interaction.

Enfin, la dernière colonne (Système) indique si la notation permet l'expression d'actions ou de réactions du système.

La structure la plus courante pour décrire l'interaction parmi les notations présentées est l'arbre de tâches. En effet, 11 des 25 notations utilisent un arbre de tâches, tandis que seulement 6 utilisent un flot de travail et 7 un flot de données. L'intérêt principal des arbres de tâches par rapport au flot de travail et au flot de contrôle réside dans la décomposition hiérarchique qui maintient un lien fort entre interaction abstraite et tâche élémentaire. Plusieurs notations visant à décrire l'interaction concrète, telles qu'ICARE, ICOM utilisent des représentations de flot de données selon un point de vue plus proche de l'implémentation. Enfin quelques notations telles qu'UML ou TaskMODL permettent d'utiliser diagrammes d'état-transition afin de décrire finement l'interaction avec un objet interactif. L'utilisation de tableau est exceptionnelle puisque seules UAN (et NUAN) décrivent l'interaction de cette manière.

Toutes les notations qui utilisent une structure d'arbres de tâches, de flot de travail, de flot de contrôle et de diagramme d'état-transition permettent de décrire la composition de tâches au moyen de relations de base que sont l'alternative, la séquence, le parallélisme et l'absence d'ordre. Les notations de flot de donnée ne décrivent pas explicitement ces relations, cependant, une notation telle qu'ICARE permet de décrire des combinaisons d'interaction selon les propriétés CARE. Celles-ci permettent de décrire certaines formes de parallélisme et d'alternative. Parmi les notations présentées, les notations reposant sur la notation CTT proposent de décrire des relations plus complexes entre les tâches, telles que suspendre-reprendre ou encore la désactivation.

En ce qui concerne les concepts du domaine, plusieurs notations telles que TOUCHE ou CIAN, proposent une représentation dédiée aux concepts du domaine, qui repose le plus souvent sur un diagramme de classe UML. La notation TaskMODL adopte la même approche mais utilise la notation RML. Le lien entre les tâches et les objets peut être décrit avec la plupart des notations présentées. Cela peut s'effectuer au sein d'une représentation spécifique telles qu'un diagramme de

responsabilité CIAN ou un diagramme d'interaction Dynamo-AID. Le lien peut également être décrit directement au sein des diagrammes décrivant l'interaction comme c'est le cas pour la notation TaskMODL.

Notation	Structure						Composition		Objet			Contexte			Système
	AT	FT	FC	FD	ET	Ta	B	A	S	L	I	S	L	I	
UML	-	X	X	-	X	-	X	-	X	p	p	X	X	X	X
UML-G	-	X	X	-	X	-	X	-	X	p	p	X	X	X	X
BPMN	-	-	-	X	-	-	X	-	X	X	X	-	na	na	-
CTT	X	-	-	-	-	-	X	X	-	X	p	-	na	na	X
K-MAD	X	-	-	-	-	-	X	-	X	X	p	X	X	-	X
TaskMODL	X	-	-	X	X	-	X	-	X	X	X	-	na	na	X
CTML	X	-	-	-	-	-	X	X	-	X	p	-	na	na	X
WTM	X	-	-	-	-	-	X	X	-	p	p	-	na	na	X
SpieLan	-	-	X	-	-	-	X	-	X	X	X	-	na	na	X
UAN	-	-	-	-	-	X	X	X	-	X	X	-	na	na	X
IOG	-	-	-	-	X	-	X	-	X	X	X	-	na	na	X
GTA	X		X	-	-	-	X	-	X	X	p	X	X	-	X
MABTA	X	X	-	-	-	-	X	-	-	p	p	-	na	na	-
CIAN	X	X	-	-	-	-	X	X	X	X	X	-	na	na	X
TOUCHE	X	-	-	-	-	-	X	X	X	X	p	-	na	na	X
CUA	X	X	-	-	-	-	X	-	-	p	p	X	X	X	-
Orchestra	-	X	-	-	-	-	X	-	X	X	X	X	X	X	-
ICOM	-	-	-	X	-	-	na	na	-	-	-	-	na	na	-
ICO	-	-	-	-	X	-	na	na	-	X	X	-	na	na	-
ICARE	-	-	-	X	-	-	na	na	-	-	-	-	na	na	-
Dynamo-AID	X	-	-	-	-	-	X	X	X	X	X	X	X	X	X
CIS	-	-	X	-	-	-	X	-	-	p	p	-	na	na	X
Azur	-	-	-	X	-	-	na	na	X	X	X	X	X	X	X
IRVO	-	-	-	X	-	-	na	na	X	X	X	X	X	X	X
MIM	-	-	-	X	-	-	na	na	X	X	X	X	X	X	X

Tableau 8 : Synthèse des caractéristiques des notations existantes pour la description de l'interaction.

Peu de notations proposent de décrire explicitement le contexte dans lequel se déroule l'interaction, parmi elles, la plateforme Dynamo-Aid, qui propose une représentation spécifique des contextes, ainsi qu'une vue globale pour lier les contextes aux tâches. La notation Orchestra permet quant à elle de décrire des situations d'interaction dans lesquelles des tâches sont possibles. Enfin les notations pour la réalité augmentée ASUR et IRVO permettent de décrire plusieurs éléments du contexte pour la réalisation d'une tâche donnée.

Enfin, la plupart des notations permettent de décrire les actions du système. Parmi celles qui ne le permettent pas, nous notons ICOM et ICARE qui sont centrés sur l'interaction multimodale en entrée, ou encore MABTA qui se focalise sur la collaboration entre les utilisateurs.

3.3 Synthèse pour la spécification de l'interaction multiutilisateur

La synthèse des caractéristiques des notations concernant la dimension multiutilisateur est résumée au sein du Tableau 9. Celui ci comporte 6 colonnes.

La première colonne précise le nom ou l'acronyme de la notation.

Les deuxième et troisième colonnes précisent si les notations prennent en charge la coopération et/ou la collaboration.

La quatrième colonne indique si les notations permettent d'exprimer le partage d'objets entre les utilisateurs.

Les cinquième et sixième colonnes indiquent si la notation permet de décrire les relations spatiales et temporelles entre les utilisateurs.

Notation	Coopération	Collaboration	Objet partagé	Spatiale	Temporelle
UML	X	-	X	-	-
UML-G	X	-	X	X	-
BPMN	X	-	X	-	-
CTT	X	-	-	-	-
K-MAD	X	-	X	-	-
TaskMODL	X	-	X	-	-
CTML	X	-	-	-	-
WTM	X	-	-	-	-
SpieLan	X	-	X	-	-
UAN	-	-	-	-	-
IOG	-	-	-	-	-
GTA	X	-	X	-	-
MABTA	X	-	-	-	-
CIAN	X	X	X	-	-
TOUCHE	X	-	X	-	-
CUA	X	X	-	-	-
Orchestra	X	-	X	X	X
ICOM	-	-	-	-	-
ICO	-	-	-	-	-
ICARE	-	-	-	-	-
Dynamo-AID	X	-	X	-	-
CIS	-	-	-	-	-
Azur	-	-	-	X	-
IRVO	X	-	-	X	X
MIM	-	-	-	-	-

Tableau 9 : Synthèse des caractéristiques des notations existantes pour l'interaction multiutilisateur.

La description des activités coopératives est possible avec les notations du domaine du Génie Logiciel, de l'Interaction Homme-Machine (sauf UAN et IOG), et de la TCAO. Les notations du domaine de la Multimodalité ou de la Réalité Augmentée ne traitent pas ces aspects de l'interaction à l'exception de la plateforme Dynamo-Aid qui repose sur CTT. Pour ces notations, le travail est divisé

en tâches élémentaires (par des arbres de tâches, des diagrammes d'activité ou des flots de travail). Celles-ci sont réparties entre les utilisateurs sur la base de rôles.

En revanche, la description des activités collaboratives n'est possible qu'avec deux notations d'arbre de tâches : CIAN et CUA. Dans les deux cas, il s'agit d'utiliser le concept de tâche abstraite collaborative, similaire au concept de tâche abstraite coopérative. Néanmoins ces notations n'autorisent pas l'assignation de rôles aux tâches élémentaires de sa décomposition en sous-tâches, ce qui limite les possibilités d'expression de l'interaction multiutilisateur.

Les activités coopératives et collaboratives doivent être traitées conjointement car elles sont les deux facettes de l'interaction multiutilisateur. La division explicite de l'interaction entre coopération et collaboration n'est pas une bonne solution car elle peut conduire à décomposer des spécifications de l'interaction de façon artificielle.

Un grand nombre de notations (11 sur 25) permettent de décrire les objets qui sont partagés entre les utilisateurs. Cela implique soit une description spécifique de ces concepts pour UML-G ou TaskMODL, soit une intégration de ces concepts au sein des descriptions de l'interaction sous forme d'arbre de tâches (CIAN) ou de flot de travail (Orchestra).

Nous avons souligné à la section précédente que peu de notations prennent en charge la description du contexte. Un aspect du contexte spécifique à l'interaction multiutilisateur consiste justement à identifier la répartition des lieux l'interaction et des utilisateurs dans l'espace et dans le temps. Quatre notations, UML-G, Orchestra, ASUR, et IRVO prennent en charge explicitement la spécification de la répartition spatiale des utilisateurs, alors que deux seulement, Orchestra et IRVO permettent de décrire la répartition temporelle des utilisateurs.

3.4 Synthèse pour la spécification de l'interaction multimodale

Le Tableau 10 constitue une synthèse des caractéristiques des notations concernant la description de l'interaction concrète et multimodale. Dans ce tableau, nous n'avons pas reporté les notations qui ne traitent explicitement aucun des caractéristiques liées à l'interaction concrète et multimodale. Le tableau comporte six colonnes dont la première correspond au nom ou à l'acronyme de la notation.

La deuxième colonne indique si la notation prend en charge la description de l'interaction concrète.

La troisième colonne indique la capacité d'une notation à exprimer l'usage d'une modalité d'interaction pour la réalisation d'une tâche.

La quatrième colonne précise si la notation permet d'exprimer les dispositifs d'interaction utilisés pour l'interaction, tandis que la cinquième colonne indique si la notation permet de décrire le langage d'interaction correspondant à ces dispositifs.

Enfin, la sixième colonne indique si les notations permettent d'exprimer la combinaison de modalités d'interaction pour la réalisation d'une tâche donnée.

Notation	Interaction concrète	Modalité	Dispositif	Langage	Combinaison
K-MAD	-	-	X	-	-
TaskMODL	X	-	X	-	-
SpieLan	X	-	-	-	-
UAN	X	-	X	X	X
IOG	X	-	-	-	-
GTA	X	-	X	X	X
MABTA	X	-	-	-	-
Orchestra	-	-	X	-	-
ICOM	X	X	X	-	X
ICO	X	X	X	-	X
ICARE	X	X	X	-	X
Dynamo-AID	X	X	-	-	X
CIS	X	X	X	X	X
Azur	X	-	X	-	-
IRVO	X	-	X	-	-
MIM	X	X	X	-	X

Tableau 10 : Synthèse des caractéristiques des notations existantes pour l'interaction concrète et multimodale.

L'interaction concrète et multimodale est un aspect qui est partiellement pris en charge par de nombreuses notations (16 sur les 25 présentés). Cependant, les possibilités d'expression qu'offrent ces notations varient beaucoup. Pour certaines, il s'agit de décrire les dispositifs d'interaction liés aux tâches (K-MAD, MABTA, TaskMODL, Orchestra). Pour d'autres, le concept de modalité est intégré aux descriptions, sous une forme où l'on peut retrouver explicitement ou implicitement le couple <dispositif, langage d'interaction> (ICARE, ICOM, ASUR, IRVO), en tant qu'entité élémentaire (Dynamo-Aid), ou servir de base à la description d'interaction (CIS, UAN, NUAN utilisé dans GTA).

Néanmoins, la description des dispositifs et des langages d'interaction est toujours effectuée grâce à de simples étiquettes. Ainsi aucune notation n'explicite les actions captées par les dispositifs ni les interprétations en commandes qui sont faites de ces actions. Il résulte que les notations existantes permettent de désigner quel dispositif d'interaction ou quelle modalité est utilisée pour réaliser une tâche donnée, mais pas comment le dispositif ou la modalité est utilisé.

De plus, certaines notations permettent de décrire des combinaisons de modalités pour une tâche donnée. Pour la notation ASUR, IRVO et ICOM, l'ordre des modalités n'est pas défini au sein des combinaisons. Pour les notations ICARE et la plateforme Dynamo-AId en revanche, les combinaisons de modalités sont décrites au moyen des propriétés CARE de complémentarité, de redondance ou d'équivalence. Dans tous les cas, les relations entre modalités ne peuvent être décrites de manière précise et nous n'avons pu spécifier finement certaines interactions multimodales de l'application WCCM.

Il est donc nécessaire d'étendre la prise en charge de l'interaction multimodale telle qu'elle est proposée par les notations existantes afin de décrire non seulement quelle modalité ou combinaison de modalités est utilisée pour réaliser une tâche, mais également comment une modalité ou une combinaison de modalités doit être utilisée.

3.5 Conclusion

Parmi les notations existantes pour la spécification de l'interaction multiutilisateur et/ou multimodale qui ont été présentées dans ce chapitre, aucune ne permet de décrire à la fois l'interaction multiutilisateur coopérative, l'interaction multiutilisateur collaborative, et l'interaction multimodale. Cependant, chaque notation couvre au moins un de ces aspects.

Sur l'ensemble des notations présentées, plusieurs permettent de décrire l'interaction multiutilisateur :

- L'interaction coopérative peut être décrite avec plusieurs notations, notamment CTT, TaskMODL, GTA, TOUCHE et CIAN. Elles utilisent un modèle hiérarchique de tâches, où les tâches élémentaires sont réparties entres les utilisateurs sur la base de rôles.
- L'interaction collaborative peut être décrite par les notations CIAN et CUA grâce à l'introduction du concept de tâche collaborative.
- Les relations logiques et temporelles sont décrites entre les tâches par des opérateurs, dont les plus précis sont ceux de la notation CTT.
- Les objets partagés entres les utilisateurs et le contexte d'utilisation peuvent être décrit avec UML-G et liés aux tâches explicitement comme le propose la notation TaskMODL.

Ces différentes notations doivent donc être combinées pour traiter complètement la description de l'interaction coopérative, et être améliorées/étendues pour permettre la spécification de certaines interactions collaboratives. En effet, l'interaction coopérative et l'interaction collaborative doivent s'articuler de manière cohérente au sein d'une même représentation afin de ne pas briser la continuité de description entre l'interaction abstraite et les tâches élémentaires.

Les notations reposant sur l'usage d'un modèle de tâches sont de trop haut niveau d'abstraction pour décrire l'interaction concrète. En effet, pour la plupart d'entre elles, les tâches feuilles des arbres de tâches sont des tâches élémentaires telles que *Sélectionner un soldat*, ou *Confirmer un choix*. Ces tâches élémentaires peuvent être détaillées au sein d'autres représentations tels qu'un diagramme NUAN, une configuration d'entrées ICOM ou un diagramme ICARE. La notation d'arbre de tâches utilisée dans la plateforme Dynamo-AID propose ce niveau de détail mais directement intégré au sein de l'arbre de tâches. Ce type de représentation intégré est particulièrement intéressant puisqu'il accompagne la transition entre la description de l'interaction de l'interaction abstraite et celle de l'interaction concrète.

Enfin, si ces notations permettent de décrire quelles modalités sont utilisées, elles ne décrivent pas comment elles doivent être utilisées en termes de primitives de leurs langages d'interaction pour une tâche donnée. D'autre part, la spécification de combinaisons de modalités d'interaction réalisées sur la base des propriétés CARE avec les notations Dynamo-Aid ou ICARE n'est pas assez précise. Elles doivent être étendues pour permettre de décrire plus finement les combinaisons temporelles.

Motivé par ces constats issus de l'étude des notations existantes, notre objectif est de proposer une notation de spécification de l'interaction multiutilisateur et multimodale qui articule de façon cohérente les concepts des notations existantes et les étend pour permettre la description

1. de la collaboration,
2. de l'usage de modalités d'interaction et de leurs combinaisons.

La notation de spécification que nous proposons est intitulée COMM et fait l’objet du chapitre suivant.

Chapitre 3 : Notation COMM

Dans ce chapitre, nous présentons notre contribution conceptuelle : la notation d'arbre de tâches COMM (COllaborative et MultiModale). Celle-ci répond à notre objectif de spécification de l'interaction multiutilisateur et multimodale. Elle s'appuie sur l'intégration de concepts présents dans les notations existantes et propose de décrire de manière continue l'interaction abstraite et concrète, par raffinement successif. La notation COMM prend en charge la description de l'interaction multiutilisateur coopérative et collaborative, ainsi que l'interaction concrète multimodale.

Ce chapitre est décomposé en trois sections. Afin de bien cerner notre contribution, la première section présente en détail nos objectifs d'élaboration d'une notation. La section suivante est consacrée à l'élaboration de la notation COMM et l'introduction des différents concepts clefs de la notation, tels que les tâches modales et les rôles interactifs. La troisième section constitue une synthèse de la notation COMM dans sa version définitive.

Dans ce chapitre, nous reprenons pour illustrer nos contributions, l'application WCCM (Warcraft Collaboratif Coopératif et Multimodal) introduite au Chapitre 1 et qui a servit de fil conducteur au Chapitre 2 pour illustrer l'usage des notations existantes.

1 Objectif de la notation

Le rôle d'une notation est de permettre la description de solutions d'interaction conçues par le concepteur, en amont de leur réalisation, comme l'illustre la Figure 73. Notre analyse des notations existantes présentée au Chapitre 2 souligne qu'il n'existe aucune notation prenant en charge conjointement les dimensions multiutilisateur et multimodale. Nous avons également noté que plusieurs notations couvrent au moins partiellement l'un de ces deux aspects. Aussi, notre principal objectif est de définir une notation pour spécifier l'interaction multiutilisateur et multimodale.

Pour cela, nous souhaitons réutiliser autant que possible les notations existantes, tant dans leurs concepts que dans leurs présentations plutôt que de définir une notation entièrement nouvelle. L'analyse des notations existantes présentée au Chapitre 2 constitue un socle à ce travail de réutilisation. La notation COMM doit donc être considérée comme l'extension de notations existantes, plus qu'une nouvelle notation. En particulier, il s'agit d'articuler les notations existantes de manière cohérente afin de couvrir les dimensions multiutilisateur et multimodale. Cette articulation de notation ne doit pas impliquer un découpage de nos préoccupations en plusieurs représentations, comme cela peut être le cas avec certaines notations existantes comme GTA où l'interaction abstraite est décrite avec un arbre de tâches et l'interaction concrète est décrite avec la notation NUAN. Aussi, nous souhaitons fournir une notation qui permette de décrire conjointement ces différents aspects plutôt qu'un ensemble de représentations disjointes. Ainsi, la notation visée doit prendre charge la description de l'interaction abstraite, qu'elle soit individuelle, coopérative ou collaborative ainsi que la description de l'interaction concrète multimodale, et cela de manière intégrée.

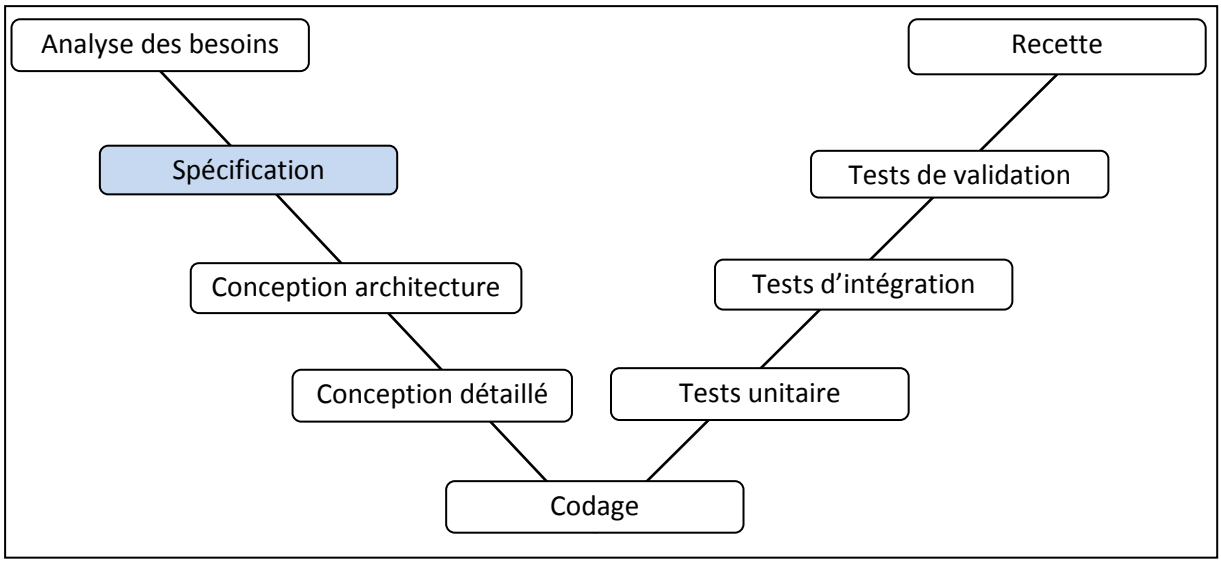


Figure 73 : Positionnement de la notation COMM dans le cycle de vie en V.

En ce qui concerne l'aspect multiutilisateur, il s'agit de proposer un moyen de décrire l'interaction entre plusieurs utilisateurs, qu'elle soit coopérative ou collaborative, au sens des définitions adoptées à la section 1.3 du Chapitre 1. Plusieurs notations proposent de décrire la dimension coopérative de l'interaction multiutilisateur, mais peu proposent de décrire la dimension collaborative. Parmi elles, nous avons identifié au Chapitre 2 les deux notations CIAN [Molina 2006] et CUA [Pinelle 2006]. Pour ces notations, l'intégration des deux aspects est réalisée en proposant des tâches de types explicitement différents : des tâches coopératives et des tâches collaboratives. Cette solution n'est pas satisfaisante vis-à-vis de notre objectif puisqu'elle conduit à décrire plusieurs représentations distinctes de l'interaction. Aussi, notre solution doit permettre de décrire à la fois l'interaction coopérative et l'interaction collaborative sans briser la continuité de la modélisation.

Pour l'aspect multimodal, il s'agit de prendre en charge la description de l'interaction concrète en termes d'usage de modalités d'interaction, au sens de la définition adoptée à la section 2.1 du Chapitre 1. La notation doit donc permettre de décrire quel dispositif physique est utilisé et comment il est utilisé. La multimodalité doit quant à elle être décrite en termes de combinaison de modalités d'interaction pour la réalisation d'une tâche donnée. Nous avons présenté au Chapitre 2 plusieurs notations telles qu'ICARE [Bouchet 2006] ou Dynamo-Aid [Clerckx 2005] qui proposent de décrire l'usage de modalités d'interaction ou de combinaisons de modalités en s'appuyant sur les propriétés CARE [Coutaz 1995]. Cette solution constitue une piste intéressante, que nous proposons de reprendre et d'enrichir notamment pour préciser plus finement les relations temporelles entre les modalités, et le langage d'interaction utilisé pour chaque modalité.

Le besoin d'une articulation cohérente et explicite au sein d'une unique représentation provient de l'analyse des différents niveaux de couplage présentés à la section 1.6 du Chapitre 1. De ce travail, nous retenons que l'interaction multiutilisateur peut être modélisée aussi bien à très haut niveau d'abstraction, qu'au niveau de l'interaction concrète, et ce, que ce soit dans le cas de l'interaction coopérative ou collaborative. En effet, dans le cas d'une interaction coopérative, la division du travail entre les utilisateurs peut aussi bien être opérée au niveau des tâches, ou en termes de répartition des modalités d'interaction pour réaliser une tâche donnée. Ceci est naturellement vrai pour la dimension collaborative. Les dimensions multiutilisateur et multimodale sont alors étroitement

mêlées. Aussi, ces deux aspects doivent être traités conjointement. Pour cela, notre notation doit permettre de décrire au sein d'une représentation unique l'aspect multiutilisateur et l'aspect multimodal au niveau de l'interaction concrète et abstraite. Le concepteur pourra ainsi décrire les niveaux de couplage qu'il souhaite mettre en œuvre dans son application. Pour résumer, notre objectif est de proposer une notation permettant au sein d'une représentation uniforme de décrire l'interaction multiutilisateur et multimodale. Cette notation doit s'inspirer des notations existantes et prendre en charge notamment la description :

- de l'interaction coopérative et collaborative ;
- des modalités d'interaction en termes de dispositif physique et de langage d'interaction ;
- du couplage de l'interaction multiutilisateur et multimodale;

La section suivante est consacrée à l'élaboration de notre solution qui répond à cet objectif : la notation COMM.

2 Définition de la notation COMM

Le processus d'élaboration de la notation COMM passe par plusieurs étapes que nous décrivons dans cette section. Tout d'abord, nous définissons une forme de base pour la notation, et pour cela, nous adoptons une notation existante : la notation CTT [Paterno 1997]. Ensuite, nous étudions les besoins particulier à la spécification de la dimension multiutilisateur, puis multimodale. Nous analysons les réponses apportées à ces besoins par les notations existantes, puis nous proposons une solution d'amendement à notre forme de base qui répond à ces besoins. Ces propositions peuvent consister en l'intégration de concepts d'autres notations, en l'évolution de concepts existants, ou en l'introduction de nouveaux concepts. Des récapitulatifs réguliers sont proposés afin de montrer l'évolution progressive de la notation lors de notre démarche de conception de la notation.

De par la manière de décrire l'évolution de la notation face aux besoins, cette section contient de nombreuses variations de notation. Ces variations constituent des supports à la réflexion et ne doivent pas servir de référence. La section suivante (section 3) présente une synthèse de la notation COMM dans sa forme définitive. C'est cette synthèse qui constitue la référence pour toutes utilisations ultérieures de la notation COMM.

2.1 Spécification de l'interaction avec un arbre de tâches

Comme nous l'avons mis en avant dans le Chapitre 2, parmi les notations existantes, nombreuses sont celles qui reposent sur l'utilisation d'un modèle de tâche hiérarchique pour décrire l'interaction. La méthode consiste à décrire une tâche racine, puis de décomposer celle-ci en sous-tâches. Ces dernières sont également décomposées en sous-tâches de plus en plus concrètes jusqu'au niveau des tâches dites élémentaires ou atomiques. Celles-ci correspondent à une interaction élémentaire sur les concepts de l'application, par exemple *Supprimer un document*, et ne décrivent pas les actions de l'utilisateur pour les réaliser, tels que *Cliquer sur un document* puis *Cliquer sur le bouton supprimer*.

Une représentation de modèle de tâche hiérarchique présente plusieurs avantages [Caffiau 2009] :

- Elle induit une division en modules sur la base des tâches abstraites. En effet, la spécification s'effectue selon une approche descendante (top-down), de la tâche la plus abstraite à la tâche la plus concrète. La division s'opère donc à haut niveau d'abstraction, et sépare l'interaction concrète en module.
- Elle est adaptée à l'activité de conception, puisqu'elle repose sur un processus de raffinements progressifs des spécifications. De plus, cela maintient un lien très fort entre les tâches de haut niveau d'abstraction correspondant aux fonctionnalités offertes par l'application, et les tâches de plus bas niveau d'abstraction qui correspondent à la mise en œuvre de ces fonctionnalités.
- De plus, elle centralise le lieu des modifications dans les spécifications. Ainsi, une modification de l'application entraine le plus souvent l'ajout de quelques tâches dans la hiérarchie, et l'ajout de relations temporelles reliant ces nouvelles tâches à celles déjà présentes dans l'arbre.
- Les auteurs de [Caffiau 2009] identifient également le fait qu'ils ne dépendent pas de technique d'interaction particulière. Ceci car les tâches élémentaires correspondent à des opérations sur les concepts de l'application et ne décrivent pas comment la tâche est réalisée concrètement en terme d'usage de modalité.

Par contraste avec les notations à base modèles de tâches hiérarchiques, plusieurs notations utilisent des diagrammes d'activités : UML [UML], SpieLAN [Bergh 2006] ou des diagrammes de flot de travail : BPMN [BPMN], Orchestra [David 2006]. Au sein de ceux-ci sont représentés des tâches similaires aux tâches élémentaires d'un modèle de tâches hiérarchiques. Les tâches abstraites ne sont pas représentées. Ce type de diagramme brise donc le lien entre interaction abstraite et concrète. C'est pourquoi, le modèle de tâche hiérarchique nous semble plus adapté pour décrire l'interaction, vis-à-vis de nos objectifs.

Parmi les modèles de tâches hiérarchiques, certains adoptent une forme d'arbre : CTT [Paterno 1997], K-MAD [Lucquiaud 2005, Baron 2006], GTA [Veer 2000], tandis que d'autres adoptent une forme à plat : MABTA [Lim 2004], CIAN [Molina 2006]. Dans une forme arborescente, les sous-tâches d'une tâche sont représentées comme ses nœuds fils. Au sein d'une représentation à plat, les sous-tâches d'une tâche sont représentées à l'intérieur de celle-ci. Représenter une décomposition en sous-tâches à l'intérieur des tâches décomposées est intéressante puisqu'elle permet de délimiter des frontières entre des ensembles de tâches. Cependant, lorsque plusieurs niveaux de décomposition interviennent, les représentations à base de rectangles imbriqués deviennent illisibles. En effet, le cadre de tâches de haut niveau devient alors immense, et il est difficile à rattacher visuellement au nom de cette tâche. A l'inverse, une forme arborescente décrit par ses relations père-fils le lien entre des tâches de haut niveau et les tâches feuilles. Il suffit en effet de suivre les relations père-fils en montant pour monter en niveaux d'abstraction, et en descendant pour descendre en niveaux d'abstraction. Aussi, la forme arborescente nous semble donc plus adaptée pour décrire la décomposition de l'interaction.

Parmi les notations existantes, la notation CTT [Paterno 1997] est très utilisée. Elle propose une forme d'arbre n-aires et de nombreux opérateurs pour décrire les relations entre les tâches. Le consensus autour de cette notation est tel que plusieurs autres notations et méthodologies reposent sur l'utilisation de CTT. C'est le cas de CTML [Wurdel 2008a], CIAN [Molina 2006], TOUCHE [Penichet 2006] et du Framework Dynamo-Aid [Clerckx 2005]. Le succès de cette notation peut s'expliquer par :

- L'utilisation de cinq types de tâches clairement identifiés par des icônes (utilisateur, interactive, système, abstraite et coopérative). Un arbre CTT fait donc apparaître clairement le type de ces tâches ce qui améliore sa lisibilité par rapport à un arbre de tâches tel que celui de GTA [Veer 2000], de CUA [Pinelle 2003] ou de MABTA [Lim 2004].
- La présence de nombreux opérateurs logiques et temporels pour décrire les relations entres les tâches. Ils permettent d'exprimer des relations communes à toutes les notations telles que la séquence, l'alternative ou le parallélisme, mais également d'exprimer des relations plus particulières tels que suspendre-reprendre, la désactivation ou encore des échanges d'information entre tâches. Ces relations permettent de démarquer CTT de notations telles que K-MAD [Lucquiaud 2005, Baron 2006], ou TaskMODL [Treatteberg 2002].
- Le placement des opérateurs qui permet d'exploiter au mieux la structure n-aires de l'arbre. En effet, un opérateur CTT se présente comme une relation entre deux sous-tâches et non au niveau d'un nœud, comme cela est proposé dans K-MAD ou GTA. Il est ainsi possible d'utiliser plusieurs opérateurs différents pour une décomposition, tandis qu'avec K-MAD, il faut plusieurs niveaux de décomposition pour décrire la même interaction. Des règles de priorité entre les opérateurs permettent de désambiguïser une décomposition présentant de multiples opérateurs. Les arbres produits peuvent donc être plus concis.
- L'existence de l'outil CTTE [Mori 2002], permettant d'éditer des spécifications selon la notation CTT est également une contribution importante.

Sur la base de ces constatations, nous choisissons d'adopter pour la notation COMM, la forme d'un arbre de tâches n-aires, et s'appuyant sur un ensemble d'opérateur de la notation CTT. La notation CTT ainsi que ses extensions comme CIAN et le framework Dynamo-Aid présente néanmoins des lacunes vis-à-vis de nos objectifs. Les lacunes identifiées concernent d'une part l'interaction multiutilisateur coopérative et collaborative, et d'autre part l'interaction concrète et notamment l'interaction multimodale.

Les éléments de la notation COMM ainsi que sa structure (pour l'instant équivalente à CTT) sont récapitulés au sein de la Figure 74. Elle comporte ainsi 5 types de tâches différents : utilisateur, système, interactive, abstraite et coopérative. Elle utilise les opérateurs binaires CTT : activation, alternative, désactivation, suspendre-reprendre, concurrence, ordre indéfini, concurrence avec échange d'information, activation avec passage d'information, ainsi que les opérateurs unaires suivants : optionnel, itération et itération finie. Les types de tâches et les opérateurs CTT sont décrits en détail dans le Chapitre 2.

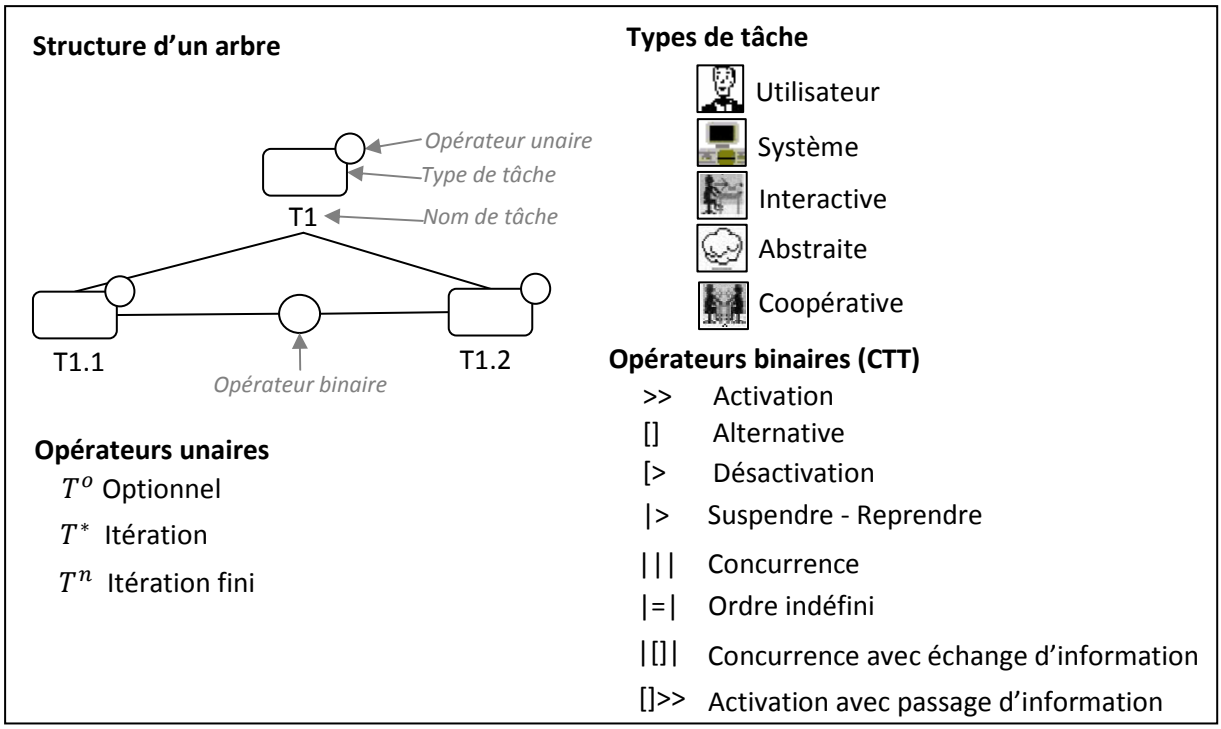


Figure 74 : Eléments et structure de la notation initiale.

2.2 Spécification de l'interaction multiutilisateur

Comme nous l'avons défini au Chapitre 1, les activités multiutilisateur peuvent être de deux types : coopérative ou collaborative. Notre objectif est de couvrir la spécification de ces deux types d'activités.

Pour ce qui concerne l'interaction coopérative, plusieurs notations présentées au Chapitre 2 proposent une solution : elle consiste à décrire des tâches coopératives sous la forme d'une composition de tâches individuelles. Ces tâches individuelles sont alors associées à des rôles. Cette manière de décrire l'interaction coopérative satisfait la définition que nous adoptons pour la coopération. En effet, les rôles constituent le moyen d'exprimer la division du travail.

La description de la collaboration n'est abordée que par deux notations : CUA [Pinelle 2006] et CIAN [Molina 2006]. Sur la base des définitions de la coopération et de la collaboration de [Dillenbourg 1996], ces notations introduisent deux types de tâches : les tâches coopératives et les tâches collaboratives. Celles-ci sont des tâches abstraites. Une tâche collaborative peut être composée de tâches collaboratives et individuelles, tandis qu'une tâche coopérative peut être composée de tâches coopératives, collaboratives et individuelles. De plus, la décomposition d'une tâche coopérative fait intervenir des rôles pour décrire la division du travail. Ces rôles sont associés aux sous-tâches individuelles. A l'opposé, une tâche collaborative peut être associé à un ou plusieurs rôles mais ses sous-tâches non, et ce par respect de la définition de la collaboration.

Ces approches de modélisation de l'interaction coopérative et collaborative présentent plusieurs limites que nous exposons dans la section suivante. Face à ces limites, et dans le but de permettre la description de l'interaction multiutilisateur, nous proposons ensuite une solution de modélisation, reposant sur les concepts de tâche multiutilisateur, de rôle métier, d'utilisateur et de rôle interactif. Cette solution constitue une extension de la notation adoptée comme base de travail.

2.2.1 Limites des notations existantes

Nous présentons ici les limites des notations existantes en termes de description des activités coopératives et collaboratives en s'appuyant sur les deux exemples reposant sur l'application WCCM, décrits au Chapitre 1. Ces limites concernent :

- La distinction des utilisateurs jouant un rôle identique, impliqués dans la réalisation d'une tâche coopérative ou collaborative ;
- L'expression d'un nombre minimum d'utilisateurs nécessaires à la réalisation d'une tâche coopérative ou collaborative ;
- L'effort de modélisation induit par la distinction explicite entre une tâche de nature coopérative et une tâche de nature collaborative.

Pour illustrer le premier point, considérons le cas où le concepteur souhaite que plusieurs utilisateurs jouent le rôle de *Sergent*, tandis qu'un seul peut jouer le rôle de *Chef*. Le concepteur peut alors décrire une tâche *Gérer les troupes* comme la composition des trois sous-tâches : les deux tâches *Déplacer un soldat* et *Former un groupe sont* associées au rôle de *Sergent* et la tâche *Déplacer un groupe est* associé au rôle de *Chef*. Cette décomposition en trois sous-tâches est illustrée à la Figure 75.

Puisque le travail est divisé entre les utilisateurs sur la base de rôle, la tâche *Gérer les troupes* est coopérative. Cependant, n'importe lequel des utilisateurs jouant le rôle *Sergent* peut effectuer les tâches *Déplacer un soldat* et *Former un groupe*, ainsi, la tâche *Gérer les troupes* peut être considérée comme collaborative. Il devient alors difficile au concepteur de choisir le bon type de tâche pour la tâche *Gérer les troupes*. De plus, rien n'indique dans l'arbre de tâches que plusieurs utilisateurs peuvent jouer le rôle de *Sergent*.

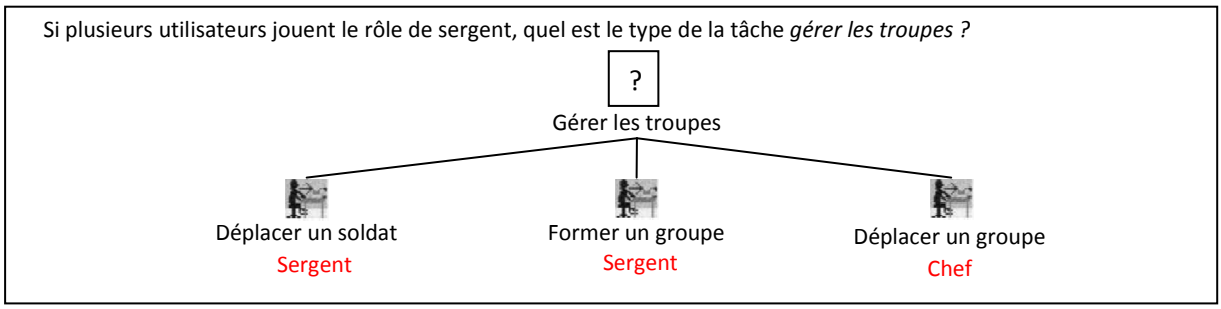


Figure 75 : Limite de la description de la coopération avec un rôle joué par plusieurs utilisateurs.

Pour résoudre ce problème, le concepteur peut introduire une tâche abstraite supplémentaire comme l'illustre l'arbre de tâches de la Figure 76. Celui-ci comporte la tâche coopérative *Gérer les troupes*, décomposée en deux sous-tâches : la tâche collaborative *Gérer les soldats*, et la tâche individuelle *Déplacer un groupe*. La tâche *Gérer les soldats* est alors décomposée en deux sous-tâches individuelles : *Déplacer un soldat* et *Former un groupe*. Si cette solution permet de bien distinguer les aspects coopératifs et collaboratifs de l'activité, elle présente deux défauts : d'une part, la taille de la spécification augmente en ajoutant une tâche abstraite, et d'autre part, rien n'indique dans la spécification que deux utilisateurs peuvent jouer le rôle de *Sergent*.

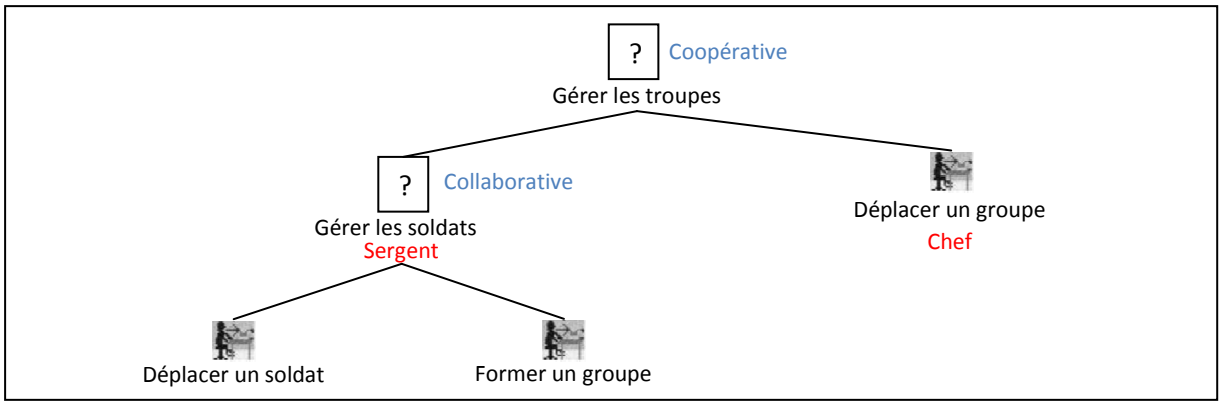


Figure 76 : Introduction des tâches abstraites pour décrire une coopération avec un rôle joué par plusieurs utilisateurs.

Maintenant que nous avons souligné le manque des notations existantes pour décrire le nombre d'utilisateurs jouant un rôle donné, nous considérons un deuxième exemple qui nous permet d'illustrer la difficulté que peut introduire la description de la collaboration. Pour cet exemple, un seul utilisateur joue le rôle de *Sergent*.

Considérons la tâche qui consiste à *Gérer un baraquement* pour produire un soldat. Au sein de la Figure 77, cette tâche est décomposée en trois sous-tâches individuelles : les tâches *Sélectionner un baraquement*, et *Créer un soldat* qui doivent être réalisées par un même utilisateur, et la tâche *confirmer* qui doit être réalisée par un autre utilisateur. La tâche *Gérer un baraquement* peut être considérée comme collaborative, puisque sa décomposition en sous-tâches ne fait intervenir aucun rôle.

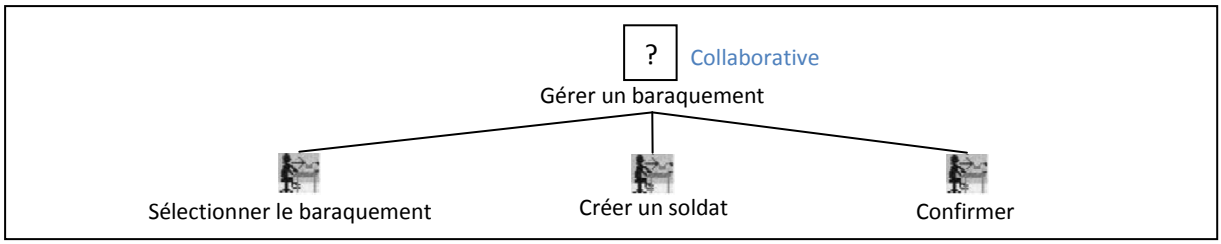


Figure 77 : Tâches collaboratives de gestion d'un baraquement.

Néanmoins, l'arbre de tâches de la Figure 77 ne modélise pas le fait qu'un utilisateur doive réaliser les deux premières sous-tâches et que l'autre utilisateur doit réaliser la troisième. Pour cela, nous proposons l'arbre de tâches de la Figure 78. Il s'agit de dupliquer les sous-tâches pour former deux groupes séparés par un opérateur d'alternative. Enfin, il s'agit d'assigner les rôles de sergent et de chef aux groupes de tel manière que dans tous les cas, les tâches *Sélectionner le baraquement* et *Créer un soldat* soient réalisées par un même utilisateur, et que la tâche *Confirmer* soient assignée à un autre utilisateur. Si l'arbre de tâches de la Figure 78 décrit une solution, elle présente deux inconvénients : d'une part, il apparait comme assez surprenant de modéliser une collaboration entre utilisateurs sous la forme d'une coopération entre des rôles, puisque les définitions de coopération et de collaboration diffèrent. D'autre part, la taille de la modélisation a fortement augmenté. Elle a presque doublé pour deux rôles. Elle serait multipliée par six pour trois rôles différents, et par douze pour quatre rôles. Ce type de solution n'est donc pas une bonne solution en vue de la concision de la modélisation obtenue.

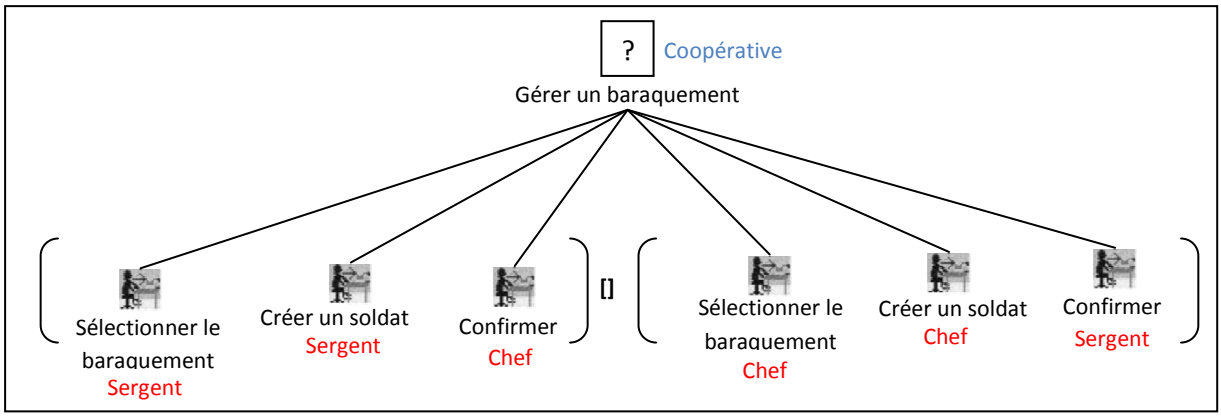


Figure 78 : Utilisation de tâche coopérative pour palier aux limites de la modélisation de la collaboration.

Pour cette tâche *Gérer un baraquement*, considérons maintenant l'hypothèse que le concepteur décide d'introduire une répartition hiérarchique entre le rôle de *Chef* et le rôle *Sergent* pour la tâche la production d'unités en imposant que ce soit toujours un utilisateur jouant le rôle de *Chef* qui réalise la tache de confirmation. Une solution immédiate consiste à assigner le rôle de *Chef* à la tâche *Confirmer*. Ceci n'est pas prévu dans les notations CUA [Pinelle 2006] et CIAN [Molina 2006], car cela va à l'encontre de la définition de la collaboration. Une solution conforme à ces notations est décrite à la Figure 79. Il s'agit de décrire la tâche *Gérer un baraquement* comme coopérative, et d'introduire une sous-tâche collaborative que nous nommons *Proposer la création d'un soldat*. Celle-ci est décomposée en deux sous-tâches : *Sélectionner un baraquement* et *Créer un soldat*.

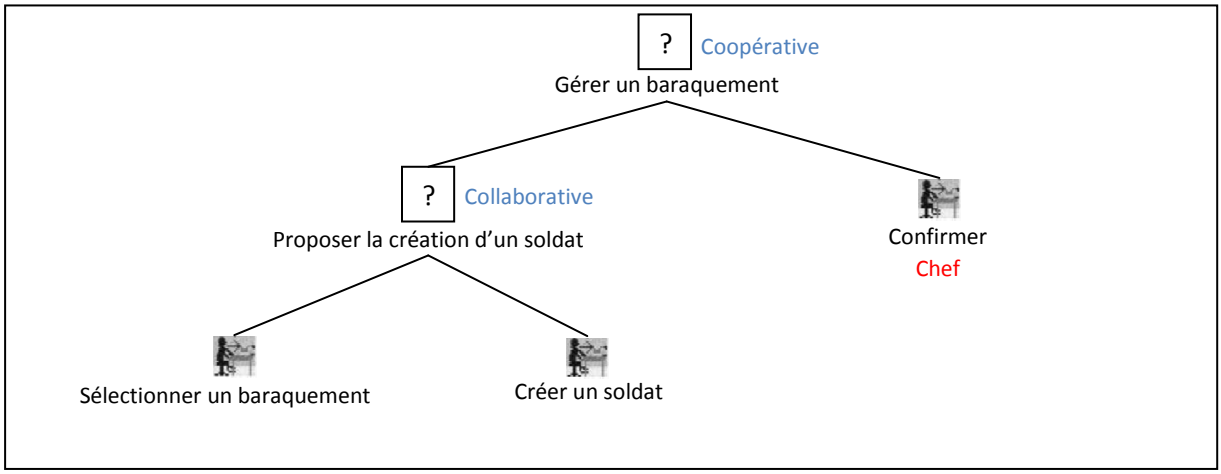


Figure 79 : Introduction de tâche assignée à un rôle au sein d'une collaboration.

La solution décrite par la Figure 79 ne permet pas d'indiquer qu'un utilisateur unique doit réaliser les tâches de *Sélectionner un baraquement* et de *Créer un soldat*. Pour décrire cela, il est nécessaire de revenir à la solution inadaptée que nous avons utilisée plus haut, c'est-à-dire décrire la collaboration sous la forme de tâche coopérative. La Figure 80 décrit exactement ce que souhaite spécifier le concepteur. La tâche *Proposer la création d'un soldat* disparait, et la forme générale de l'arbre est celle de la Figure 78. En effet, la tâche *Gérer un baraquement* est décomposée en deux groupes de deux sous-tâches qui sont assignées tantôt au rôle de *Sergent*, tantôt au rôle de *Chef*. La présence des parenthèses permet d'éviter une décomposition en sous-tâches.

Cette fois encore, la solution consiste à modéliser des tâches collaboratives comme des tâches coopératives et oblige à multiplier le nombre de tâches. Ces solutions ne sont pas satisfaisantes car la description produite présente des redondances qui nuisent à sa concision et à sa lisibilité.

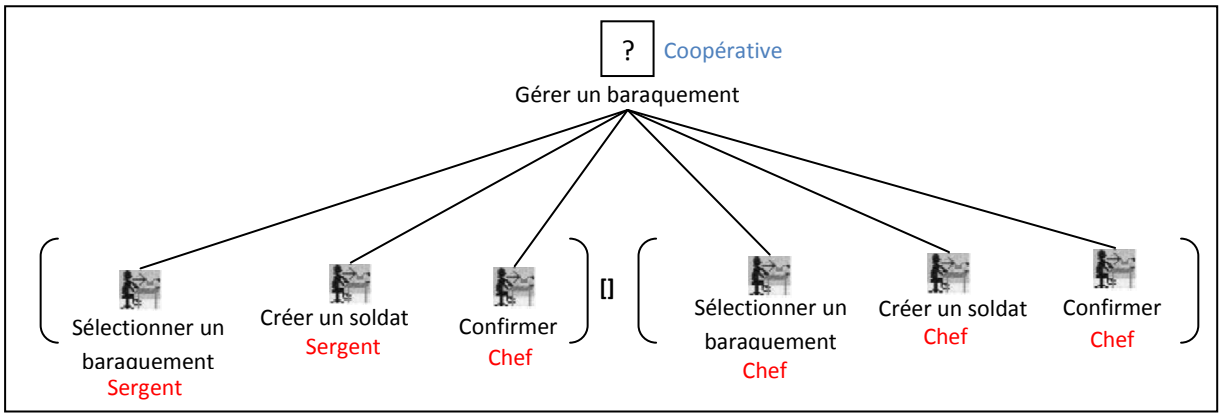


Figure 80 : Utilisation de tâche coopérative pour palier aux limites de la modélisation de la collaboration.

Nous avons illustré les limites que présentent certaines notations existantes en termes de modélisation de la coopération et de la collaboration en considérant des exemples. Nous avons en particulier souligné le problème de concision des modélisations obtenues avec les notations existantes. De plus, l'utilisation des notations existantes nous a conduit à décrire la collaboration entre les utilisateurs sous la forme de coopération entre les rôles. Dans les sous-sections suivantes, nous présentons notre solution pour pallier à ces limites.

2.2.2 Notre solution

Les limites en termes de spécification des activités coopératives et collaboratives que nous avons illustrées dans la sous-section précédente ont pour conséquences principales d'augmenter la taille des spécifications et de compliquer la tâche du concepteur qui doit revisiter ses descriptions et représenter la collaboration sous la forme de coopération. Face à ce constat, nous proposons une solution qui s'articule en plusieurs points :

- Tout d'abord, nous proposons d'introduire un nouveau type de tâche plus générique : les tâches multiutilisateurs afin de remplacer et d'englober les tâches coopératives et les tâches collaboratives.
- Ensuite, il s'agit d'associer pour chaque tâche le ou les rôles qui peuvent la réaliser, et de compléter cette description par le nombre d'utilisateurs qui peuvent jouer ces rôles.
- Enfin, nous introduisons le concept de rôle interactif en contraste des rôles que l'on peut qualifier de rôles métiers présents dans les notations existantes.

Ces trois points qui constituent notre solution pour décrire l'interaction multiutilisateurs sont détaillés dans les trois sous-sections suivantes.

2.2.2.1 Tâche multiutilisateur

Outre la difficulté à spécifier finement la tâche (comme l'illustre la Figure 80 par exemple), le problème est que la spécification obtenue sera exploitée pour la réalisation logicielle de l'application.

Or si les tâches sont dupliquées pour décrire la collaboration sous la forme de coopération, cela peut conduire :

- soit à une application développée avec des portions redondantes, pour respecter la spécification,
- soit un travail supplémentaire pour factoriser la spécification avant de la développer.

Dans les deux cas, cela implique un effort supplémentaire lors des phases de conception logicielle.

De plus, lors de la conception logicielle, la nature coopérative ou collaborative d'une tâche n'a pas grande importance. En effet, lors du développement, il s'agit de savoir quel rôle donne accès à quelles fonctionnalités, c'est-à-dire à quelles tâches. Ainsi, lors d'une activité multiutilisateur, il s'agit de savoir quels rôles peuvent y participer, c'est-à-dire quels rôles peuvent réaliser les tâches qui la décrivent. La nature coopérative ou collaborative de la tâche n'a pas d'importance pour le développement. Il nous apparaît donc comme inutile de décrire explicitement la nature coopérative ou collaborative d'une tâche pour des spécifications devant servir de référence au développement.

Nous proposons donc d'éviter aux concepteurs de modéliser la nature de la tâche afin qu'ils se concentrent sur la question des rôles pouvant les réaliser. Ainsi, nous introduisons le concept de tâche multiutilisateur qui est une tâche pouvant être réalisée par un ou plusieurs utilisateurs. Une tâche multiutilisateur n'implique aucune hypothèse quant à la nature coopérative ou collaborative de la tâche. La décomposition en sous-tâches et les décorations de la notation, tels que les rôles assignés aux tâches renseignent sur la nature coopérative ou collaborative sans qu'il soit utile de le préciser explicitement. En introduisant le concept de tâche multiutilisateur, nous supprimons également le concept de tâches coopératives. La Figure 81 illustre les éléments de la notation COMM après l'introduction de ce nouveau type de tâche : les tâches multiutilisateurs, et la suppression des tâches coopératives.

Lors de l'utilisation de la notation CTT [Mori 2002], la coopération est décrite au sein d'un arbre de tâches coopératives et les tâches individuelles associées à chaque rôle sont décrites dans un arbre spécifique (un par rôle). Cette décomposition en plusieurs arbres est moins pratique pour décrire la collaboration, puisque les sous-tâches individuelles ne sont associées à aucun rôle. Aussi, nous proposons de décrire toutes les tâches au sein d'un arbre de tâches unique qui comporte aussi bien des tâches individuelles que multiutilisateurs (correspondant elles-mêmes à des tâches coopératives ou collaboratives).

Pour conclure cette section, les trois amendements à notre notation (Figure 81) sont :

1. La suppression du type de tâche : coopérative
2. L'ajout du type de tâche : multiutilisateur
3. La spécification de l'interaction au sein d'un arbre unique (au lieu d'un arbre coopératif et de plusieurs arbres individuels).

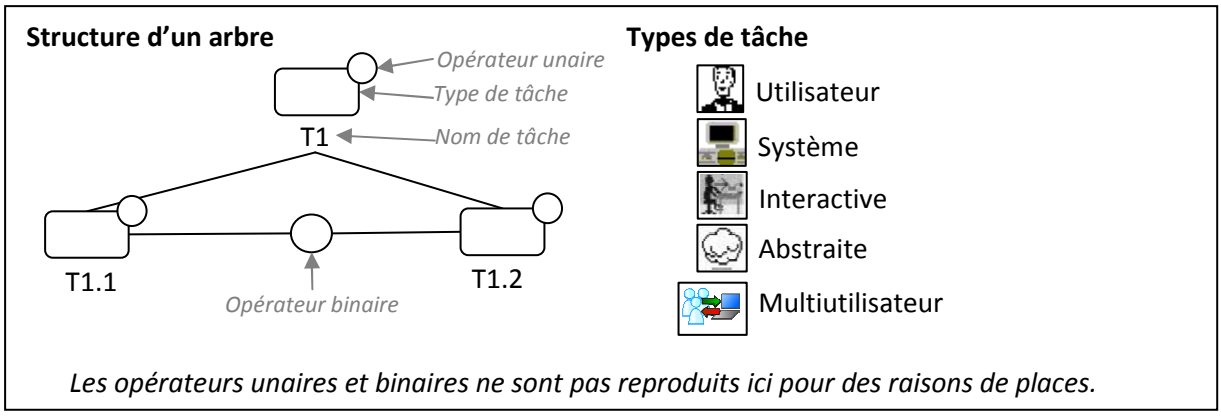


Figure 81 : Eléments de la notation COMM après l'introduction de la tâche multiutilisateur.

2.2.2.2 Rôle métier et nombre d'utilisateurs

Dans la plupart des notations actuelles, y compris CTT [Mori 2002], seules les tâches feuilles d'un arbre de tâches peuvent être associées à un rôle. La notation CIAN [Molina 2006] introduit une nuance pour les arbres de tâches collaboratives. Il s'agit de décrire au niveau de la tâche collaborative de plus haut niveau, les rôles qui peuvent participer à la collaboration.

Nous avons choisi de ne pas utiliser de tâche explicitement collaborative, cependant, assigner des rôles à un autre niveau que celui des tâches feuilles est pertinent pour trois raisons :

1. La première raison est la capacité de factorisation que cela induit. Celle-ci est illustrée à la Figure 82. Considérons une tâche T décomposée en deux sous-tâches *T1* et *T2* associées au *Rôle 1*. Une factorisation simple permise avec l'association de rôle à une tâche non-feuille consiste à associer le *Rôle 1* directement à la tâche *T* plutôt qu'aux deux sous-tâches.

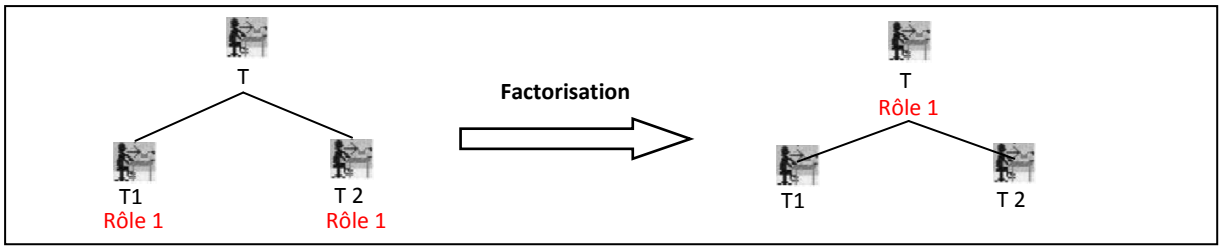


Figure 82 : Factorisation de la distribution des rôles dans un arbre de tâche.

2. La deuxième raison est le renforcement de l'approche descendante (*top-down*) qu'introduit la spécification d'un arbre de tâches. En effet, spécifier un arbre de tâche consiste à décrire une tâche de très haut niveau d'abstraction, qui est raffinée successivement en sous-tâches jusqu'au niveau des tâches concrètes. Associer des rôles aux tâches de haut niveau d'abstraction permet donc de se poser en amont de l'interaction concrète la question de la répartition des tâches. Cette répartition se prolonge tout au long des raffinements successifs jusqu'au niveau des tâches concrètes comme l'illustre la Figure 83. A la première étape, le concepteur associe les rôles *R1* et *R2* à la tâche racine *T*. Lors de la deuxième étape, le concepteur décompose la tâche *T* en deux sous-tâches : *T1* et *T2*, puis leurs associe le rôle *R1* à *T1* et les rôles *R1* et *R2* à *T2*. Lors de la troisième étape, il/elle décompose la tâche *T2* en deux sous-tâches, *T2.1* et *T2.2,* auxquelles il/elle associe respectivement les rôles *R1* et

R2.L'association des rôles s'effectue de manière continue et cohérente lors des raffinements successifs.

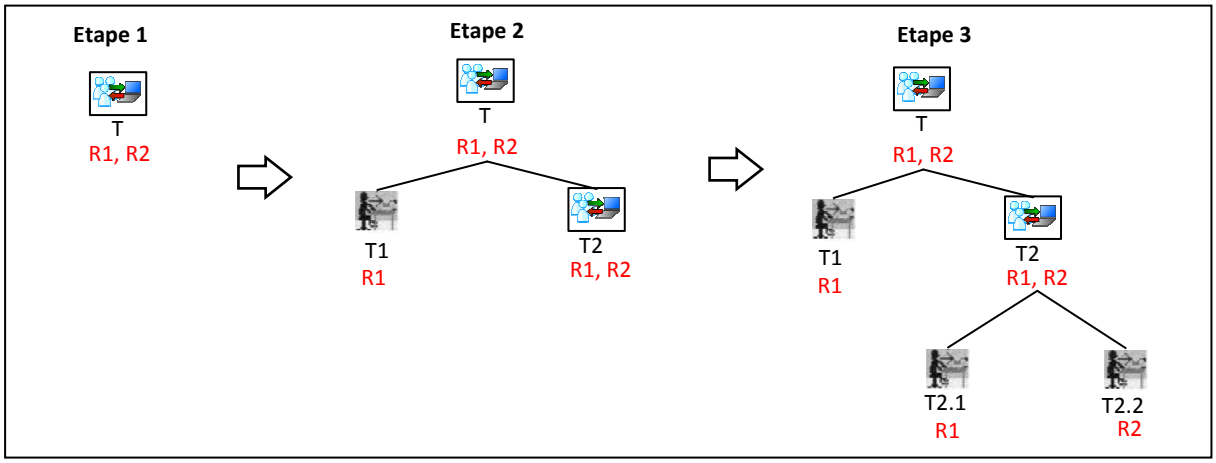


Figure 83 : Association des rôles aux tâches à chaque étape du raffinement en sous-tâches.

3. Enfin, associer des rôles à des tâches non feuilles permet de préciser quel est le nombre d'utilisateurs qui jouent un rôle donné. En effet, si nous reprenons l'exemple de la tâche *Gérer les troupes* de l'application WCCM avec plusieurs utilisateurs jouant le rôle de *Sergent* et un seul utilisateur jouant le rôle de *Chef*, nous pouvons décrire simplement la présence de deux utilisateurs jouant le rôle *Sergent*. La Figure 84 reprend la modélisation de cette tâche, initialement proposée à la Figure 75, adaptée aux amendements proposés. Ainsi, la tâche *gérer les troupes* est associée à un ou deux utilisateurs jouant le rôle de *Sergent* et à un utilisateur jouant le rôle de *Chef*. Les sous-tâches *Déplacer un soldat* et *Former un groupe* sont associées à un utilisateur jouant le rôle de *Sergent.* Enfin, la tâche *Déplacer un groupe* est associée au rôle de *Chef.*

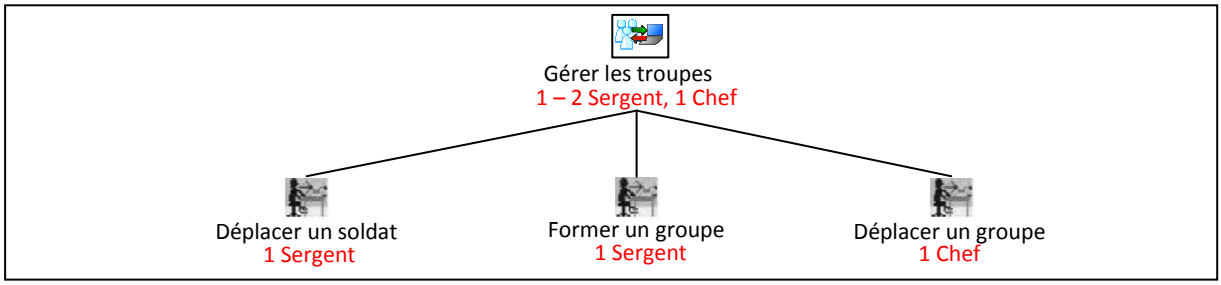


Figure 84 : Tâche gérer les soldats avec la spécification des rôles et du nombre d'utilisateur.

Néanmoins, la description des rôles et des utilisateurs à tous les niveaux ne présente pas que des avantages. En effet, cela implique pour le concepteur de se poser la question pour chaque tâche : quels sont les rôles pouvant la réaliser, et combien d'utilisateurs peuvent jouer chaque rôle. Pour une application de grande taille, cela peut être fastidieux. Aussi, cette étape dans la spécification est optionnelle si le concepteur juge qu'il n'y a pas d'ambiguïté.

Pour résumer, l'extension de notre notation (Figure 85) consiste à décrire pour chaque tâche le ou les rôles requis pour réaliser la tâche, ainsi que le nombre d'utilisateurs pouvant jouer ces rôles. Par défaut, lorsqu'un seul utilisateur jouant un rôle donné est requis, il n'est pas utile de préciser le nombre d'utilisateurs.

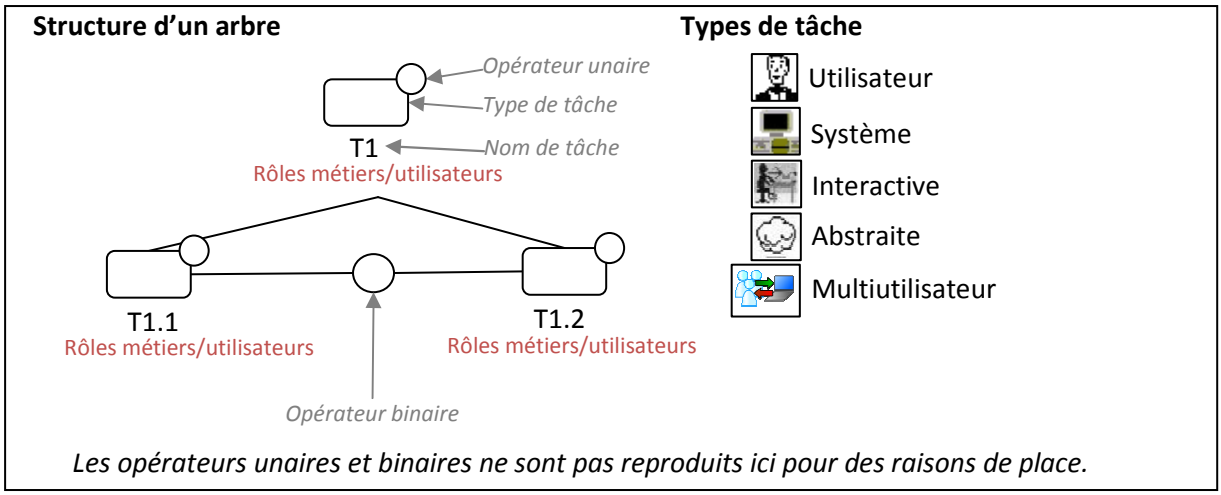


Figure 85 : Eléments de la notation COMM après l'introduction des rôles et du nombre d'utilisateurs pour les tâches.

2.2.2.3 Rôle interactif

L'analyse des différentes notations montre que la structuration de l'activité de groupe repose sur la notation de rôle métier. Cette approche est restrictive, en particulier pour les tâches collaboratives. En effet, pour ces dernières, il est nécessaire de se placer du point de vue de l'utilisateur. Nous nous focalisons ici sur un utilisateur et non un rôle comme à la section précédente.

L'usage d'une tâche multiutilisateur autorise la décoration de celle-ci par plusieurs rôles comme l'illustre la Figure 83. L'approche descendante décrire dans la section précedente a pour conséquence de générer un arbre dont les feuilles sont décorées avec un seul rôle. La répartition des tâches en fonction des rôles est alors figée dès la conception et ne laisse aucune souplesse lors de l'exécution. Il s'agit alors d'une tâche nécessairement coopérative et nous avons montrée les limites d'une telle approche.

Une tâche collaborative est au contraire opportuniste et la répartition ne peut être figée au moment de la conception. Cette répartition se fait à l'exécution et est décidée par les utilisateurs jouant les rôles associés à la tâche collaborative. Aussi, ce type de situation peut être décrit si l'on raisonne en termes d'utilisateur, et non plus en termes de rôle. Pour cela, nous proposons le concept de rôle interactif, en contraste au concept classique de rôle, que nous qualifions de rôle métier.

Une tâche *T2* peut être décorée par un rôle interactif si celle-ci est une sous-tâche d'une tâche multiutilisateur T1. Un rôle interactif est une décoration de tâche qui indique qu'un lien s'établit dynamiquement entre un utilisateur et un ensemble de tâche. Il permet d'indiquer l'association et l'exclusion dynamique de plusieurs tâches à un utilisateur donné. Un rôle interactif est associé à l'utilisateur lorsque celui-ci réalise une tâche qui est décorée par ce rôle. Le rôle reste lié à l'utilisateur jusqu'à ce que l'instance de la tâche mère se termine. Un utilisateur lié à un rôle interactif est alors le seul à pouvoir réaliser les autres tâches décorées par ce même rôle interactif.

Ceci permet d'indiquer l'association dynamique d'un ensemble de tâches à un utilisateur donné. Par ailleurs, cet utilisateur ne peut pas réaliser une tâche décorée par un rôle interactif différent, permettant d'exclure dynamiquement un utilisateur de la réalisation d'un ensemble de tâches données.

Nous illustrons à la Figure 86 l'usage de rôle interactif pour la tâche multiutilisateur : *Gérer un baraquement*. Elle est associée à un rôle métier de *Sergent* et un rôle métier de *Chef*, ce qui signifie que deux utilisateurs sont requis pour la réaliser. Nous décrivons trois sous-tâches : les tâches *Sélectionner un baraquement* et *Créer un soldat* qui sont associées à un rôle interactif *<Créateur>* et la tâche *Confirmer* qui est associée à un rôle interactif *<Validateur>.*

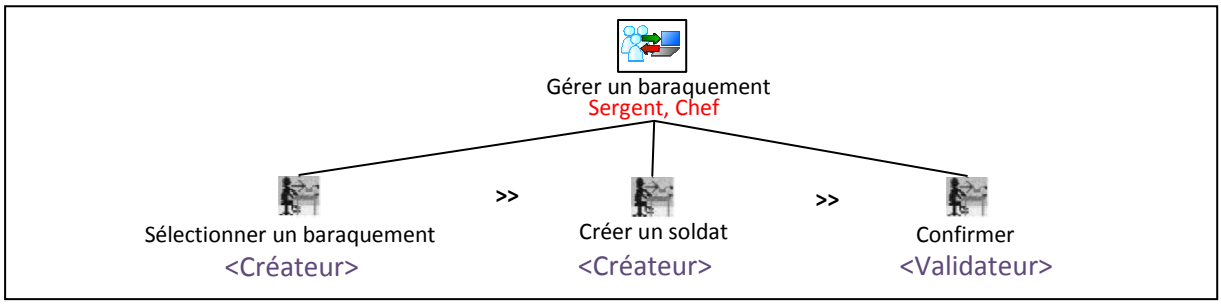


Figure 86 : Utilisation de rôle interactif pour décrire la tâche de gestion d'un baraquement.

L'arbre de tâches de la Figure 86 signifie que lorsqu'un utilisateur (jouant soit le rôle métier de *Sergent*, soit le rôle métier de *Chef*) réalise la tâche *Sélectionner le baraquement*, il devient le seul à pouvoir réaliser la tâche *Créer un soldat*. En effet, lorsqu'il sélectionne le baraquement, le rôle interactif de *<Créateur>* lui est dynamiquement assigné. La tâche *Créer un soldat* étant décorée également par ce rôle interactif, il devient le seul à pouvoir la réaliser. De la même manière, cet utilisateur ne pourra pas réaliser la confirmation de la création du soldat. En effet, la tâche *Confirmer* est décorée du rôle interactif *<Validateur>*, alors que l'utilisateur est devenu un *<Créateur>* pour cette instance de tâche *Gérer un baraquement*. Seul l'autre utilisateur pourra donc réaliser la tâche *Confirmer*. S'il le fait, le rôle de *<Validateur>* lui est dynamiquement assigné jusqu'à la fin de l'instance en cours de la tâche Gérer un baraquement, c'est-à-dire, juste après qu'il a réalisé la tâche de confirmation. En effet, la tâche *Gérer un baraquement* étant considérée comme réalisée, l'assignation dynamique des rôles interactifs aux deux utilisateurs est rompue. Une autre instance de la tâche *Gérer un baraquement* peut alors être commencée pour peu qu'un des deux utilisateurs décide de *sélectionner le baraquement* et de devenir ainsi un *<Créateur>*.

La comparaison de la Figure 86 avec la Figure 80 souligne la concision qu'apporte l'utilisation de rôle interactif à la spécification. Dans ce cas précis, cela permet de décrire une tâche collaborative complexe, sans recourir à la duplication de tâches ou à l'introduction de tâche abstraite supplémentaires. Les décorations sous forme de rôle interactif suffisent.

Outre la concision de la spécification, considérons un autre exemple pour illustrer le pouvoir d'expression du rôle interactif.

Considérons le cas où le concepteur souhaite que seul l'utilisateur jouant le rôle métier de *Chef* puisse réaliser la confirmation. La spécification conserve la même forme et ne subit qu'une modification des décorations de tâches, comme l'illustre la Figure 87. La tâche *Confirmer* perd alors

son rôle interactif de <*Validateur*> et est décorée du rôle métier de *Chef*. La spécification indique comme précédemment qu'un utilisateur (soit le *Sergent*, soit le *Chef*) peut réaliser les deux premières sous-tâches, tandis que seul l'utilisateur jouant le rôle métier de *Chef* peut réaliser la confirmation.

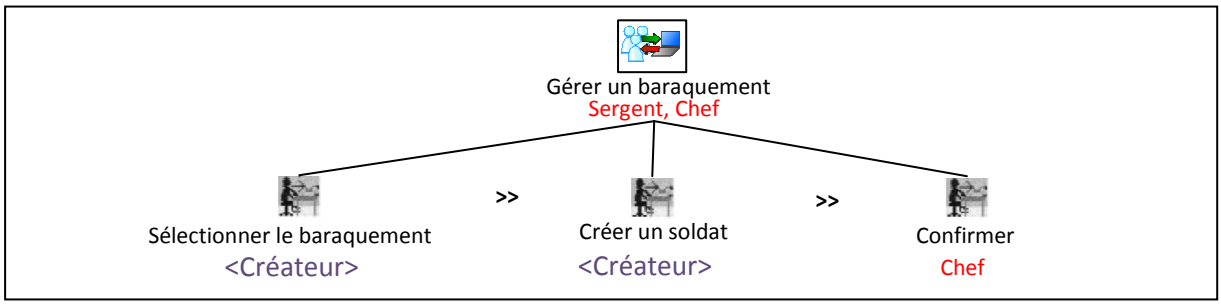


Figure 87 : Utilisation conjointe de rôle métier et de rôle interactif pour la tâche de gestion d'un baraquement.

Pour conclure cette section, l'extension de notre notation consiste à introduire le concept de rôle interactif. Nous précisons également que pour décrire la notion de rôle présente dans les notations existantes, nous utiliserons dorénavant le terme de rôle métier. Les éléments de la notation COMM prennent alors la forme illustrée à la Figure 88. De plus, un rôle interactif n'a de sens que dans le cadre d'une décomposition en sous-tâches, c'est pourquoi nous avons indiqué la possibilité de décrire un rôle interactif uniquement pour les sous-tâches et non pour la racine.

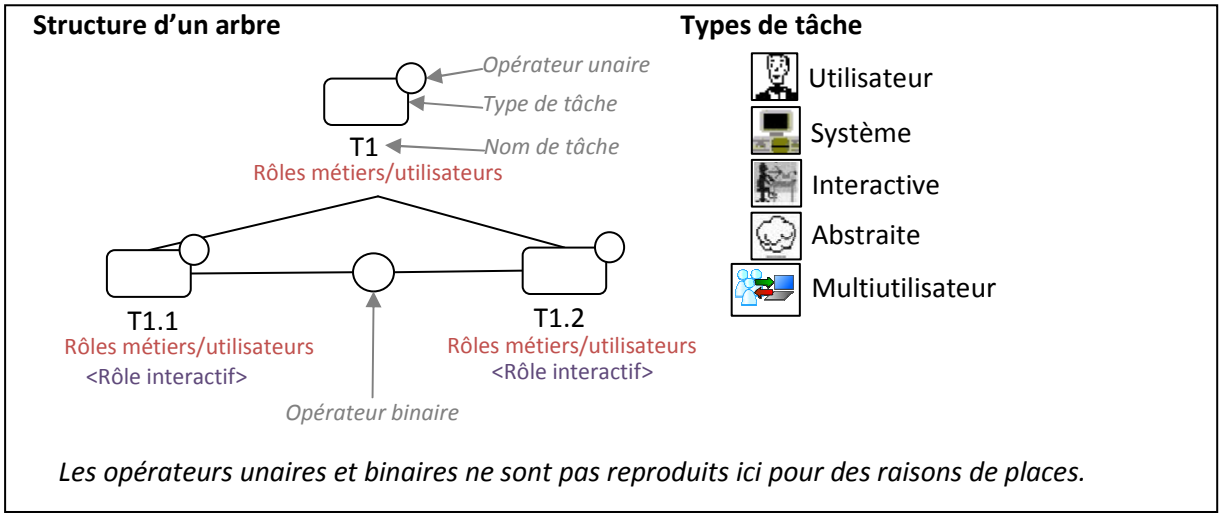


Figure 88 : Eléments de la notation COMM après introduction du concept de rôle interactif.

2.2.3 Mise en cohérence de la notation

Ayant introduit de nouveaux concepts dédiés à la spécification de l'interaction multiutilisateur, et en particulier un nouveau type de tâche (tache multiutilisateur), il convient de s'assurer de la cohérence de la notation résultante vis-à-vis de nos objectifs. Nous considérons ainsi les concepts hérités de CTT, notamment, les différents types de tâches et les opérateurs unaires d'itération.

2.2.3.1 Types de tâches

L'introduction d'un type de tâche multiutilisateur implique un questionnement sur les autres des types de tâche de la notation. Pour cela nous présentons trois dimensions orthogonales : le niveau d'abstraction, le(s)auteur(s) et l'interactivité, qui se combinent au sein des définitions de types de tâches de la notation : utilisateur, système, interactive, abstraite, multiutilisateur.

2.2.3.1.1 Première dimension : niveau d'abstraction

Considérons tout d'abord le niveau d'abstraction des tâches. Pour cela, il convient d'abord de définir ce qu'est le niveau d'abstraction d'une tâche.

Une première définition du niveau d'abstraction s'appuie sur la structure hiérarchique du modèle de tâche, et l'opposition avec le concept de tâche élémentaires (ou tâches atomiques). Une tâche abstraite peut ainsi être définie comme une tâche non-élémentaire. Cependant, puisqu'une tâche élémentaire est une tâche que le concepteur n'as pas souhaité décomposer, une définition de la tâche abstraite reposant sur l'opposition avec les tâches élémentaires est trop fragile.

Les notations CTT et K-MAD modère cette approche en considérant qu'une tâche est abstraite s'il elle est décomposée en sous-tâches de plusieurs de types différents. Une tâche décomposées en plusieurs sous-tâches du même type prend alors ce type. Cette approche est illustrée à la Figure 89. L'arbre de tâches (A) est composées de deux sous-tâches T1 et T2 de type interactif. La tâche T mère de ces sous-tâches est donc de type interactif. A l'inverse, dans l'arbre de tâches (B), comme les sous-tâches T1 et T2 sont de type différent, alors la tâche T est de type abstrait. Cette approche permet de limiter la fragilité de la définition du type de tâche abstraite reposant sur l'opposition avec les tâches élémentaires. En effet, Si une tâche élémentaire d'un type donné est décomposée en sous-tâches du même type alors elle n'en devient pas pour autant abstraite.

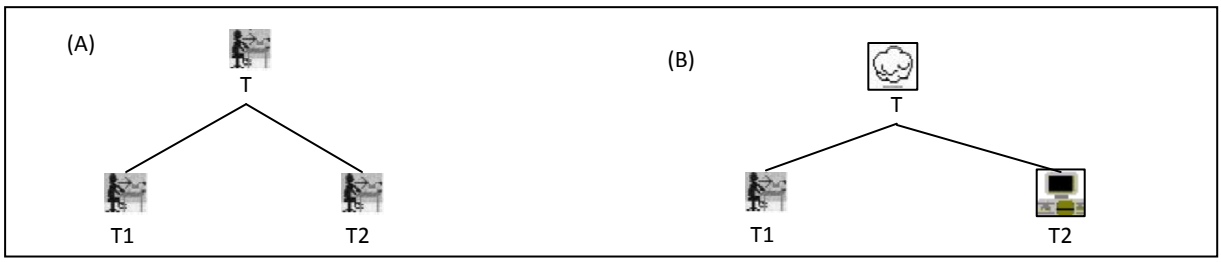


Figure 89 : Typage des tâches non-feuille dans CTT.

Considérons maintenant une deuxième définition du niveau d'abstraction qui s'appuie sur les décompositions opérées dans le modèle ARCH [Bass 1992] entre le contrôleur de dialogue, et l'interface. Sur la base de ce modèle, nous pouvons définir que les tâches abstraites décrivent le contrôleur de dialogue, tandis que les tâches élémentaires décrivent l'interface. Cependant, la frontière entre le dialogue et l'interface peut varier selon le métamodèle Slinky [Bass 1992]. Ceci n'est pas très cohérent avec le concept de tâche élémentaire puisque celle-ci sont élémentaire de part le fait qu'elles ne sont pas décomposées. Aussi, faut t'il effectuer une distinction entre des tâches abstraites correspondant au contrôleur de dialogue et des tâches concrètes correspondant à l'interface.

Cette distinction tâche abstraite/concrète est cohérente avec la définition des types de tâches adoptées par les notations CTT et K-MAD si l'on considère que les règles de typages comme une

référence pour déterminer l'appartenance d'une tâche au niveau abstrait/concret. Cette approche ne permet en revanche pas de faire varier le niveau d'abstraction selon le métamodèle Slinky. En effet, le niveau d'abstraction est intégré dans le type de tâche avec d'autres aspects comme l'auteur de la tâche.

De plus, nous constatons une redondance partielle dans l'expression du niveau d'abstraction entre les tâches de types multiutilisateur (qui sont des tâches abstraites obligatoirement décomposées en sous-tâches) uniquement par des tâches de type abstraite. Cette redondance partielle est également présente entre les tâches coopérative et abstraite de CTT [Mori 2002].

Nous adoptons la définition du niveau d'abstraction comme une frontière définie selon le modèle ARCH entre les tâches abstraites décrivant le contrôleur de dialogue et des tâches concrètes décrivant l'interface. Enfin, le point de vue lors de la réalisation de spécification est un choix du concepteur, c'est pourquoi, conformément au métamodèle Slinky, nous devons laisser la liberté au concepteur de réaliser la distinction entre tâches abstraites/concrètes.

2.2.3.1.2 Deuxième dimension : auteur(s) de la tâche

La définition des types de tâches dans la notation COMM est fondée en partie sur la distinction de l'auteur de la tâche, c'est-à-dire QUI peut réaliser la tâche. Nous pouvons en effet distinguer :

- les tâches de types utilisateurs et interactive comme des tâches réalisable par un utilisateur seul,
- les tâches de type multiutilisateurs comme des tâches réalisables par plusieurs utilisateurs,
- les tâches de type système réalisable par le système.

La dimension de l'auteur de la tâche est orthogonale à la dimension du niveau d'abstraction présenté à la section précédente. En effet, il est possible de décrire des tâches réalisables par un utilisateur seul aussi bien comme une tâche abstraite tels que la tâche *Gérer les paysans* de l'application WCCM définit au Chapitre 1, ou comme une tâche élémentaire tels que la tâche *Sélectionner un soldat*.

2.2.3.1.3 Troisième dimension : interactivité

La troisième dimension qui contribue à définir les types de tâches actuelles est la question de l'interactivité de la tâche, c'est-à-dire le fait qu'une tâche indique ou non une interaction entre le(s) utilisateur(s) et le système.

Ainsi, parmi les types de tâches de la notation, les types interactive, multiutilisateur et système permettent d'exprimer une interaction entre le (ou les) utilisateur(s) et le système. En revanche, les tâches utilisateurs indiquent par leur définition une absence d'interaction de l'utilisateur avec le système. Enfin, les tâche de type abstraite ne peuvent être définie a priori comme décrivant un dialogue ou non entre le(s) utilisateur(s) et le système. En effet, c'est la décomposition en sous-tâches qui permet d'indiquer s'il s'agit ou non d'un dialogue. Ainsi, si la décomposition d'une tâche abstraite fait apparaitre au moins une tâche interactive ou multiutilisateur, alors elle peut être considérée comme décrivant un dialogue.

La dimension interactive d'une tâche est orthogonale à son niveau d'abstraction, puisque le niveau d'abstraction est définit par le concepteur selon le modèle ARCH [Bass 1992] et le métamodèle Slinky [Bass 1992].

La dimension interactive d'une tâche est également orthogonale à la dimension des acteurs, puisqu'aussi bien les utilisateurs que le système peuvent être l'auteur d'une tâche interactive.

2.2.3.1.4 Nouveaux types et propriétés de tâches

Considérons maintenant la conjugaison des dimensions de l'interactivité et des auteurs dans la définition des types de tâches de la notation COMM. Le Tableau 11 illustre la répartition des types de tâches selon ces deux dimensions. Nous notons que pour un utilisateur seul, une dichotomie est opérée selon la dimension de l'interaction. En effet, les tâches utilisateurs et interactives sont toutes les deux des tâches mono-utilisateurs, mais qui sont soit non-interactives soit interactives. En revanche, les tâches multiutilisateurs et les tâches systèmes sont utilisées indifféremment qu'ils s'agissent de tâches interactives ou non.

	Mono-utilisateur	Multiutilisateur	Système
Sans interaction	Utilisateur	Multiutilisateur	Système
Interaction	Interactive	Multiutilisateur	Système

Tableau 11 : Répartition actuelle des type de tâches COMM selon la dimension interactives et des auteurs.

Cette non-distinction existe également dans CTT, comme le montre la Tableau 12 qui illustre la répartition des types de tâches CTT selon ces mêmes dimensions.

	Mono-utilisateur	Multiutilisateur	Système
Sans interaction	Utilisateur	Coopérative	Système
Interaction	Interactive	Coopérative	Système

Tableau 12 : Répartition des types de tâches CTT selon la dimension interactives et des auteurs.

Pour lever ces ambigüités, nous créons 6 types de tâches distincts correspondant à nos deux dimensions, comme le montre le Tableau 13. Ainsi une tâche multiutilisateur peut être soit une tâche de groupe entre les utilisateurs sans interaction avec le système, soir correspondre à une interaction de groupe. Les tâches systèmes peuvent correspondre à des tâches de calculs ou à des tâches de présentations.

	Mono-utilisateur	Multiutilisateur	Système
Sans interaction	Utilisateur	Groupe	Calcul
Interaction	Interaction	Interaction de groupe	Présentation

Tableau 13 : Nouvelle types de tâches COMM et répartitions selon la dimension du QUI et du QUOI.

Considérons maintenant la troisième dimension qui concerne le niveau d'abstraction de la tâche.

Tout d'abord, afin d'éviter d'utiliser un type général abstrait comme c'est le cas dans les notations CTT et K-MAD, nous proposons d'adopter la convention décrite au sein du Tableau 14. Chaque case du tableau correspond au type de la tâche mère de deux tâches dont les types sont définis en abscisse et en ordonnée. Ainsi par exemple, si une tâche *T* est décomposée en deux sous-tâches dont l'une est de type Interaction et l'autre de type Interaction de Groupe (IG), alors le type de *T* est Interaction de Groupe. Cette convention de typage permet d'éliminer de la notation le concept de tâche abstraite telle qu'il est défini dans la notation CTT. Toute tâche COMM doit donc avoir un type défini parmi les 6 types : calcul, utilisateur, présentation, interaction, groupe et interaction de groupe.

	Calcul	Utilisateur	Présentation	Interaction	Groupe	Interaction de groupe
Calcul (C)	C	U	I	I	IG	IG
Utilisateur (U)		U	I	I	IG	IG
Présentation (P)			P	I	IG	IG
Interaction (I)				I	IG	IG
Groupe (G)					G	IG
Interaction de groupe (IG)						IG

Tableau 14 : Type de la tâche mère d'une décomposition comprenant deux sous-tâches.

Avec ces nouvelles définitions de type de tâche et la suppression du type abstraite, nous supprimons une distinction qui pouvait exister au sein des arbres de tâches : c'est-à-dire la délimitation entre

- un niveau de l'arbre que l'on peut qualifier d'abstrait dans le sens où les tâches qui sont utilisées permettent d'articuler de manière cohérente les sous-tâches, et correspondant au contrôleur de dialogue de ARCH [Bass 1992].
- et un niveau de tâche que l'on peut qualifier de concrète dans le sens où les tâches correspondent à des opérations sur les concepts du domaine.

Nous proposons un moyen d'exprimer le niveau abstrait/concrète des tâches sans pour autant remettre en question les nouveaux type de tâches. Le type d'une tâche considérée comme abstraite est entouré d'un nuage à l'instar de sa représentation dans la notation CTT, tandis que le type d'une tâche considérée comme concrète est entourée par un rectangle. Les tâches abstraites peuvent être décrites en partant de la racine de l'arbre, jusqu'à un niveau de décomposition où figure ensuite des tâches concrètes.

La frontière entre tâches abstraites et tâches concrètes est variable dans un arbre de tâches COMM selon la volonté du concepteur afin de respecter le phénomène de Slinky [Bass 1992] et ce, aussi bien verticalement, qu'horizontalement.

- Verticalement, elle peut intervenir après un nombre quelconque de décompositions, qui varie selon les branches de l'arbre.
- Horizontalement, il est possible qu'une même décomposition introduise à la fois des tâches concrètes et des tâches abstraites.

Pour illustrer la distinction entre tâche abstraite et concrète, considérons à nouveau l'application WCCM. La tâche *Gérer les soldats* de la Figure 90 peut être qualifiée d'abstraite puisqu'elle sert uniquement à articuler les trois sous-tâches : *Déplacer un soldat*, *Former un groupe* et *Déplacer un groupe*. Les tâches *Déplacer un soldat* et *Former un groupe* peuvent être considérées comme concrètes puisqu'elles correspondent à des opérations sur les concepts du domaine : les soldats et les groupes. Enfin la tâche *Déplacer un groupe* peut également être considérée comme concrète car elle décrit également une opération sur le concept de groupe. Elle est malgré cela décomposée en deux tâches qui décrivent comment la réaliser, par la *Sélection du groupe* suivi du *Choix d'une destination*.

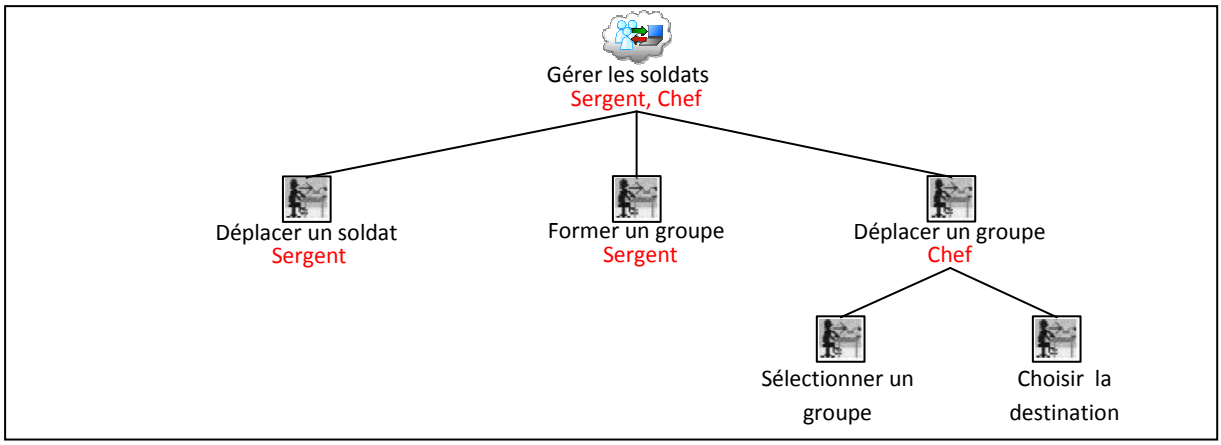


Figure 90 : Modélisation des tâches abstraites et des tâches concrètes.

Pour conclure, nous synthétisons les modifications de la notation pour introduire les nouveaux types de tâches et la distinction de l'interaction abstraite/concrète (Figure 91) :

- Le type de tâche système est supprimé de la notation et remplacé par deux nouveaux types de tâche : les tâches de présentation qui correspondent à une tâche de présentation d'information par le système aux utilisateurs, et les tâches de calcul, qui correspondent à une tâche de calcul du système sans interaction avec l'utilisateur.
- Le type de tâche multiutilisateur est supprimé de la notation et remplacé par deux nouveaux types de tâche : les tâches de groupe qui correspondent à une interaction entre les utilisateurs sans interaction avec le système, et les tâches d'interaction de groupe, qui correspondent à une interaction des utilisateurs avec le système.
- Le type de tâche abstraite est supprimé.
- Les différents types de tâche peuvent être utilisés aussi bien pour des tâches feuilles de l'arbre que pour des tâches internes de l'arbre. Le type d'une tâche dépend du type de ses sous-tâches selon la convention définie par le Tableau 14.
- Une nouvelle propriété de tâche est définie : il s'agit de son niveau d'abstraction qui peut être abstrait ou concret. Une tâche abstraite peut avoir des tâches filles abstraites et concrètes, tandis qu'une tâche concrète peut uniquement avoir des tâches filles concrètes. Nous considérons qu'une tâche est concrète si elle correspond à une opération sur les concepts du domaine.

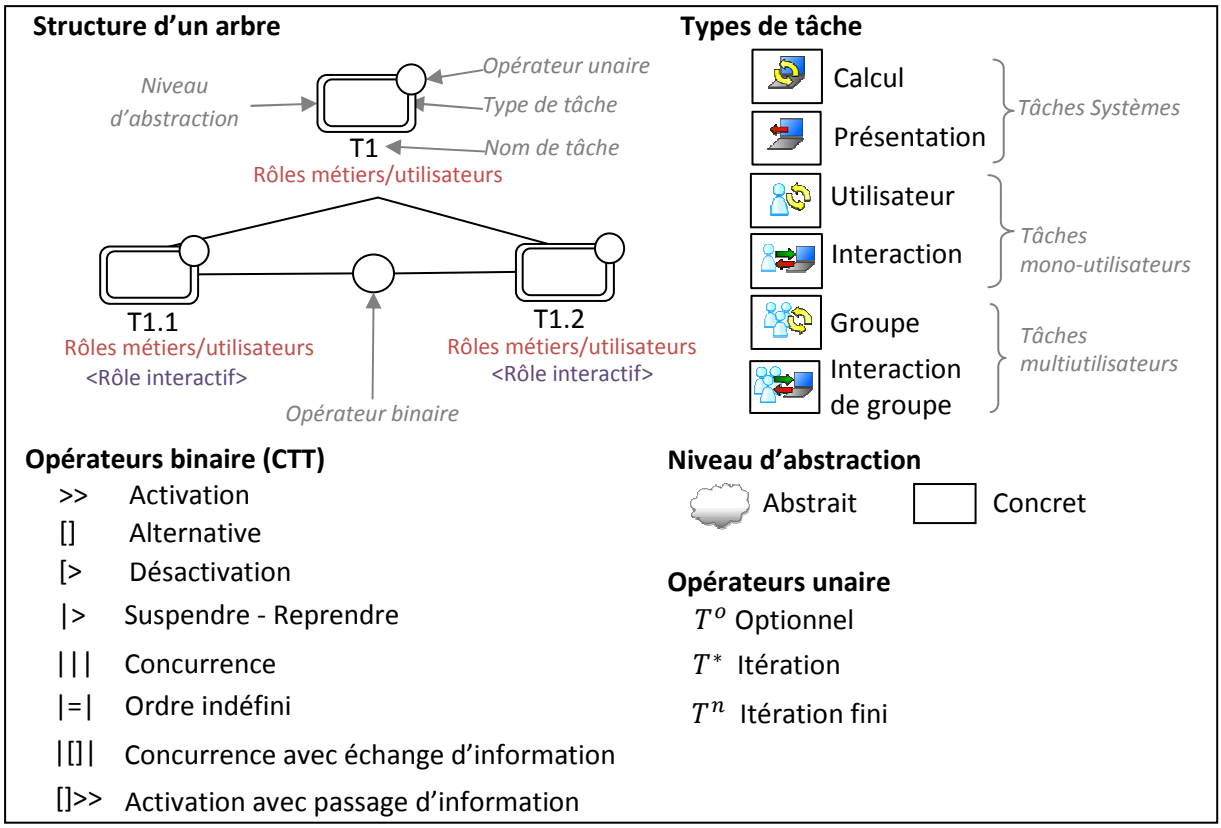


Figure 91 : Eléments de la notation COMM après la mis en cohérence des différents type de tâches.

2.2.3.2 Opérateurs unaires d'itération

La plupart des notations proposent trois opérateurs de base pour lier les tâches : l'alternative, la séquence et le parallélisme. Plusieurs notations dont la notation CTT [Mori 2002] proposent une palette plus large d'opérateurs binaires. Ces opérateurs supplémentaires que l'on peut qualifier de complexes enrichissent les notations et permettent d'exprimer de nombreux cas d'interaction. En revanche, ils ne sont que des variations des opérateurs d'alternative, de séquence et de parallélisme. Ainsi, par exemple, l'opérateur de suspension-reprise de CTT peut être vu comme une alternative entre la tâche en cours et la tâche qui la suspend.

Les notations existantes proposent également plusieurs opérateurs unaires pouvant s'appliquer à une tâche donnée. Les opérateurs unaires les plus courants sont : optionnel et l'itération. Ces opérateurs unaires peuvent être reliés aux opérateurs binaires de base. Ainsi, l'optionalité constitue une alternative entre réaliser une tâche et ne pas la réaliser. De la même manière, l'opérateur unaire d'itération est similaire à l'opérateur séquence, puisqu'il définit la possibilité pour l'utilisateur de répéter la réalisation d'une tâche lorsque celle-ci est terminée. En revanche, il n'y a pas d'opérateur unaire qui puisse être comparé à l'opérateur binaire de parallélisme.

Néanmoins, lorsqu'une tâche peut être répétée plusieurs fois, la répétition peut être séquentielle comme cela est proposé dans les notations existantes, mais également parallèle. Ainsi, dans notre exemple, le concepteur peut choisir de donner la possibilité au rôle métier de *Sergent* de déplacer plusieurs soldats à la fois par glisser-déposer, par exemple grâce à une interaction bi-manuelle sur une surface tactile. Cet exemple ne peut être représenté avec l'opérateur d'itération tel qu'il est défini par les notations existantes.

Les auteurs de [Sinnig 2007] ont souligné ce problème et proposent l'opérateur d'itération parallèle (#), en plus de l'opérateur d'itération séquentielle (*) tel qu'il existe dans la notation CTT. Avec un tel opérateur unaire, il est possible d'exprimer qu'une tâche est répétable plusieurs fois de manière parallèle.

Si l'on considère les opérateurs unaires de base tels qu'ils sont définis dans CTT [Mori 2002] ou K-MAD [Lucquiaud 2005, Baron 2006] auxquels on ajoute l'opérateur d'itération parallèle (#), le concepteur peut spécifier les itérations suivantes :

- T^1 ou T : tâche réalisable une fois
- T^o ou $[T]$: tâche optionnelle, réalisable au plus une fois
- T^n : tâche devant être réalisée *n* fois séquentiellement
- T^* : tâche réalisable un nombre quelconque de fois séquentiellement, y compris zéro
- $T^\#$: tâche réalisable un nombre quelconque de fois parallèlement, y compris zéro

Cet ensemble de possibilité n'est pas complet. Ces opérateurs ne suffisent pas pour décrire une tâche qui peut être réalisé exactement un nombre fini de fois parallèlement alors qu'il est possible de décrire qu'une tâche peut être réalisée exactement un nombre fini de fois séquentiellement. De plus, il est possible de décrire qu'une tâche peut être réalisée jusqu'à une fois, mais pas jusqu'à deux fois, que ce soit dans le cas d'itération parallèle ou séquentielle.

Aussi, nous proposons de distinguer la capacité d'itération d'une tâche selon deux dimensions. Il s'agit de définir d'une part la nature de l'itération et d'autre part sa multiplicité. La nature d'une itération est définie entre séquentielle (>) et parallèle (||), tandis que sa multiplicité est définie par un nombre ou un symbole. Précisons que lorsqu'une tâche ne peut être répétée plus d'une fois, il est inutile de décrire la nature séquentielle ou parallèle de la répétition. La multiplicité d'une tâche est décrite selon les huit expressions suivantes :

- T ou T^1 : exactement une fois
- T^{1*} : jusqu'à une fois
- T^{1+} : au moins une fois
- T^n : exactement *n* fois
- T^{n*} : jusqu'à *n* fois
- T^{n+} : au moins *n* fois
- $T^{n..m}$: entre *n* et *m* fois
- T^* : un nombre quelconque de fois

La Figure 92 illustre l'utilisation du nouvel opérateur d'itération pour la tâche *Gérer les troupes* de notre *exemple*. La sous-tâche *Déplacer un soldat* est définie comme pouvant être répétée jusqu'à deux fois en parallèle par l'utilisateur jouant le rôle métier de *Sergent*. Les tâches *Former un groupe* et *Déplacer un groupe* ne peuvent quant à elle être répétées plusieurs fois séquentiellement. Notons qu'entre les tâches *Déplacer un soldat* et *Former un groupe*, nous utilisons l'opérateur d'alternative. Ainsi l'utilisateur jouant le rôle métier de *Sergent* a le choix entre déplacer un soldat ou former un groupe. Nous utilisons l'opérateur parallèle (III) entre les deux premières sous-tâches et la tâche *Déplacer un groupe* afin d'indiquer que les utilisateurs jouant les rôles métier de *Sergent* et de *Chef* peuvent participer en même temps à la gestion des troupes.

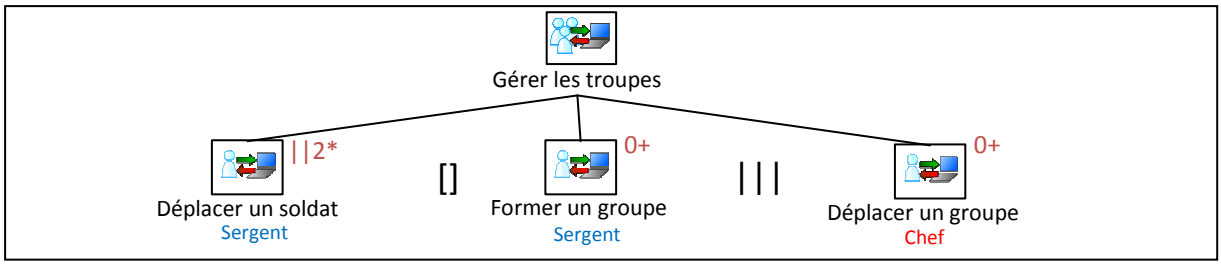


Figure 92 : Répétition parallèle de la tâche déplacer un soldat.

Pour résumé, la modification de notre notation consiste donc à supprimer les opérateurs unaires : optionnel et itération, et à les remplacer par un opérateur unaire unique d'itération définissant la nature de l'itération et sa multiplicité (Figure 93).

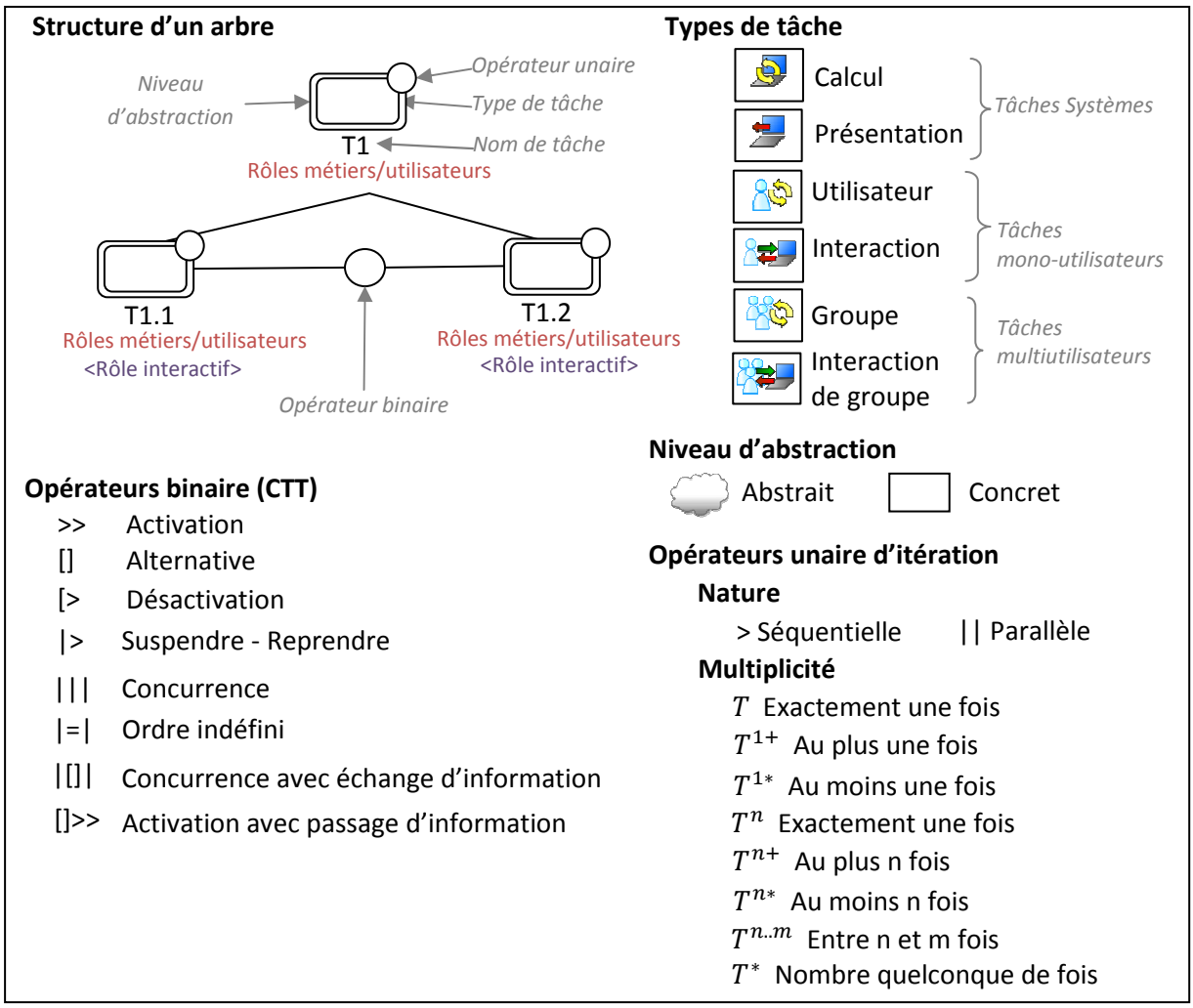


Figure 93 : Elément de la notation COMM après la refonte des opérateurs unaires d'itération.

2.2.4 Synthèse

Dans cette section, nous avons abordé la spécification de l'interaction multiutilisateur de façon incrémentale. A chaque étape de l'élaboration de la notation COMM, nous avons présenté les limites

des notations existantes, et avons proposé plusieurs amendements à la notation COMM pour palier ces limites.

Nous avons jusqu'ici principalement traité de la question de la description de l'interaction abstraite, et n'avons pas abordé la question de la spécification de l'interaction concrète et multimodale. Ceci est l'objet de la section suivante, et nous apportons des évolutions à la notation COMM tout en conservant sa cohérence.

2.3 Spécification de l'interaction multimodale

Cette section est consacrée au deuxième objectif que nous poursuivons dans la définition de notre notation : spécifier l'interaction multimodale. Il s'agit d'exprimer la possibilité d'utiliser des modalités d'interaction, mais également d'exprimer la combinaison de modalités d'interaction pour une tâche donnée.

Parmi les notations qui prennent en compte la dimension multimodale, les notations ICARE [Bouchet 2006] et ICON [Dragicevic 2001] ainsi que la notation du Framework Dynamo-Aid [Clerckx 2005] proposent une approche relativement proche : il s'agit de décrire pour chaque tâche d'une application, la modalité ou la combinaison de modalités qui permet de la réaliser. Tandis que l'approche précise QUELLE est la modalité d'interaction ou QUELLE combinaison de modalités est utilisée pour réaliser une tâche donnée, elle n'indique pas COMMENT cette modalité ou combinaison de modalités est utilisée. Le COMMENT recouvre deux questions : d'une part la manière d'utiliser une modalité donnée pour réaliser la tâche et d'autre part la relation logique et temporelle entre les modalités au sein d'une combinaison.

Sur la base de ces notations, nous proposons une solution en deux volets comprenant :(1) l'introduction du concept de tâche modale inspirée du framework Dynamo-Aid, et permettant de décrire l'usage d'une modalité d'interaction pour réaliser une tâche donnée, et (2) l'intégration d'opérateurs logiques et temporels reposant sur les relations temporelles de Allen pour spécifier avec précision la combinaison de modalités.

Dans les sections suivantes, nous détaillons d'abord les limites des notations existantes, puis nous décrivons notre solution pour répondre à ces limites qui consiste à étendre la notation COMM.

2.3.1 Limite des notations existantes

La notation ICON [Dragicevic 2001] détaillée au Chapitre 2 décrit l'utilisation de modalités d'interaction pour une tâche grâce à un assemblage de composants. Ces assemblages permettent d'indiquer les dispositifs physiques utilisés sous la forme de boîte munie de ports de sortie. Ils permettent également de décrire les tâches d'une application sous la forme de boîte munie de ports d'entrée. Enfin, des composants nommés transducteurs permettent de transformer les données sortantes des dispositifs en entrés compréhensibles par les tâches. ICON présente cependant une limite forte : le comportement interne des composants n'est pas décrit explicitement.

En introduisant la notation ICARE [Bouchet 2006] au Chapitre 2, nous avons décrit la tâche de déplacement d'un soldat. L'assemblage produit comporte un composant de complémentarité reliant une modalité gestuelle et une modalité vocale décrites chacune par deux composants : un composant caméra et un composant reconnaissance de geste pour la modalité gestuelle, et un

composant microphone et un composant reconnaissance de parole pour la modalité vocale. Lors d'un tel assemblage, ni le composant de reconnaissance de geste, ni le composant de reconnaissance de parole n'est décrit avec plus de détails. Aussi, il n'est pas possible d'indiquer quel geste est reconnu, ni quel énoncé est reconnu. L'utilisation d'une modalité ne peut donc être exprimée que partiellement avec ICARE. D'autre part, le composant de complémentarité est défini selon les propriétés CARE [Coutaz 1994]. La complémentarité indique ainsi l'usage conjoint des deux modalités pour réaliser la tâche. Cependant, cela n'indique pas l'ordre dans lequel ces modalités doivent être utilisées. Dans le cas de notre exemple, l'usage de la modalité vocale doit être réalisé pendant l'usage de la modalité gestuelle, c'est-à-dire que l'utilisateur doit désigner un point du doigt, dire « aller ici », et peut enfin arrêter de désigner le point du doigt. Les combinaisons de modalités ne peuvent être exprimées que partiellement avec les propriétés CARE telles qu'elles sont utilisées dans ICARE.

Avec le framework Dynamo-AID [Clerckx 2005] décrit au Chapitre 2, une modalité ou combinaison de modalités peut être associée aux tâches élémentaires d'un arbre de tâche de type CTT. La notation est illustrée au Chapitre 2 (Figure 66) pour la tâche *déplacer un soldat*. La combinaison des modalités vocale et gestuelle est décrite par une étiquette *C[gestuelle, vocale]* reliée par un trait à la tâche *déplacer le soldat*. Le C indique la propriété CARE de complémentarité, tandis que les étiquettes *gestuelle* et *vocale* indiquent les modalités complémentaires. Une première limite de cette approche est que les étiquettes *gestuelle* et *vocale* n'indiquent pas les dispositifs physiques qui captent le geste ou la parole des utilisateurs. Ces étiquettes ne précisent pas non plus les gestes ou les paroles qui doivent être faits ou prononcés pour réaliser la tâche. En cela, l'approche ne décrit que partiellement l'usage d'une modalité d'interaction. Ensuite, à l'instar de la notation ICARE, l'usage des propriétés CARE sans autre précision au sein du framework Dynamo-AID ne permet pas d'indiquer l'ordre dans lequel les modalités doivent être utilisées. Dynamo-AID permet donc d'exprimer partiellement les combinaisons de modalités.

Les notations UAN [Hartson 1990] et NUAN [Venema 2000] proposent une approche assez proche de celle d'ICARE ou de Dynamo-AID, c'est-à-dire de décrire pour une tâche donnée, la ou les modalités d'interaction qui interviennent. UAN et NUAN vont cependant beaucoup plus loin en termes de précision. En effet, l'usage d'un dispositif physique est décrit en terme d'actions élémentaires, tel qu'un clic souris. La combinaison de plusieurs actions élémentaires sur plusieurs dispositifs physiques permet ainsi d'exprimer une combinaison de modalités. La modélisation avec UAN et NUAN est en revanche complètement déconnectée de l'arbre de tâches. En effet, la spécification s'effectue au sein d'un tableau, où chaque ligne du tableau correspond à une tâche concrète. La continuité avec un arbre de tâches est donc difficile à maintenir. De plus, les notations UAN et NUAN se limitent en termes de dispositifs à l'usage du clavier et de la souris. Elles doivent être étendues pour prendre en compte de nouveaux dispositifs d'interaction. Si l'usage de modalités d'interaction et de combinaisons de modalités peut être décrite de manière précise avec UAN et NUAN, la forme de tableau et son absence de prise en compte des nouvelles modalités d'interaction nous incite à ne pas adopter cette approche.

D'autres notations tiennent compte de l'utilisation de modalités ou de dispositifs physiques d'interaction. Pour les notations ASUR [Dubois 2002a] ou IRVO [Chalon 2005], il s'agit de décrire les relations entres des dispositifs physiques et d'autres éléments intervenant dans l'interaction, tels qu'un composant logiciel ou un utilisateur. Une relation peut être l'utilisation d'un dispositif par un

utilisateur, sans qu'aucune précision supplémentaire ne soit indiquée. Ce type de notation permet de fournir une représentation des dispositifs disponibles, et leur utilisation par les utilisateurs, mais ne permet pas de décrire l'interaction multimodale concrète.

La notation CIS [Appert 2007] propose de décrire l'interaction concrète sous la forme de chemin d'actions concrètes sur des dispositifs tels qu'un clic souris. Cette notation est néanmoins dédiée aux interfaces WIMP et les modalités vocales et gestuelles ne sont pas considérées. D'autre part les relations temporelles prise en compte sont uniquement la séquence, l'alternative et le parallélisme. Ainsi, par exemple, la relation de parallélisme est de trop haut niveau d'abstraction pour spécifier la tâche multimodale de déplacement d'un soldat de WCCM, qui consiste en la combinaison complémentaire d'une modalité vocale et gestuelle.

Pour conclure, les limitations des notations existantes peuvent être résumées à la difficulté de décrire COMMENT doit être utilisée une modalité, et COMMENT doit être utilisée une combinaison de modalités.

2.3.2 Notre solution

Pour répondre à notre objectif qui vise à prendre en charge dans la notation COMM la spécification de la multimodalité, en prenant en compte les limitations des notations existantes, nous adoptons une approche mixte qui s'inspire du framework Dynamo-Aid et de la notation ICARE.

Nous proposons de décrire l'usage d'une modalité ou d'une combinaison de modalités directement au sein d'un arbre de tâches COMM. Ainsi un lien fort est maintenu entre l'interaction abstraite, l'interaction concrète décrite par des tâches élémentaires, et l'interaction multimodale décrite en termes de combinaisons de modalités d'interaction. Pour cela, nous introduisons d'une part le concept de tâche modale afin de décrire l'usage d'une modalité, et d'autre part un ensemble d'opérateurs logiques et temporels issus des relations temporelles de Allen [Allen 1983] et étendus dans [Vernier 2000] afin de décrire précisément les combinaisons de modalités. Notre solution pour décrire la multimodalité s'intègre de manière cohérente dans la notation COMM et permet de décrire aussi bien l'usage d'une modalité d'interaction que l'usage d'une combinaison de modalités d'interaction pour la réalisation d'une tâche donnée.

Nous décrivons d'abord en détail le concept de tâche modale, puis les nouveaux opérateurs temporels. Ensuite, nous présentons des évolutions de la notation nécessaires pour conserver sa cohérence après l'ajout de ces concepts spécifiques à l'interaction multimodale.

2.3.2.1 Tâches modales

Nous introduisons le concept de tâche modale pour spécifier l'usage d'une modalité lors de la réalisation d'une tâche donnée. Une tâche modale décrit QUEL dispositif est utilisé et COMMENT il est utilisé.

Au sein des notations existantes, les modalités sont décrites soit par l'intermédiaire de dispositifs physiques soit en tant qu'entités, mais l'utilisation du dispositif n'est pas décrite. Pourtant si l'on considère la définition d'une modalité [Coutaz 1995] comme un couple <dispositif, langage d'interaction>, l'un des deux éléments du couple est le langage d'interaction. Pour une modalité

vocale, le couple <microphone, langage pseudo naturel> définit le microphone comme dispositif d'interaction et le langage pseudo naturel comme langage d'interaction.

Cette notation de langage d'interaction est explicite dans ICARE sous la forme d'un composant, et implicite dans ICON sous la forme de transducteurs. Néanmoins, pour une application donnée, il convient de définir les éléments du langage d'interaction. Par exemple, pour la reconnaissance vocale, il convient de spécifier l'ensemble des mots reconnus, de même pour une interaction gestuelle, l'ensemble des gestes reconnus. De plus, il est important de décrire la correspondance entre les séquences d'actions reconnues et la tâche correspondante, c'est-à-dire quels sont les éléments du langage d'interaction qui permettent de réaliser une tâche donnée. Ainsi, pour la tâche *Déplacer d'un soldat* de notre exemple, l'utilisateur doit prononcer « aller ici » conjointement à une désignation du doigt pour réaliser la tâche de déplacement. Cet élément du langage d'interaction doit être décrit. Aucune des notations existantes ne propose de solution pour décrire les éléments du langage d'interaction d'une modalité mis en jeu pour la réalisation d'une tâche.

Pour prendre en compte les deux facettes d'une modalité (dispositif, langage), une tâche modale dans COMM est définie sous la forme d'un couple <dispositif d'interaction, primitive du langage d'interaction>. Une tâche modale est utilisée dans un arbre de tâches COMM comme une tâche fille d'une tâche feuille et une tâche modale ne peut avoir de fils. Ainsi, toutes les tâches d'un arbre de tâches qualifiées d'élémentaires ou de feuilles pour les notations existantes peuvent être affinées sous forme de tâches modales avec la notation COMM.

Pour représenter ces tâches modales au sein de la notation COMM, nous nous sommes inspirés du framework dynamo-AID [Clerckx 2005] comme l'illustre la Figure 94. Celle-ci décrit la tâche élémentaire Déplacer un soldat et son raffinement par une tâche modale pour laquelle est spécifiée d'une part le dispositif utilisé, un microphone, et d'autre part la primitive d'utilisation qui consiste à dire « Aller ici ». La spécification indique ainsi QUEL dispositif est utilisé pour réaliser la tâche *déplacer le soldat*, et COMMENT il doit être utilisé. L'interaction est ainsi décrite de manière concrète en termes d'usage d'une modalité d'interaction.

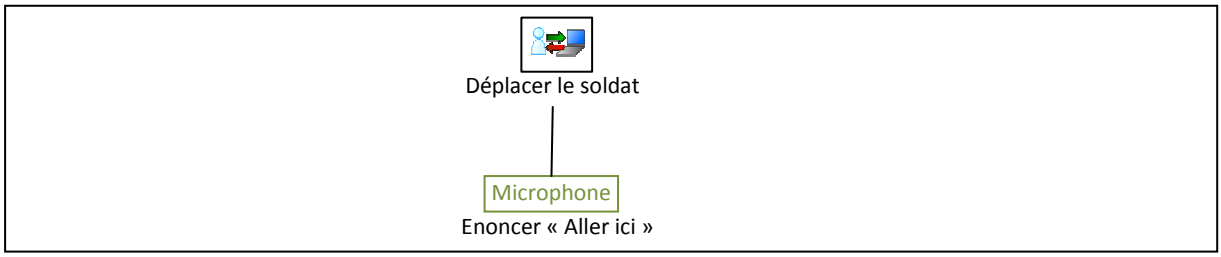


Figure 94 : Utilisation d'une tâche modale pour déplacer un soldat.

Avec une notation telle que NUAN, il est nécessaire de décrire pour chaque tâche une ligne dans un tableau, ce qui implique une discontinuité de la représentation de l'interaction (arbre + tableau). Avec la notation COMM et le concept de tâche modale, la spécification de l'interaction est uniforme puisqu'elle se fait en effectuant un raffinement supplémentaire des tâches de l'arbre de tâches. De plus, un seul affinement est nécessaire, puisque les tâches modales n'ont pas de pas de fils. De plus, le raffinement d'un niveau ne remet pas en cause la structure de l'arbre de tâches, puisqu'il s'agit uniquement de décrire comment chaque tâche est réalisée concrètement. La Figure 95 illustre l'ajout de la spécification de l'interaction concrète avec des modalités d'interaction au sein d'une

spécification d'arbre de tâches existante (Figure 87). Les trois sous-tâches de la tâche *Gérer un baraquement* : *Sélectionner le baraquement*, *Créer un soldat*, et *Confirmer* sont raffinées avec des tâches modales. Ainsi, pour réaliser la tâche *Sélectionner un baraquement*, il faut *Pointer du doigt le baraquement*, ce qui est capté par le système au moyen d'une *Caméra*. Ensuite, pour réaliser la tâche *Créer un soldat*, l'utilisateur doit *Dire « créer un soldat »*, ce qui est capté par un *Microphone*. Enfin, la tâche *Confirmer* est réalisée lorsque la tâche modale *Dire « ok »* est réalisée. La structure de l'arbre n'a pas été remise en cause par la description des modalités utilisées.

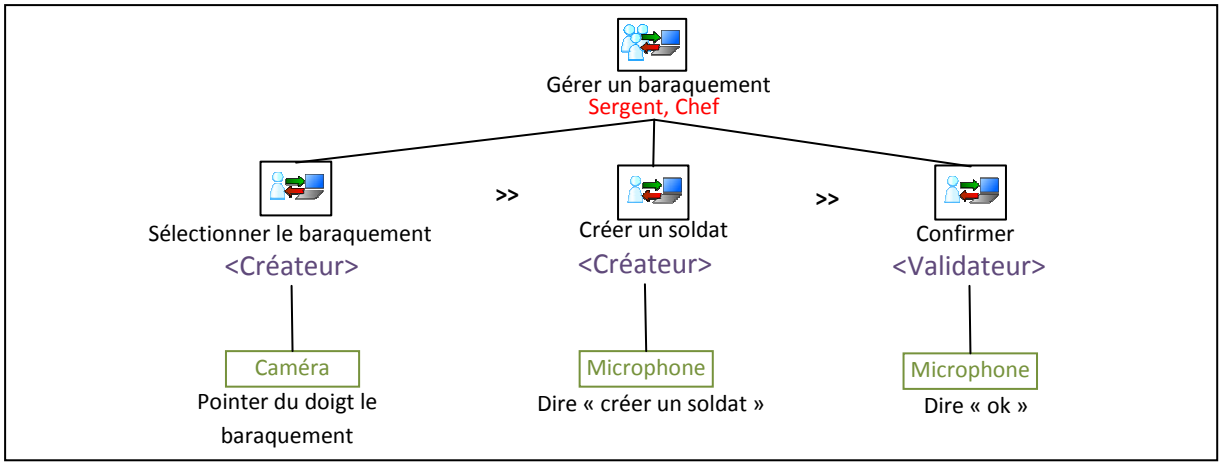


Figure 95 : Utilisation du concept de tâche modale pour la tâche de gestion d'un baraquement.

Ayant défini une tâche modale, nous étudions comment ce nouveau concept s'insère dans la notation COMM.

Comme le définit la notation COMM, une tâche doit posséder un type parmi calcul, présentation, utilisateur, interaction, groupe et interaction de groupe. Une approche immédiate serait alors de définir un nouveau type pour les tâches modales. Néanmoins, comme les tâches modales constituent des raffinements des tâches existantes, elles peuvent être caractérisées par les types de tâches actuelles. Une tâche modale est ainsi soit de type interaction (modalité en entrée), soit de type présentation (modalité en sortie). La Figure 96 illustre l'utilisation des tâches modales en reprenant l'arbre de tâches de la Figure 95. Dans cette figure, chaque tâche modale possède l'un des types de la notation COMM.

Une tâche modale peut également être de type interaction de groupe et ainsi exprimer la redondance des actions de plusieurs utilisateurs avec une même modalité. Ainsi cela permet de décrire que deux utilisateurs doivent énoncer « ok » pour confirmer une action à l'aide d'une tâche modale comme l'illustre la Figure 97 (B) de manière plus concise que l'arbre de la Figure 97 (A). Cependant, ce type de représentation empêche de spécifier des relations temporelles entre l'usage de la modalité par les utilisateurs alors qu'elles peuvent être exprimées avec l'arbre de tâches de la Figure 97 (A).

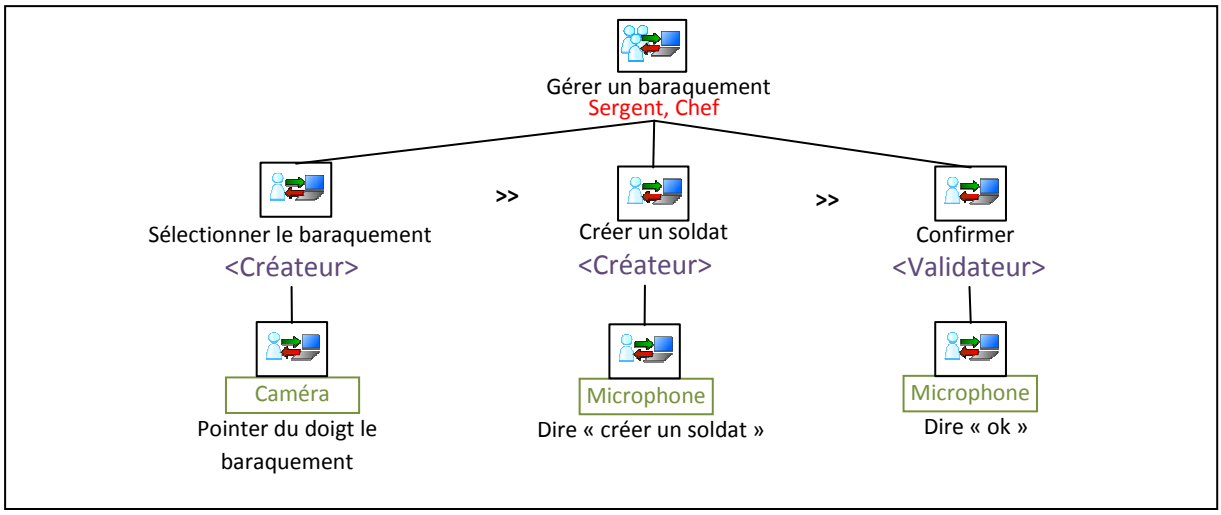


Figure 96 : Utilisation des tâches modales pour la tâche de gestion d'un baraquement.

Aussi, pour lever cette ambiguïté, nous considérons que la relation d'utilisation conjointe d'une modalité exprimée par une tâche modale de type interaction de groupe est une concurrence. De plus, nous recommandons au concepteur d'adopter la forme concise Figure 97 (C) lorsque c'est possible. Cette forme est la factorisation de celle de la Figure 97 (A), sans perte d'information car l'opérateur unaire d'itération (ici ||2) permet l'expression de la relation temporelle.

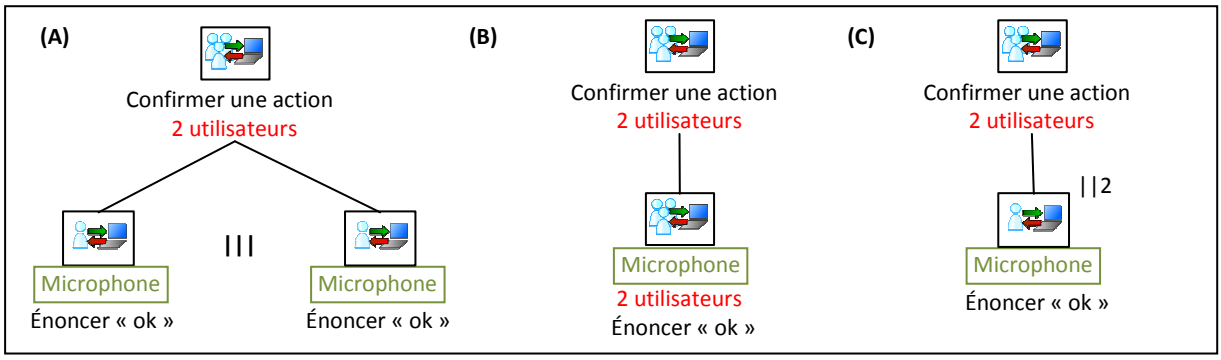


Figure 97 : Utilisation de tâche modale de type interaction de groupe pour exprimer la redondance.

Les interactions concrètes modales en entrée et en sortie peuvent être exprimées en termes de tâches modales. La Figure 98 illustre une spécification plus détaillée de la tâche interactive *sélectionner un soldat* qui est décomposée en deux sous-tâches modales : La première consiste pour l'utilisateur à *pointer du doigt le soldat*, et la deuxième consiste pour le système à *jouer le son « oui, mon seigneur »*. Notons que dans un souci de cohérence entre le concept de tâches modales et les icônes de type de tâches, nous utilisons le même code couleur pour définir un dispositif en entrée et en sortie. Ainsi, un dispositif d'entrée est défini avec un cadre vert comme la flèche verte définit une entrée utilisateur au sein de l'icone, et un dispositif de sortie est défini avec un cerclage rouge comme la flèche rouge définit une sortie du système.

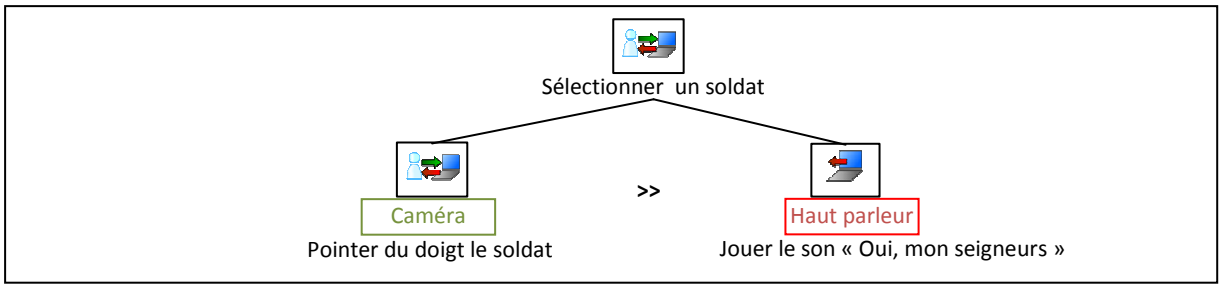


Figure 98 : Tâches modales en entrée et en sortie pour la sélection d'un soldat.

Pour résumer, nous étendons la notation COMM (Figure 99) en ajoutant le concept de tâche modale : Toute tâche feuille de l'arbre dont le type est interaction, présentation et interaction de groupe peut être affinée en tâche modale.

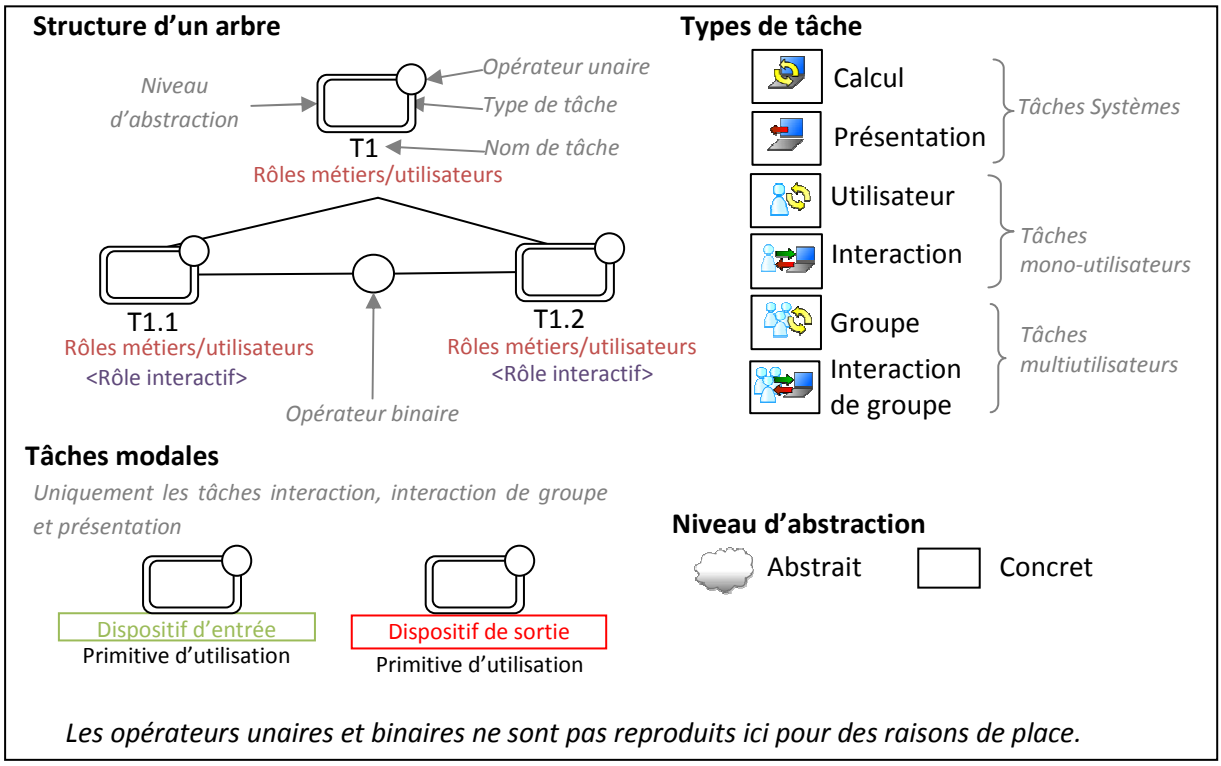


Figure 99 : Eléments de la notation COMM après l'adoption du concept de tâche modale.

2.3.2.2 Opérateur de Allen

Une tâche modale peut impliquer une combinaison de plusieurs modalités. Nous introduisons des opérateurs temporels reposant sur les relations temporelles de Allen [Allen 1983] et leurs raffinements [Vernier 2000] dans le but de permettre la spécification de la composition de modalités.

Une application est multimodale si elle dispose de plusieurs modalités d'interaction en entrée et/ou en sortie. Les notations existantes décrivent l'association des modalités aux tâches d'une application sur la base des propriétés CARE [Coutaz 1995]. Ainsi, la propriété d'Assignation (A de CARE) permet d'exprimer qu'une modalité est associée de manière unique à une tâche donnée (comme à la Figure 96), tandis que pour les trois autres propriétés, une combinaison de modalités est associée à une

tâche. Dans le cas d'une combinaison Complémentaire (C de CARE), deux modalités fournissant des informations distinctes sont nécessaires pour réaliser une tâche. Dans le cas d'une équivalence, une seule modalité est nécessaire pour réaliser une tâche parmi un ensemble fini de modalités. Enfin, dans le cas d'une Redondance (R de CARE), l'emploi de plusieurs modalités fournissant la même information est nécessaire pour réaliser une tâche. Ces propriétés sont utiles pour décrire le type de combinaisons de modalités associées à une tâche mais elles ne définissent pas comment s'articule concrètement l'usage des modalités. Ainsi pour une tâche donnée, un assemblage complémentaire d'une modalité vocale et d'une modalité gestuelle peuvent prendre plusieurs formes. En effet, l'utilisateur peut d'abord utiliser la modalité vocale puis la modalité gestuelle, ou, inversement, utiliser d'abord la modalité gestuelle, puis la modalité vocale. Par ailleurs, l'utilisateur peut également utiliser la modalité vocale pendant qu'il utilise la modalité gestuelle. Ces différentes possibilités étudiées dans [Oviat 2003] correspondent toutes à un usage complémentaire des modalités. Or, le concepteur doit spécifier comment s'organise l'usage des modalités avec le plus de précision possible puisque c'est sur la base de ces spécifications que le système est implémenté. La propriété de complémentarité seule est ainsi insuffisante pour décrire finement les relations d'usage entre plusieurs modalités. La question des relations temporelles entre modalités se pose de la même manière pour les propriétés CARE de redondance et d'équivalence. Il est ainsi nécessaire d'identifier un moyen de spécifier précisément les combinaisons de modalités que ce soit en équivalence, en redondance, ou en complémentarité.

Les opérateurs binaires de la notation CTT que nous avons repris dans la notation COMM permettent d'exprimer certaines combinaisons de modalités entre des tâches modales. La Figure 100 illustre plusieurs combinaisons de modalités exprimées avec les opérateurs CTT. Les arbres de tâches a) et b) expriment tous les deux la complémentarité de deux modalités pour la tâche *T*. En a) il s'agit d'utiliser d'abord la modalité 1, puis la modalité 2, tandis qu'en b), les deux modalités peuvent être utilisées de manière concurrente. Les arbres de tâches c) et d) expriment deux cas de redondance en utilisant les mêmes opérateurs CTT. Enfin, l'opérateur d'alternative est utilisé en e) pour exprimer l'équivalence de deux modalités pour la tâche *T*. Néanmoins des combinaisons de modalité ne peuvent être exprimées avec les opérateurs CTT, comme l'usage d'une modalité pendant l'usage d'une autre. Cet exemple d'usage imbriqué de modalité correspond au cas de la tâche *Déplacer un soldat* dans notre exemple.

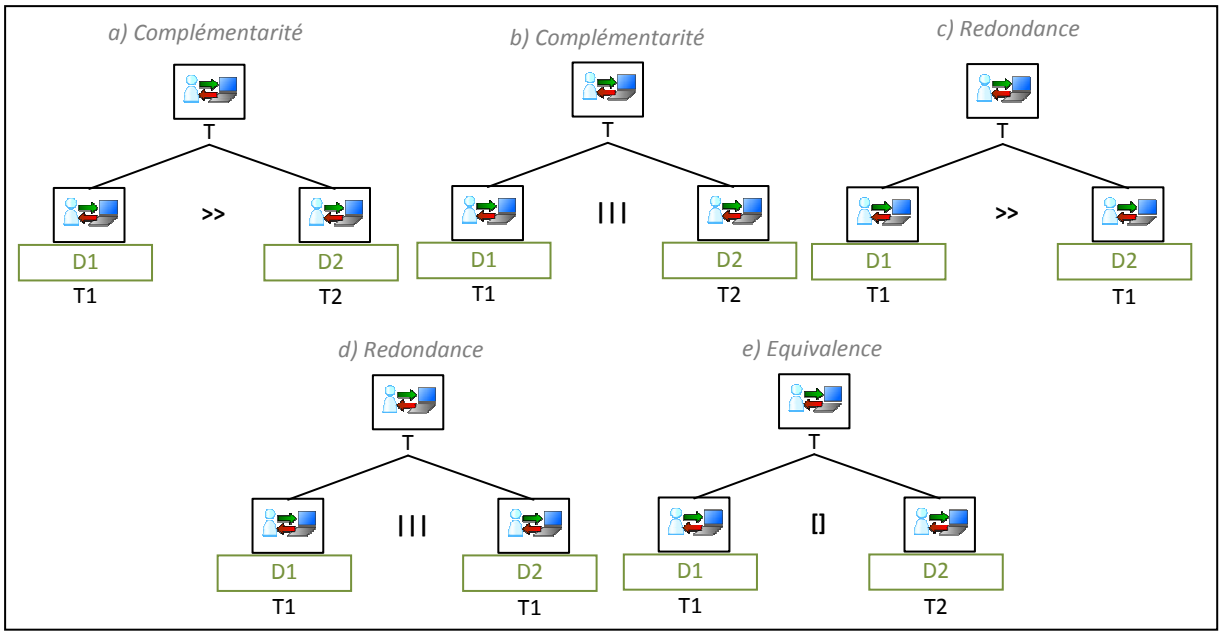


Figure 100 : Utilisation des opérateurs CTT pour exprimer l'usage complémentaire, redondant ou équivalent de modalité.

Afin d'améliorer la précision des spécifications des combinaisons de modalités, nous proposons d'ajouter à la notation un ensemble d'opérateurs issus des relations temporelles de Allen [Allen 1983] comme définit par [Vernier 2000] pour l'interaction multimodale. Allen définit 5 relations temporelles entre des intervalles de temps qui sont illustrée à la Figure 101 : l'anachronisme, la coïncidence, la concomitance, la séquence, et le parallélisme. Le nom de ces relations est défini selon [Vernier 2000]. Chaque relation est illustrée par une représentation graphique des intervalles considérés selon une ligne de temps non représenté qui s'écoule de gauche à droite. Ainsi, par exemple pour l'anachronisme, les deux intervalles sont-ils disposés l'un après l'autre.

Figure 101 : Les 5 relations temporelles de Allen.

D'après Allen, en considérant la décomposition de la relation coïncidence, et les inverses de toutes les relations, il est possible d'obtenir les 13 relations temporelles utilisées comme opérateurs binaires. La Figure 102 illustre les 13 opérateurs correspondant aux 13 relations temporelles. Les intervalles *A* et *B* introduits dans le nom des opérateurs correspondent au temps nécessaire pour réaliser les sous-tâches *A* et *B*. Etant très précis, ces opérateurs laissent peu de place à l'ambiguïté quant à l'ordre d'utilisation des modalités.

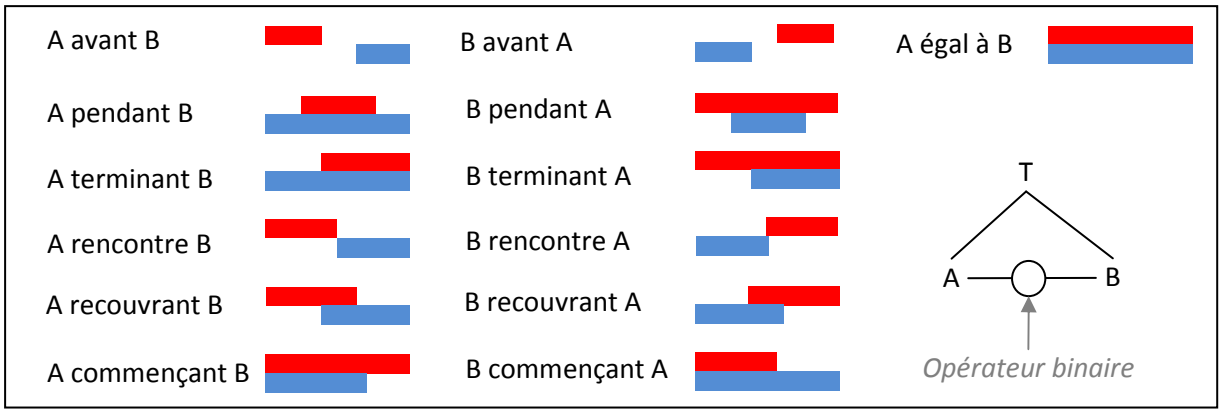


Figure 102 : Les 13 opérateurs issus des relations temporelles de Allen.

La Figure 103 illustre l'utilisation des tâches modales, des opérateurs CTT et des opérateurs de Allen pour la tâche *Déplacer un groupe* de notre exemple. La tâche *Déplacer un groupe* est décomposée en deux sous-tâches : *Sélectionner un groupe*, puis *Déplacer le groupe*. La sélection d'un groupe peut être réalisée soit en pointant du doigt un groupe, soit en énonçant le nom du groupe. Pour représenter cette équivalence, la tâche *Sélectionner un groupe* est décomposée en deux sous-tâches modales équivalentes : *Pointer du doigt un groupe* et *Enoncer « le nom d'un groupe »*, utilisant respectivement un dispositif *Caméra* et *Microphone*. L'équivalence est ici représentée par un opérateur d'alternative. Ensuite, la tâche *Déplacer le groupe* consiste à énoncer « aller ici » pendant que l'on désigne du doigt la destination. Cette interaction multimodale est représentée par deux tâches modales : *Pointer du doigt la destination* et *Enoncer « aller ici »*. Ces deux tâches modales sont liées par un opérateur *B pendant A*, ce qui signifie que la tâche à gauche de l'opérateur, la tâche *Enoncer « aller ici »,* doit être réalisée pendant que la tâche à droite de l'opérateur est en train d'être réalisée, c'est-à-dire pendant la désignation du doigt.

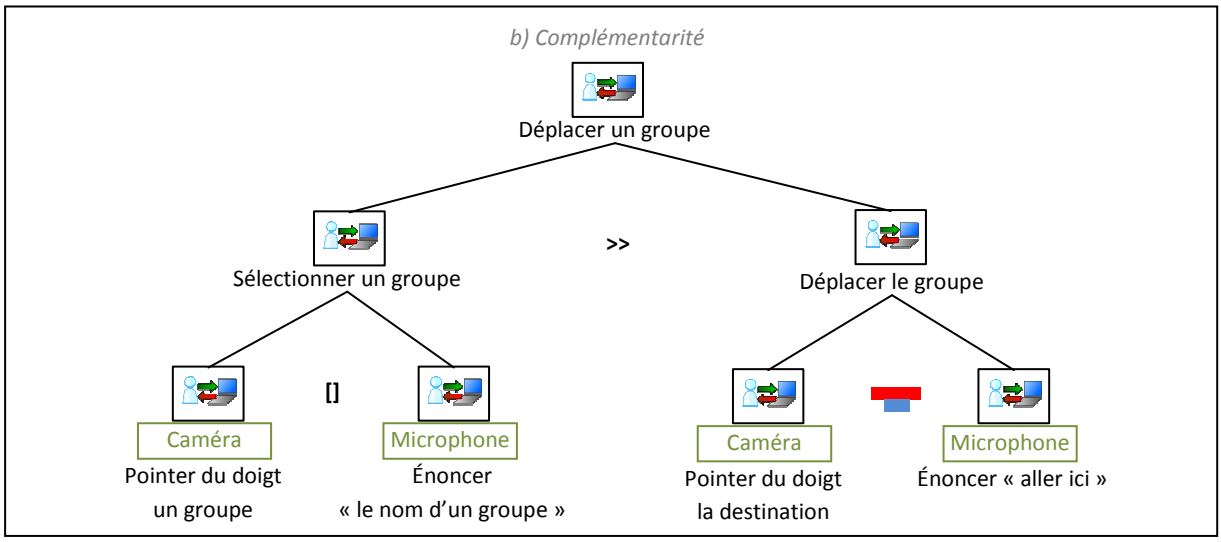


Figure 103 : Spécifier l'interaction multimodale avec un opérateur de Allen.

Les opérateurs de Allen décrivent des relations temporelles plus précises que les opérateurs binaires CTT et affinent la séquence et le parallélisme. Nous les recommandons pour spécifier l'interaction multimodale comme dans [Vernier 2000]. Cependant, ils peuvent être utilisés à plus haut niveau d'abstraction pour décrire des relations temporelles entre tâches non-modales. Mais la précision

qu'impliquent ces opérateurs nous semble peu pertinente entre tâches non-modales (par exemple la distinction entre l'anachronisme et la séquence à peu d'importance au niveau de la tâche abstraite).

Pour résumé, nous étendons les opérateurs binaires de la notation COMM en ajoutant les 13 opérateurs issus des relations temporelles de Allen (Figure 104)

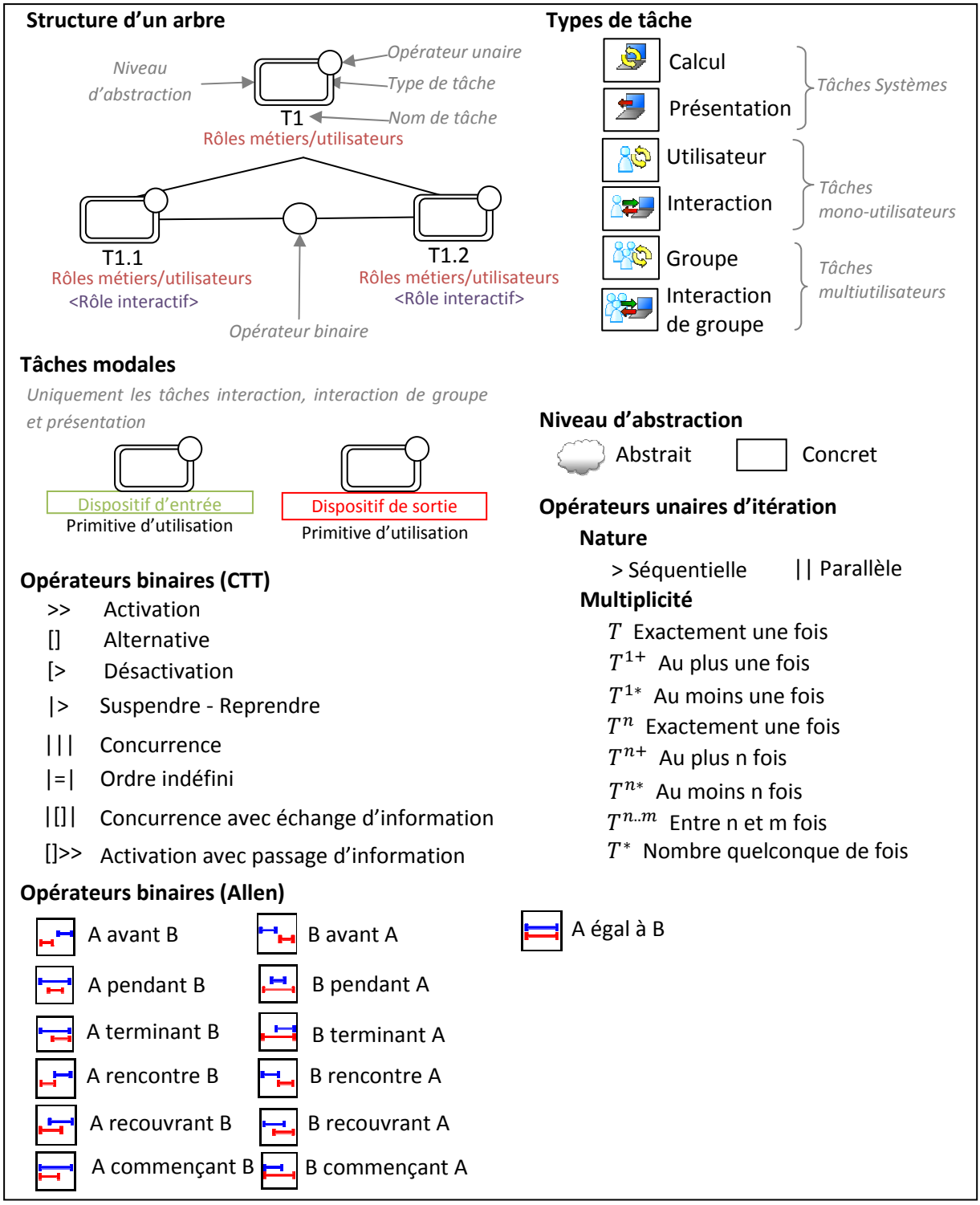


Figure 104 : Eléments de la notation COMM après adoption des opérateurs issus des relations temporelles de Allen.

2.3.3 Mise en cohérence de la notation

L'introduction des tâches modales dans la notation induit la possibilité de raffiner une tâche d'interaction ou d'interaction de groupe en sous-tâches modales. Une tâche d'interaction peut être décomposée en tâches d'interaction, mais également en tâches de présentation. Ces dernières permettent de décrire par exemple la réaction du système à l'action de l'utilisateur. Or, lorsque l'on décompose une tâche d'interaction en deux sous-tâches modales, comme c'est le cas au sein de l'arbre de la Figure 97, nous devons utiliser pour ces sous-tâches, un des types disponibles dans la notation : interaction pour l'action de l'utilisateur et présentation pour la réaction du système. Ainsi, avec la notation COMM, le concepteur peut être précis et décrire à la fois l'action et la réaction du système pour des cas particuliers. Dans le cas le plus général, une tâche d'interaction sera affinée par une seule tâche modale qui décrit l'action de l'utilisateur. La réaction du système n'étant pas décrite explicitement. Cependant, une ambiguïté subsiste. En effet, une tâche de type interaction peut être utilisée aussi bien pour décrire une interaction avec le système (ce qui inclut des réactions implicites) et pour décrire une action de l'utilisateur (sans réaction du système).

Pour lever cette ambiguïté, nous proposons la création d'un nouveau type de tâche : les tâches d'action utilisateur. Celles-ci correspondent à la réalisation d'une action de la part de l'utilisateur sans réaction attendue du système que ce soient des retours d'information-réaction ou des réponses système [Beaudoin 2004]. La Figure 105 illustre la description de la tâche *Sélectionner un soldat* de notre exemple. Celle-ci est affinée en (A) par une tâche modale de type interaction laissant implicite la réaction du système. En (B), elle est décomposée en deux sous-tâches modales. La première est une tâche d'action *Pointer du doigt le soldat*, et la deuxième est un retour d'action décrit de manière explicite par une tâche de présentation : *Jouer le son « Oui, mon seigneur »*.

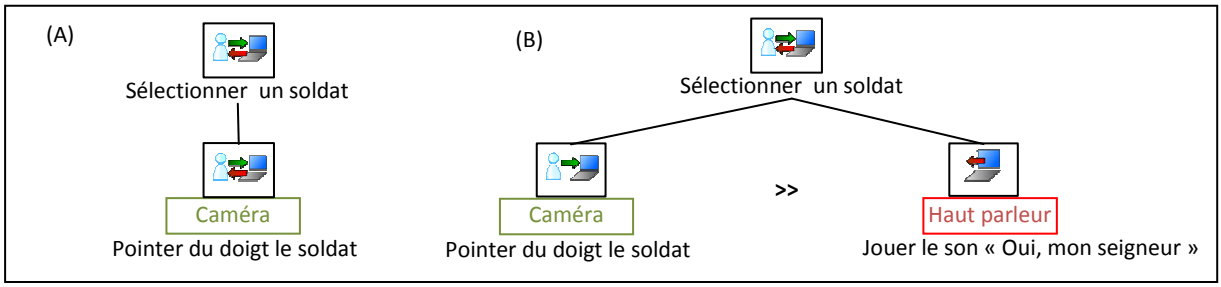


Figure 105 : Utilisation d'une tâche d'action pour spécifier la sélection d'un soldat.

De manière symétrique, nous proposons de décomposer une tâche de type interaction de groupe en plusieurs sous-tâches afin de décrire d'une part les actions des utilisateurs sans réaction du système, et d'autre part les réactions du système. Nous introduisons donc un nouveau type de tâche : les tâches d'actions de groupe. Le type de tâche de présentation suffit à décrire les réactions du système.

La notation COMM dispose ainsi de 8 types de tâches : calcul, présentation, utilisateur, action, interaction, groupe, action de groupe et interaction de groupe. La convention de typage des tâches internes de l'arbre de tâches COMM doit être modifiée pour prendre en compte les deux nouveaux types de tâches. Le Tableau 15 décrit cette nouvelle convention.

	Calcul	Utilisateur	Présentation	Action	Interaction	Groupe	Action de groupe	Interaction de groupe
Calcul (C)	C	U	I	I	I	IG	IG	IG
Utilisateur (U)		U	I	I	I	IG	IG	IG
Présentation(P)			P	I	I	IG	IG	IG
Action (A)				A	I	IG	IG	IG
Interaction (I)					I	IG	IG	IG
Groupe (G)						G	IG	IG
Action de Groupe (AG)							AG	IG
Interaction de Groupe (IG)								IG

Tableau 15 : Nouvelle convention des types de tâche mère d'une décomposition comprenant deux sous-tâches.

Pour résumer, nous adoptons les modifications suivantes à notre notation (Figure 106) afin d'introduire les concepts de tâche d'action et de tâche d'action de groupe :

- Ajout d'un nouveau type de tâche : la tâche d'action qui représente une tâche d'action de l'utilisateur sans réaction du système.
- Ajout d'un nouveau type de tâche : la tâche d'action de groupe qui représente une tâche d'action de groupe sans réaction du système.

De plus, la convention de typage des tâches définie par le Tableau 14 est remplacée par le Tableau 15 pour tenir compte de ces nouveaux types de tâche.

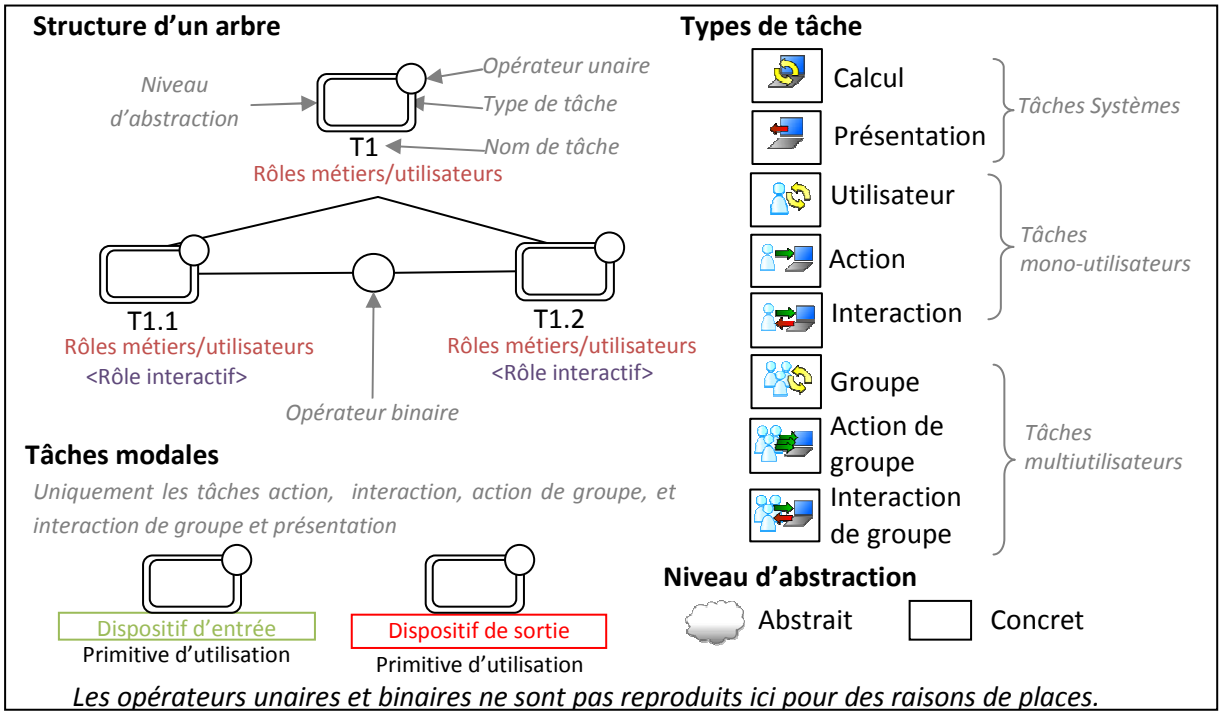


Figure 106 : Eléments de la notation COMM après l'adoption des tâches d'action et d'action de groupe.

2.3.4 Synthèse

Dans cette section, nous avons introduit les concepts de tâches modales et l'utilisation d'opérateurs reposant sur les relations temporelles de Allen pour décrire l'interaction multimodale. La notation COMM à ce stade prend donc en charge la spécification de l'interaction multimodale et multiutilisateur au sein d'une représentation unique d'arbre de tâches. Les arbres de tâches COMM permettent ainsi de décrire l'interaction abstraite et concrète, qu'elle soit individuelle ou multiutilisateur.

2.4 Au delà de l'arbre de tâches : autres éléments de spécification

Nous présentons des éléments complémentaires pour la notation d'arbre de tâches COMM.

Ces éléments concernent la factorisation des arbres de tâches par un mécanisme de template, la spécification de contextes d'utilisation, et l'intégration de concepts courant parmi les autres notations d'arbre de tâche : les pré-conditions et les effets.

2.4.1 Définition et utilisation de *Template*

De par la forme arborescente, les spécifications réalisées avec la notation COMM atteignent rapidement des dimensions importantes. Pour le processus de modélisation, il s'agit d'un problème moindre. En revanche, il peut nuire à la lisibilité de la spécification produite et engendrer une difficulté à obtenir une vision d'ensemble. De plus, un arbre de tâches COMM, et notamment sa partie concrète multimodale, peut présenter de nombreuses répétitions, par exemple parce que plusieurs interactions reviennent à effectuer un glisser-déposer par exemple. Les solutions que nous envisageons pour réduire la taille des spécifications sont d'une part de décomposer la spécification en plusieurs arbres, et d'autre part d'adapter à une notation d'arbre de tâches un mécanisme de factorisation tel qu'il en existe dans les notations telles qu'UML [UML] ou DiaMODL [Treatteberg 2002].

Une solution au problème de la taille des spécifications consiste à diviser l'arbre de tâches en portion lors des raffinements comme l'illustre la Figure 107. La racine *T* de l'arbre est défini en a). Elle est décomposée en deux sous-tâches *T1* et *T2*. La décomposition des tâches *T1* et *T2* en sous-tâches est réalisée respectivement en b) et c) plutôt que directement dans l'arbre a). Ce type de décomposition des spécifications peut être utile pour disposer de plusieurs spécifications de petite taille plutôt qu'une grande spécification. Néanmoins, la taille totale des spécifications est augmentée légèrement puisqu'à chaque division, une tâche est répétée. Par ailleurs, ce type de solution peut briser le lien entre l'interaction abstraite et l'interaction concrète. Une telle division doit donc être effectuée avec habileté en faisant en sorte, par exemple, que chaque portion d'arbre de tâches correspond à un ensemble cohérent de fonctionnalités de l'application.

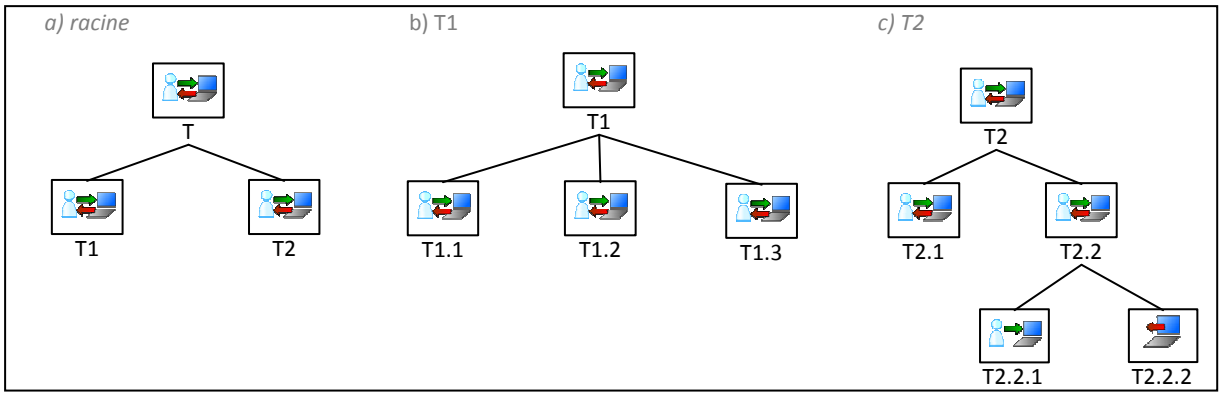


Figure 107 : Division horizontale d'un arbre de tâches.

Les actions élémentaires ou combinaisons d'action élémentaire pour réaliser une tâche (décrit avec la notation COMM par des tâches modales) sont souvent semblables. En effet, la sélection d'un item dans un menu, est semblable à la sélection d'un bouton dans une palette, ou à la sélection d'un objet dans un éditeur de dessin. De plus, le raffinement de ces trois tâches par des tâches modales conduit souvent (dans le cadre d'une interface WIMP classique) a être décrit par un clic souris dans une zone rectangulaire. Dans le but de limiter ce problème de répétition des tâches concrètes, ainsi que pour permettre une réduction de la taille des arbres de tâches, nous proposons d'adapter un mécanisme de factorisation pour notre notation.

La question de la factorisation des spécifications d'arbre de tâches est abordée dans [Thevenin 2001] dans le cadre d'applications exécutables dans plusieurs contextes d'utilisation. L'auteur propose de factoriser les spécifications à haut niveau d'abstraction, et de les différencier à bas niveau d'abstraction en fonction des contextes d'exécution. La partie supérieure d'un arbre de tâches est ainsi réutilisée pour l'ensemble des contextes d'utilisation, tandis que seule la partie inférieure dupliquée lorsqu'une différence doit être décrite entre deux contextes d'utilisation. Ce type d'approche est utilisable avec la notation COMM notamment car elle définit l'association de contextes d'utilisation aux tâches. Cependant ce type de solution ne répond pas à la question de la répétition de l'interaction concrète, c'est-à-dire de la partie inférieure de l'arbre de tâches.

La notation UML [UML] propose un moyen d'exprimer une préoccupation récurrente au sein de ses spécifications par le biais des mécanismes de *Template*. Pour la représentation de diagramme de classe, il s'agit de définir une classe paramétrée (*Template*). L'utilisation d'un *Template* sur une classe doit fixer la valeur des différents paramètres. La Figure 108 illustre ainsi la définition (1) d'une classe paramétrée List dont un paramètre est *P*. Les méthodes *add* et *get* sont définies avec ce paramètre. L'utilisation pour un paramètre de type *String* est illustrée en (2). Le paramètre prend la valeur *String* ce qui est indiquer par *P->String*. Aussi, la méthode *add* et *get* vont manipulent des *String*.

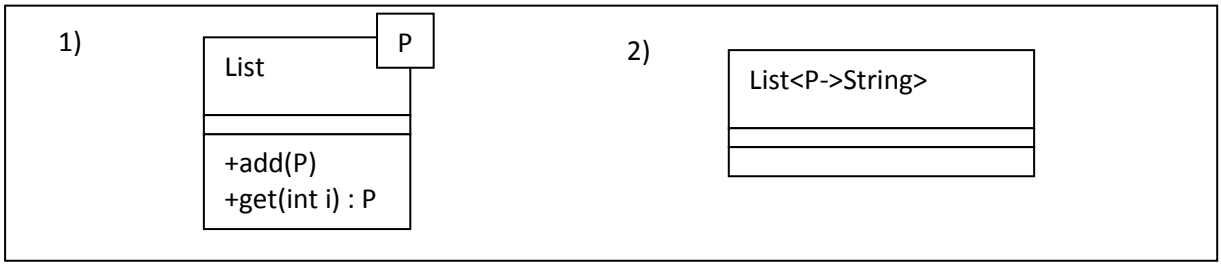


Figure 108 : Définition (1) et utilisation (2) d'un template dans un diagramme de classe UML.

La notation de modèle de dialogue DiaMODL [Tratteberg 2002] qui accompagne la notation TaskMODKL propose une approche similaire reposant sur des composants paramétrés.

Nous proposons d'adapter le mécanisme de *Template* à une notation d'arbre de tâches, et ainsi permettre au concepteur de définir d'une part des arbres de tâches paramétrés et de les réutiliser au sein d'autres arbres de tâches en rappelant leur nom, et en fixant leurs paramètres. La Figure 109 illustre la définition et l'utilisation d'un *Template* pour effectuer des glisser-déposer. La définition consiste à décrire un arbre de tâches muni de deux paramètres, ici *<Shape 1>* et *<Shape 2>*. L'utilisation de ce *Template* consiste à ajouter à une tâche un hexagone pour signifier que la concrétisation de cette tâche utilise un *Template*, et également à ajouter une ligne en dessous du nom de la tâche décrivant le nom du *Template* utilisé, et les objets transmis en paramètres. Ici, pour la tâche déplacer un soldat, il s'agit de faire glisser le *<soldat>* sur la *<carte>*. Il est entendu que les objets soldat et carte sont des « Shape » (forme) conformément à la définition du *Template*.

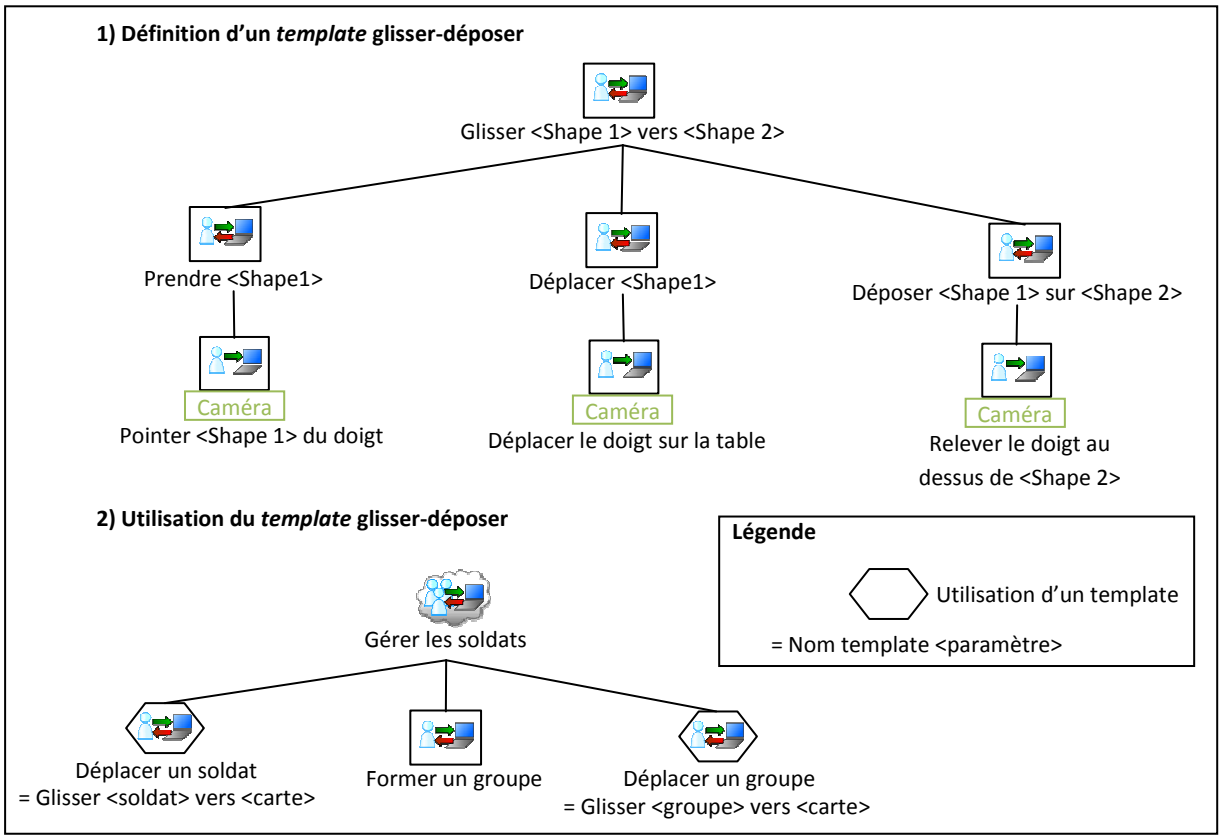


Figure 109 : Définition (1) et utilisation (2) de template sur les arbres de tâches.

Au sein de la Figure 109, nous avons défini un arbre paramétrée avec des paramètres correspondant à des concepts métier de notre exemple. Cependant, il ne faut pas limiter l'utilisation des paramètres aux concepts métier de l'application. En effet, il est possible d'utiliser un mécanisme de template avec des paramètres définis sur les noms des tâches ou sur les noms de dispositifs. Ainsi, n'importe quel élément textuel décrit comme paramètre lors de la définition d'un template peut être remplacé par une valeur lors de son utilisation dans un arbre. Le paramétrage selon un dispositif par exemple permet de remplacer facilement une modalité par une autre. Ce mécanisme apporte donc une grande liberté au concepteur pour factoriser ses spécifications, tout en étant compréhensible par un développeur habitué au principe du paramétrage.

2.4.2 Spécification de contexte d'utilisation

La spécification du contexte d'utilisation est importante notamment pour la multimodalité [Ray 2005]. En effet, offrir des choix de modalités équivalentes à l'utilisateur constitue une solution pour maintenir un bon niveau d'utilisabilité dans des contextes d'utilisations variables.

Peu de notations proposent un moyen de spécifier le contexte dans lequel se déroule l'interaction. Parmi elles, le framework Dynamo-Aid [Clerckx 2005] propose une représentation complémentaire de l'arbre de tâches qui permet de décrire des contextes d'utilisation, puis une représentation générale qui relie : les contextes et les tâches. Cette approche sous forme de deux représentations complémentaires est intéressante puisqu'elle permet de décrire les contextes de manière isolée de l'arbre. En revanche, la représentation générale est très difficile à lire de par les nombreux liens entre les concepts qui y sont représentés. De plus, le lien entre tâches et contextes n'est pas représenté dans l'arbre de tâches.

La notation ASUR [Dubois 2002a] permet pour une tâche donnée de décrire le contexte d'utilisation en termes de dispositifs, de systèmes et d'utilisateurs présents. Les relations entre ces éléments sont détaillées pour décrire la tâche. Il est ainsi possible de décrire de décrire par exemple les liens physiques entre les dispositifs et les utilisateurs, ou les échanges d'informations. Ce type de représentation se focalise sur une tâche donnée. Aussi, elle peut être utilisée pour décrire certaines tâches mais pas pour décrire l'ensemble des tâches d'un arbre.

D'autres notations telles que CTT [Mori 2002], K-MAD [Lucquiaud 2005, Baron 2006], GTA [Veer 2000], ou encore TaskMODL [Trætteberg 2002] proposent de décrire le contexte sous une forme annexe pour chaque tâche. Pour les trois premières notations, le contexte n'est pas explicitement représenté dans l'arbre ce qui ne rend pas le lien perceptible. Pour TaskMODL, il s'agit de décrire des diagrammes RML illustrés au Chapitre 2 directement au sein de l'arbre, ce qui fait exploser sa taille.

A l'instar des notations d'arbre de tâches tel que K-MAD et GTA, nous proposons de décrire de manière textuelle un ou plusieurs contextes d'utilisation. Ces contextes peuvent faire l'objet de représentations complémentaires ce qui n'est pas l'objet de nos travaux. Nous lions ensuite explicitement les contextes ainsi définis avec les tâches de l'arbre en utilisant des étiquettes. Chaque contexte défini a une étiquette numérotée : 1, 2, etc. Un ou plusieurs contextes peuvent alors être associés à une tâche en spécifiant à gauche de la tâche la ou les étiquettes de contexte. La Figure 110 illustre la définition de contextes et leur utilisation au sein d'un arbre de tâches COMM. Tout d'abord, trois contextes sont définis (1) : le premier est le contexte par défaut, et les deux autres correspondent à un environnement bruyant et à une situation de mobilité des utilisateurs. Ensuite

les contextes sont utilisés (2) dans l'arbre de tâches. La tâche *T1* est associée au contexte par défaut, tandis que la tâche *T2* est associée à un contexte mobile et bruyant.

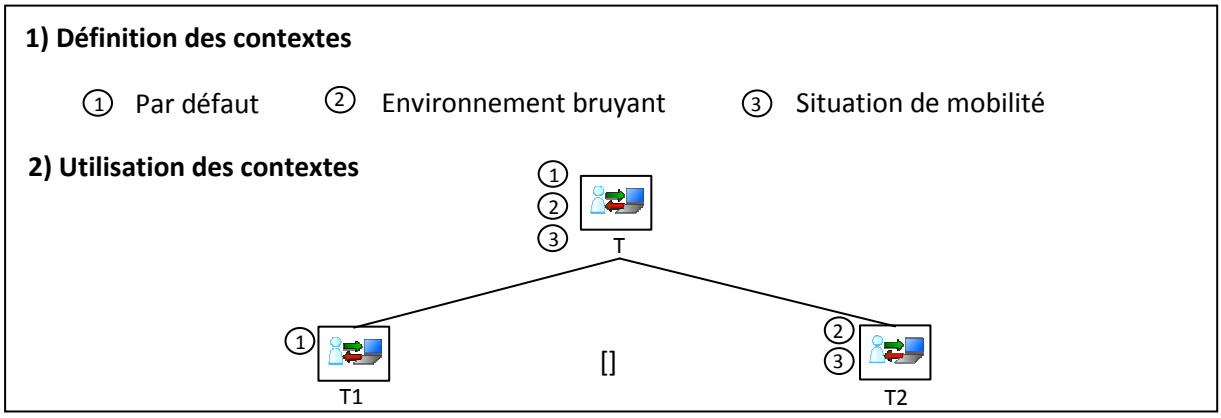


Figure 110 : Définition (1) de contextes et utilisation (2) au sein d'un arbre de tâches.

2.4.3 Spécification de pré-conditions et d'effets

Nous avons montré au le Chapitre 2 que la plupart des notations de modèle de tâche propose de décrire des pré-conditions pour pouvoir réaliser une tâche. Ces pré-conditions s'expriment sur les concepts du domaine. Pour les notations K-MAD et CTT, celles-ci sont décrites textuellement en tant que propriété de la tâche et ne sont pas explicitement représentées au sein des arbres de tâches. La notation TaskMODL adopte une autre approche en proposant de décrire explicitement la pré-condition au sein d'un arbre de tâches, sous la forme d'insertion de diagrammes RML. Ces derniers permettent de représenter des classes d'objets et des instances de classes d'objets. Comme nous l'avons montré au Chapitre 2, insérer des diagrammes au sein d'autres diagrammes augmente la taille de la spécification et la rend difficile à lire.

A l'instar de TaskMODL, nous adoptons une représentation des pré-conditions directement dans l'arbre afin de favoriser la compréhension de l'interaction. En revanche, nous souhaitons préserver la lisibilité des arbres de tâches COMM, c'est pourquoi nous proposons de décrire ces pré-conditions textuellement. De plus, afin de rester cohérent avec la description des concepts du domaine (concepts qui peuvent être décrits de manière complémentaire par des diagrammes de classes UML ou des diagrammes RML), nous proposons d'utiliser les classes et instances décrites au sein de ces diagrammes. Le texte d'une pré-condition est écrit en italique afin de le distinguer des autres étiquettes de la tâche. La Figure 111 illustre l'utilisation de pré-condition au sein d'un arbre de tâches COMM. La première pré-condition est définie sur la tâche *Gérer les soldats* : *Soldats s*. Elle indique qu'au moins une instance de la classe soldat doit exister pour pouvoir réaliser cette tâche. De manière similaire, la tâche *Déplacer un groupe* définit la condition *Groupe g* qui indique qu'un groupe doit exister pour être réalisé.

De plus comme pour les pré-conditions, plusieurs notations existantes dont la notation CTT [Mori 2002] proposent de décrire les effets ou post-condition en temps que propriétés d'une tâche. Un effet permet de décrire une réaction du système qui peut être visible à l'utilisateur et/ou de décrire explicitement une action de calcul du système. La notation COMM prévoit un type de tâche explicitement pour décrire ces deux types d'effets. Il s'agit respectivement des tâches de présentation et des tâches de calcul. Cependant, dans de nombreux cas, le concepteur peut ne pas

souhaiter décrire l'effet d'une tâche sous la forme d'une tâche pour des raisons de concision. Aussi, nous proposons de décrire également les effets d'une tâche directement au sein de l'arbre de tâches en tant que propriété d'une tâche, de manière similaire aux pré-conditions. Le texte suit un symbole "=>" pour le distinguer d'une pré-condition. La Figure 111 illustre l'utilisation d'un effet pour la tâche *Former un groupe* : *=> new Groupe().* L'effet consiste à créer un nouveau groupe.

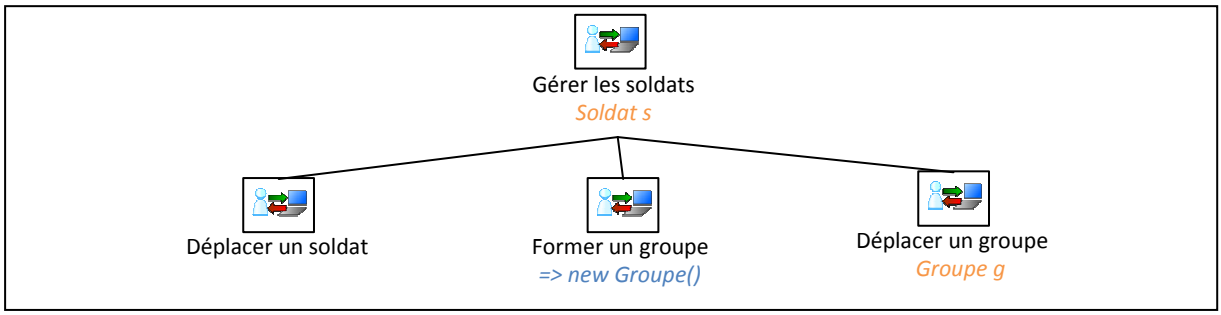


Figure 111 : Utilisation de pré-condition au sein d'un arbre de tâche COMM.

3 Synthèse de la notation COMM

Dans ce chapitre, nous avons présenté et justifié de manière incrémentale notre notation COMM pour la spécification de l'interaction multiutilisateur et multimodale. Ces deux aspects de l'interaction sont intégrés au sein d'une représentation uniforme d'arbre de tâches dont les éléments de base sont issus de CTT et K-MAD. La Figure 112 synthétise les éléments de la notation COMM. Une tâche COMM est ainsi définie par :

- Un nom.
- Un type parmi les tâches de calculs et de présentations qui correspondent aux tâches systèmes, les tâches utilisateurs, action et interaction qui correspondent aux tâches mono-utilisateurs, et les tâches de groupes, d'actions de groupe et interactions de groupe qui sont les tâches multiutilisateurs.
- Un niveau d'abstraction qui peut être : concret ou abstrait selon que la tâche corresponde ou non à une opération sur les concepts du domaine.
- Un opérateur unaire d'itération qui décrit la capacité de la tâche à être répétée. Cet opérateur est défini selon la nature séquentielle ou parallèle de la répétition, et par sa multiplicité.
- Un dispositif physique, auquel cas la tâche devient une tâche modale qui doit être une feuille de l'arbre, et dont le nom correspond à une primitive d'utilisation du dispositif.
- Les utilisateurs requis et les rôles métiers qu'ils doivent jouer pour pouvoir réaliser une tâche et ainsi décrire la division coopérative du travail.
- Un éventuel rôle interactif pour préciser l'interaction collaborative.
- Un ou plusieurs contextes d'utilisation.
- Une pré-condition pour décrire une condition à la réalisation d'une tâche.
- Un effet pour décrire de manière synthétique une action du système (sans introduire une tâche).

Une tâche COMM peut être décomposée en sous-tâches (à l'exception des tâches modales). Ces sous-tâches sont reliées par des opérateurs qui peuvent être soit les opérateurs logiques et temporels de la notation CTT, soit les opérateurs issus des relations temporelles de Allen. Nous recommandons l'usage des opérateurs CTT pour la description de l'interaction de haut niveau d'abstraction et l'usage des opérateurs de Allen pour la description des combinaisons de tâches modales. Par ailleurs, la convention de typage des tâches définie au Tableau 14 complète les éléments de la notation COMM. Elle permet d'identifier le type d'une tâche mère dont les deux tâches filles sont de types différents.

Enfin, le mécanisme de *Template* vient compléter la notation COMM et permet de réduire la taille de l'arbre de tâches en le décomposant. Ce mécanisme peut également être utilisé au sein d'autres notations d'arbre de tâches comme K-MAD [Lucquiaud 2005, Baron 2006] ou TaskMODL [Traetteberg 2000].

Les principales contributions de la notation COMM à l'état de l'art des notations de spécification de l'interaction incluent :

- L'introduction du concept de tâches multiutilisateurs (groupe, action de groupe et interaction de groupe) pour décrire les interactions multiutilisateurs coopératives et collaboratives.
- L'introduction du concept de rôle interactif pour décrire l'association utilisateur-tâche dans le cas d'interactions collaboratives.
- L'introduction du concept de tâches modales et d'opérateurs issus des relations temporelles de Allen pour décrire l'interaction multimodale.

Ces contributions ainsi que les autres concepts intégrés à la notation COMM (contexte, pré-condition, effet, rôle métier, utilisateurs, et niveau d'abstraction) peuvent être repris et intégrés au sein d'autres notations selon les besoins en termes de spécification. En particulier, nous envisageons son intégration à UML [UML].

Dans le chapitre suivant, nous présentons e-COMM, l'éditeur de spécification COMM. Nous abordons dans le Chapitre 5 les aspects de validation de la notation COMM et de son éditeur e-COMM.

Structure d'un arbre

Niveau d'abstraction
Opérateur unaire
Type de tâche
T1
Nom de tâche
Rôles métiers/utilisateurs

T1.1
Rôles métiers/utilisateurs
<Rôle interactif>

T1.2
Rôles métiers/utilisateurs
<Rôle interactif>

Opérateur binaire

Tâches modales

Uniquement les tâches action, interaction, action de groupe, et interaction de groupe et présentation

Dispositif d'entrée
Primitive d'utilisation

Dispositif de sortie
Primitive d'utilisation

Opérateurs binaire (CTT)

- >> Activation
- [] Alternative
- [> Désactivation
- |> Suspendre - Reprendre
- ||| Concurrence
- |=| Ordre indéfini
- |[]| Concurrence avec échange d'information
- []>> Activation avec passage d'information

Opérateurs binaire (Allen)

- A avant B
- B avant A
- A pendant B
- B pendant A
- A terminant B
- B terminant A
- A rencontre B
- B rencontre A
- A recouvrant B
- B recouvrant A
- A commençant B
- B commençant A
- A égal à B

Types de tâche

Tâches Systèmes
- Calcul
- Présentation

Tâches mono-utilisateurs
- Utilisateur
- Action
- Interaction

Tâches multiutilisateurs
- Groupe
- Action de groupe
- Interaction de groupe

Niveau d'abstraction

- Abstrait
- Concret

Opérateurs unaire d'itération

Nature

> Séquentielle || Parallèle

Multiplicité

- T Exactement une fois
- T^{1+} Au plus une fois
- T^{1*} Au moins une fois
- T^{n} Exactement une fois
- T^{n+} Au plus n fois
- T^{n*} Au moins n fois
- $T^{n..m}$ Entre n et m fois
- T^{*} Nombre quelconque de fois

Définition de *Template*

Décrire un arbre de tâche avec des paramètres : <paramètre>.

Utilisation de *Template*

Tache
= Nom *template* <paramètre>

Figure 112 : Eléments de la version définitive de la notation COMM.

Chapitre 4 : editeur e-COMM

Dans ce chapitre, nous présentons un outil que nous avons développé pour compléter la notation COMM : un éditeur de spécification selon la notation COMM, nommé e-COMM [Jourde 2010a] [Jourde 2010b]. L'éditeur permet la spécification d'arbres de tâches COMM, ainsi que la spécification des éléments liés aux tâches COMM que sont les rôles métiers et les contextes.

La première section de ce chapitre décrit nos objectifs pour la réalisation d'un outil d'édition d'arbres de tâches COMM et présente la démarche que nous avons adoptée pour réaliser l'éditeur e-COMM. La deuxième section présente les éditeurs qui accompagnent les notations d'arbre de tâches introduites au chapitre 2 et met en lumière leurs avantages et leurs limites. Enfin, la quatrième section présente notre éditeur e-COMM.

1 Objectif et démarche

L'intérêt pour une notation de disposer d'outils logiciels a été souligné au sein du chapitre 2 : les outils structurent le processus de conception [Veer 2000], aident le concepteur à utiliser une notation [Mori 2002] et aident à la diffusion d'une notation [Molina 2009]. Sur la base de ces trois arguments, il nous apparaît comme indispensable de doter la notation COMM d'un outil logiciel.

1.1 Objectif

Les outils logiciels accompagnant les notations existantes sont des éditeurs de spécifications, des outils d'aide à l'analyse des spécifications produites et des générateurs de code. Les deux derniers types d'outils complètent le premier type d'outils, les éditeurs de spécifications. En effet, un éditeur permet de réaliser des spécifications qui peuvent ensuite être analysées et servir de référence pour générer du code. Une première étape pour outiller la notation COMM est donc de réaliser un éditeur de spécifications.

Notre ambition est de réaliser un éditeur qui couvre l'ensemble des concepts de la notation COMM. Il s'agit de permettre la spécification d'arbres de tâches, mais également la spécification de contextes et de rôles métier pouvant être liés aux tâches des arbres.

Nous souhaitons que la diffusion de l'éditeur soit la plus large possible. Aussi, l'éditeur d'arbres de tâches et la description de la notation COMM doivent être accessibles publiquement sur internet. Par ailleurs, le déploiement de l'éditeur chez les utilisateurs doit être le plus simple possible.

Enfin, nous souhaitons produire un éditeur d'arbres de tâches qui nécessite un faible temps d'apprentissage, et doté de fonctionnalités aisées à comprendre et à utiliser. C'est pourquoi, nous visons à développer un éditeur dont les mécanismes d'interaction avec l'utilisateur reposent autant que possible sur la manipulation directe [Schneiderman 1997].

Pour résumer, notre objectif est de développer un éditeur pour la notation COMM qui permette la spécification d'arbres de tâches COMM, de contextes et de rôles métier. Cet éditeur doit être

publiquement accessible par internet et facile à déployer pour l'utilisateur afin de favoriser sa diffusion. Enfin, il doit être muni d'une interface homme machine simple à comprendre et à utiliser.

1.2 Démarche

Dans une première étape, nous étudions les éditeurs d'arbres de tâches dont disposent les notations présentées au Chapitre 2. Il est naturel qu'aucun éditeur ne couvre la notation COMM. En revanche, ces éditeurs peuvent nous inspirer pour la conception de notre éditeur. Nous analysons leurs qualités et leurs limites vis-à-vis de nos objectifs : facilité de diffusion/déploiement et facilité d'utilisation.

Nous présentons ensuite l'éditeur e-COMM, tel que nous l'avons conçu et développé. Pour cela, nous détaillons l'architecture logicielle d'e-COMM et les technologies que nous avons utilisées en les justifiant en regard de nos objectifs et des conclusions de notre analyse des éditeurs existants. E-COMM est ainsi une combinaison d'une application web dynamique développée en Java, Javascript, HTML et CSS et d'un client internet riche développé en C#/Silverlight. Puis nous présentons l'interface homme machine conçue pour e-COMM ainsi que les avantages qu'apporte cette interface par rapport à celles des éditeurs de tâches existants.

2 Editeurs d'arbres de tâches

Pour chaque éditeur, nous considérons leur capacité à répondre à nos objectifs : facilité de diffusion et facilité d'utilisation. De plus, si l'information est disponible, nous indiquons les technologies utilisées pour réaliser ces éditeurs.

2.1 CTTE

L'outil logiciel CTTE (*Concur Task tree Environnement*) [Mori 2002] permet de spécifier des arbres de tâches selon la notation CTT [Parteno 1997]. CTTE est publiquement téléchargeable sous forme d'archive sur le site http://giove.isti.cnr.it/tools/ctte/ qui est consacré à la notation CTT. CTTE est développé en Java pour permettre sa portabilité. De ce fait, l'installation de CTT est réalisée en décompressant les fichiers de l'application présents dans l'archive téléchargeable, parmi lesquels se trouve l'exécutable qui permet de lancer CTTE. Par ailleurs, CTTE requiert l'installation préalable d'une machine virtuelle Java.

L'interface de CTTE est décomposée en plusieurs grandes parties (Figure 113) : Une surface d'édition dans laquelle est représenté l'arbre de tâche en cours d'édition, un ensemble de barre d'outils, une mini-vue, et une barre de propriétés de la tâche sélectionnée.

L'utilisation de CTTE n'est pas aisée à comprendre et peu pratique, puisque les tâches principales (ajout de tâche et définition des opérateurs) repose sur une logique de sélection. En effet, pour ajouter une sous-tâche, il faut : sélectionner une tâche de l'arbre, puis cliquer sur un bouton du type de tâche présent dans la barre latérale. La tâche est alors ajoutée comme sous-tâche de la tâche sélectionnée. De même, pour ajouter un opérateur entre deux sous-tâches, il faut sélectionner la tâche située le plus à gauche, puis cliquer sur un bouton d'opérateur de la barre latérale.

De plus, même si la notation CTT prévoit que deux sous-tâches sont toujours reliées par un opérateur, l'éditeur CTTE ne le prend pas en considération. En effet, les liens entre deux sous-tâches tâches ne sont pas automatiquement tracés et sont ajoutés par l'utilisateur en même temps que les opérateurs.

Enfin, la version actuelle de l'éditeur comporte plusieurs bugs majeurs impliquant des fermetures intempestives de l'application (sans sauvegarde de l'arbre en cours d'édition) et des problèmes de placements des tâches. Certaines tâches sont déplacées automatiquement lorsque l'utilisateur les sélectionne. Si ces problèmes techniques peuvent être résolus, en l'état, ils constituent un frein à l'utilisation de CTTE, et donc à la large diffusion de CTT, comme son utilisation en enseignement de l'IHM.

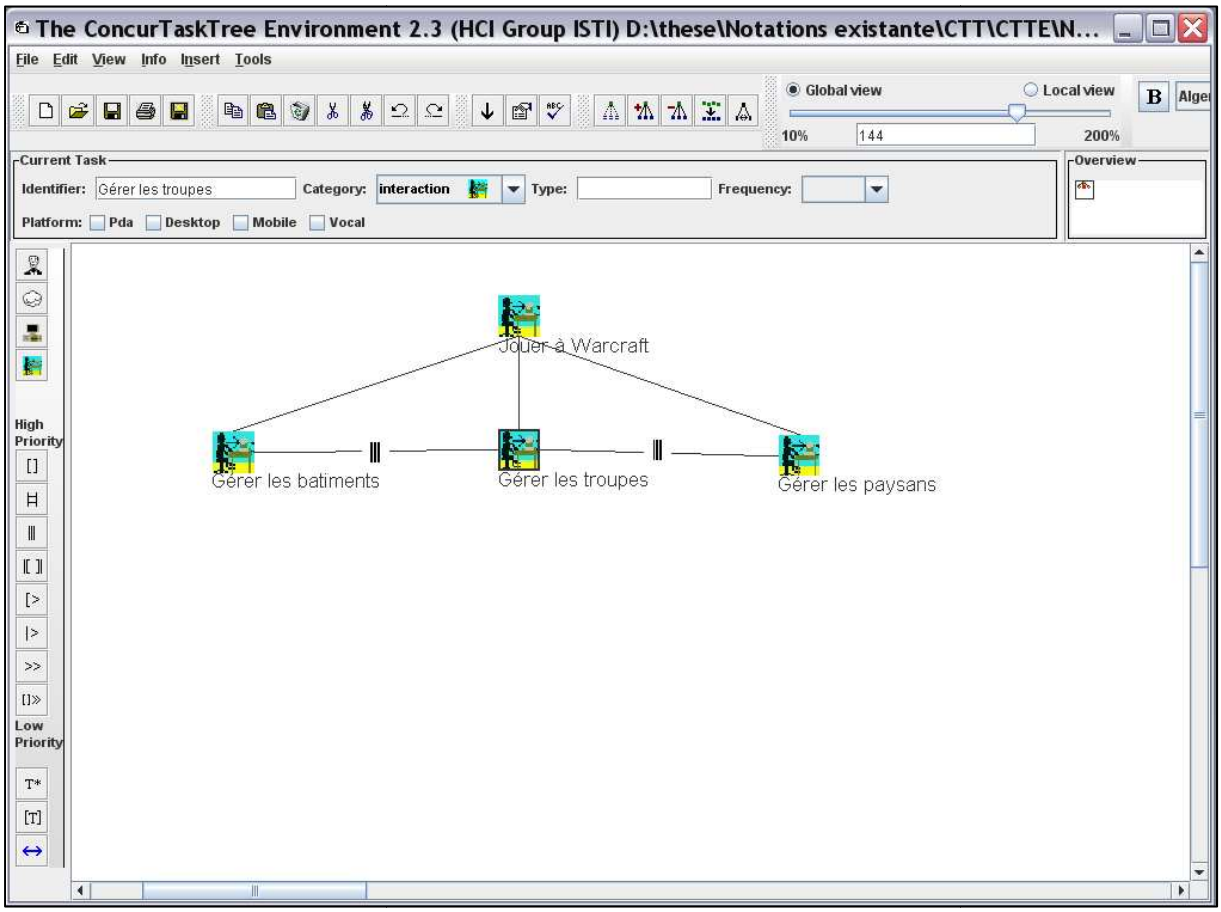


Figure 113 : Interface de CTTE (http://giove.isti.cnr.it/tools/ctte/).

2.2 K-MADe

L'éditeur K-MADe (*Kernel of Model for Activity Description environment*) [Baron 2006] développé en Java, permet la modélisation selon la notation K-MAD, basée sur la notation N-MDA [Lucquiaud 2005]. Cet éditeur permet la description d'arbres de tâches K-MAD, mais également la spécification d'objets abstraits et concrets, des utilisateurs et de leurs rôles et d'événements sous la forme de tableaux.

K-MADe est téléchargeable sous forme d'archive à l'adresse suivante : http://kmade.sourceforge.net/. L'éditeur s'installe par décompression des fichiers de l'archive. Comme pour CTTE, K-MADe nécessite l'installation préalable d'une machine virtuelle Java.

L'interface de l'éditeur (Figure 114) est composée d'une zone d'édition de l'arbre, d'une barre d'outils pour l'édition de l'arbre, d'une zone de caractéristiques de la tâche sélectionnée, et d'une mini-vue sur l'arbre. Le style d'interaction proposé est proche d'un éditeur de dessin vectoriel. En effet, pour ajouter une tâche, l'utilisateur doit d'abord sélectionner un type de tâche puis sélectionner un emplacement sur la surface d'édition.

Notons que lorsqu'une tâche est ajoutée, elle n'est reliée à aucune autre tâche de l'arbre. L'utilisateur doit manuellement décrire les liens entre les tâches. Pour ce faire, elle/il trace une ligne entre les ports de composition de tâches (représentés par des petits rectangles bleus). Cette opérations bien que simple peut se révéler fastidieuse lorsqu'elle doit être répétée de nombreuses fois pour spécifier de grands arbres de tâches.

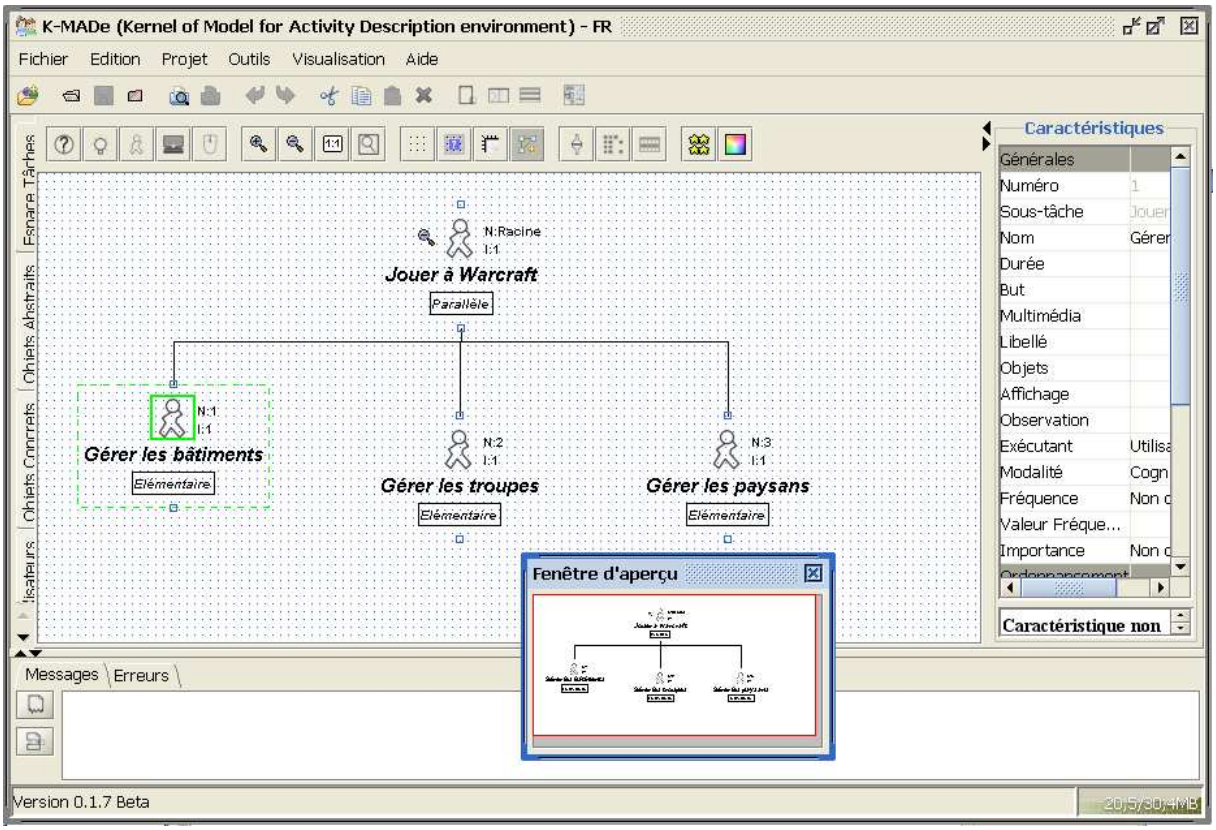


Figure 114 : Interface de K-MADe (http://kmade.sourceforge.net/).

La version de l'éditeur K-MAD, présentée à la Figure 114, est celle que nous avons utilisée comme référence dans la conduite de nos travaux, en particulier lors de la conception de l'éditeur e-COMM. Depuis, une nouvelle version de l'éditeur K-MAD est disponible, celle-ci comprend notamment l'usage d'un menu circulaire.

2.3 TOUCHE Case

TOUCHE Case [Penichet 2010] est l'outil reposant sur la méthode de conception TOUCHE [Penichet 2006]. A l'instar de CTTE et K-MADe, TOUCHE Case est développé en Java et peut être téléchargé sur le site http://www.penichet.net/.

L'interface relative à l'édition de l'arbre de tâche est illustrée à la Figure 115. Son utilisation est proche de celle introduite par CTTE. En effet, l'ajout d'une tâche s'effectue en sélectionnant une tâche puis en cliquant sur les boutons d'ajout de tâches. Le nombre de possibilités d'ajout (et donc de boutons dans la barre) est plus important que dans CTTE, car on peut ajouter soit des tâches sœurs (à gauche ou à droite), soit des sous-tâches.

Contrairement à CTTE, le lien entre les tâches sœurs est créé lors de l'ajout de tâches. En revanche, le lien n'est pas associé à un opérateur. L'ajout d'un opérateur s'effectue en sélectionnant un lien, puis en cliquant sur un opérateur dans la barre d'outils.

Enfin, pour changer le nom d'une tâche, l'utilisateur doit sélectionner une tâche, spécifier le nouveau nom dans le champ *Name*, puis valider en cliquant sur le bouton *Apply.*

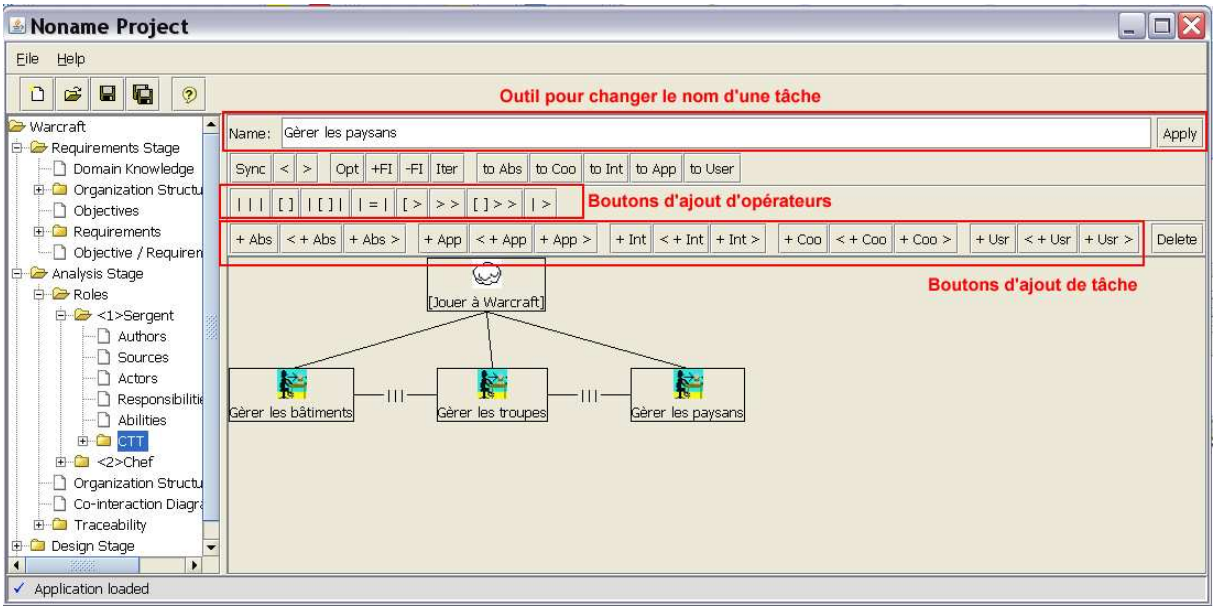


Figure 115 : Interface de l'éditeur d'arbre de tâches intégré dans TOUCHE Case (http://www.penichet.net/).

2.4 EUTERPE

L'outil EUTERPE repose sur la notation GTA [Veer 2000]. EUTERPE est une application pour Windows 95 et Window NT, mais fonctionne encore avec Window XP. EUTERPE est téléchargeable à l'adresse suivante : http://www.cs.vu.nl/~gerrit/gta/euterpe.html.

Le fonctionnement d'EUTERPE pour spécifier un arbre de tâches, est assez proche de CTTE. En effet, pour ajouter une tâche, l'utilisateur doit sélectionner une tâche existante (sauf pour créer la racine), puis cliquer sur l'un des deux boutons d'ajout : ajout d'une tâche fille, ajout d'une tâche sœur, comme souligné à la Figure 116.

Un opérateur de composition pour une tâche peut être spécifié en définissant textuellement une valeur pour la propriété *Constructor* de la tâche. Les propriétés d'une tâche sont accessibles sous forme d'une fenêtre modale s'ouvrant lors d'un double-clic sur une tâche. La fenêtre modale et le champ *Constructor* sont illustrés à la Figure 116.

EUTERPE permet également de décrire les autres concepts de la notation GTA (Objet, Rôle, Agent, Evénement) au sein de représentations distinctes.

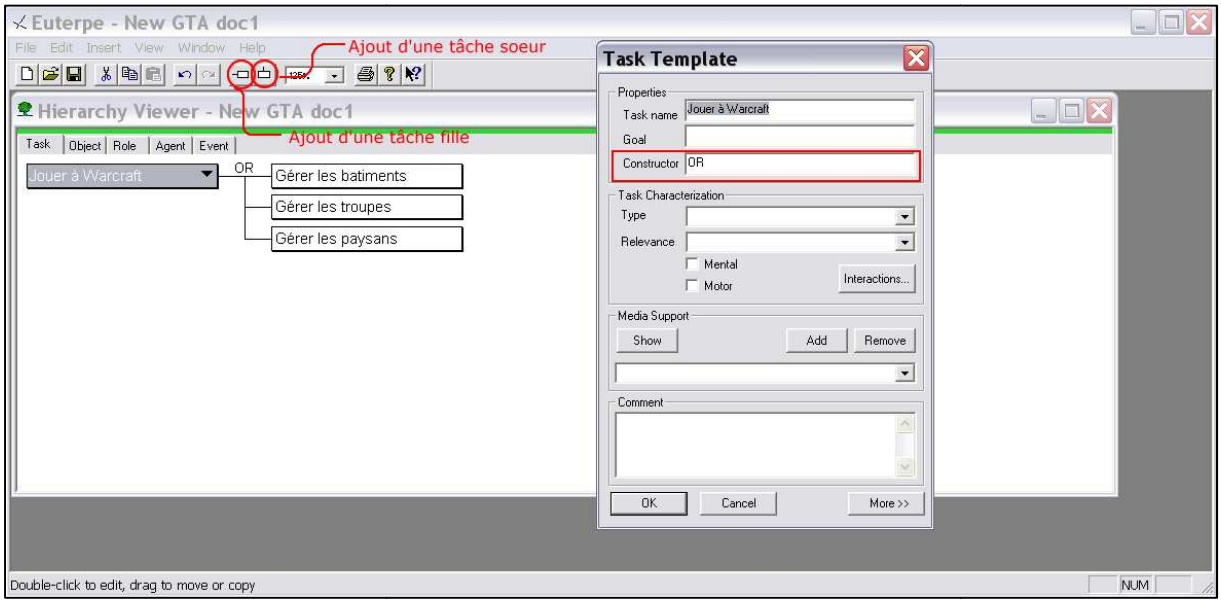


Figure 116 : Interface d'EUTERPE (http://www.cs.vu.nl/~gerrit/gta/euterpe.html).

2.5 Dynamo-Aid Tool

Dynamo-Aid [Clerckx 2005, Clerckx 2008] dispose d'un outil d'édition de spécification que nous n'avons pu tester. D'après les auteurs de [Clerckx 2005], celui-ci est développé en Java. L'illustration de la Figure 117 est issue de [Clerckx 2005].

Pour la spécification d'arbre de tâches, Dynamo-Aid repose sur la notation CTT [Paterno 1997]. La Figure 117 illustre l'édition d'un arbre de tâche CTT avec l'outil Dynamo-Aid Tool. L'outil Dynamo-Aid Tool propose également de décrire des représentations complémentaires à l'arbre de tâches, tels que les contextes d'utilisation.

Le fonctionnement de Dynamo-Aid Tool pour les tâches d'ajout de tâches et de spécification des opérateurs sont les mêmes que pour CTTE. Notons que l'éditeur ne dispose pas d'une mini-vue comme dans K-MADe. Enfin, l'éditeur utilise des animations pour donner une impression de fluidité dans l'interaction, notamment dans le cas de zoom et de déplacement de la vue.

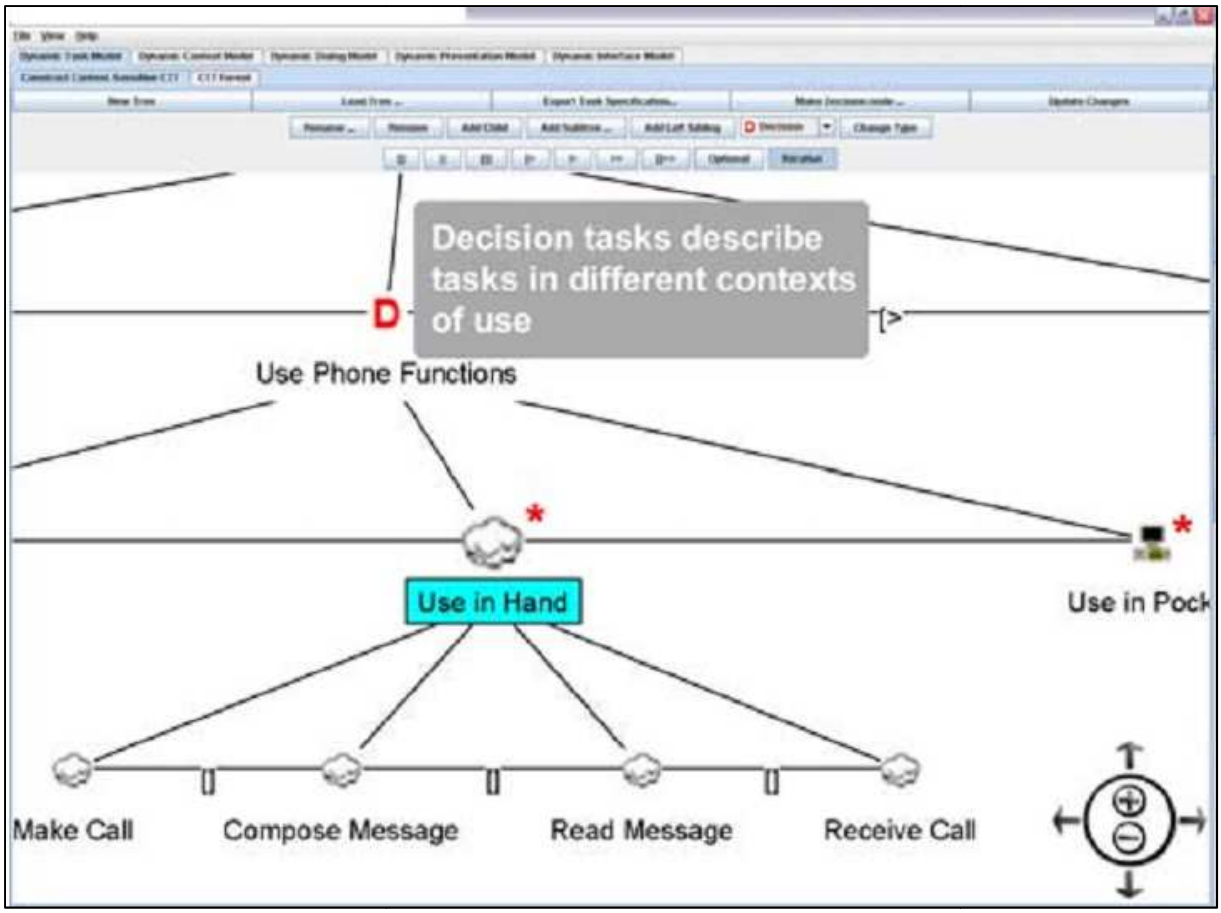


Figure 117 : Editeur d'arbre de tâches de Dynamo-Aid, illustration tirée de [Clerckx 2005].

2.6 CTML

L'éditeur de spécification CTML [Wurdel 2008a, Wurdel 2008b, Wurdel 2008c] est développé sous la forme d'une extension pour l'environnement de développement intégré Eclipse. Il requiert donc l'installation d'une machine virtuelle java et l'installation d'Eclipse.

Nous n'avons pas pu nous procurer l'éditeur de spécification CTML. Aussi, l'illustration de la Figure 118 provient de [Wurdel 2008c]. La notation CTML repose sur la notation CTT, aussi, la forme générale de l'outil d'édition est proche de CTTE. Cependant d'après les auteurs, l'ajout de tâches dans l'arbre est réalisée par glisser-déposer depuis une palette de types de tâches (non visible dans la Figure 118). Enfin, les auteurs ne précisent pas la manière de spécifier un opérateur entre les tâches.

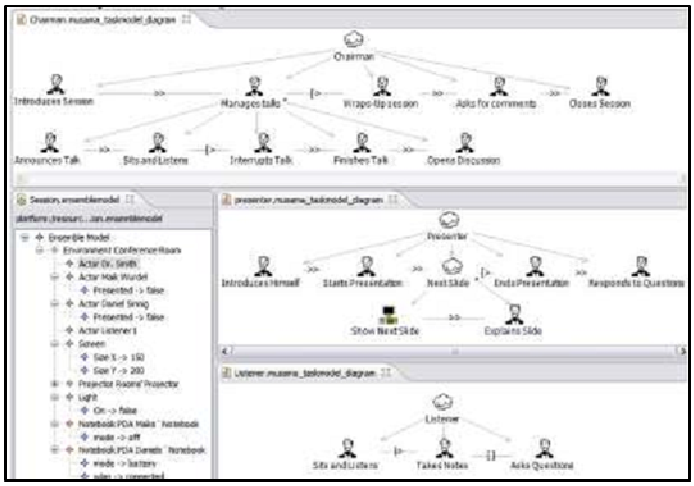


Figure 118 : Editeur d'arbre de tâche CTML, illustration issue de [Wurdel 2008].

2.7 Synthèse

Nous avons présenté les éditeurs d'arbres de tâches dont disposent les notations introduites au Chapitre 2. Le Tableau 16 présente un récapitulatif de ces éditeurs. Les deux premières colonnes indiquent le nom du logiciel et la notation correspondante. La troisième colonne concerne les moyens mis en œuvre pour la diffusion de l'outil et de la notation. La quatrième colonne concerne la facilité d'utilisation de l'éditeur.

Pour ce qui concerne la diffusion des éditeurs (et par conséquent des notations), nous soulignons l'importance de la disponibilité en téléchargement sur internet des éditeurs. Cette importance a été soulignée dans [Veer 2000], [Mori 2002] et [Molina 2009]. Néanmoins, deux des six éditeurs ne sont pas téléchargeables, ce qui constitue un frein naturel à leur diffusion.

Cinq des six éditeurs sont développés en Java pour des questions de portabilité. Cette portabilité augmente les possibilités d'utilisation de l'outil. Toutefois, elle impose aux utilisateurs d'installer au préalable une machine virtuelle java, ce qui n'est pas toujours possible (notamment au sein d'entreprise où les droits administrateurs sont réservés au service des équipements informatiques). EUTERPE, l'éditeur le plus ancien est développé spécifiquement pour Windows. Enfin, les six éditeurs sont développés sous la forme d'application locale que l'utilisateur doit déployer sur sa machine avant de l'utiliser. Le changement de version doit être réalisé manuellement par les utilisateurs, ce qui est un frein à l'évolution de l'éditeur et de la notation correspondante.

Nous constatons qu'à l'exception de K-MADe, les éditeurs proposés ne permettent pas la manipulation directe des tâches. De plus la création d'ue tâche implique une sélection préalable de la tâche mère. L'interaction offerte dans ces éditeurs complexifie et contraint la spécification d'un arbre dans un ordre imposé.

L'interaction que propose K-MADe est proche d'un éditeur de dessin vectoriel. En effet, les tâches sont ajoutées par sélection d'un outil et placement sur la surface d'édition. En revanche, les liens entre tâches et sous-tâches ne sont pas automatiquement ajoutés. Cela constitue un travail fastidieux pour l'utilisateur qui pourrait être automatisé.

Outils	Notation	Diffusion	Simplicité d'utilisation
CTTE	CTT	Téléchargeable : http://giove.isti.cnr.it/tools/ctte/ Multiplateforme (Java)	Manque de stabilité (bug)
K-MADE	K-MAD	Téléchargeable : http://kmade.sourceforge.net/ Multiplateforme (Java)	Interaction proche d'un éditeur vectoriel de dessin
TOUCHE Case	TOUCHE	Téléchargeable : http://www.penichet.net/ Multiplateforme (Java)	Peu de manipulation directe, peut de flexibilité (ordres des actions imposées)
Dynamo-Aid	Dynamo-Aid	Non téléchargeable	Peu de manipulation directe, peut de flexibilité (ordres des actions imposées)
CTMLE	CTML	Non téléchargeable Multiplateforme (Java) reposant sur l'IDE Eclipse	-
EUTERPE	GTA	Téléchargeable : http://www.cs.vu.nl/~gerrit/gta/euterpe.html Logiciel pour Window 95/NT	Peu de manipulation directe, peut de flexibilité (ordres des actions imposées)

Tableau 16 : Récapitulatif des éditeurs d'arbres de tâches.

Notre étude des éditeurs existants souligne que l'approche la plus communément adoptée consiste à réaliser un éditeur multiplateforme en Java installé en local. Cette solution permet une large diffusion. Néanmoins, les mises à jour de l'éditeur doivent être faites manuellement et une machine virtuelle Java doit être installée. Pour pallier à ces problèmes, tout en garantissant une large diffusion, nous nous sommes orientés vers une application web dynamique.

Nous retenons aussi de cette étude qu'il est important que l'éditeur soit accompagné d'une description de la notation sous-jacente comme pour CTTE et K-MADe.

Enfin, notre expérience pratique avec les éditeurs existants nous a permis d'identifier des difficultés dans l'interaction. Notre objectif est de permettre la manipulation directe sur l'arbre afin que le concepteur se focalise sur la spécification et d'offrir une grande flexibilité dans la façon de construire l'arbre tout en minimisant les tâches répétitives.

3 L'éditeur e-COMM

Nous présentons d'abord l'architecture d'e-COMM, puis les technologies utilisées pour le réaliser avant de détailler son interface utilisateur.

3.1 Architecture de l'éditeur e-COMM

Pour garantir une large diffusion, nous avons souligné l'intérêt d'un éditeur multiplateforme reposant sur une machine virtuelle. Néanmoins la mise à jour de l'éditeur, par exemple suite à une modification ou à l'extension de la notation, est difficile à mettre en œuvre. Pour pallier à ce problème, nous avons identifié trois solutions :

1. La première solution consiste à développer un outil de mise à jour automatique pour un éditeur qui s'installe localement. L'outil de mise à jour peut alors vérifier l'existence d'une version plus récente au démarrage de l'éditeur et l'installer le cas échéant.
2. La deuxième solution consiste à utiliser une solution désormais courante : développer l'éditeur sous la forme d'une application web dynamique. L'application est déployée sur un serveur et les utilisateurs s'y connectent avec un navigateur web. Ainsi, une mise à jour du serveur est répercutée immédiatement pour tous les utilisateurs.

3. La troisième solution consiste à développer l'éditeur sous la forme d'une application internet riche (RIA en anglais pour *Rich Internet Application*) qui s'exécute au sein d'une page web. Le navigateur web de l'utilisateur télécharge automatiquement la dernière version de l'éditeur ce qui règle le problème de mise à jour.

La première solution peut être écartée, s'il s'agit de développer un éditeur spécifique à chaque plateforme. Dans le cas d'un développement pour une machine virtuelle Java, la première solution est en revanche acceptable.

Les deuxièmes et troisièmes solutions sont très intéressantes car elles résolvent le problème de mise à jour, et évitent de plus aux utilisateurs l'installation de l'éditeur sur leurs machines. La troisième solution est meilleure que la deuxième pour les raisons suivantes :

- L'interface des applications web dynamique repose pour l'interface client sur les langages *HTML/CSS* et *Javascript*. Ceux-ci ne permettent pas aujourd'hui de réaliser des applications vectorielles fonctionnant pour tous les navigateurs. Tracer un simple trait entre deux tâches est un défi qui peut être relevé au détriment des performances.
- Les RIA permettent la création d'applications aussi dynamiques qu'une application classique. Par ailleurs, les technologies les plus établies pour les RIA, telles que : *Java*, *Flash/Flex* de Adobe, ou *Silverlight* de Microsoft offrent la possibilité d'installer en local les RIA, afin de s'exécuter en mode déconnecté. Aussi, une RIA développée dans le cadre de la troisième solution correspond également à la première solution envisagée.

En revanche, la deuxième solution présente un avantage sur les deux autres : Les données d'une application web dynamique sont stockées sur un (ou plusieurs) serveur(s). L'application ne fait que modifier ces données. Aussi, les données de l'utilisateur sont accessibles partout quel que soit la machine avec laquelle elle/il utilise l'éditeur (à condition bien sûr de disposer d'un accès internet).

A partir de cette analyse, la solution que nous avons adoptée est la composition de la deuxième et la troisième solution. Il s'agit de développer une application web dynamique qui permet à l'utilisateur de s'identifier, et de gérer un ensemble de projets de spécifications. De plus, cette application web dynamique intègre une RIA qui permet d'éditer les spécifications d'arbre de tâches. Pour cela, le serveur sur lequel l'application web est déployée comporte également un ensemble de services web auxquels accède le RIA.

La Figure 119 illustre l'architecture de notre solution et son déploiement. Sur un serveur s'exécute un serveur d'applications, nécessaire à l'hébergement de notre application web dynamique et de nos services web. Le serveur dispose également d'une base de données. L'application web et les services web peuvent accéder à la base de données. Les services web, l'application web et la base de données peuvent être déployés sur des machines distinctes pour augmenter les performances.

L'utilisateur dispose d'une machine munie d'un navigateur web. Il navigue sur les pages de l'application web dont celles contenant l'éditeur réalisé sous la forme d'un RIA. Le RIA appelle les services web selon les actions de l'utilisateur.

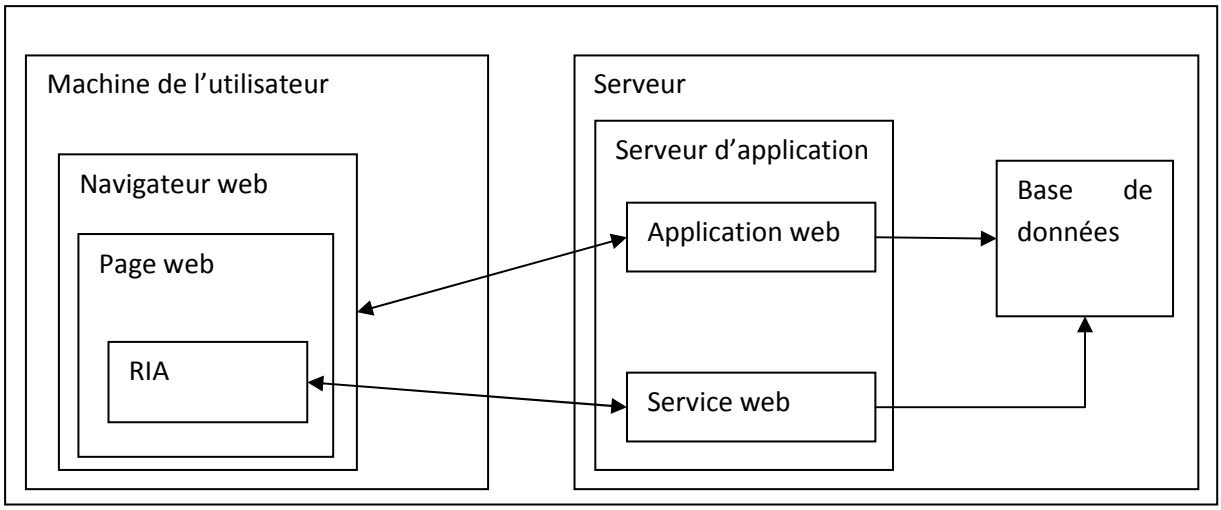


Figure 119 : Déploiement de notre solution lors de l'exécution.

Cette solution offre les avantages suivants:

- Elle permet l'identification des utilisateurs et donc un stockage en ligne de leurs données (c'est-à-dire de leurs spécifications). Cela permet aussi de collecter des données statistiques sur l'utilisation de l'éditeur (et donc de la notation), telles que le nombre d'arbres réalisés avec l'éditeur, ou la taille moyenne de ces arbres en nombre de tâches.
- L'application RIA offre un degré de dynamisme équivalent à une application interactive classique.
- Seule l'installation d'une machine virtuelle permettant le RIA est nécessaire.
- La mise à jour de l'application web dynamique et du RIA est automatique pour les utilisateurs, ce qui simplifie l'évolution de l'éditeur et de sa notation.

3.2 Technologies utilisées

Pour la réalisation des différents éléments logiciels qui composent notre solution, nous avons choisi plusieurs technologies.

3.2.1 Partie serveur

Nous avons choisi de réaliser l'application web dynamique en *Java* pour la partie serveur en utilisant les *framework Struts* et *Hibernate*.

Le *framework Struts* organise le développement d'applications web. *Struts* sépare les préoccupations selon le patron Modèle-Vue-Contrôleur. Les vues sont les pages web de l'application, et sont développées avec les langages *HTML*, CSS, et *Javascript,* ainsi qu'avec un jeu de balises spécifique à *Struts*. Le contrôleur et le modèle sont développés en Java. Les objets décrits dans le modèle peuvent être consultés depuis les pages web grâce aux balises spécifiques de *Struts*.

Pour la gestion de la persistance, nous avons utilisé le *framework Hibernate*. Celui-ci permet d'abstraire la base de données sous la forme d'un ensemble de classes. L'accès à la base de données est ainsi réalisé grâce à la manipulation d'objets, instances de ces classes.

Le choix du langage Java et de ces deux *frameworks* est justifié par le temps de réalisation logicielle de l'outil. Nous disposons des compétences techniques et de l'expérience nécessaires pour réaliser rapidement l'application web dynamique à l'aide de ces technologies. Le temps gagné sur la réalisation de cette partie a été consacré à la réalisation du RIA et des services web.

Nous avons également décidé de développer l'ensemble des services web en *Java,* car ces derniers utilisent la même base de données que l'application web. Comme nous disposons pour la base de données d'une couche de persistance développée en *Java* (à l'aide d'*Hibernate*), la réutiliser pour les services constitue un gain de temps important.

Afin de gagner encore du temps de développement pour la partie serveur, nous avons choisi d'utiliser un framework pour la réalisation des services : *Restlet*. Ce dernier permet de réaliser rapidement des services *Rest*. En effet, celui-ci abstrait la prise en charge des requêtes et permet au développeur de se concentrer sur l'écriture du code métier.

Pour le serveur d'applications sur lequel doit s'exécuter l'application web dynamique et les services web, nous avons choisi *Tomcat* qui repose sur un serveur *Apache*.

Enfin, pour les mêmes raisons, nous avons choisi une base de données *MySQL*.

3.2.2 Partie RIA

Pour le développement du RIA, nous avons choisi la technologie *Silverlight* de Microsoft. Nous ne disposions initialement d'aucunes compétences particulières dans les trois technologies RIA existantes: *Flash/Flex*, *Java*, et *Silverlight*. Pour réaliser notre choix, nous avons essayé chacune des technologies pour la réalisation d'un prototype de notre éditeur.

La technologie *Silverlight* s'est révélée être la plus simple à mettre en œuvre. En effet, en une semaine, nous avons pu développer un prototype qui rassemble les fonctionnalités minimales que nous avions définies pour l'éditeur : zoom animé, déplacement rapide de la vue, déplacement des tâches par manipulation directe, et communication avec le serveur. Tout d'abord *Silverlight* permet de réaliser l'accès au service web *Rest* du serveur de façon très simple. De plus, la réalisation d'interface *Silverlight* repose sur la description de composants graphiques de forme quelconque, auquel un comportement peut être associé. Les composants et leurs comportements peuvent être décrits manuellement (avec XAML pour les composants, C# ou VB pour le comportement) ou grâce à un éditeur vectoriel. *Flash/Flex* propose la même approche mais impose l'usage d'un éditeur vectoriel dont l'utilisation s'est révélée trop compliquée. Enfin, en Java, nous avons essayé de développer un prototype avec la boîte à outils Swing et un autre avec Java/FX. L'utilisation de Swing est une tâche fastidieuse comparée à l'usage de Silverlight. *Java/FX* adopte le même principe que *Silverlight* et *Flash/Flex*, c'est-à-dire la description d'éléments graphiques et l'association de comportements. Cependant, la syntaxe de *Java/FX* est tellement compliquée que nous avons renoncé avant d'obtenir un prototype satisfaisant.

3.2.3 Synthèse des technologies

Pour conclure, le choix des technologies utilisées pour la réalisation d'e-COMM repose d'une part sur nos compétences et notre expérience pour ce qui concerne le développement du module serveur, et

d'autre part sur la simplicité de réalisation pour le développement du RIA. Ces choix sont récapitulés au sein du Tableau 17.

Module logiciel	Technologie choisit	Justification
Application web dynamique	*HMTL/css/javascript, Java, Struts, Hibernate*	Compétences existantes, gain de temps.
Services web	*Java, Restlet*	Couche persistance déjà développée en Java avec *Hibernate*
RIA	*Silverlight*	Technologie RIA la plus rapide à prendre en main, et n'impose pas l'utilisation d'éditeur vectoriel.
Base de données	*MySQL*	Compétences existantes, gain de temps.
Serveur d'application	*Tomcat*	Compétences existantes, gain de temps.

Tableau 17 : Technologies utilisées pour chaque module de l'éditeur e-COMM.

3.3 IHM du portail e-COMM

Nous présentons le portail dédié à la diffusion de la notation COMM et de l'éditeur e-COMM. Nous illustrons chaque page de ce portail avec des captures d'écran.

Figure 120 : Portail e-COMM (www.e-comm.fr.nf).

Le portail est illustré à la Figure 120. Il est composé de 4 parties principales : Une barre en haut contenant le nom de l'outil ainsi qu'un mini-formulaire pour s'identifier. Un menu comportant 5

éléments que nous détaillons par la suite. Une barre d'accès rapide situé à droite. Et, enfin, un zone centrale qui contient la page web actuellement visitée (la page d'accueil à la Figure 120).

3.3.1 Barre d'accès rapide

La barre d'accès rapide rassemble des liens et des informations contextuelles, qui varient selon que l'utilisateur est identifié ou non. Elle a pour but de faciliter l'utilisation de e-COMM en limitant les efforts de navigation.

Lorsque l'utilisateur n'est pas identifié, la barre d'accès rapide adopte la forme illustrée à la Figure 121. Elle propose un lien vers la création d'un compte, des informations statistiques concernant la diffusion d'e-COMM, et un lien pour recommander e-COMM.

Lorsque l'utilisateur est identifié, la barre d'accès rapide prend la forme illustrée à la Figure 121. Elle est divisée en deux parties. La première partie permet à l'utilisateur de recommander e-COMM en envoyant des invitations. La deuxième partie permet à l'utilisateur de gérer ses projets. Il peut ainsi soit créer un nouveau projet, soit charger l'un des projets qu'elle/il a déjà créé.

Figure 121 : Barre d'accès rapide lorsque l'utilisateur est connecté au portail e-COMM.

3.3.2 Menus et pages du portail

Le menu comporte 5 liens qui renvoient aux 5 pages principales du portail accessibles sans identification : La page d'accueil illustrée à la Figure 121 propose une synthèse de nos objectifs, et introduit la notation COMM ainsi que l'éditeur e-COMM. La page de nouvelles permet de se tenir informé sur les nouveautés de COMM et d'e-COMM. La page contact présente l'équipe à l'origine du projet. La page "Notation COMM" présenté à la Figure 122 présente l'ensemble des éléments de la notation COMM.

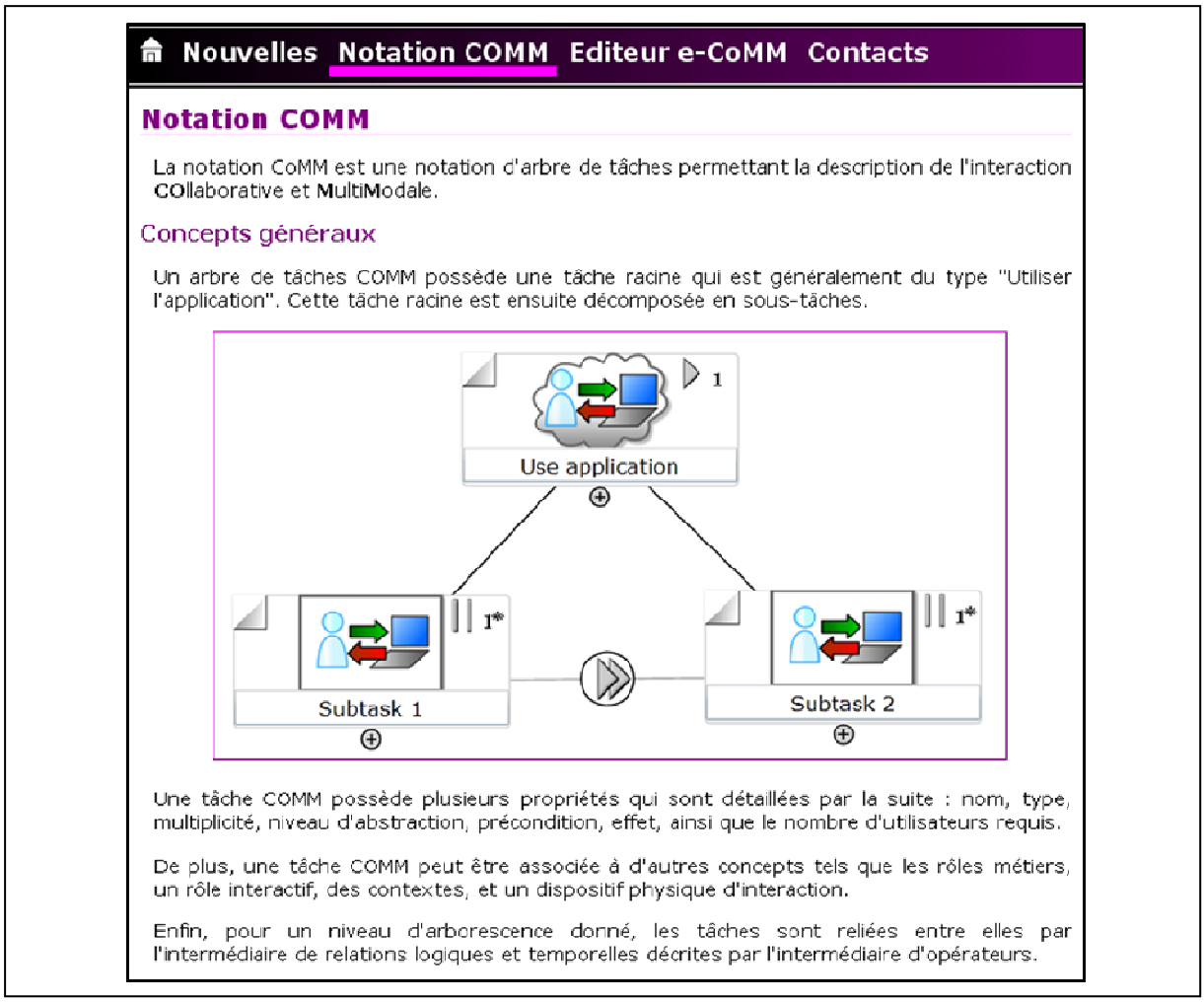


Figure 122 : Page "Notation COMM" du portail e-COMM.

Enfin, la page "Editeur e-COMM" présentée à la Figure 123 propose une version de démonstration de l'éditeur qui ne requiert pas d'identification, et qui sauvegarde un projet pendant 30 jours par cookie.

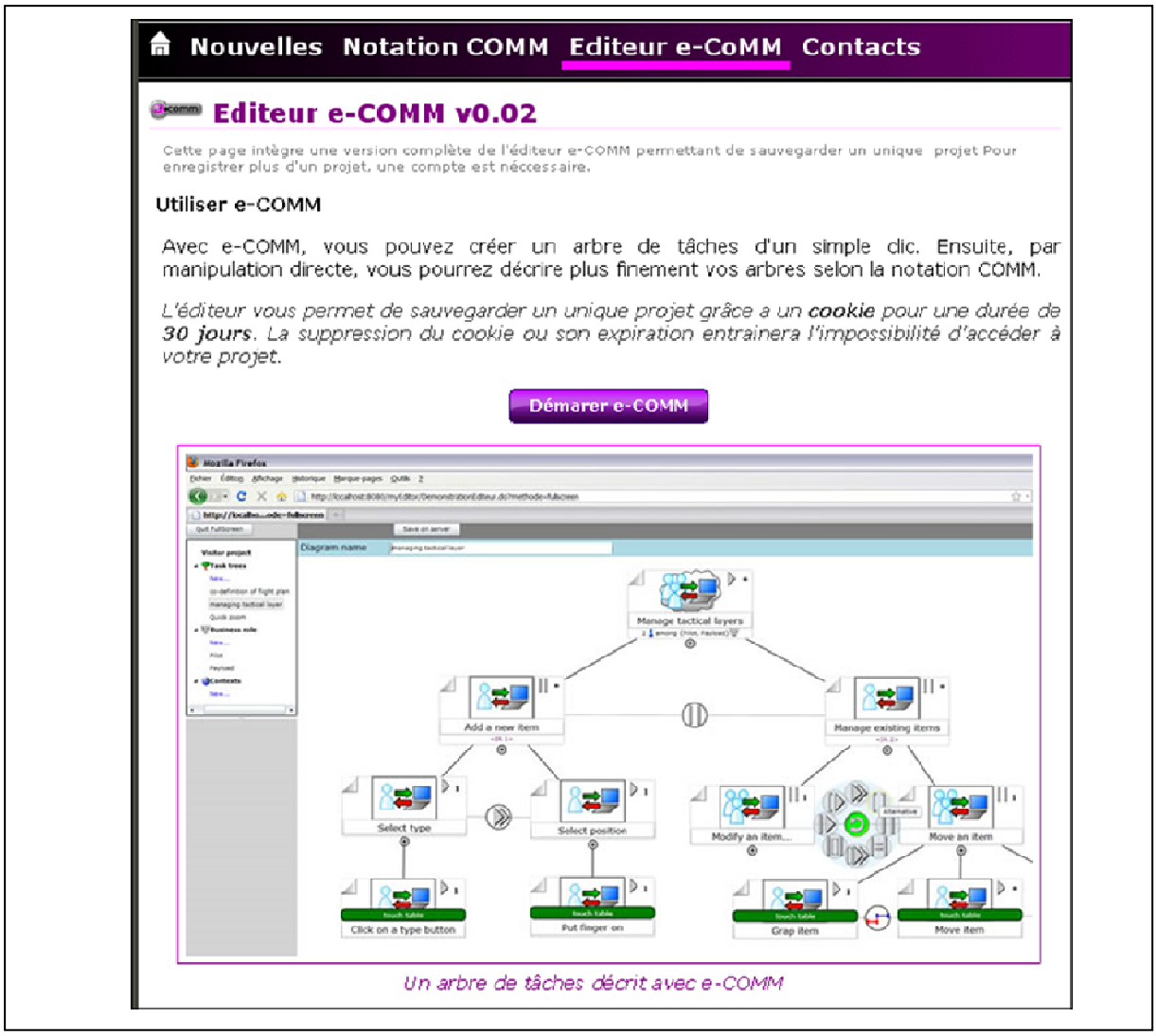


Figure 123 : Page "Editeur e-COMM" du portail e-COMM.

Lorsque l'utilisateur est identifié, le menu est légèrement différent. La page "Editeur e-COMM" telle qu'elle est illustrée à la Figure 123 disparaît pour être remplacée par une page "Editeur e-COMM : vos projets". Cette nouvelle version de la page est illustrée à la Figure 124. Elle comporte un tableau rassemblant les projets créés par l'utilisateur, et fournit des liens pour les supprimer ou les éditer. Elle propose également un lien pour créer un nouveau projet.

Figure 124 : Page "Editeur e-COMM : vos projets" du portail e-COMM (requiert une identification).

Lorsque l'utilisateur sélectionne un projet dans la liste présentée dans la page "Editeur e-COMM : vos projets", ou sur la barre d'accès rapide, ou encore "créer un nouveau projet", elle/il est obtient la page de gestion du projet présentée à la Figure 125. Elle/il peut alors modifier les propriétés des projets ou les éditer.

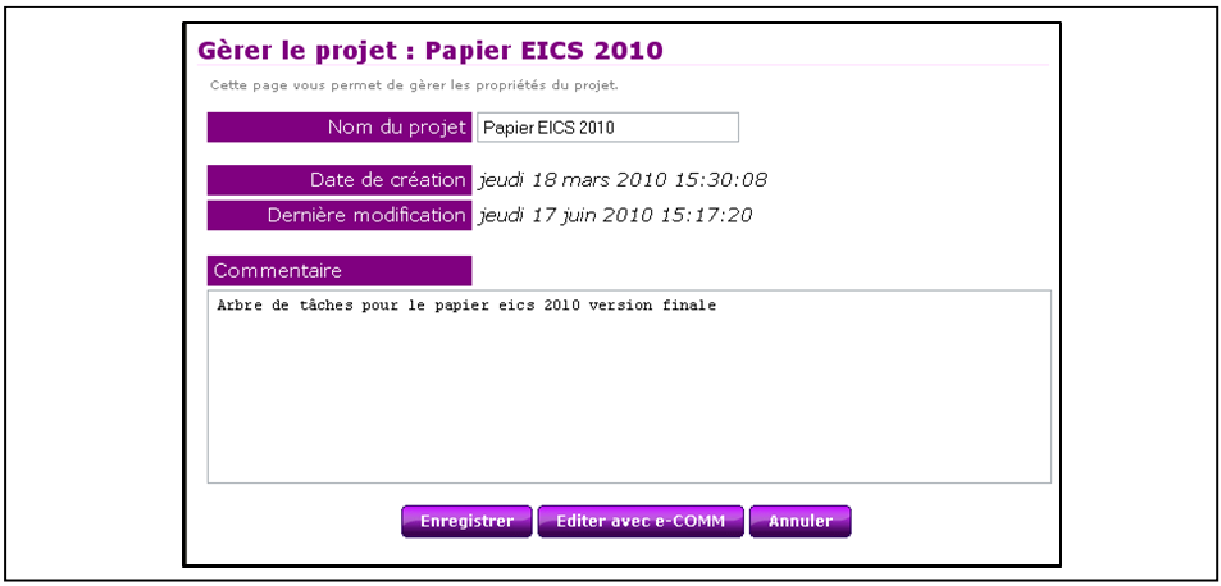


Figure 125 : Page de gestion d'un projet du portail e-COMM.

Cliquer sur le bouton "Editer avec e-COMM" de la page de gestion de projet (Figure 125), ou cliquer sur l'icône e-COMM d'une ligne du tableau de la page "Editeur e-COMM : vos projets" (Figure 124) ouvre l'application internet riche qui constitue l'éditeur e-COMM. Cette RIA est décrite en détail dans la section suivante.

3.4 IHM du RIA e-COMM

Nous présentons ici l'application internet riche qu'est l'éditeur d'arbre de tâches e-COMM développé avec la technologie *Silverlight* de Microsoft.

Lorsqu'un utilisateur utilise e-COMM pour éditer un projet, celui-ci prend les dimensions du navigateur web comme l'illustre la Figure 126. L'interface de l'éditeur est divisée en trois parties.

La première partie est une barre d'outils générale qui propose plusieurs boutons : le premier bouton permet de fermer l'éditeur pour revenir au portail, le deuxième permet d'enregistrer le projet en cours d'édition sur le serveur et le troisième permet de télécharger le projet sous la forme d'un fichier xml.

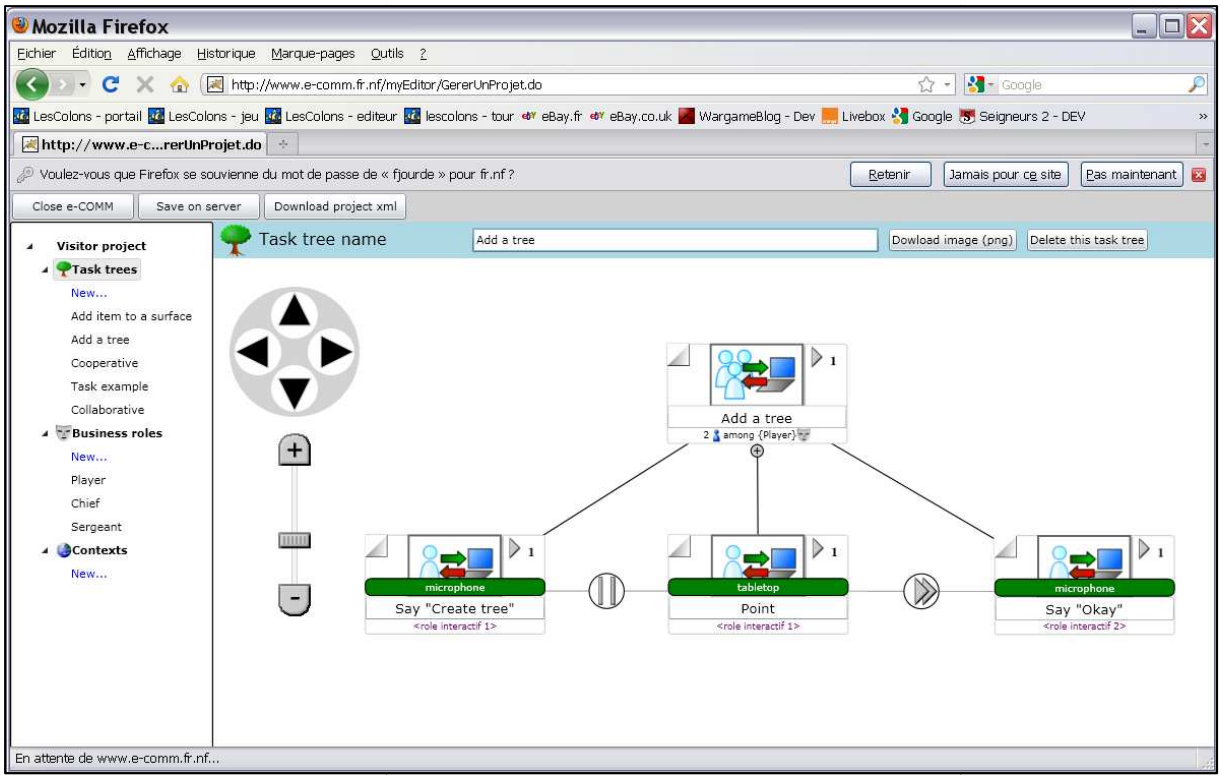


Figure 126 : Un projet ouvert e-COMM.

La deuxième partie de l'interface (à gauche) est une arborescence qui présente l'ensemble des modèles créés dans le projet et des liens pour créer de nouveaux modèles. Un projet comporte trois types de modèles : les arbres de tâches (*Task trees*), les rôles métiers (*Business roles*) et les contextes (*Contexts*). Les modèles déjà créés sont répartis selon ces trois catégories. Chaque catégorie de modèle dispose également d'un lien "*new…*" pour créer un nouveau modèle du type correspondant.

La troisième partie constitue la zone d'édition des modèles. Celle-ci est composée de deux parties : la barre en bleu propose des outils qui concernent le modèle en cours d'édition, tandis que le reste constitue la zone de spécification du modèle. Pour un arbre de tâches, il s'agit d'une feuille blanche sur laquelle sont représentées les tâches de l'arbre, comme l'illustre la Figure 126. Avant de détailler la construction de l'arbre de tâches par manipulation directe, nous présentons comment spécifier un rôle métier puis un contexte.

3.4.1 Description des rôles métiers

L'éditeur e-COMM permet de décrire des rôles métiers (Figure 127). Chaque rôle métier possède un nom qui l'identifie dans l'arborescence et une description textuelle.

Une fois défini, un rôle métier peut être associé à un nombre quelconque de tâches. Les dernières peuvent être situées dans n'importe quel arbre de tâches du projet. Les rôles métiers peuvent ainsi être considérés comme des variables globales du projet.

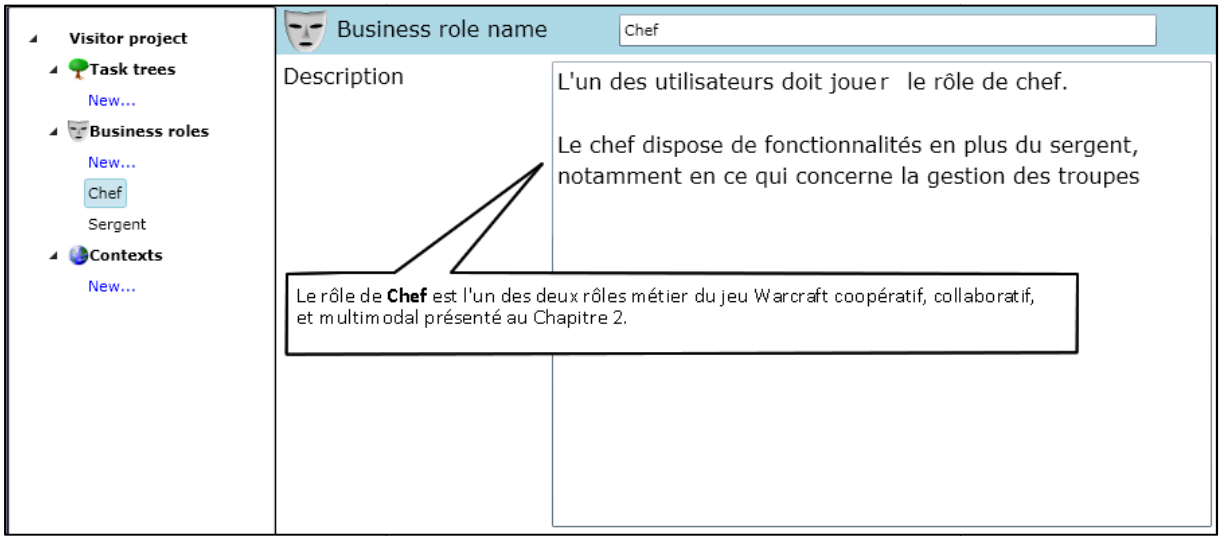


Figure 127 : Spécification d'un rôle avec e-COMM.

3.4.2 Description des contextes

L'éditeur e-COMM propose de décrire des contextes sous une forme proche des rôles métiers. Un contexte dispose ainsi d'un nom, d'une étiquette numérotée, et d'une description textuelle, comme l'illustre la Figure 128.

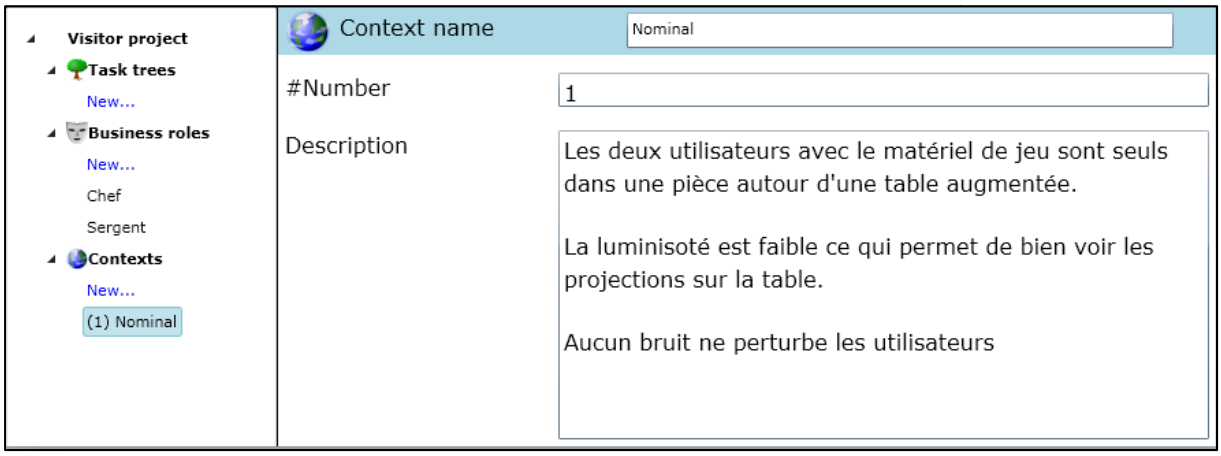


Figure 128 : Spécification du contexte d'utilisation avec e-COMM.

Une fois défini, un contexte peut être associé à un nombre quelconque de tâches présentes dans n'importe quel arbre de tâches du projet. A l'instar des rôles métiers, les contextes peuvent être considérés comme des variables globales du projet.

3.4.3 Description des arbres de tâches

L'éditeur d'arbres de tâches est illustré à la Figure 129 suite à la création d'un nouvel arbre de tâches. Son interface est composée de deux parties : la partie supérieure comporte le nom de l'arbre, un bouton pour télécharger l'arbre au format image, et un bouton pour le supprimer. La partie inférieure de l'interface est composée d'une zone d'édition présentant l'arbre nouvellement créé.

L'éditeur e-COMM propose de décrire les arbres de tâches COMM uniquement par manipulation directe. C'est pourquoi l'interface de l'éditeur e-COMM illustrée à la Figure 129 ne propose aucune barre ou palette d'outils d'édition. Ainsi, lorsqu'un nouvel arbre est créé, il comporte toujours une tâche racine.

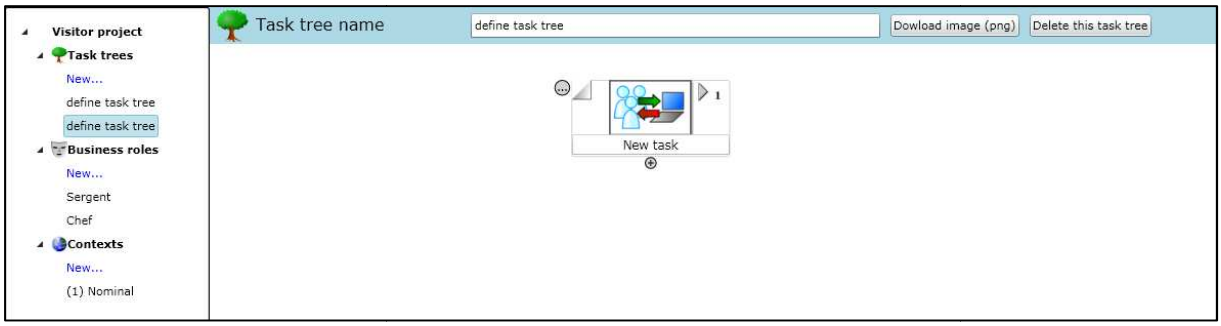


Figure 129 : Interface de l'éditeur d'arbre de tâches e-COMM suite à la création d'un nouvel arbre de tâches.

Nous détaillons dans la suite les différentes fonctionnalités de l'éditeur sous une forme proche d'un scénario d'usage classique.

3.4.3.1 Ajouter une tâche

Lorsqu'un arbre de tâches COMM est créé avec l'éditeur, il contient une tâche racine. La Figure 130 illustre la manière d'ajouter deux sous-tâches à cette tâche en quatre étapes.

1. La première étape consiste pour l'utilisateur à amener le curseur sur le bouton d'ajout de sous-tâches représenté par un "+" sous la représentation d'une tâche.
2. Lorsque son curseur atteint le bouton d'ajout, le curseur prend la forme d'une main communément utilisée pour désigner un objet cliquable et le bouton "+" s'entoure de jaune et grossit.
3. L'utilisateur effectue un premier clic sur le bouton "+", ce qui a pour effet immédiat de créer une sous-tâche. La tâche racine est directement reliée à la tâche nouvellement créée par un lien, à l'instar de ce que l'on peut obtenir avec CTTE et contrairement à K-MADe où les liens doivent être tracés.
4. Lorsque l'utilisateur effectue un deuxième clic sur le bouton "+", une deuxième sous-tâche vient se positionner à droite de la première. Celle-ci est également liée à la tâche racine comme l'était la première. De plus, un lien est automatiquement ajouté entre les deux sous-tâches. Ce lien affiche l'opérateur "?" qui signifie que la relation entre les deux sous-tâches n'est pas encore définie. La création automatique du lien constitue une amélioration notable par rapport à CTTE, puisqu'il incite l'utilisateur à décrire une relation entre les sous-tâches. La création automatique du lien entre sous-tâches était déjà présente dans TOUCHE Case. Nous l'avons renforcée avec l'insertion automatique d'un opérateur par défaut pour renforcer

l'incitation à la définition de la relation. Nous montrons plus loin comment un utilisateur peut définir l'opérateur de relation entre tâches.

L'ajout d'une sous-tâche dans e-COMM est ainsi très simple et requiert peu d'actions de la part de l'utilisateur (trajectoire d'interaction réduite). L'ajout de tâches avec les éditeurs existants est plus compliqué. Outre la réduction de la trajectoire d'interaction, l'éditeur évite des allers-retours entre plusieurs zones de travail, comme dans les autres éditeurs. Avec CTTE et TOUCHE case, l'ajout se fait par deux sélections, la sélection d'une tâche de l'arbre en cours d'édition puis la sélection d'un type de tâche à ajouter dans la palette. Avec K-MADe, la création se fait par la sélection d'un type de tâche dans la palette puis la sélection d'un emplacement dans la zone d'édition, et enfin, d'un glisser déposer pour relier la tâche mère avec la tâche nouvellement créée.

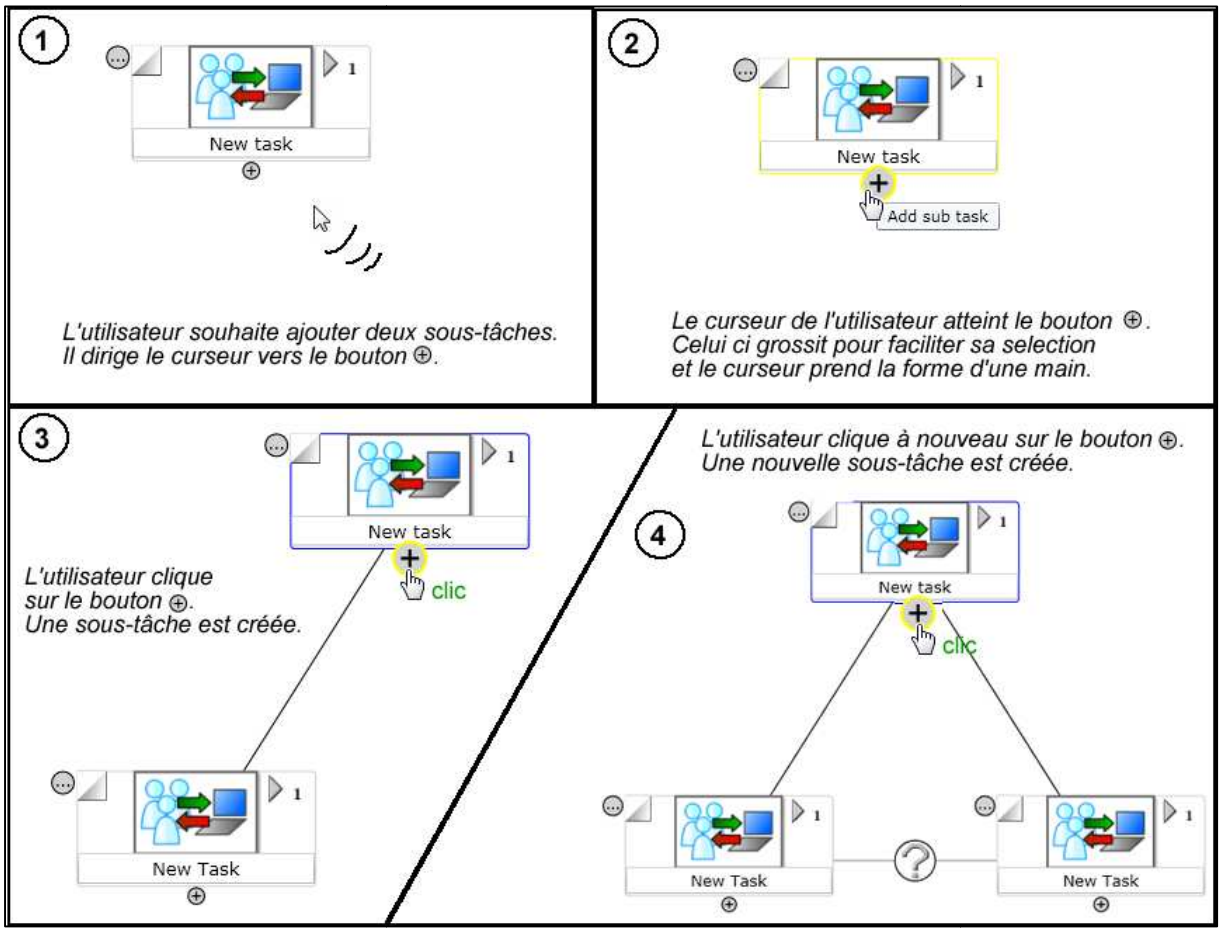


Figure 130 : Création de deux sous-tâches.

3.4.3.2 Spécifier un opérateur CTT entre deux tâches

Pour spécifier un opérateur entre deux tâches, nous proposons d'utiliser un menu circulaire. L'un des apports des menus circulaires est de garantir la même distance à parcourir pour chaque item contrairement à un menu linéaire [Bailly 2009]. L'utilisation du menu pour spécifier un opérateur issu de la notation CTT est illustrée en six étapes à la Figure 131.

1. L'utilisateur amène le curseur vers l'opérateur par défaut (?) défini entre les deux sous-tâches.
2. Lorsque celui-ci est atteint, il est mis en relief et le curseur prend la forme d'une main.

3. L'utilisateur clique alors sur l'opérateur et le menu circulaire s'ouvre avec une animation.
4. Le menu est ouvert.
5. L'utilisateur peut alors déplacer son curseur puis cliquer sur l'un des opérateurs, par exemple activation (*activate*).
6. Après la sélection, le menu disparaît et l'opérateur entre les deux sous-tâches est remplacé par l'opérateur sélectionné, par exemple activation.

Cette interaction est plus efficace que celles proposées par les éditeurs existants car elle est localisée au niveau de l'opérateur à remplacer. CTTE et TOUCHE Case imposent un aller-retour vers une barre d'outils tandis que EUTERPE et K-MADE reposent sur l'ouverture de la fenêtre de propriétés de la tâche mère des sous-tâches.

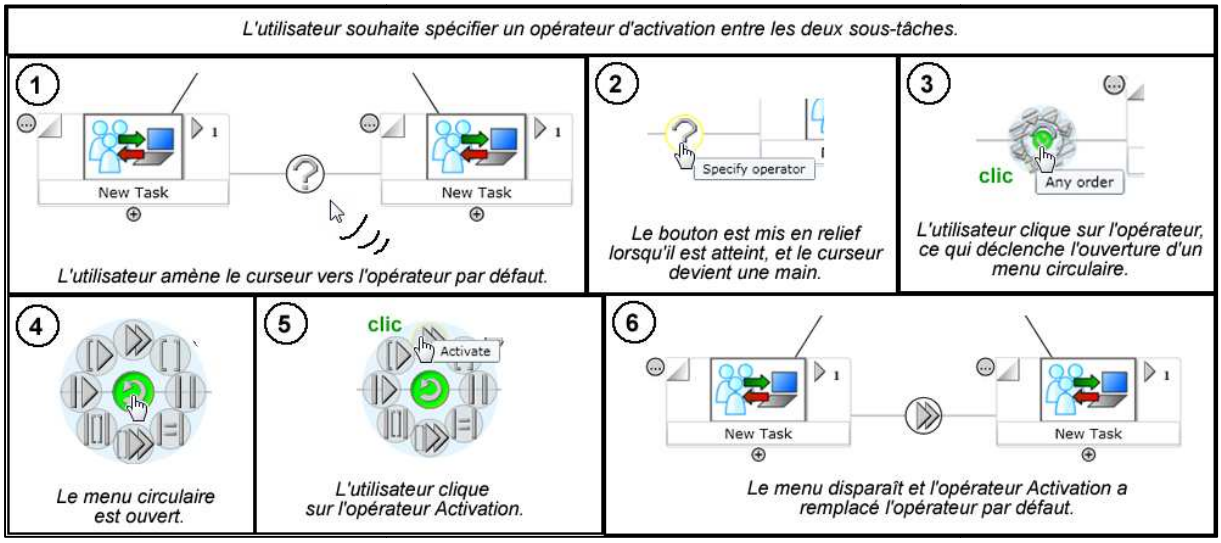


Figure 131 : Spécification d'un opérateur CTT entre deux sous-tâches.

3.4.3.3 Spécifier un opérateur de Allen entre deux tâches.

Un autre type d'opérateur peut être spécifié entre deux sous-tâches COMM : il s'agit des opérateurs issus des relations temporelles de Allen. Leur introduction dans la notation COMM visent principalement à permettre l'expression de combinaisons de modalités d'interaction, c'est-à-dire exprimer une relation temporelle entre deux tâches modales. Aussi, nous illustrons à la Figure 132, le scénario d'usage qui permet à l'utilisateur de l'éditeur e-COMM de spécifier un opérateur de Allen entre deux sous-tâches modales. Celui-ci est composé de six étapes.

1. L'utilisateur amène son curseur vers l'opérateur à remplacer.
2. L'utilisateur ouvre le menu circulaire de choix des opérateurs de CTT puis clique en son centre.
3. Le menu circulaire des opérateurs de Allen s'ouvre alors dans une animation.
4. Le menu est ouvert.
5. L'utilisateur clique sur un opérateur, par exemple "A recouvre B" (*A overlaps B*).
6. Le menu circulaire disparaît et l'opérateur entre les deux tâches modales est remplacé par l'opérateur sélectionné, par exemple "A recouvre B".

La trajectoire d'interaction est un peu plus longue que pour la sélection d'un opérateur CTT, puisqu'elle nécessite un clic de plus. Cependant, nous avons ajouté un accès direct au menu

circulaire des opérateurs de Allen par un double-clique sur l'opérateur à modifier. Ainsi, l'utilisateur n'a pas à attendre le chargement du menu circulaire de CTT. Par ailleurs, l'interaction reste localisée là où la modification doit être réalisée dans l'arbre de tâches et ne nécessite pas l'utilisation d'une barre d'outils.

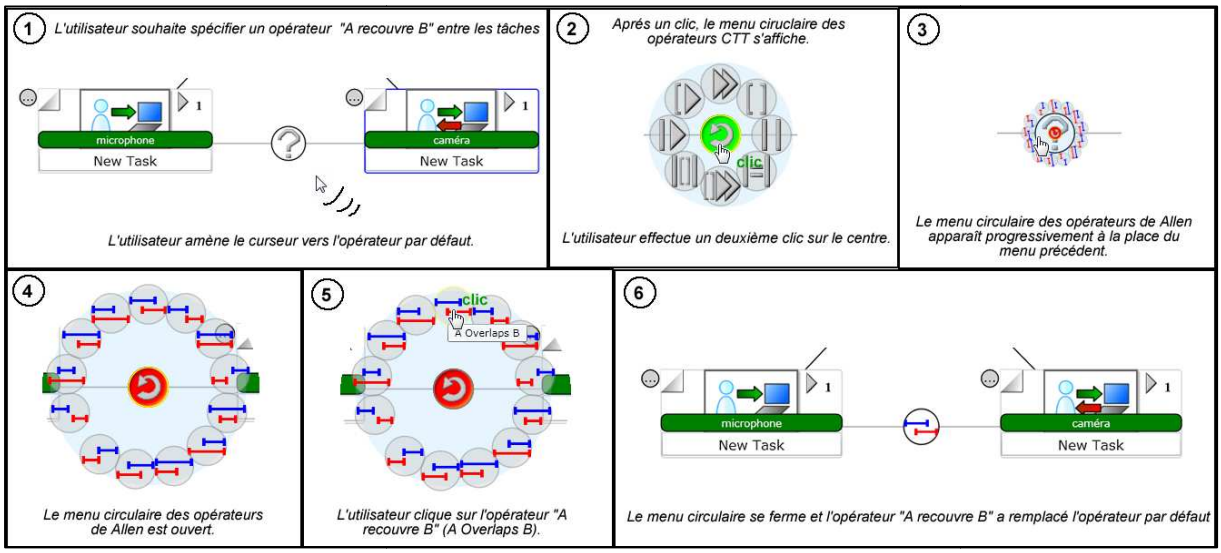


Figure 132 : Spécification d'un opérateur de Allen entre deux sous-tâches modales.

3.4.3.4 Sélectionner une tâche

La sélection d'une tâche se fait par manipulation directe, comme cela est illustré en trois étapes à la Figure 133.

1. L'utilisateur amène le curseur vers la tâche.
2. La tâche est mise en relief, le curseur devient une main, puis l'utilisateur clique sur la tâche.
3. La tâche est sélectionnée.

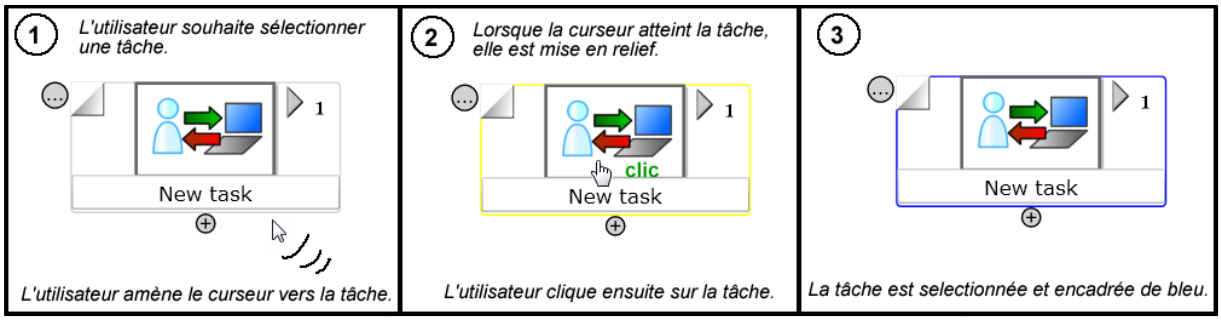


Figure 133 : Sélectionner une tâche.

3.4.3.5 Sélectionner plusieurs tâches

L'éditeur e-COMM autorise la sélection de plusieurs tâches simultanément. Cela permet notamment de déplacer plusieurs tâches. La sélection de plusieurs tâches peut s'effectuer de deux manières différentes.

La première méthode consiste à sélectionner les tâches unes à unes en maintenant le bouton CRTL enfoncé. La sélection avec la touche CTRL enfoncé d'une tâche déjà sélectionnée la retire de la

sélection. L'utilisation de la touche CTRL pour sélectionner/désélectionner des objets est présente dans de nombreux outils tels que des éditeurs de dessin.

La deuxième méthode consiste à utiliser un lasso comme l'illustre la Figure 134. Un lasso peut être tracé avec la souris en effectuant un glisser-déposer entre deux points (Origine et Destination) dans la zone d'édition. La rétroaction au cours de l'utilisation du lasso est assurée par le tracé d'un rectangle bleu dont un coin est situé à l'origine, et le coin opposé suit le curseur de la souris. Lorsque l'utilisateur relâche la souris, toutes tâches complètement ou partiellement comprise dans le rectangle sont sélectionnées (et par là même, encadrées de bleu).

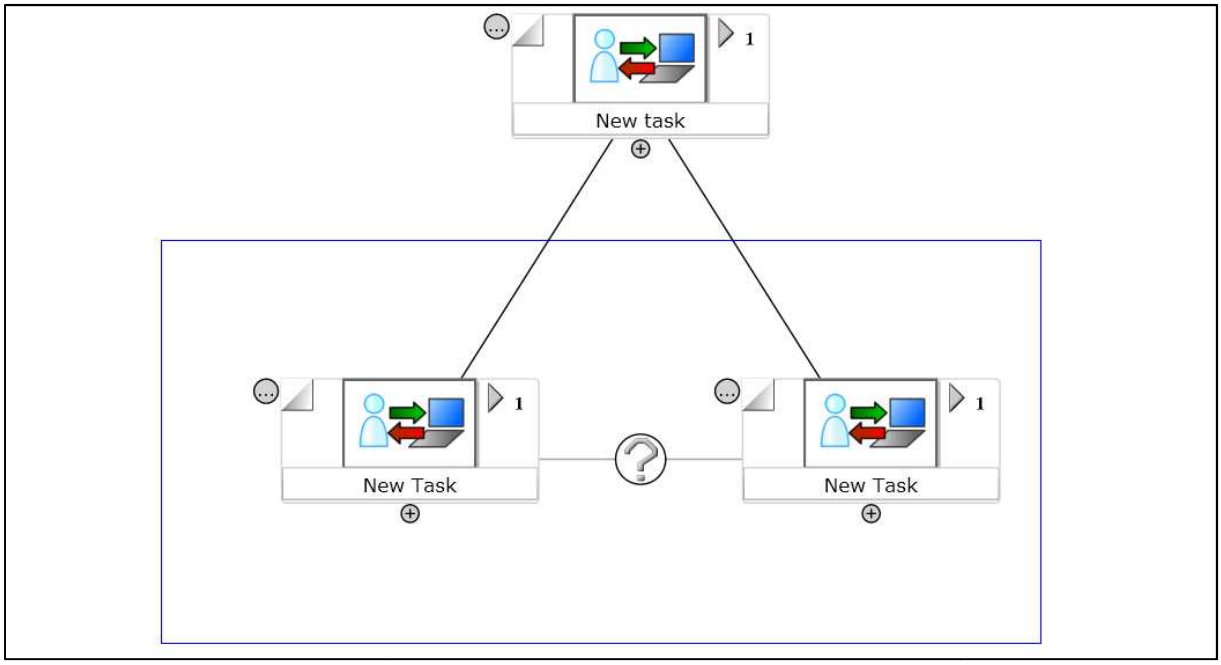


Figure 134 : Utilisation du lasso pour sélectionner plusieurs tâches.

3.4.3.6 Déplacer des tâches

A la manière de K-MADe, l'éditeur d'arbres de tâches e-COMM laisse à l'utilisateur le soin d'organiser visuellement les tâches de l'arbre. Le placement des tâches est ainsi entièrement libre. Cependant, lors de l'ajout de nouvelles tâches, un placement automatique est proposé à l'utilisateur.

Le choix de cette méthode de placement est motivé par la grande souplesse laissé au concepteur en train de spécifier un arbre de tâches. Le déplacement d'une tâche par l'utilisateur peut correspondre à l'un des objectifs suivants :

- Le premier objectif consiste à déplacer une tâche uniquement pour assurer un meilleur rendu visuel de l'arbre pour l'utilisateur. La structure de l'arbre n'est pas affectée par ce déplacement.
- Le deuxième raison de déplacer une tâche dans l'arbre est de modifier sa position dans l'arbre. Ce type de déplacement implique donc une modification de la structure de l'arbre.

Nous avons souhaité définir une méthode d'interaction unique pour ces deux types de déplacements aux objectifs distincts. Ces deux interactions s'effectuent par un glisser-déposer d'une tâche vers une destination.

Pour déplacer une tâche sans modifier la structure de l'arbre, l'utilisateur peut réaliser un glisser-déposer d'une tâche dans la zone d'édition. Lorsque la tâche est prise, elle apparaît en semi transparence, jusqu'à ce qu'elle soit déposée.

Pour déplacer une tâche dans la structure de l'arbre, l'utilisateur réalise également un glisser-déposer. La dépose doit en revanche être effectuée sur des cibles spécifiques que nous nommons points d'insertion. Les points d'insertion apparaissent à plusieurs endroits de l'arbre lorsque l'utilisateur commence à déplacer une tâche. La Figure 135 illustre l'apparition des points d'insertion représentés par des cercles rouges dans l'arbre, dans le cas où l'utilisateur déplace la tâche *T22* (qui est semi-transparente). Les points d'insertion apparaissent dans l'arbre là où la tâche peut être insérée. L'apparition des points d'insertion est régie par les règles suivantes :

a. Lorsqu'une tâche n'a pas de sous-tâches, un point d'insertion est affiché en dessous d'elle.
b. Lorsqu'une tâche n'a pas de tâches sœur à gauche ou à droite, un point d'insertion est respectivement placé à gauche ou à droite.
c. Enfin, un point d'insertion est placé sur tous les opérateurs, pour signifier qu'une tâche peut être insérée entre deux tâches. Notons que l'opérateur lié à la tâche en cours de déplacement n'est pas concerné puisqu'il va disparaître suite au déplacement.

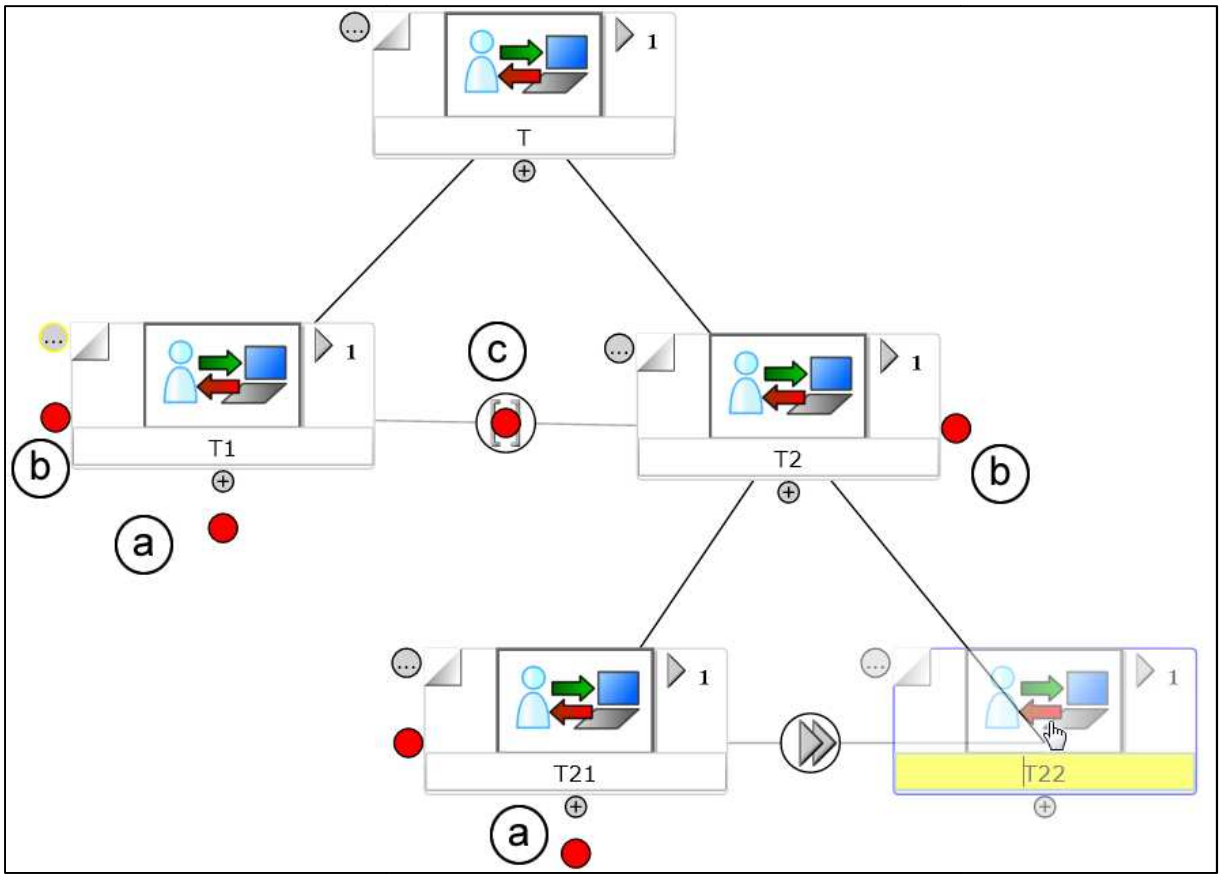


Figure 135 : Apparition des points d'insertion lors du déplacement d'une tâche.

L'utilisation d'un point d'insertion est illustrée à la Figure 136. Nous considérons que l'utilisateur souhaite déplacer la tâche *T22* pour qu'elle devienne une sous-tâche de la tâche *T21*. Lors des étapes (1) et (2), l'utilisateur fait glisser la tâche *T22* en direction du point d'insertion situé sous la tâche *T21*. Lorsque la tâche atteint le point d'insertion, celui-ci grossit pour inciter l'utilisateur à déposer la

tâche (3). Enfin, l'utilisateur dépose la tâche *T22* (4). Les points d'insertion disparaissent et la tâche *T21* est insérée comme une sous-tâche de la tâche *T21*. Le déplacement de tâches dans la structure de l'arbre est équivalent à un couper-coller. C'est pourquoi, nous n'avons pas implémenté de fonction couper-coller dans l'éditeur e-COMM.

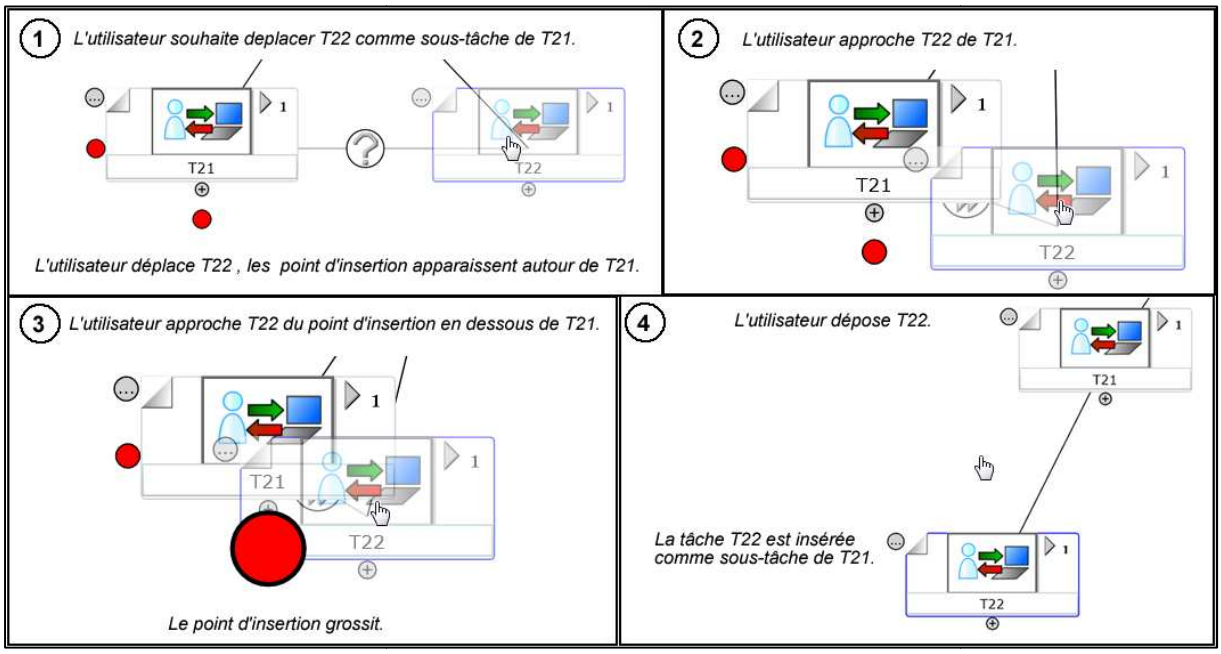


Figure 136 : Déplacement d'une tâche dans un arbre en utilisant un point d'insertion.

3.4.3.7 Manipuler la vue (facteur de zoom et position).

Un arbre de tâches qu'il soit réalisé ou non avec la notation COMM peut avoir des dimensions conséquentes. Aussi, il est souvent difficile de représenter l'arbre entier sur un écran. Comme les éditeurs CTTE et K-MADe, nous avons adopté le principe d'une vue mobile sur l'arbre en cours d'édition. L'utilisateur peut déplacer la vue et en modifier le facteur de zoom.

Pour déplacer la vue, l'utilisateur dispose de deux possibilités. La première consiste à utiliser les touches fléchées de son clavier. Ainsi, par exemple, la touche ↑ permet de déplacer la vue vers le haut. La deuxième consiste à déplacer la vue par glisser-déposer. Pour cela, l'utilisateur doit maintenir la touche CTRL enfoncé, puis effectuer un glisser-déposer de la zone d'édition (le curseur étant placé sur le fond blanc). La vue se déplace ainsi de manière fluide en fonction du mouvement imprimé par l'utilisateur.

Dans la première version de l'éditeur, le seul moyen de modifier le facteur de zoom pour l'utilisateur est d'utiliser la roulette de la souris. Le zoom est initialement réglé sur 100%. Le facteur de zoom augmente ou diminue de 10% lorsque l'utilisateur fait rouler la roulette de la souris respectivement vers le haut ou le bas. La valeur du facteur de zoom est comprise dans l'intervalle [1% ; 100%].

Lors de l'évaluation de l'éditeur qui est détaillée au Chapitre 5, plusieurs défauts dans la navigation ont été mis en lumière les fonctions de modification du facteur zoom et de déplacement par CTRL + glisser-déposer étaient peu intuitives. Pour corriger ce défaut d'utilisabilité, nous avons ajouté les deux artefacts graphiques illustrées à la Figure 137 :

- Une rose des vents qui permet de déplacer la vue dans une des quatre directions cardinales.
- Une glissière de zoom, qui représente le facteur de zoom actuel et qui permet de le modifier. L'augmentation ou la diminution du zoom s'effectue en cliquant respectivement sur les boutons "+" et "-", ou en déplaçant le curseur de la glissière respectivement vers le haut ou le bas.

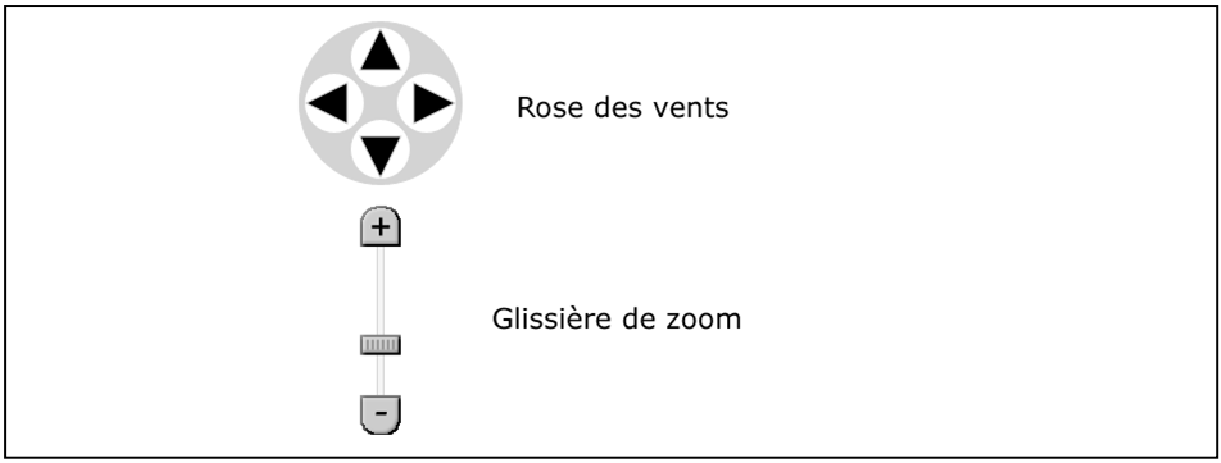


Figure 137 : Rose des vents pour le déplacement et glissière pour le réglage du facteur de zoom.

3.4.3.8 Modifier les propriétés d'une tâche

Les tâches COMM disposent de plusieurs propriétés décrites au Chapitre 3 que l'utilisateur doit pouvoir spécifier. Pour les modifier, l'utilisateur dispose d'une fenêtre modale de propriétés comme présentée à la Figure 138. La fenêtre de propriétés s'affiche lorsque l'utilisateur clique dans le coin supérieur gauche d'une tâche, ou lorsqu'il double-clique sur une tâche.

Nous avons privilégié l'utilisation d'une fenêtre modale s'ouvrant au centre de la vue plutôt qu'une barre de propriétés située à droite de la surface d'édition comme c'est le cas par exemple dans K-MADe. Ce choix est guidé par deux raisons :

- La première raison est qu'une barre de propriété est souvent vide (donc une surface d'affichage perdue), lorsqu'aucune tâche n'est sélectionnée, ou quand plusieurs tâches sont sélectionnées.
- La deuxième raison de ce choix est de minimiser la distance à parcourir entre la tâche et les propriétés.

Notons que dans les deux cas, une fenêtre modale ou une barre de propriété, il y a changement de contexte d'interaction, de la zone d'édition pour sélectionner la tâche vers la fenêtre ou la barre de propriétés.

La fenêtre comporte les différentes propriétés définies par la notation COMM :

- Le nom de la tâche (*Task name*) spécifié textuellement.
- Le type de la tâche (*Type*) choisi parmi les huit types de la notation COMM grâce à une liste déroulante.
- Le nombre d'utilisateurs (*Number of user*) pré-rempli selon le type de tâche et modifiable lorsqu'il s'agit d'une tâche multiutilisateur.

- Les rôles métiers (*Role*) choisis parmi ceux définis préalablement, au sein d'une liste déroulante à choix multiples.
- L'opérateur unaire d'itération défini selon la nature de l'itération et le nombre d'itérations. A la Figure 138, la nature et le nombre sont définis respectivement comme "Parallèle" et "Exactement N fois".
- Le niveau d'abstraction (*Abstraction*) qui peut être abstrait ou concret.
- Un rôle interactif (*Interactive role*). Une liste déroulante permet de choisir un rôle interactif existant, tandis qu'un formulaire intégré (*new role)* permet de créer un nouveau rôle interactif.
- Un dispositif physique (*Modality device*) peut être spécifié à condition que la tâche soit une feuille de l'arbre de tâches.
- Et enfin un champ de texte pour décrire la pré-condition de la tâche, ses effets, ou de simples commentaires. A la Figure 138, nous spécifions que le nombre d'itération N prend la valeur 2.

L'utilisateur peut modifier les propriétés de la tâche, puis sélectionner sur le bouton "Appliquer" (*Apply)* pour enregistrer ces modifications ou sélectionner le bouton "Annuler" (*Cancel*) pour quitter la fenêtre de propriétés sans enregistrer les modifications.

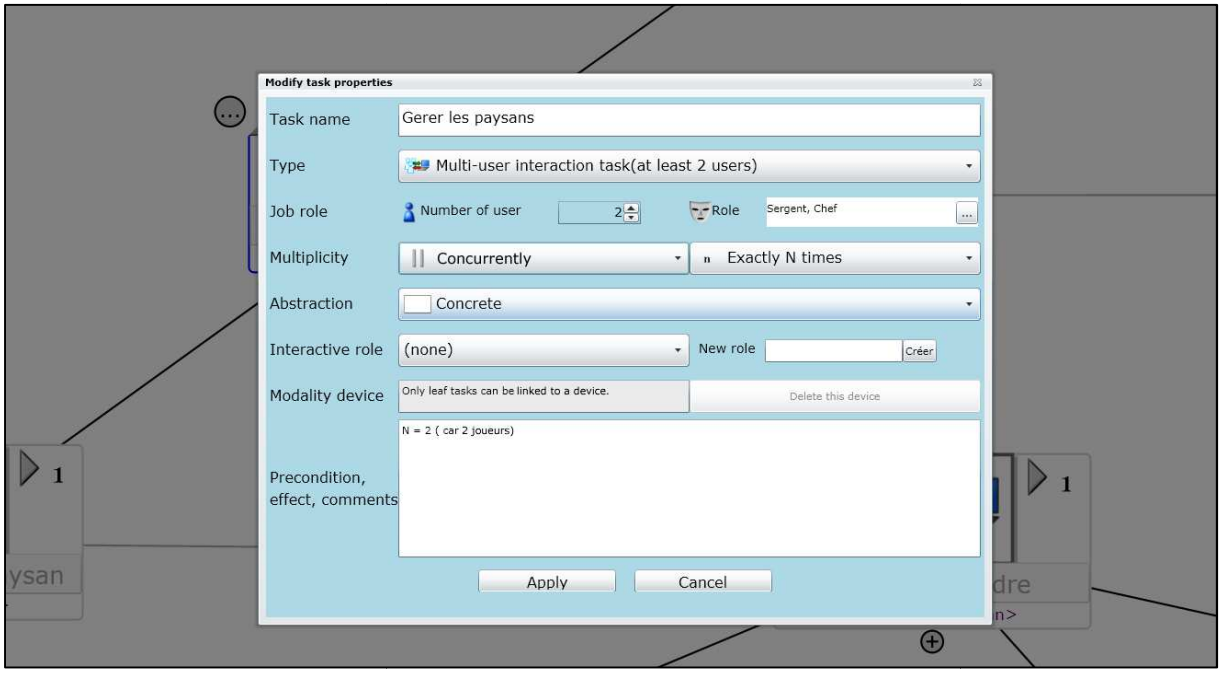


Figure 138 : Fenêtre modale de propriétés d'une tâche.

3.4.3.9 Associer un contexte à une tâche

Une tâche COMM peut être associée à un ou plusieurs contextes comme cela est décrit à la section 2.4.2 du Chapitre 3. Au sein de l'éditeur, les contextes doivent être définis préalablement à leur association à une ou plusieurs tâches. Si au moins un contexte est défini, alors les tâches de l'arbre comportent un bouton (…) à gauche de leur cadre comme l'illustre la Figure 139.

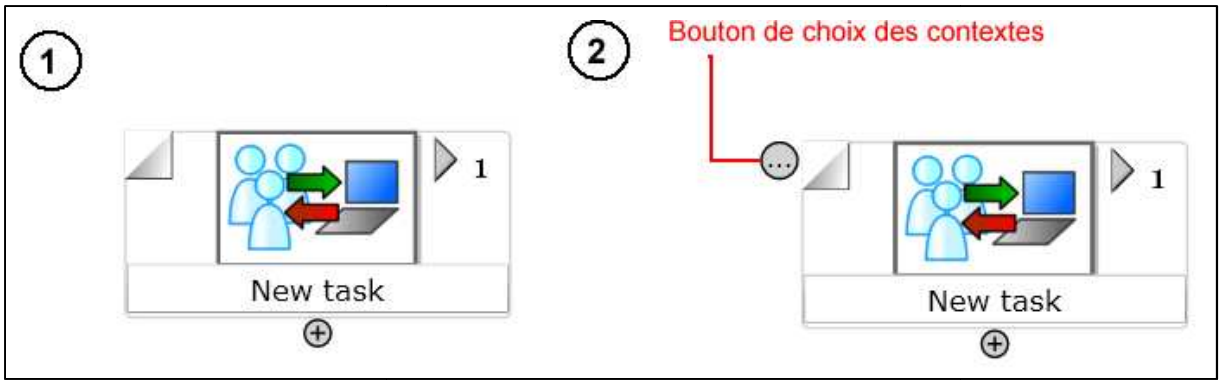


Figure 139 : Tâche (1) lorsqu'aucun contexte n'est défini, (2) lorsqu'au moins un contexte est défini.

Lorsque l'utilisateur clique sur ce bouton (…), une fenêtre modale de choix des contextes s'affiche. Cette fenêtre présentée à la Figure 140 comporte la liste des contextes. Un contexte est identifié par son étiquette numérotée et par son nom. Chacun des contextes est muni d'une case à cocher, qui peut être cochée pour indiquer que la tâche est associée à ce contexte.

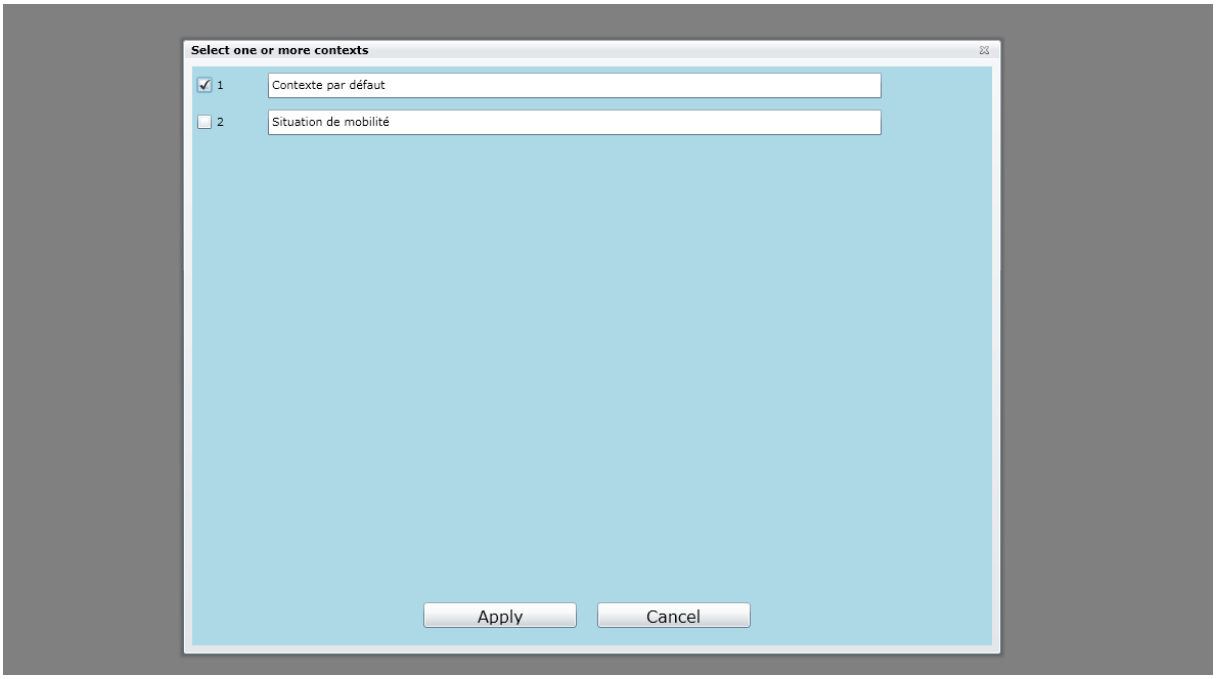


Figure 140 : Fenêtre modale de sélection des contextes.

3.4.4 Système de sauvegarde automatique

L'expérience d'utilisation des autres éditeurs existants nous a motivé à fournir un système de sauvegarde automatique.

L'architecture client RIA – serveur que nous avons choisie ne nous permet pas de modifier les fichiers sur la machine de l'utilisateur sans son accord explicite par une fenêtre modale. Aussi, la version en cours d'édition au sein du RIA ne peut être sauvegardée de manière régulière en local sur la machine de l'utilisateur.

Aussi, nous avons implémenté un système de sauvegarde automatique qui envoie avec une grande régularité les modifications vers le serveur où le projet est sauvegardé. Cette sauvegarde à lieu après chaque modification d'un élément d'un modèle.

Ainsi, en cas de fermeture intempestive du navigateur ou d'erreur du programme entrainant sa fermeture, l'utilisateur peut retrouver après reconnexion à e-COMM son projet dans un état le plus proche possible de celui avant l'arrêt intempestif.

Cette protection contre les erreurs renforce l'utilisabilité de l'éditeur e-COMM.

4 Conclusion

Dans ce chapitre, nous avons présenté le besoin de disposer d'un éditeur de spécification pour la notation d'arbre de tâches COMM afin d'aider à sa diffusion. Nous avons également énoncé nos objectifs pour la réalisation de cet éditeur : il doit couvrir l'ensemble de la notation COMM et doit être facile à déployer et à utiliser.

Pour cela, nous avons dans un premier temps étudié les éditeurs d'arbres de tâches existants associés aux notations introduites au Chapitre 2. Cette étude nous a permis d'identifier les atouts et les limitations des éditeurs qui ont guidé la conception de notre éditeur e-COMM. Nous n'avons ainsi identifié que certaines interactions principales des éditeurs existants étaient trop complexe, telles que l'ajout de nouvelles tâches.

Nous avons ensuite réalisé l'éditeur e-COMM. Il s'agit d'une application web dynamique (développé en Java) comprenant, pour la partie d'édition, une application internet riche (développée avec Silverlight). Le choix de cette combinaison de technologie a été guidé par notre objectif de facilité de déploiement pour l'utilisateur. L'éditeur e-COMM est ainsi publiquement accessible sur internet et exécutable directement au sein d'un navigateur.

Afin de faciliter l'utilisation de l'éditeur et sa prise en main, nous avons favorisé au maximum la manipulation directe pour la spécification de l'arbre de tâches lors de la conception de l'interaction. Dans le chapitre suivant consacrée aux évaluations expérimentales menées, nous revenons sur cette facilité d'utilisation et de prise en main de l'éditeur e-COMM.

Enfin, la réalisation de l'éditeur e-COMM constitue une contribution logicielle importante, car e-COMM couvre l'ensemble des concepts de la notation COMM et est ainsi un éditeur d'arbre de tâche complet.

Chapitre 5 : Evaluation de COMM et e-COMM

Nous avons présenté au Chapitre 3 la notation COM et au Chapitre 4 l'éditeur e-COMM. Aussi ce chapitre est dédié à l'évaluation de ces deux contributions, COMM et e-COMM. Notre objectif d'évaluation de COMM consiste à montrer d'une part que la notation répond bien au problème énoncé, spécifier l'interaction multiutilisateur et multimodale, et d'autre part qu'elle est compréhensible et utilisable par la population ciblée par la notation : les concepteurs d'IHM.

Nous abordons tout d'abord les problèmes liés à l'évaluation d'une telle notation et la démarche adoptée, et nous présentons ensuite les études que nous avons menées, autant analytiques qu'expérimentales.

1 Difficultés d'évaluation et démarche adoptée

Evaluer une notation de spécification est une tâche difficile. Comme souligné dans [Serrano 2010] et [Coutrix 2009], il existe une différence entre l'évaluation de techniques d'interaction (et autres produits "simples") et l'évaluation de notations ou modèles ("produits plus complexes"). En effet les techniques d'interaction, comme une nouvelle technique de menu, peuvent être évaluées par des évaluations contrôlées selon des métriques [Francone 2010]. Un exemple de métrique classique est la performance (rapidité, précision) en considérant des tâches à réaliser. Dans le cas d'une notation ou d'un modèle d'interaction, les méthodes traditionnelles doivent faire face à une complexité accrue :

- La tâche de spécification est une tâche complexe et parfois longue.
- Les mesures sont vagues : Utilité ? Qualité du système final développé à partir des spécifications ? Adéquation du système développé aux spécifications ? Difficulté de spécification (pourvoir d'expression) ?
- Les utilisateurs ont de profils hétérogènes : concepteurs, ergonomes, informaticiens.

Toutes ces difficultés d'évaluation sont soulignées dans [Olsen 2007] qui remet en cause nombreuses évaluations de type contrôlé qui ont été effectuées dans le domaine de l'IHM. Face à ces difficultés et dans le temps imparti d'une thèse, nous sommes conscients que nos évaluations sont donc partielles. Nous avons adopté une approche d'évaluation à la fois conceptuelle (ou analytique) et expérimentale (ou empirique) afin de mener des évaluations en largeur traitant différents aspects de la notation à évaluer.

L'évaluation conceptuelle (section 2), ne reposant donc pas sur l'expérience, c'est-à-dire l'utilisation de la notation, a consisté à étudier la notation COMM au regard de critères de la littérature et en particulier ceux de Green [Green 2000] et Olsen [Olsen 2007].

Les évaluations expérimentales ont consisté à spécifier un système existant ou à développer. Pour cela :

- Nous avons d'abord éprouvé la notation en spécifiant notre étude de cas WCCM (Warcraft Collaboratif, Coopératif et Multimodal) (section 3) : cette forme d'évaluation expérimentale (par un expert de la notation) permet de souligner le pouvoir d'expression de COMM, puisque nous avons montré dans notre état de l'art (Chapitre 2) que les notations existantes ne permettaient pas de spécifier complètement et de manière satisfaisante WCCM. En effet de nombreuses notations traitent soit la dimension multiutilisateur de l'interaction, soit la dimension multimodale de l'interaction : nous avons mis en relief les limites des notations existantes selon chacune de ces deux dimensions dans le Chapitre 2 et nous avons exposé au Chapitre 3 comment la notation COMM répondait aux besoins identifiés pour la dimension multiutilisateur de l'interaction, qu'elle soit coopérative ou collaborative, mais aussi pour la dimension multimodale de l'interaction.
- Nous avons ensuite fait utiliser la notation par des concepteurs, utilisateurs cibles de la notation (section 4). Le système à spécifier est un éditeur de posters ou d'affiches et nos utilisateurs cibles étaient des élèves ingénieurs en Master 2 Professionnel Génie Informatique de l'Université de Grenoble.
- Enfin nous avons utilisé la notation COMM tout au long du projet PEA FH/PA (Facteurs Humains et Partage d'Autorité) (section 5). Le système à spécifier est un poste de commande de drones multiutilisateur et multimodal. Cette expérimentation montre tout d'abord la capacité de la notation à passer à l'échelle dans le cadre de la spécification d'un système complexe (au-delà d'un cas d'étude plus simple comme WCCM). Cette expérimentation montre aussi la capacité de la notation COMM à être intégrée dans un processus de conception itératif, puisque que nous avons utilisé la notation à chaque cycle du projet qui a duré trois ans.

Nous notons enfin que nos évaluations se focalisent sur la notation COMM. Néanmoins nous avons fait utiliser l'éditeur e-COMM dès ses premières versions. Au Chapitre 4, nous avons montré l'utilité d'e-COMM pour aider à la diffusion de la notation et pour structurer l'utilisation de la notation. L'éditeur e-COMM prend en charge la description de l'ensemble des concepts de la notation COMM, à l'exception du mécanisme de *Template*. Aussi, l'évaluation d'e-COMM a pour objectif d'identifier les défauts d'utilisabilité qu'il présente. Pour cela, nous avons fait utiliser l'éditeur lors de l'évaluation à des concepteurs et nous avons utilisé e-COMM lors de la phase 2 du projet PEA FH/PA. De plus, l'ensemble des arbres de tâches de ce chapitre a été produit avec l'éditeur e-COMM.

2 Evaluation conceptuelle de COMM

Pour étudier conceptuellement la notation COMM, nous utilisons d'abord la méthode d'inspection décrite dans [Green 2000], notée « les dimensions cognitives des notations ». Cette méthode met en exergue 4 critères d'une notation : viscosité, contraintes, visibilité et lisibilité des rôles. Nous détaillons la définition de ces critères ainsi que notre analyse de COMM au regard de ces critères.

- Viscosité : la viscosité représente la résistance au changement dans une spécification. La notation COMM offre une faible viscosité au niveau des tâches modales : une modalité d'interaction peut être facilement et localement modifiée dans une spécification COMM. Néanmoins au niveau des tâches abstraites, la notation COMM est une extension de la

notation CTT et plus généralement des notations d'arbre de tâches. Notons que les nouveaux concepts introduits dans COMM, tels que les tâches multiutilisateurs, les tâches modales ou encore les rôles interactifs, ne modifient pas la structure de l'arbre tout en permettant de préciser les interactions multiutilisateur et multimodale. COMM hérite donc du problème des arbres de tâches pour ce qui concerne les tâches abstraites : en effet une modification à un haut niveau d'abstraction dans l'arbre de tâches peut impliquer de nombreuses modifications dans les sous-arbres et donc une forte viscosité.

- Contraintes : les contraintes concernent l'ordre de conception. La notation COMM offre la possibilité de spécifier toutes les tâches abstraites puis de focaliser sur l'aspect concret et multimodal de l'interaction ou de spécifier complètement de façon abstraite et concrète une partie des tâches. Malgré cette flexibilité de conception, correspondant à deux pratiques de conception, COMM impose néanmoins une approche descendante (top-down) au concepteur, héritée des notations d'arbre de tâches. Aussi il n'est pas possible de spécifier les tâches concrètes sans avoir spécifié au préalable les tâches abstraites correspondantes.
- Visibilité : la visibilité est la possibilité de voir ou trouver les éléments de la notation facilement. Nous avons construit la notation de telle façon qu'on puisse voir tous les éléments à tout instant : tâches abstraites, tâches modales, rôles et opérateurs. En particulier les niveaux abstrait et concret des tâches sont visibles sur le même schéma simultanément. De plus nous héritons des notations d'arbre de tâches la décomposition hiérarchique qui permet de facilement localiser une partie de la conception.
- Lisibilité des rôles : la lisibilité des rôles représente la facilité avec laquelle la fonction d'un élément de la notation est déduite par les concepteurs. Cette lisibilité est forte dans COMM grâce à la classification des tâches en 8 types représentés graphiquement différemment, à la distinction graphique des tâches modales et à la représentation distincte des rôles métiers et interactifs. Pour approfondir ce critère, il convient de mener une évaluation expérimentale avec des concepteurs (paragraphe 4) pour confirmer le design graphique des éléments de la notation.

Une autre liste de critères complémentaires est proposée dans [Olsen 2007]. La méthode conceptuelle d'évaluation proposée dans [Olsen 2007] concerne les outils logiciels. Elle peut néanmoins s'appliquer à un modèle ou notation d'interaction [Coutrix 2009]. La méthode considère quatre critères que sont la viscosité, l'importance, le pouvoir unificateur et le passage à l'échelle. La viscosité est un critère déjà étudié ci-dessus dans le cadre de la méthode d'inspection de Green [Green 2000]. Nous considérons les trois autres critères :

- Importance : l'importance de la notation est définie dans le contexte d'un STU (Situation, Tâche, Utilisateur) visé : un STU constitue donc un cadre dans lequel la notation peut être évaluée. En effet lors de l'évaluation, il convient de considérer les utilisateurs cibles, pour un ensemble de tâches donné, dans certaines situations. La notation COMM vise un STU clairement défini : un ou des concepteurs qui souhaitent spécifier l'interaction multiutilisateur et multimodale d'un système. La situation (S) est donc la phase de spécification dans le cycle de vie du logiciel, la tâche (T) est la spécification de la solution conçue et les utilisateurs (U) sont les concepteurs du système. Dans le contexte de cet STU donné, l'importance de la notation se justifie par le fait qu'aucune notation existante ne permet de spécifier complètement et de façon intégrée l'interaction multiutilisateur et multimodale (Chapitre 2). Or nous avons souligné au Chapitre 1 que les systèmes

multiutilisateurs et multimodaux sont de plus en plus nombreux, et ce constat est dû à la multiplicité des modalités d'interaction disponibles aujourd'hui (comme les tables interactives multi-points). Le problème traité par la notation (tâche du STU) est donc important. Nous complétons cette évaluation conceptuelle de l'importance de la notation au paragraphe 3, en montrant expérimentalement que la notation COMM permet de spécifier l'étude de cas WCCM, utilisée aussi pour étudier les notations existantes au Chapitre 2.

- Pouvoir unificateur : le pouvoir unificateur désigne la simplicité d'interconnexion entre les approches existantes. Nous soulignons ici que la notation COMM est une extension de la notation d'arbre de tâches CTT. De plus les extensions proposées dans COMM permettent de spécifier l'interaction multiutilisateur et multimodale contrairement aux notations existantes. Nous soulignons ce point expérimentalement à la section 3 en considérant la spécification de l'étude de cas WCCM.
- Passage à l'échelle : le passage à l'échelle désigne la capacité de l'approche à être utilisée pour des problèmes plus larges. Héritée des notations d'arbres de tâches, la forme arborescente des spécifications réalisées avec la notation COMM peut atteindre rapidement des dimensions importantes. Nous avons présenté au Chapitre 3 le mécanisme de Template afin d'encapsuler des sous-arbres et ainsi décomposer la spécification en parties plus petites. De plus, sans exploiter le mécanisme de Template non implémenté dans e-COMM, nous avons étudié expérimentalement le passage à l'échelle en considérant la spécification avec COMM et e-COMM d'un système de grande taille dans le cadre du projet PEA FH/PA (paragraphe 5).

Les critères de Green [Green 2000] et Olsen [Olsen 2007] permettent de valider conceptuellement une notation et d'éviter des erreurs. Elles doivent se compléter par des expériences d'utilisation de la notation (évaluations expérimentales). Les critères énoncés dans ce paragraphe, comme l'importance ou le passage à l'échelle, servent de guide aux évaluations expérimentales que nous avons menées. Les trois paragraphes suivant présentent nos évaluations expérimentales de la notation COMM et de l'éditeur e-COMM.

3 Evaluation expérimentale : spécification de WCCM

Nous présentons la spécification de WCCM (Warcraft Collaboratif, Coopératif et Multimodal) à l'aide de la notation COMM. Il ne s'agit pas ici de réaliser une spécification complète du système, mais d'illustrer les points clefs de la spécification au regard des notations existantes (Chapitre 2). Cette modélisation est faite par les auteurs de la notation COMM en utilisant e-COMM.

Pour plus de lisibilité dans la mise en page, nous avons décomposé l'arbre de tâches COMM de l'étude de cas WCCM verticalement et horizontalement. Ce découpage nous permet d'illustrer chaque point clef de la notation COMM avec une seule figure.

La Figure 141 illustre la racine de l'arbre de tâches COMM de l'application WCCM. Elle comporte une tâche multiutilisateur *Jouer à WCCM* associée à deux rôles métier : *Sergent* et *Chef*. Elle est décomposée en trois sous-tâches concurrentes : les deux tâches multiutilisateurs *Gérer les bâtiments* et *Gérer les troupes* et la tâche individuelle *Gérer les paysans*. Nous détaillons par la suite les tâches

Gérer les troupes et *Gérer les bâtiments* qui correspondent respectivement à des activités coopératives et collaboratives.

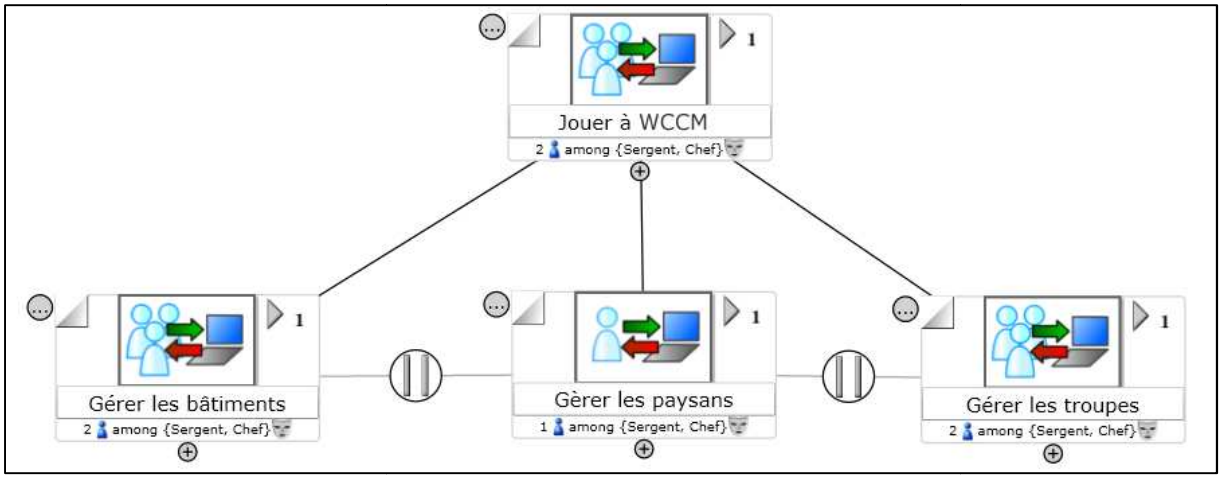


Figure 141 : Tâche Jouer à WCCM (Warcraft Collaboratif, Coopératif et Multimodal).

La tâche de gestion des troupes est illustrée au sein de la Figure 142. Elle est décomposée en trois sous-tâches individuelles : *Déplacer un soldat*, *Former et groupe* et *Déplacer un groupe*. La répartition des tâches entre les utilisateurs sur la base des deux rôles *Sergent* et *Chef* permet d'exprimer la coopération. Les opérateurs indiquent que l'utilisateur jouant le rôle métier de *Sergent* peut soit *Déplacer un soldat* soit *Former un groupe*. La tâche *Déplacer un groupe* peut quant à elle être réalisée en concurrence des deux autres. Nous détaillons par la suite ces trois sous-tâches.

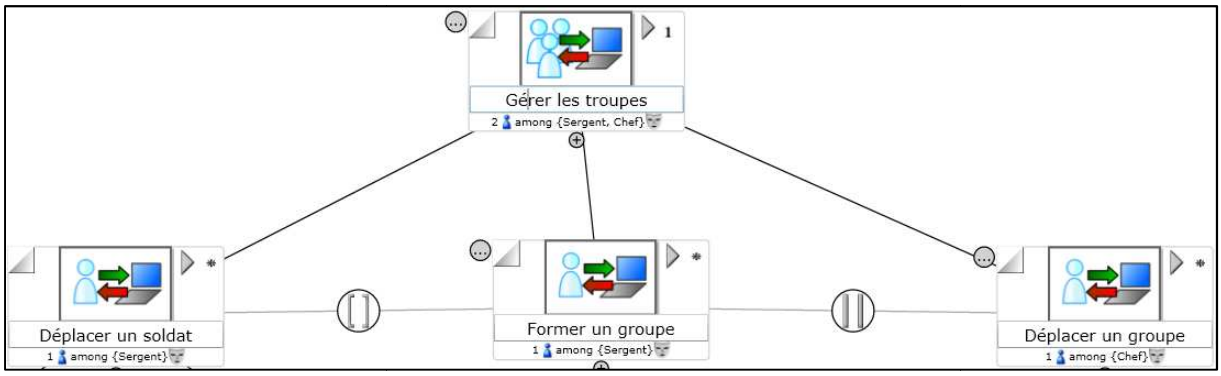


Figure 142 : Tâche Gérer les troupes dans WCCM.

3.1 Interaction individuelle et multimodale

La tâche individuelle qui consiste à déplacer un soldat est modélisée à la Figure 143. Elle est décomposée en deux sous-tâches : *Sélectionner un soldat* et *Déplacer un soldat*. Pour sélectionner un soldat, l'utilisateur jouant le rôle de *Sergent* doit *Pointer du doigt un soldat*. Nous indiquons une réaction du système qui consiste à mettre en relief le soldat sous la forme d'une tâche modale en sortie. Pour déplacer un soldat sélectionné, l'utilisateur doit dire « aller ici » pendant qu'il désigne du doigt un point de la carte, ce que nous avons modélisé par deux tâches modales reliées par un opérateur *A pendant B.* La combinaison de modalités décrit une complémentarité au sens des propriétés CARE [Coutaz 1994].

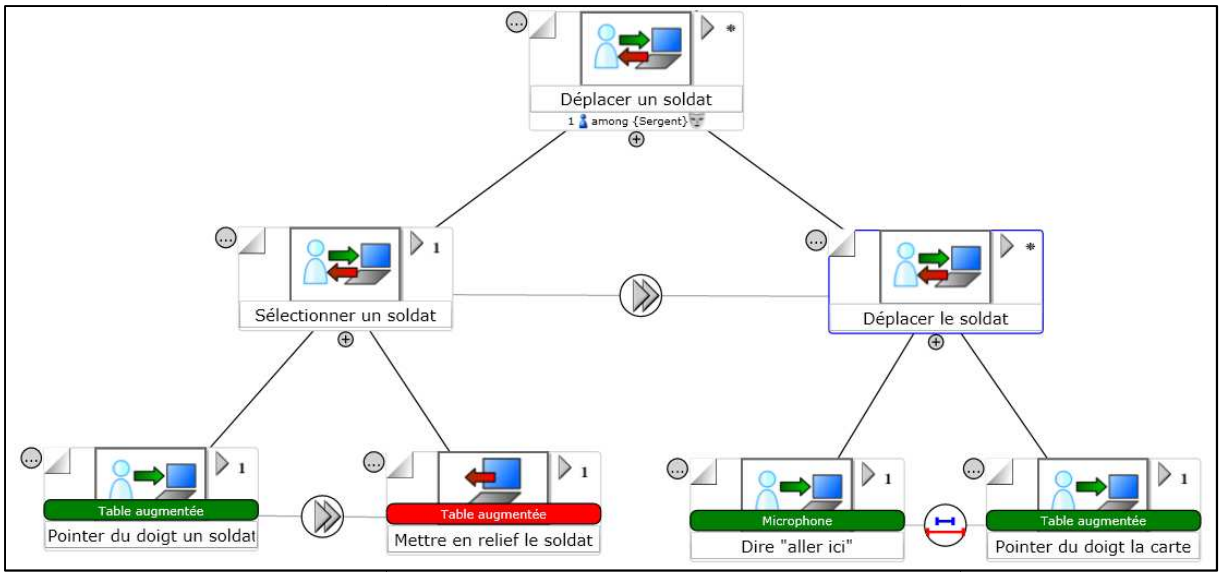


Figure 143 : Tâche Déplacer un soldat dans WCCM.

3.2 Interaction coopérative et multimodale

La Figure 144 présente la modélisation des tâches *Former un groupe* et *Déplacer un groupe*. La tâche *Former un groupe* est décomposée en deux sous-tâches séquentielles : *Sélectionner des soldats* et *Définir comme groupe*. La première est réalisée par une tâche modale qui consiste à *Délimiter une zone de la table avec les mains*, tandis que la deuxième est réalisée par une tâche modale consistant à indiquer oralement le nom du groupe. L'utilisateur jouant le rôle de *Chef* peut déplacer les groupes ainsi créés. Pour cela, il doit d'abord *Sélectionner un groupe*. La sélection d'un groupe peut être réalisée par l'usage de l'une des deux modalités d'interaction décrites par les tâches modales suivantes : *Pointer du doigt un groupe*, *Dire {un nom d'un groupe}*. Il s'agit ici d'une combinaison Equivalente de deux modalités au sens des propriétés CARE [Coutaz 1994]. Ensuite, l'utilisateur jouant le rôle de *Chef* peut déplacer le groupe par une combinaison de deux modalités l'une vocale et l'autre gestuelle représentées par deux tâches modales : l'utilisateur (*Chef*) doit réaliser la tâche modale *Dire « aller ici »*, pendant qu'il pointe du doigt la carte. Il s'agit une nouvelle fois d'une combinaison complémentaire de modalités.

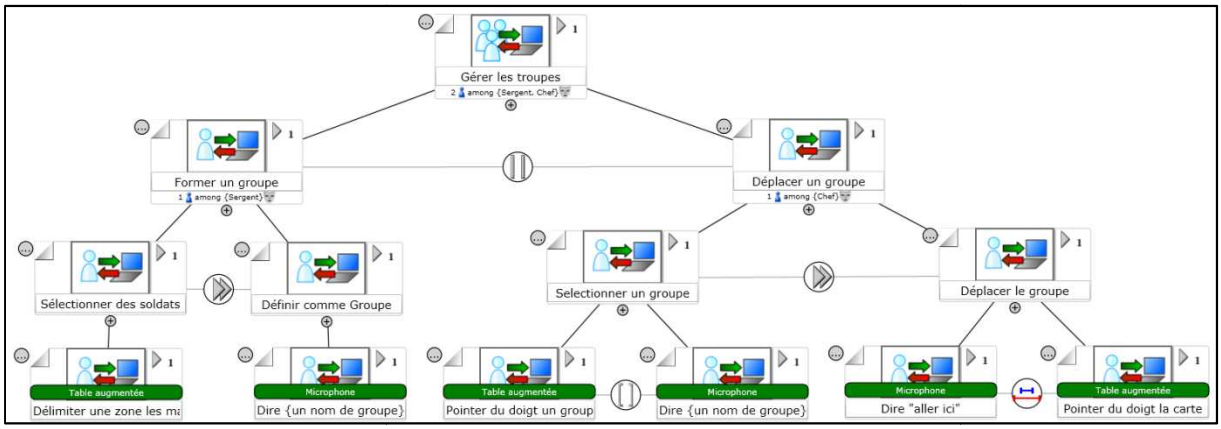


Figure 144 : Tâches Former et Déplacer un groupe dans WCCM.

3.3 Interaction collaborative et multimodale

Nous modélisons la tâche collaborative de gestion des bâtiments. La Figure 145 illustre la spécification de la tâche *Gérer les bâtiments*, introduite à la Figure 141. Elle est décomposée en trois sous-tâches alternatives : la tâche multiutilisateur *Gérer un baraquement* et les deux tâches individuelles *Gérer un hôtel de ville* et *Gérer une Ferme*. Nous avons déjà illustré la spécification d'une tâche individuelle à la Figure 143 pour la gestion des soldats, aussi, nous ne décrivons pas en détail les tâches de gestion d'un hôtel de ville et d'une ferme.

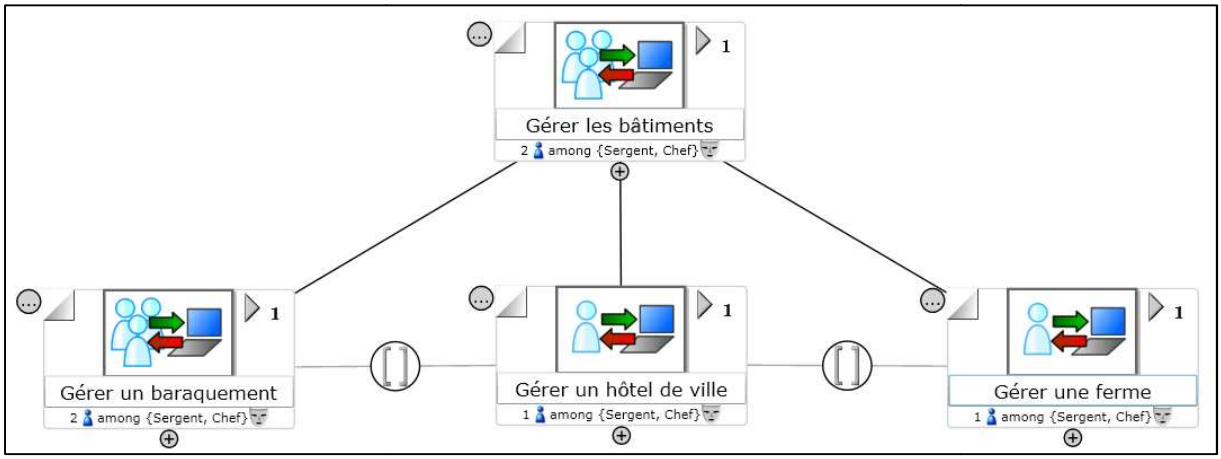


Figure 145 : Tâche Gérer les bâtiments dans WCCM.

La tâche de gestion d'un baraquement qui correspond à une tâche collaborative est détaillée au sein de la Figure 146. Elle est décomposée en sous-tâches séquentielles : *Sélectionner un baraquement*, *Créer un soldat* et *Confirmer la création*. Les tâches *Sélectionner un baraquement* et *Créer un soldat* sont associées à un rôle interactif <Créateur> tandis que la tâche *Confirmer la création* est associée à un rôle interactif <Validateur>. Cela impose que les deux premières sous-tâches soient réalisées par un même utilisateur tandis que la validation est réalisée par un autre utilisateur.

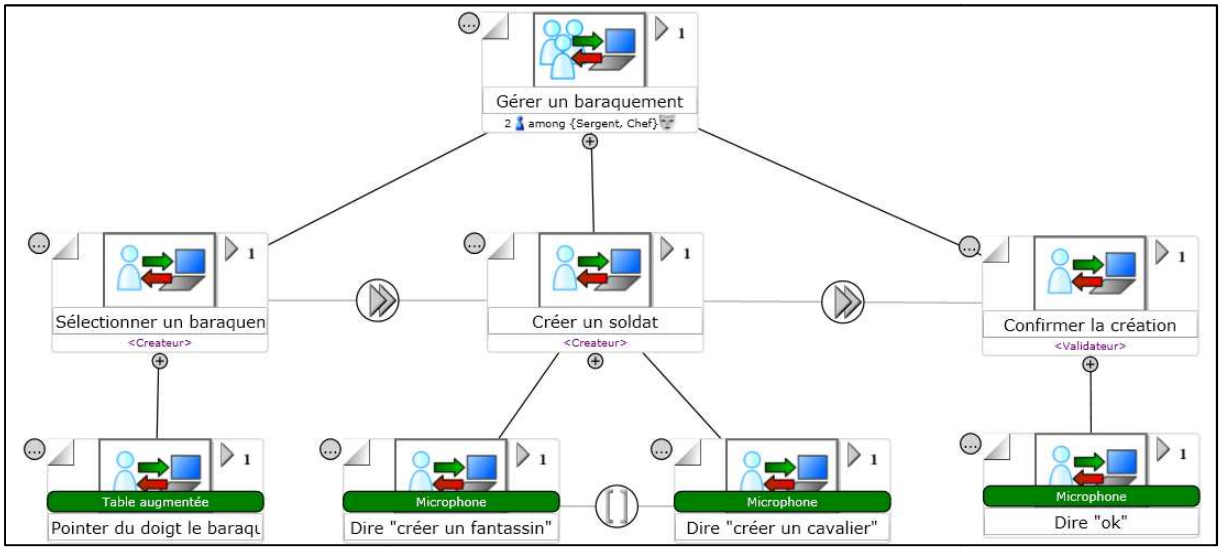


Figure 146 : Tâche Gérer un baraquement dans WCCM.

3.4 Conclusion

L'utilisation de la notation COMM pour la spécification de l'interaction de l'application WCCM souligne son pouvoir d'expression au regard des notations existantes. En effet au Chapitre 2, l'application WCCM a été utilisée pour illustrer le pouvoir d'expression et les limites de chaque notation existante. Cette utilisation de la notation COMM correspond donc au critère d'importance comme défini [Green 2000]. En particulier nous avons présenté la spécification :

- de tâches multiutilisateurs qui relèvent de coopération entre les utilisateurs telles que la tâche *Gérer les troupes*,
- de tâches multiutilisateurs exprimant la collaboration entre les utilisateurs telles que la tâche *Gérer un baraquement*,
- de tâches individuelles telles que la tâche *Déplacer un soldat*.

De plus, la spécification de l'interaction a pu être menée du plus haut niveau d'abstraction par la modélisation de la tâche *Jouer à WCCM*, jusqu'au niveau de l'interaction concrète avec la description de tâches modale telles que *Délimiter une zone avec les mains* ou encore *Dire "ok"*.

Nous avons également décrit différentes formes de combinaisons de modalités d'interaction pour une tâche donnée : par exemple l'équivalence des modalités vocales et gestuelles pour la sélection d'un groupe de soldats et la complémentarité des modalités vocales et gestuelles pour les tâches de déplacement d'un soldat ou d'un groupe de soldats.

Enfin, l'ensemble des spécifications illustrées dans cette section a été réalisé avec l'éditeur e-COMM.

4 Evaluation expérimentale par des concepteurs

Tandis que la spécification précédente a été faite par les auteurs de la notation, nous présentons dans cette section une expérimentation faite avec des concepteurs non experts de la notation. Cette expérimentation a consisté à faire utiliser la notation et son éditeur à un groupe de concepteurs d'IHM pour la spécification d'une application multiutilisateur et multimodale. Nous détaillons d'abord les objectifs de cette expérimentation et le protocole expérimental. Nous présentons ensuite les résultats expérimentaux.

4.1 Objectifs et protocole expérimental

Notre objectif a été de vérifier si les concepts de la notation COMM sont aisés à comprendre et à intégrer. Pour cela, nous avons choisi de faire utiliser la notation COMM et son editeur e-COMM à un groupe d'utilisateurs pour la spécification d'un système qu'ils ont conçu. Nous visions donc des retours qui nous permettent de mesurer la capacité des utilisateurs à comprendre les concepts de la notation COMM et la capacité des utilisateurs à utiliser avec succès la notation COMM.

De plus, cette expérimentation avait aussi comme objectif de faire tester l'éditeur e-COMM afin d'identifier des problèmes d'utilisabilité (apprentissage de l'outil et difficultés d'utilisation) et des fonctionnalités manquantes.

Pour le protocole expérimental, nous avons choisi de faire une expérience de conception/spécification sur plusieurs semaines, sans contrôle des périodes de travail, en nous focalisant sur les artéfacts produits par les sujets. En effet le travail de conception et spécification de l'interaction est une activité longue. Ce constat est encore plus vrai pour les systèmes multiutilisateurs où les interactions doivent être conçues selon le point de vue de chaque utilisateur, et pour les systèmes multimodaux, où l'interaction concrète doit être conçue et finement décrite pour chaque tâche. La longueur de l'activité peut largement dépasser une séance de quelques heures que l'on peut requérir de la part de sujets lors d'une expérimentation contrôlée.

Les 16 participants à cette expérience sont les étudiants de Master 2 Professionnelle spécialité Génie Informatique. Les participants ont entre 22 et 31 ans. L'ensemble de ces étudiants a suivi dans sa scolarité universitaire une formation en Interaction Homme-Machine au cours de laquelle la conception centrée utilisateur et les arbres de tâches ont été abordé. Les étudiants ont participé à l'expérience d'utilisation de la notation COMM et de l'éditeur e-COMM dans le cadre de l'unité d'enseignement sur les collecticiels (36h). Dans ce contexte, l'expérience incluait un cours et des travaux dirigés pour un total de 15h en présence des étudiants. L'organisation de l'expérience a été la suivante :

Une séance préparatoire a été proposée aux étudiants, pendant laquelle :

- Nous avons effectué un rappel des concepts de l'analyse de la tâche et de la modélisation avec un arbre de tâches.
- Nous avons introduit la notation COMM sous la forme d'un cours d'une heure pendant lequel l'ensemble des concepts de la notation (à l'exception du mécanisme de template) a été détaillé et illustré avec l'application WCCM.
- Nous avons ensuite présenté aux étudiants le cahier des charges de l'application à concevoir et avons répondu à leurs questions pendant une heure.
- Enfin, nous avons présenté les conditions de travail que nous leur imposons.

Nous avons imposé aux étudiants de réaliser le travail de conception en binôme dont la constitution a été laissée libre. Lors des 15h en présence des étudiants, nous avons répondu aux questions. Par contre le temps de travail total de conception et de spécification n'a pas été contrôlé. Les étudiants ont ainsi travaillé aussi bien pendant/qu'en dehors des heures de cours/td.

L'application à concevoir est un outil collaboratif de prototypage de posters ou d'affiches pour une société fictive d'édition de posters. L'outil à concevoir est utilisé dans le cadre de réunions de création (2 – 6 personnes) impliquant collaborateurs et clients. Au cours de ces réunions, les collaborateurs montrent des maquettes d'affiche ou de poster pour obtenir des réactions de leurs clients. Ils doivent pouvoir modifier les maquettes d'affiche au cours des réunions dans le but de répondre aux besoins de leurs clients en temps réel. Sur une maquette présentée à tous les participants, les clients ou les collaborateurs modifient la structure, la mise en forme et le contenu de la maquette. Les modifications doivent être visibles de tous et la maquette doit pouvoir être sauvegardée dans différentes versions successives afin de permettre des comparaisons. Enfin nous n'avons pas fixé de contrainte en termes d'interaction à concevoir : toutefois, nous avons encouragé l'exploration de modalités d'interaction autres que la souris et le clavier.

Enfin nous avons imposé de décrire l'interaction du système avec e-COMM. Cependant, il était possible d'utiliser des modélisations complémentaires, telles que des maquettes ou encore des diagrammes UML.

A l'issue de la période de conception, chaque binôme a exposé son travail au cours d'une présentation orale de 20 minutes. Nous avons demandé aux étudiants d'intégrer dans leurs présentations les spécifications COMM produites afin de vérifier la bonne utilisation de la notation. De plus, chaque étudiant a rempli un questionnaire composé de deux parties (Annexe 3) : la première partie est composée de huit questions ouvertes ayant pour but de recueillir des commentaires sur COMM et e-COMM. La deuxième partie consiste à exprimer l'adhésion à un ensemble de sept propositions standardisées SUS (System Usability Scale) sur l'utilisabilité de l'éditeur e-COMM.

4.2 Résultats

Nous détaillons ici les résultats de l'expérience. Nous présentons tout d'abord les résultats de l'utilisation de la notation COMM : la qualité des spécifications COMM produites. Ensuite, nous exposons l'analyse des réponses du questionnaire puis les résultats de l'évaluation SUS.

4.2.1 Qualité des spécifications

La majorité des spécifications COMM produites sont conformes à la notation COMM.

L'interaction abstraite multiutilisateur a été décrite par tous les groupes avec succès, aussi bien en termes de coopération que de collaboration.

La coopération a été décrite en utilisant le concept de tâche multiutilisateur et des rôles métiers qui sont le plus souvent : un rôle de collaborateur de la société d'édition et un rôle de client. Des tâches spécifiques sont réservées aux collaborateurs, telles que l'ouverture ou la suppression d'une maquette.

La description de la collaboration est décrite le plus souvent sous la forme d'une tâche multiutilisateur associée à plusieurs collaborateurs et décomposées en plusieurs tâches individuelles réalisables par un seul des collaborateurs à la fois. En revanche, aucun binôme n'a décrit de rôles interactifs pour préciser la collaboration entre des utilisateurs.

L'interaction concrète multimodale a été spécifiée avec succès par seulement cinq des huit groupes. Ceux-ci ont utilisé des tâches modales pour décrire la manière de réaliser des tâches élémentaires. Par exemple, une tâche *Supprimer un élément* a été décrite par une tâche modale : *Cliquer sur le bouton « supprimer un élément »* (le dispositif utilisé étant la souris). D'autres binômes ont proposé l'usage d'une table tactile pour l'interaction, et ont ainsi décrit des tâches modales telles que *Pointer du doigt un objet* pour réaliser une tâche élémentaire de sélection d'un objet de l'affiche.

Plusieurs spécifications d'interaction concrète proposent des combinaisons équivalentes de modalités d'interaction. En revanche aucune combinaison complémentaire de modalités n'a été décrite. Aussi, peu de spécification utilisent les opérateurs issus des relations temporelles de Allen.

Enfin, les erreurs les plus fréquentes dans les spécifications concernent soit l'opérateur unaire d'itération, soit le typage de tâches. Dans le premier cas, il s'agit le plus souvent d'un oubli de

spécification d'une valeur (ce qui laisse la valeur par défaut dans l'éditeur). Dans le deuxième cas, la convention de typage des tâches n'est tout simplement pas respectée.

4.2.2 Questionnaires

Le Tableau 18 présente une synthèse des réponses aux questions posées à la fin de la phase de conception/spécification avec COMM. Le questionnaire complet est fourni en Annexe 3.

La plupart des sujets (13/16) ont déjà utilisé une notation de spécification. Il s'agit le plus souvent de diagrammes UML (cas d'utilisation et diagramme de classes) mais également de notations d'arbre de tâche telles que CTT. Seuls trois sujets déclarent ne pas avoir utilisé d'autre notation de spécification avant l'expérience.

Les 13 sujets ayant déjà utilisé une notation de spécification, répondent avoir déjà utilisé trois catégories d'éditeur. Il s'agit soit d'éditeurs UML (ArgoUML ou StarUML lorsqu'ils sont nommés), de l'éditeur d'arbre de tâches CTTE, et enfin de PowerPoint. Les reproches faits à ces systèmes sont principalement la difficulté d'utilisation pour les éditeurs UML et pour CTTE qui est de plus jugé peu intuitif, et l'inadéquation à l'activité de spécification pour PowerPoint.

Pour les trois points forts de la notation COMM, les sujets ont répondu : la prise en compte de l'aspect multiutilisateur, la prise en compte des dispositifs physiques d'interaction et la facilité de démarrage avec la notation.

Le reproche principal concerne la pauvreté de l'éditeur e-COMM en termes de fonctionnalités de base, telles que le copier-coller ou faire-défaire. De plus une critique faite sur la notation COMM concerne le très grand nombre d'opérateurs possibles entre les tâches (opérateurs CTT + opérateurs Allen). Cette richesse complexifie l'apprentissage de la notation.

La faiblesse de l'éditeur e-COMM est cependant nuancée par la suite car un même nombre de sujets déclare souhaiter un éditeur présentant le même type d'interaction pour des notations telles qu'UML.

Plus de la moitié des sujets déclarent souhaiter réutiliser l'éditeur e-COMM pour spécifier de futurs projets. Tous fixent cependant la condition que celui-ci soit amélioré. Trois sujets déclarent ne pas souhaiter réutiliser e-COMM et deux parmi ces sujets soulignent le fait que les notations d'arbres de tâches ne sont pas utilisées dans l'industrie.

L'idée de sous-arbres réutilisables emporte un soutien massif des sujets, ce qui est confirme l'importance d'étendre e-COMM avec le mécanisme de template décrit au Chapitre 3.

Enfin, les utilisateurs ont proposé plusieurs améliorations à apporter à l'éditeur e-COMM. Ces améliorations concernent principalement l'ajout de fonctionnalités tels que le copier-coller et le défaire-refaire, mais également des améliorations de la navigation et du repérage dans la vue à l'aide de barre de défilement, ou l'ajout d'un panneau pour le zoom comme dans les systèmes de navigation dans une carte.

Questions	Réponses
Avez-vous déjà utilisé des notations de spécification? Si oui, lesquels ? Si oui, quelles était les principales difficultés?	13 sur 16 ont répondu oui. Les notations utilisées sont des diagrammes UML (8/11) ou des notations d'arbre de tâche (5/11)
Avez-vous déjà utilisé des éditeurs pour spécifier un système? Si oui lesquels? Si oui, quelles étaient les principales difficultés rencontrées ?	13 sur 16 ont répondu oui. Editeur UML (5/13) : pas facile d'utilisation, Power point (4/13) : pas fait pour, CTTE (4/13) : pas intuitif.
Enoncer 3 points forts de la notation COMM	aspects multiutilisateur (8), gestion dispositifs (5), facile (4)
Enoncer 3 points faibles de la notation COMM	PB avec l'éditeur (8/16) : manque de fonctionnalités de base : le copier-coller, défaire-refaire, panneau de zoom, barre de défilement, suppression. Trop d'opérateurs différents (2/16).
Est-ce que vous aimeriez utiliser ce type d'éditeur avec ce genre d'interaction pour une autre notation de spécification (UML, etc)? Pourquoi?	Oui (8/16) : car plus rapide ou plus simple que les éditeurs UML Non (2/16) : car pas pratique.
Est-ce que vous réutiliseriez l'éditeur e-COMM pour vos projets futurs ? Si possible argumenter pourquoi et pour quels types de systèmes (collecticiel, mobiles, multimodalité, système graphique classique de type écran-clavier-souris ?)	Oui (10/16), mais il faut améliorer l'éditeur Non (3/16)
Pensez-vous pertinent de pouvoir sauvegarder des sous-arbres pour les réutiliser dans un autre arbre ou un autre emplacement dans le même arbre ?	Oui (12/16)
Suggestions d'améliorations de l'éditeur ou de la notation ?	Défaire/Refaire (8/16), copier/coller (6/16), améliorer la navigation (5/16), barre de défilement (2/16), panneau de zoom (2/16)

Tableau 18 : Expérimentation : réponses aux questions ouvertes.

4.2.3 Score d'utilisabilité SUS

Le score d'utilisabilité obtenu fournit un indicateur de l'utilisabilité globale d'e-COMM. Le questionnaire SUS permet uniquement de détecter des défauts d'utilisabilité mais ne les diagnostique pas.

Les scores obtenus s'étalent entre 55 et 77,5. Ceux-ci sont plutôt regroupés dans un intervalle de 22,5 point. La moyenne des scores est quant à elle d'environ 66,5 c'est-à-dire au centre de l'intervalle [55 ; 77,5]. La répartition des scores et leurs moyennes indiquent que le résultat est plutôt fiable.

L'interprétation du score [Bangor 2008] nous indique qu'e-COMM est jugé « plutôt utilisable » par les sujets. De plus, ces mesures seront aussi utilisées lors d'une prochaine évaluation d'e-COMM après améliorations, afin de mesurer l'évolution de son utilisabilité globale.

4.3 Conclusion

De cette expérimentation avec 16 sujets, nous concluons que la notation COMM a été bien appréhendée par les sujets et semble globalement comprise aussi bien pour la description de

l'interaction abstraite multiutilisateur (coopérative et collaborative) que pour l'interaction concrète multimodale. Toutefois, le concept de rôle interactif n'a pas été utilisé lors de la description d'interaction collaborative, et aucune combinaison complémentaire de modalités n'a été décrite. Les opérateurs issus des relations temporelles de Allen ont ainsi été très peu utilisés.

Les sujets ont souligné l'intérêt d'un éditeur facile à utiliser. Plusieurs défauts majeurs d'utilisabilité ont été notés, notamment l'absence de fonctionnalités de base telles que le copier-coller ou le défaire-refaire. Ils ont également souhaité que la navigation dans l'arbre soit simplifiée, principalement en termes de gestion de la vue et de gestion du zoom. Aucun sujet n'a en revanche exprimé le besoin d'une mini-vue. La version testée de l'éditeur e-COMM a donc été améliorée en tenant compte de ces retours : les outils de navigation directionnelle et la réglette de manipulation du facteur de zoom présentés dans le Chapitre 4 sont des améliorations consécutives à cette expérimentation.

5 Evaluation expérimentale dans le cadre du projet PEA FH/PA

La notation COMM et l'éditeur e-COMM ont été utilisés et mis au point de façon itérative dans le cadre du Programme d'Etude Amont (PEA) de la Direction Générale des Armements (DGA) noté Facteurs Humains et Partage d'Autorité (FH/PA). Clos en février 2011, le projet avait pour objectif d'améliorer les postes de commande de drones militaires et notamment leurs interfaces utilisateurs. Dans ce cadre, de nombreuses solutions d'interaction ont été conçues, et certaines implémentées au sein de deux prototypes, un par itération. Les solutions d'interaction conçues dans ce projet, ainsi que l'interaction des deux prototypes développés ont été spécifiés avec la notation COMM et l'éditeur e-COMM. Ce projet a permis de mesurer la capacité de la notation à passer à l'échelle (un des critères de la méthode d'évaluation décrite dans [Olsen 2007] et présentée au paragraphe 2), en utilisant la notation dans un projet en vraie grandeur.

Nous présentons d'abord le projet et ses étapes, puis nous exposons les spécifications produites avec la notation COMM au cours de ce projet.

5.1 Projet PEA FH/PA

Pour présenter le projet PEA FH/PA, nous décrivons ses intervenants, ses objectifs, son déroulement, et un ensemble de concepts clefs du domaine métier des postes de commande de drones.

5.1.1 Intervenants du projet

Le projet regroupe quatre partenaires industriels : Bertin technologies, EADS DS (Défense et Sécurité), Sagem DS (Défense et Sécurité), et PY automation. Bertin porte le projet et apporte des compétences en ergonomie, EADS et Sagem sont les fournisseurs des systèmes de drones actuellement en usage par l'armée française et apportent donc leurs connaissances métier sur les drones et les postes de commande de drones. Enfin PY automation apporte ses compétences en simulation avec le simulateur S4D, et a en charge le développement des prototypes pour leur usage dans le cadre du simulateur.

Le Laboratoire d'Informatique de Grenoble (LIG) et l'Ecole Nationale Supérieure des Télécommunications (ENSTB) sont associés au projet dans le cadre de contrats de sous-traitance avec les partenaires industriels (Bertin technologies).

5.1.2 Objectif du projet

L'objectif du projet était d'améliorer les postes de commande de drones militaires, afin d'augmenter l'efficacité des opérateurs qui les utilisent. Pour améliorer les postes de commande, deux axes complémentaires ont été explorés. Le premier axe consiste à prendre en compte les facteurs humains en amont de la réalisation des postes de commande, et notamment des considérations ergonomiques. Le deuxième axe exploré repose sur le concept de partage d'autorité. Le partage d'autorité est une stratégie d'utilisation des ressources que sont les utilisateurs et les agents logiciels d'un système [Leal 2009]. Elle consiste à donner au système la capacité d'allouer dynamiquement les tâches aux utilisateurs ou au système selon le contexte, l'objectif de la tâche, sa satisfaction dans des conditions optimales et dans le respect de la sécurité du matériel, des hommes et de l'environnement. Au cours du projet, la réalisation des prototypes de poste de commande de drones intègre dès la conception le concept de partage d'autorité.

La conception et réalisation de prototypes d'un tel système nécessite d'élaborer de nombreuses spécifications, et notamment des spécifications de l'interaction utilisateur. C'est dans ce cadre de la conception et de la spécification de l'interaction utilisateur que se positionne la contribution du LIG au projet. Le LIG a participé à la conception de l'interaction des prototypes de postes de commande de drones, a fourni une notation de spécification de l'interaction, et a participé à la réalisation et à la validation des spécifications.

5.1.3 Déroulement du projet

L'approche adoptée dans le projet a été itérative et le projet a comporté trois étapes principales.

1. La première étape a consisté en une étude approfondie du problème et des besoins. Cette étude a reposé sur une analyse des systèmes existants : synthèse des problèmes d'utilisation réelle des drones et des postes de commande actuels et analyse des origines des problèmes répertoriés.
2. La deuxième étape est la réalisation d'un prototype de poste de commande de drones. Nous avons mené cette étape comme un projet à part entière. Les différents intervenants du projet ont conçu le poste de commande de drones et son interface utilisateur, notamment dans le cadre de séances de créativité orchestrées par Bertin. Les résultats de ces séances incluaient les solutions d'interaction, qui ont été ensuite spécifiées avec COMM. Sur la base de ces spécifications, une première version d'un prototype a été implémentée. Il s'agit d'un poste de commande de drones pour deux utilisateurs. Ce prototype a ensuite été testé en simulation par des opérateurs de l'armée.
3. La troisième étape du projet a consisté à affiner le premier prototype réalisé lors de la deuxième étape. A partir des résultats des expériences utilisateurs et de besoins non couverts par le premier prototype, nous avons conçu et spécifié un deuxième prototype de poste de commande de drones, c'est ce second prototype que nous illustrons dans la suite. Il considère notamment des interactions plus élaboré, tels que le partage d'un drone entre

plusieurs stations. Ce prototype a fait l'objet d'une seconde campagne d'expérimentation et testé en simulation par des opérateurs de l'armée.

A l'issu de la réalisation et évaluation de ces deux prototypes, des conclusions sur les solutions proposées et un ensemble de recommandations ont été émis pour la réalisation de futurs postes de commande de drones. Ces recommandations pourront alors être exploitées par les partenaires industriels EADS et SAGEM pour améliorer les futures versions de leurs postes de commande de drones.

5.1.4 Poste de commande de drones : terminologie

Nous présentons des définitions des concepts clefs du domaine métier des drones et des postes de commande de drones.

Vecteur : Un vecteur désigne l'un des drones dirigés par les opérateurs du poste de commande. Dans la suite, nous utilisons indifféremment les termes de vecteurs et de drones.

Charge utile : La charge utile désigne l'ensemble des équipements embarqués sur un drone. Pour les drones français, il s'agit d'équipements d'observation comme une caméra.

Point de passage (*waypoint*) : Il s'agit d'un point de passage prédéfini pour un vecteur. Ce point est défini en latitude, longitude et altitude. Dans la suite, nous préférons le terme de point de passage à son anglicisme waypoint sauf au sein des arbres de tâches où la version anglaise est plus courte.

Plan de vol : Un plan de vol est une route aérienne définie par un ensemble de points de passage et de chemins entre ces *points*. Un vecteur suit un plan de vol s'il tend à passer par les points de passage définis par le plan de vol.

Trajectoire : Une trajectoire désigne l'ensemble des points par lesquels passe un vecteur. L'historique d'une trajectoire est l'ensemble des points par lequel un vecteur est passé, tandis qu'une trajectoire prévisionnelle est un ensemble des points par lequel le vecteur devrait passer. Pour suivre un plan de vol, un drone doit adopter une trajectoire qui passe par les points de passage du plan de vol.

5.2 Spécification du second prototype

Nous présentons une partie de la spécification du second prototype avec la notation COMM et l'éditeur e-COMM. La spécification produite est un arbre de tâches de 12 niveaux de profondeur, et comportant un peu plus de 180 tâches. Cette exploitation de la notation COMM dans le projet PEA FH/PA montre donc sa capacité de passage à l'échelle. Au vue de la taille de la spécification obtenue, nous ne la présentons totalement. Nous montrons des arbres de tâches qui illustrent les différents concepts de la notation COMM. Néanmoins l'objectif étant d'illustrer le passage à l'échelle, il nous a semblé utile de détailler de nombreux arbres de tâches abstraites et concrètes.

La spécification du second prototype avec la notation COMM a été réalisée par l'enrichissement de la spécification du premier prototype. Cette première version des spécifications (non présentée dans ce mémoire) s'est déroulée sur une période de plusieurs mois pendant lesquels la notation a évolué. En effet la notation COMM a subi plusieurs évolutions pendant cette même période avant d'aboutir à sa forme définitive présentée au Chapitre 3. Ainsi, la notation finale n'a pas été disponible tout au

long du processus de spécification, mais uniquement lors du dernier raffinement de ces spécifications. De plus, l'éditeur e-COMM présenté au Chapitre 4, a été réalisé a posteriori de la réalisation des spécifications du premier prototype. Celles-ci ont donc d'abord été décrites informatiquement avec PowerPoint, puis transcrites avec l'éditeur e-COMM, dès qu'il a été disponible. Ce sont ces spécifications décrites avec e-COMM pour le premier prototype et enrichi pour le second prototype qui sont présentées dans la suite de cette section.

5.2.1 Prototype : objectif global, scénario et rôles

Le prototype vise à faire travailler ensemble deux opérateurs pour la conduite de missions d'observation durant lesquelles deux drones sont disponibles. Comme sur les versions actuelles de postes de commande, les deux opérateurs sont co-localisés et disposent chacun d'une console de travail. Le poste de commande ou son environnement pouvant être bruyant, les opérateurs communiquent entre eux grâce à un casque et microphone. Cet équipement permet également la communication avec l'extérieur, notamment pour transmettre des observations ou pour recevoir des ordres. Dans le cadre de la réalisation d'une version simulable supportée par le simulateur, les consoles de chaque opérateur sont composées de deux écrans larges, d'un clavier et d'une souris.

En amont de la conception du prototype du poste de commande de drones, nous avons défini un scénario de base sous la forme d'une mission que doivent remplir les opérateurs. Il s'agit d'une mission de surveillance d'un convoi par deux drones, illustrée à la Figure 147. Au cours de cette mission, de nombreux événements sont provoqués afin de rendre difficile la gestion de la trajectoire des drones par l'opérateur. Nous avons défini les fonctionnalités du prototype dans le but de couvrir à minima ce scénario, car il s'agit du scénario joué lors des tests utilisateurs avec des opérateurs de l'armée.

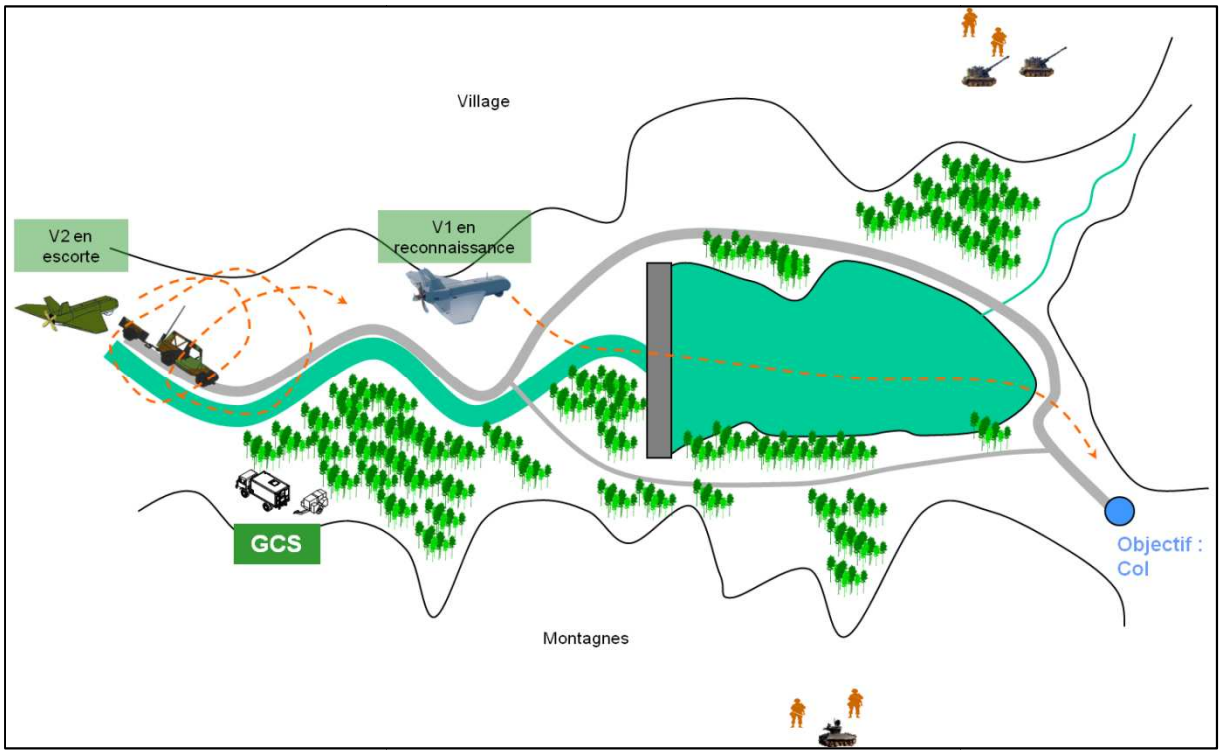


Figure 147 : Mission de surveillance d'un convoi.

Nous avons défini deux rôles métiers : un rôle d'opérateur Vecteur et un rôle d'opérateur Charge Utile.

L'opérateur Vecteur (VE) contrôle un ou plusieurs drones (appelés vecteurs). Il doit diriger les drones selon des plans de vol définis pour une mission donnée. Il peut également contrôler finement les drones, par exemple leurs orientations, leurs altitudes, ou encore leurs vitesses, et ainsi définir une trajectoire particulière.

L'opérateur Charge Utile (CU) a la responsabilité de l'utilisation des équipements d'observation et de surveillances embarqués à bord du drone. Par exemple, lorsqu'une caméra rotative est embarquée sur un drone, l'opérateur CU peut modifier l'orientation ou le facteur de zoom de cette caméra pour réaliser la mission d'observation.

Les opérateurs CU et VE travaillent ensemble pour le succès d'une mission, qui consiste à généralement à observer ou surveiller une ou plusieurs zones. L'opérateur VE est responsable des vecteurs pour permette à l'opérateur CU de réaliser la mission d'observation.

Pendant cette mission, de nombreux événements peuvent intervenir, tels qu'une demande de mission plus prioritaire d'observation d'une zone pour identifier un véhicule lance missile. L'un de ces événements consiste à mettre en relation la station effectuant la mission de suivi du convoi avec une autre station. Cette deuxième a un objectif prioritaire qui requiert pour sa réalisation que la première station lui cède le contrôle de la caméra d'un de ses deux drones. Les événements simulés correspondent soit à des situations réelles s'étant reproduites dans le contexte de la simulation, soit à des anticipations de besoins recueillies par Bertin auprès de l'armée.

5.2.2 Prototype : partage d'autorité et définition de contextes

Dans le cadre de la prise en charge du concept de partage d'autorité pour la réalisation du prototype de poste de commande de drones, nous avons défini des modes d'utilisation des systèmes notés modes opératoires. Les différents modes opératoires sont présentés en détail dans [Leal 2009] et correspondent à des niveaux d'automatisation du système. Trois modes opératoires ont été retenus (A1, A2 et A3) comme pertinents pour le scénario de mission d'observation et notamment pour la gestion de trajectoires.

Le mode A1 correspond au niveau d'automatisation le plus élevé. Le système agit de manière autonome. Dans ce mode, l'utilisateur peut avoir conscience de la situation, mais les décisions prises par le système ne sont pas expliquées et sont simplement exécutées. Toutefois, l'utilisateur conserve une capacité d'intervention : par exemple, dans ce mode, l'opérateur VE peut choisir de dérouter un drone sur un autre plan de vol.

Le mode A2 est un mode d'automatisation intermédiaire. Le système calcule et présente plusieurs possibilités à l'utilisateur dont celle choisie par le système. L'utilisateur dispose alors d'un certain temps pour choisir une des alternatives proposées. Dans le cas d'une dérive par rapport au plan de vol, le système propose par exemple 3 trajectoires dont une qu'il a choisie de suivre au cas où l'utilisateur n'intervient pas. L'utilisateur a néanmoins la possibilité de choisir une trajectoire parmi les trois proposées.

Enfin le mode A3 est le mode le moins automatisé. L'utilisateur dispose de nombreuses possibilités pour définir la trajectoire du vecteur, et notamment, la possibilité de modifier son orientation, son altitude ou sa roulis par la définition de consignes.

Chacun des trois modes opératoires définit pour les opérateurs l'accès à un ensemble de fonctionnalités, le choix du mode opératoire étant effectué par le système de partage d'autorité. Aussi, chaque mode peut être considéré comme un contexte d'utilisation. C'est pourquoi nous avons spécifié ces trois modes opératoires comme des contextes avec la notation COMM. Ainsi, le mode A1 porte l'étiquette (1), le mode A2 l'étiquette (2) et le mode A3 l'étiquette (3).

5.2.3 Prototype : maquettes graphiques

Chaque opérateur dispose de deux écrans physiques sur lesquels s'affichent les vues liées à son rôle (VE ou CU).

L'opérateur Vecteur dispose d'un système de cartographie illustré à la Figure 148 et d'un système de contrôle de vol illustré à la Figure 149.

Le système de cartographie est décomposé en deux parties : une barre d'informations située sur la gauche et la cartographie. La barre d'informations affiche les drones contrôlés par l'opérateur Vecteur, ainsi que des informations sur les drones sélectionnés. Le système de cartographie est le lieu de nombreuses manipulations. Il permet entre autre de définir des plans de vol.

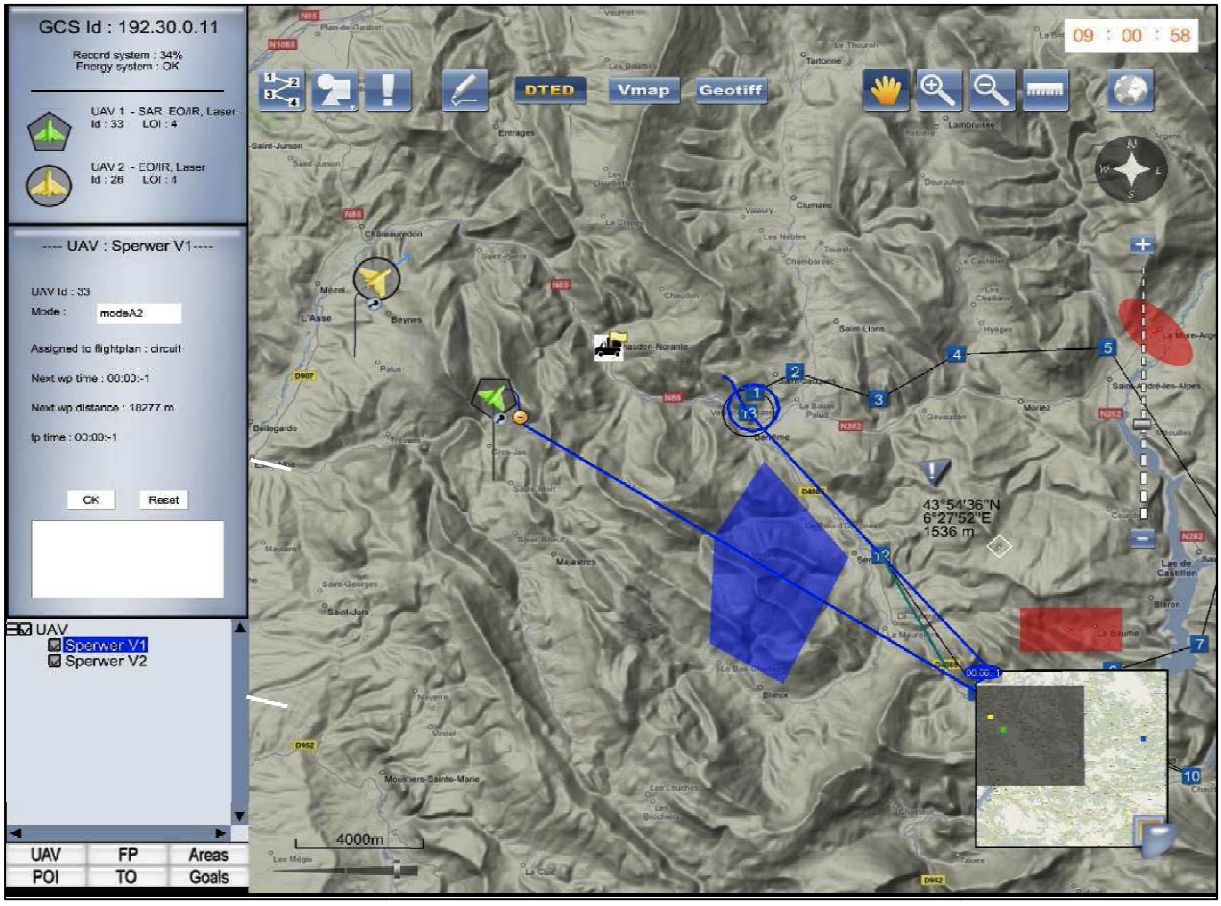


Figure 148 : Maquette : système de cartographie de l'opérateur Vecteur.

L'écran de contrôle de vol est décomposé verticalement en deux parties, une pour chaque drone pouvant être contrôlé par l'opérateur Vecteur. Pour chaque vecteur, cette vue présente des informations sur son état, tels que son orientation, sa vitesse, son altitude, ou encore son niveau d'énergie. Sur la droite de l'écran, est affiché le mode de vol (manuel ou automatique), ainsi que la liste des événements survenus, tels qu'un changement automatique de mode (A1, A2, A3).

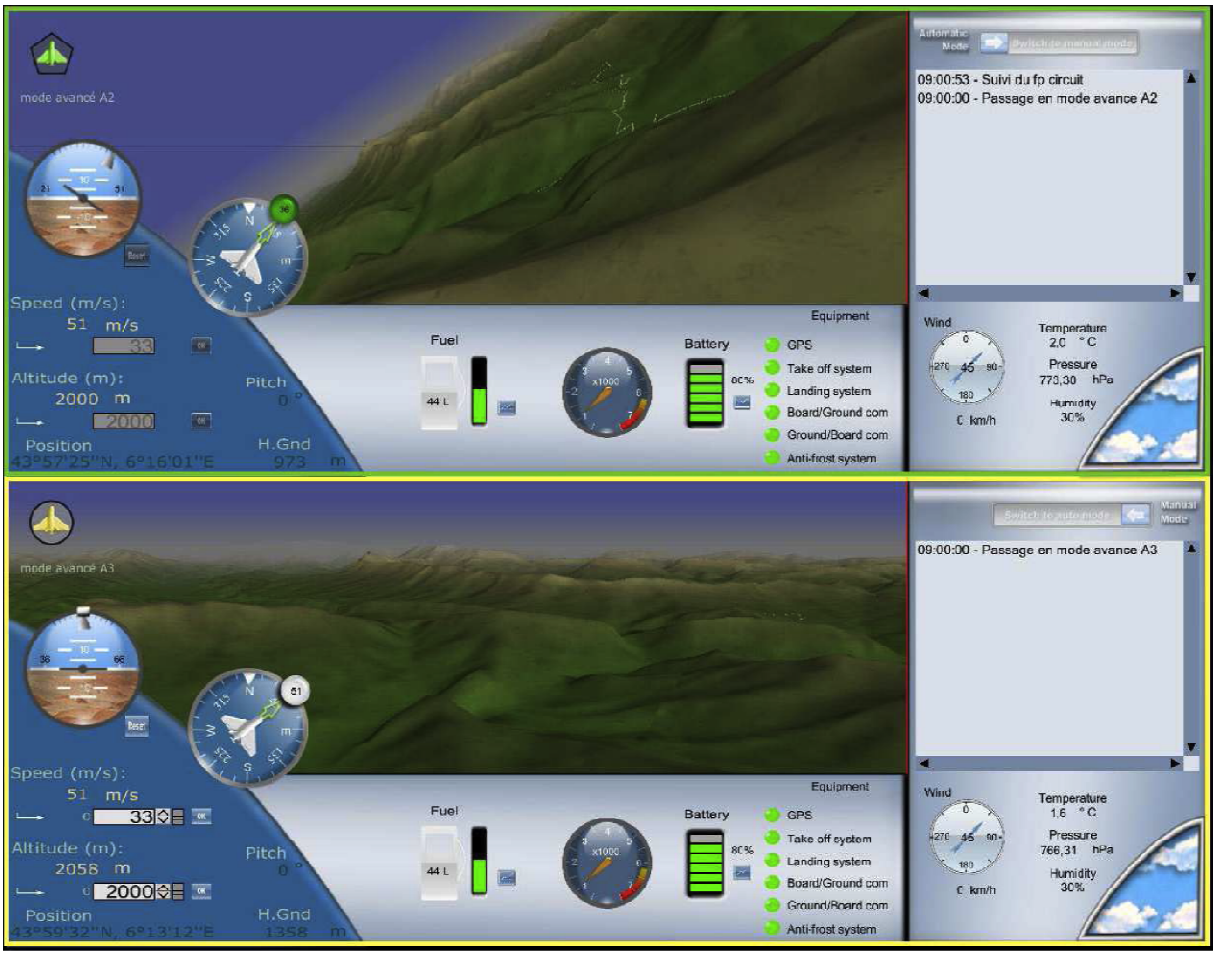


Figure 149 : Maquette : système de contrôle de vol de l'opérateur Vecteur.

L'opérateur Charge Utile dispose également d'un système de cartographie illustré à la Figure 150. Celui-ci permet à l'opérateur d'évaluer la position des drones sur la carte pour réaliser au mieux les observations. Ce système de cartographie ne dispose pas de possibilités interactionnelles de modification du plan de vol des drones. En revanche, il indique la trajectoire de chaque drone afin de permettre l'anticipation de leur position par l'opérateur Charge Utile.

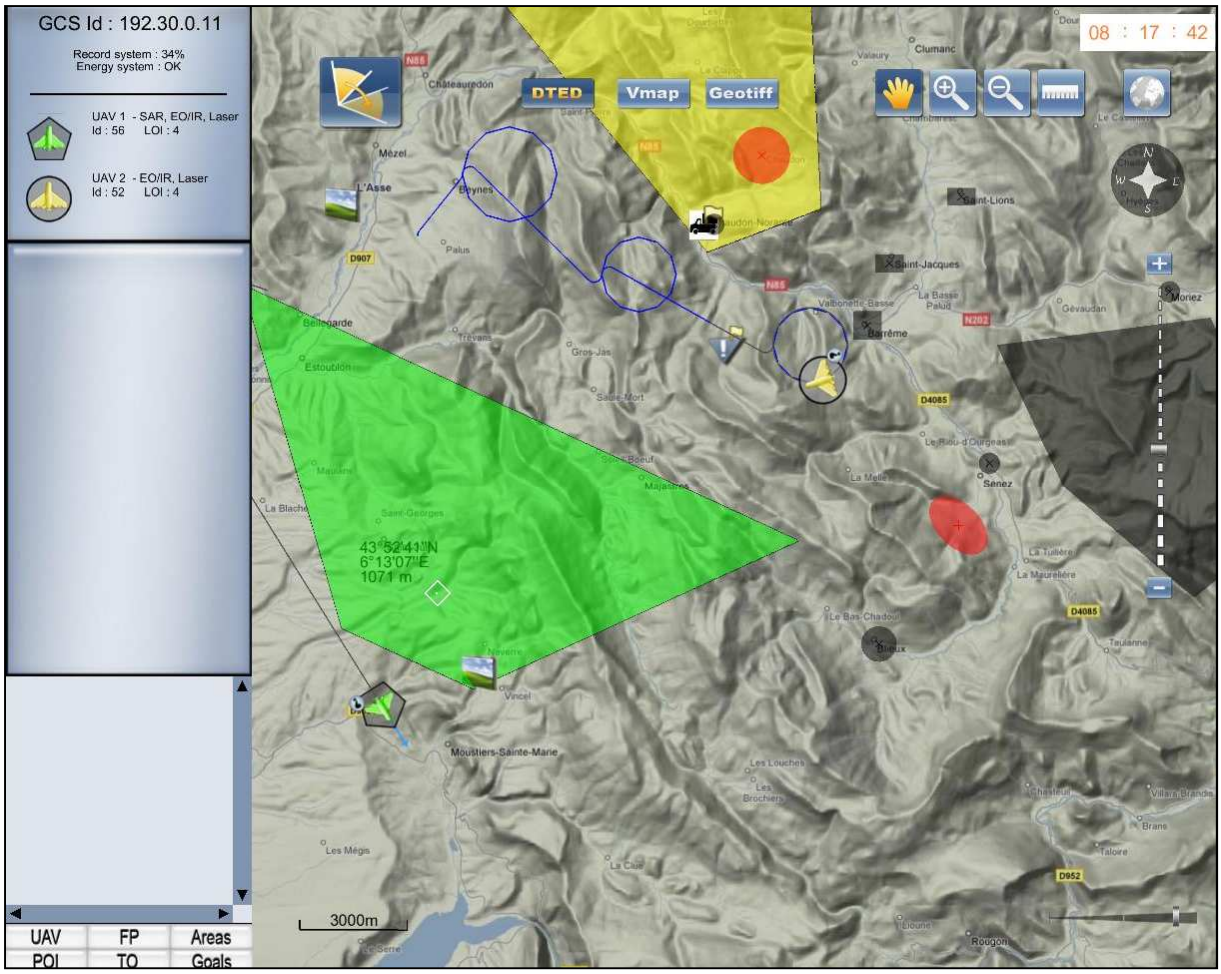


Figure 150 : Maquette : système de cartographie de l'opérateur Charge Utile.

Le deuxième écran de l'opérateur Charge Utile, présenté à la Figure 151, est un système de visualisation en 3D qui simule les équipements d'observation embarqués sur les drones. L'opérateur peut orienter la prise de vue pour balayer une zone ou pour suivre un objet. Cette vue présente également l'orientation du drone (boussole en bas à gauche) et la dernière capture d'écran (en bas à droite). Sachant que deux drones sont contrôlés simultanément, l'écran est divisée en deux parties pour permettre la visualisation simultanée des images par la charge utile de chacun des drones.

Figure 151 : Maquette : système de visualisation 3D de l'opérateur Charge Utile.

5.2.4 Spécification COMM de l'interaction de l'opérateur Vecteur

Nous présentons ici les spécifications de l'interaction proposée par le système pour l'opérateur Vecteur. Ces spécifications ont toutes été réalisées avec l'éditeur e-COMM. Nous avons décomposé les spécifications en plusieurs arbres afin de les rendre plus lisibles sur support papier. Pour la présentation de ces différents arbres, nous suivrons une approche descendante, en présentant d'abord la racine de l'arbre de tâches, puis nous parcourons les différents sous-arbres de tâches.

Ainsi, le premier arbre de tâche (Figure 152) représente la gestion d'un drone tactique (DT). Elle se décompose en deux sous-tâches pouvant être effectuées en parallèle : la gestion du drone tactique (*Contrôler état DT*), et la gestion de la trajectoire d'un drone tactique (*Contrôler trajectoire DT*). Ces deux sous-tâches correspondent aux deux interfaces, déployées sur deux écrans distincts, dont les maquettes graphiques sont présentées dans la section précédente. La première correspond au système de contrôle de vol (Figure 149) et la deuxième au système de cartographie (Figure 148). Les deux tâches sont réalisables pour tous les modes opératoires prévus et introduits précédemment. Dans la suite, les sous-tâches sont décrites dans l'ordre suivant :

- Gérer l'état du drone tactique
  - Indiquer l'état du Drone tactique
  - Modifier les paramètres de vol
- Contrôler la vue trajectoire

La tâche *Changer de mode* permet, en cliquant à la souris sur le bouton glissière correspondant, de basculer du mode manuel au mode automatique et inversement. Enfin, il convient de noter que la tâche permettant de gérer les objets n'est pas décrite car elle se réduit à une interaction classique de type WIMP : clic sur des boutons, navigation dans une arborescence de cases à cocher.

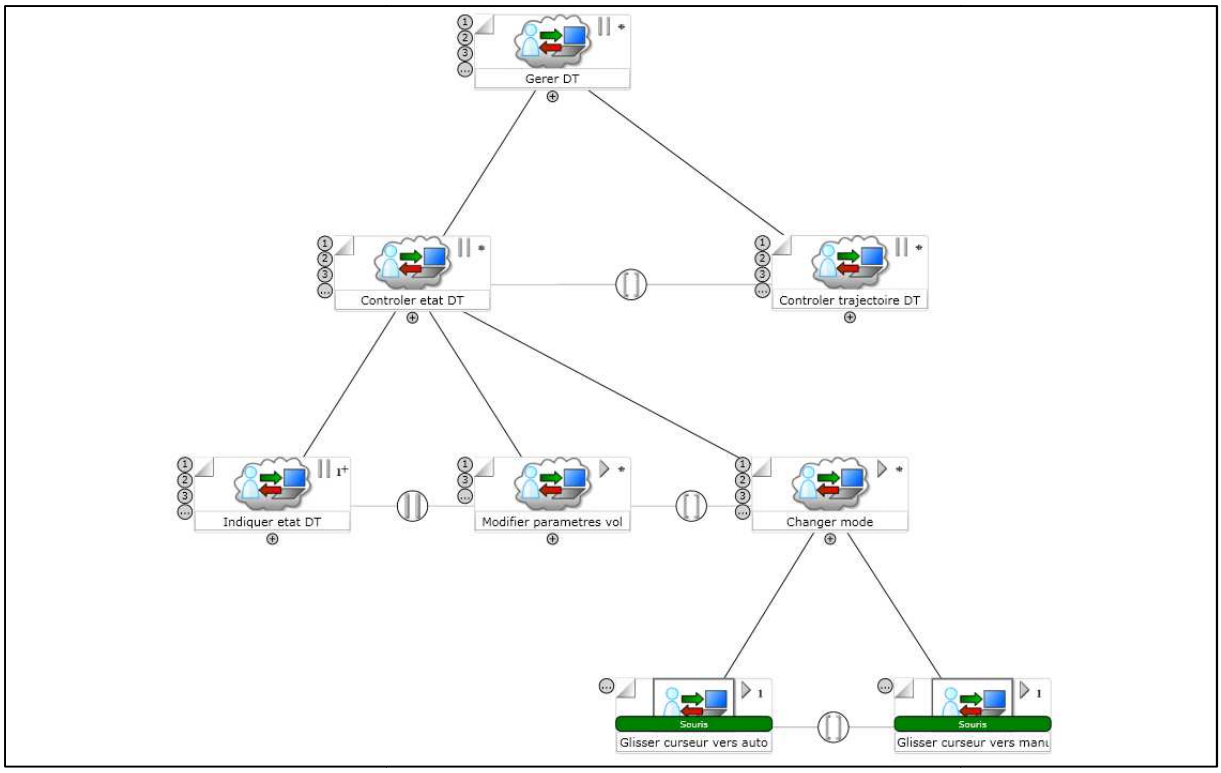


Figure 152 : Arbre de tâches pour la gestion d'un drone tactique (DT).

5.2.4.1 Indiquer l'état du drone tactique

La tâche *Indiquer l'état du drone tactique* est une tâche de présentation d'information sur l'état du drone tactique. La Figure 153 illustre la forme générale de l'arbre de tâches qui décrit cette tâche. Sur celui-ci, nous avons délimité quatre zones colorées pour décomposer l'arbre en plusieurs parties qui sont détaillées par la suite. Les parties rouge, bleue, verte et jaune sont détaillées respectivement à la Figure 154, à la Figure 155, à la Figure 156 et à la Figure 157.

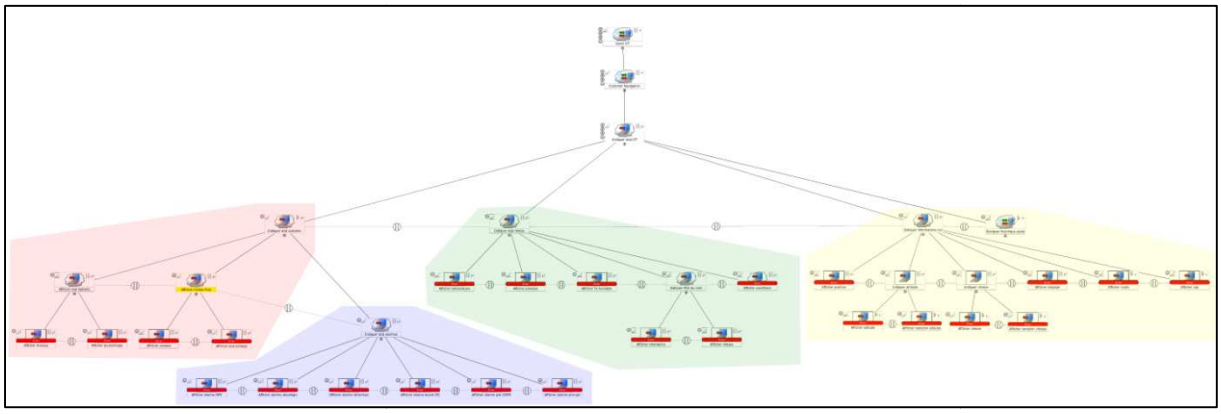


Figure 153 : Tâche : indiquer l'état du drone tactique (Indiquer état DT).

Les informations sur l'état du drone tactique sont présentées sous forme :

- numérique pour : la vitesse, l'altitude, la position, le pitch, la vitesse du vent, la température, la pression, le taux d'humidité et le mode opératoire.

- graphique pour : les niveaux de carburant (symbole de pompe à essence) et de batterie (niveaux de couleur verte), les alarmes (voyants verts ou rouges), le régime moteur, et la vitesse du vente, les conditions météorologiques (vue du ciel) et l'identification du drone (symbole).

Ces tâches sont réalisées en parallèle car toutes les informations sont disponibles simultanément. Dès que le système détecte que l'une des valeurs mesurées est modifiée, il met à jour l'affichage. L'affichage de l'état de la batterie (Figure 154) est une tâche qui est décomposée en deux tâches modales en sortie : *Afficher niveaux* et *Afficher pourcentage* qui sont les deux représentations graphiques pour présenter le niveau de charge de la batterie du drone tactique. De la même manière, l'affichage du niveau de carburant est décrit par deux tâches modales en sortie (*Afficher niveaux*, et *Afficher pourcentage*).

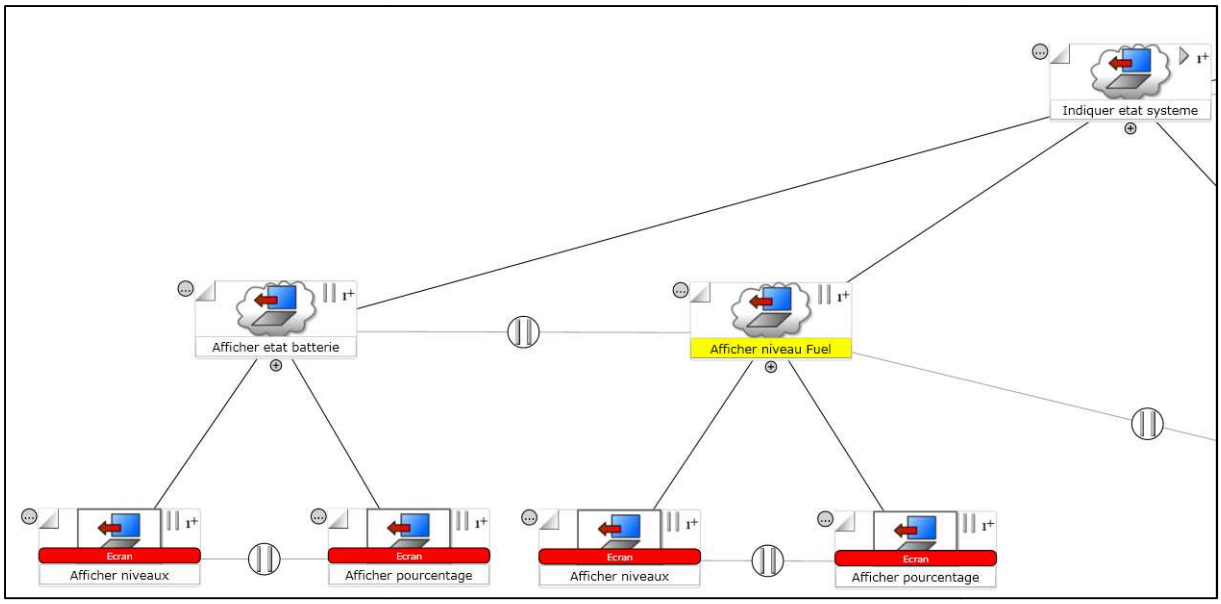


Figure 154 : Tâche : indiquer l'état du drone tactique (détail de la partie rouge de la Figure 153).

L'arbre de tâches de la Figure 155 correspond à l'affichage des différentes alarmes gérées par les systèmes, telles que le disfonctionnement du système de guidage GPS (*Afficher alarme GPS*) ou encore un défaut de fonctionnement du système antigel (*Afficher alarme antigel*).

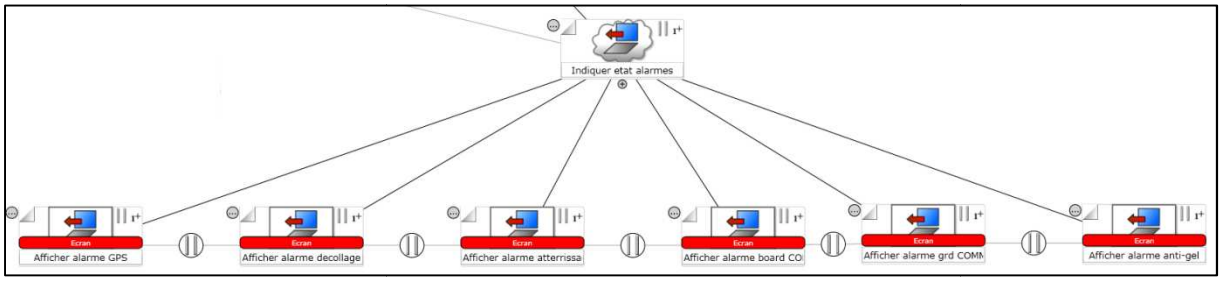


Figure 155 : Tâche : indiquer l'état du drone tactique (détail de la partie bleue de la Figure 153).

L'arbre de tâche de la Figure 156correspond à l'affichage des informations météorologiques, telles que la température (*Afficher température*) ou le taux d'humidité (*Afficher humidité*).

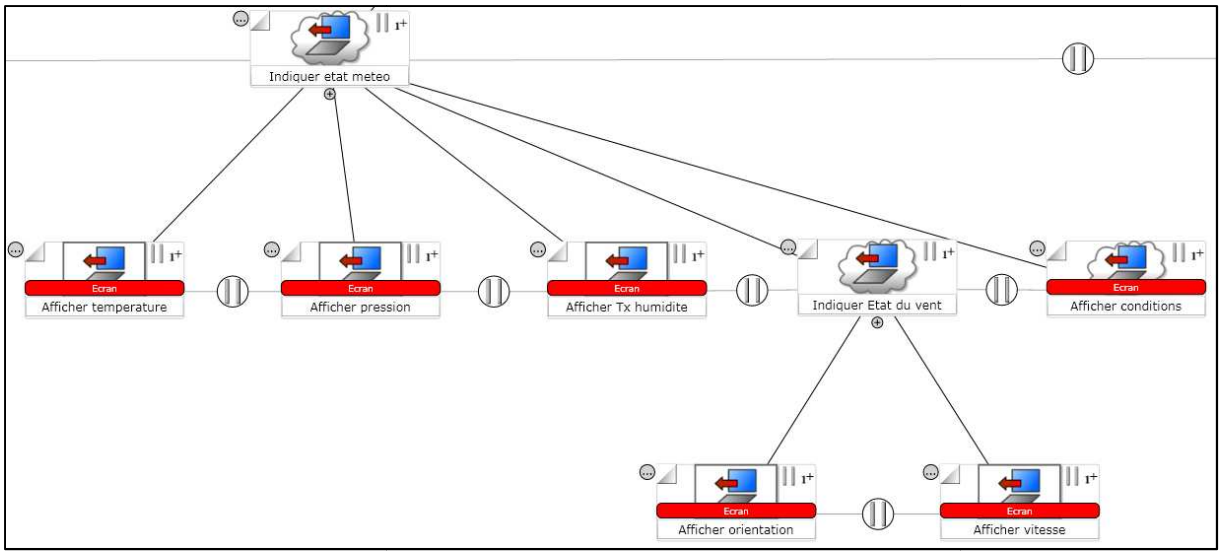


Figure 156 : Tâche : indiquer l'état du drone tactique (détail de la partie verte de la Figure 153).

L'arbre de tâches de la Figure 157 comporte un ensemble de tâches de présentation correspondant à l'affichage de l'état du drone comme sa position (*Afficher position*), son altitude (*Afficher altitude*) ou son roulis (*Afficher roulis*).

De plus, cet arbre de tâches comporte une tâche interactive : *Naviguer dans l'historique système,* alors que les sous-arbres ci-dessus ne comportaient que des tâches de présentation. Cependant, celle-ci se limitant à des tâches articulatoires[6] permettant de faire défiler l'historique à l'aide de la barre de défilement, nous ne l'avons pas détaillée ici.

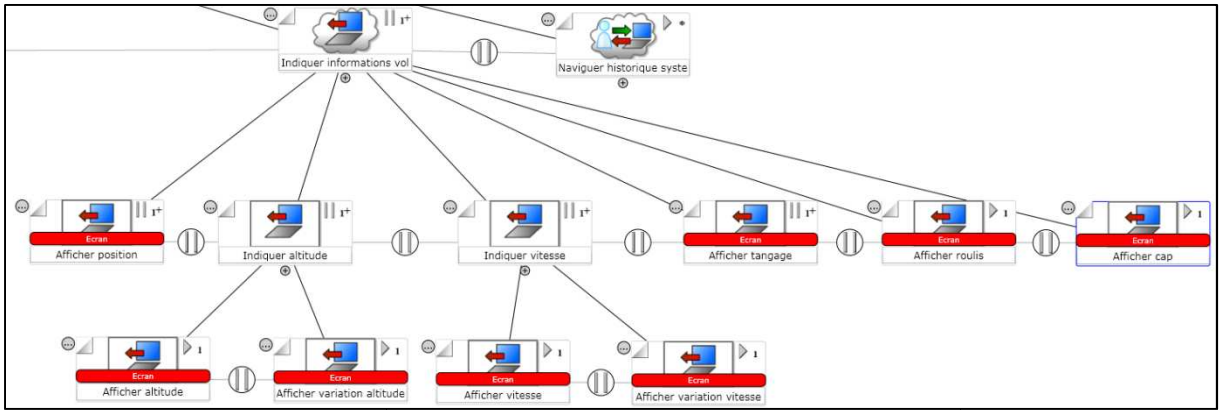


Figure 157 : Tâche : indiquer l'état du drone tactique (détail de la partie jaune de la Figure 153).

5.2.4.2 Modifier les paramètres de vol

La tâche *Modifier paramètres de vol* décrit l'interaction pour modifier la vitesse, l'altitude, l'inclinaison (pitch), le roulis et le cap du drone tactique. La Figure 158 illustre l'arbre de tâches complet pour la tâche *Modifier paramètres de vol*. La partie rouge correspond à l'interaction pour modifier la vitesse, l'altitude et l'inclinaison, la partie bleue décrit la gestion du roulis, et la partie verte décrit la gestion du cap. Nous détaillons chacun de ces sous-arbres.

[6] « Une tâche articulatoire est une tâche induite par la solution informatique. Elle ne dépend pas du domaine. Techniquement, l'accomplissement d'une tâche articulatoire ne modifie pas l'état des concepts du domaine. Les tâches de gestion des fenêtres sont de nature articulatoire. » [Coutaz 1998]

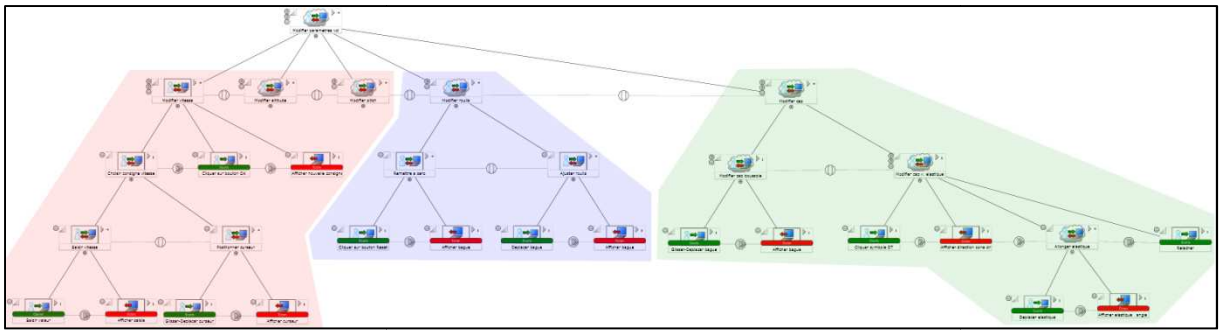


Figure 158 : Modifier les paramètres de vol.

Pour les trois premiers paramètres (vitesse, altitude, inclinaison), l'interaction repose sur la saisie d'une consigne, soit sous forme numérique par une saisie au clavier dans un champ texte, soit par manipulation directe à la souris par le déplacement d'un curseur. La consigne saisie est enregistrée dès lors que l'on clique sur le bouton *Ok* associé au paramètre. Il convient de noter que la modification de la vitesse est possible pour les modes A1 et A3 (décorations 1 et 3). La modification des deux autres paramètres n'est permise que pour le mode A3 (décoration 3). L'arbre de tâches de la Figure 159 illustre cette interaction pour la tâche *Modifier vitesse.* Nous ne décrivons pas ici les tâches *Modifier altitude* et *Modifier pitch* qui reprennent la même forme que la tâche *Modifier vitesse*. Ceci est un cas typique où l'utilisation des Templates serait intéressante : le sous-arbre décrit pour la modification de la vitesse pourrait alors être défini avec un paramètre <paramètre de vol>, qui serait remplacé lors de l'utilisation du Template par la vitesse, l'altitude ou l'inclinaison.

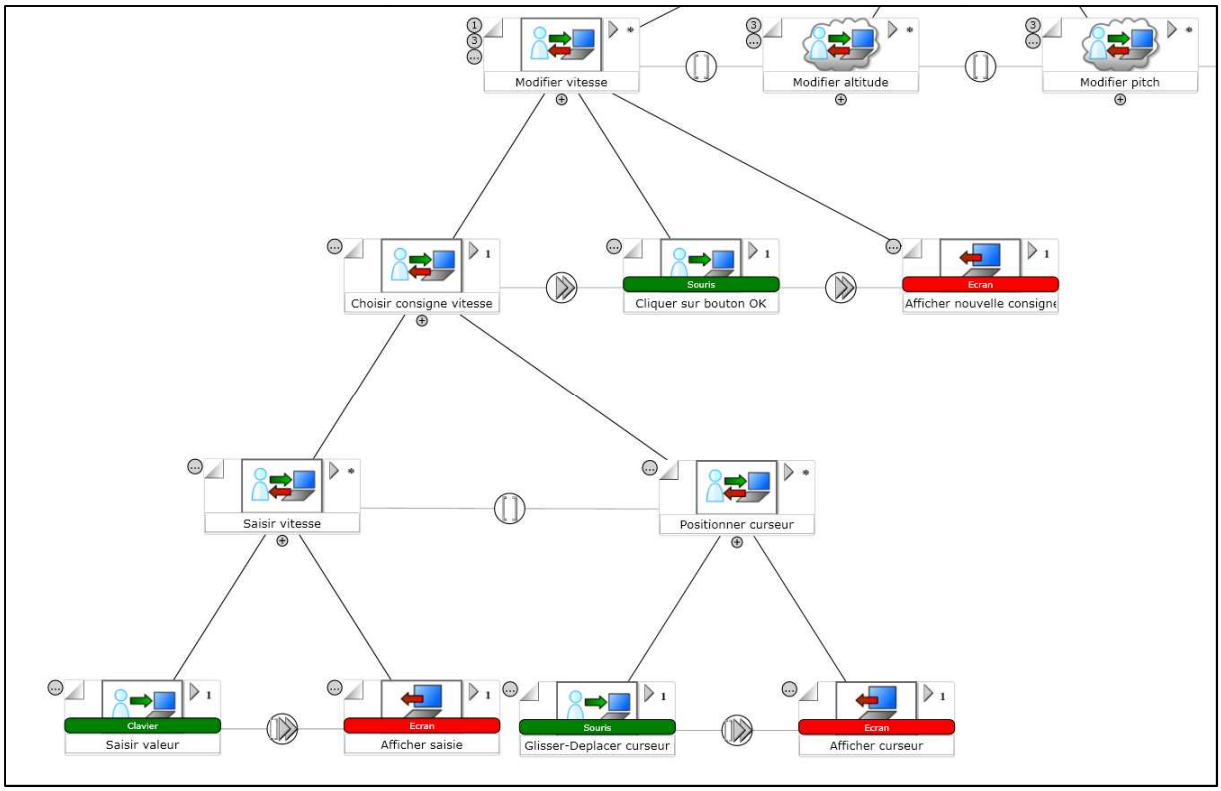


Figure 159 : Tache : modifier les paramètres de vol (détail de la partie rouge de la Figure 158).

L'arbre de tâche de la Figure 160 illustre l'interaction pour modifier le paramètre de roulis du drone tactique (*Modifier roulis*). Celle-ci repose sur le déplacement à la souris d'une bague autour du

cadran représentant l'alignement par rapport à l'horizon. Le bouton *Reset* permet la remise à zéro de la bague. Cette tâche n'est réalisable que pour le mode A3, c'est pourquoi, la tâche *Modifier roulis* est décorée avec le *Contexte 3*.

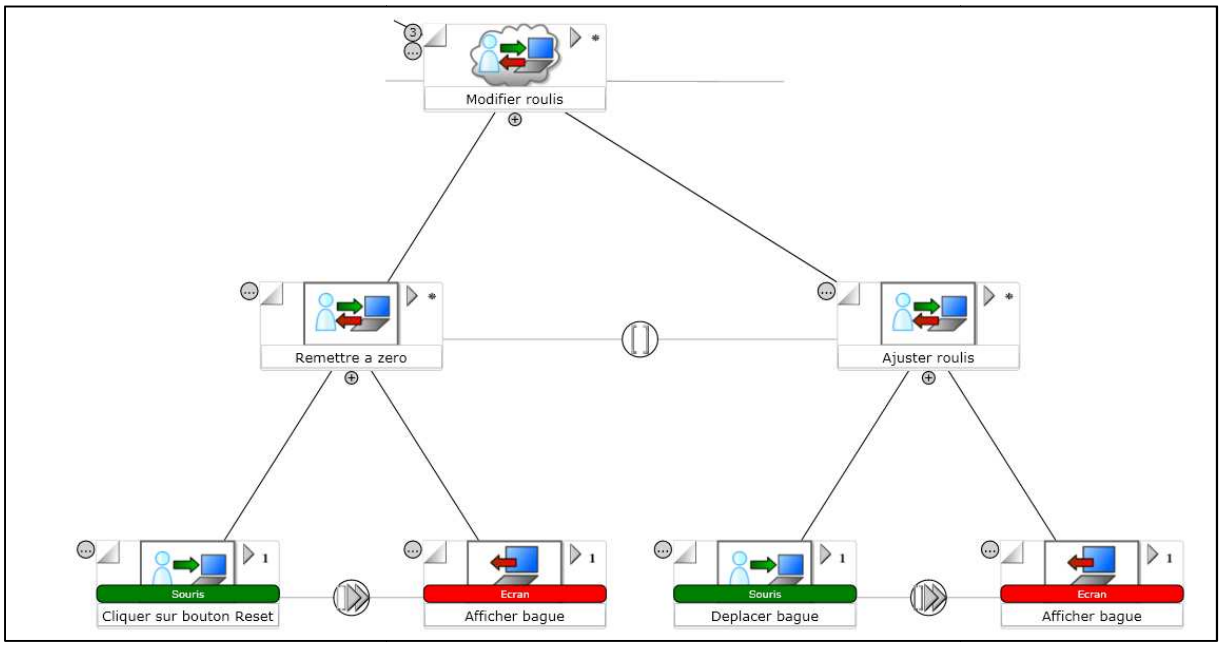


Figure 160 : Tâche : modifier les paramètres de vol (détail de la partie bleue de la Figure 158).

Enfin, l'arbre de tâches de la Figure 161 illustre l'interaction pour modifier le cap du drone tactique. Deux possibilités sont offertes à l'utilisateur. La première consiste, à partir de l'écran de vol, à déplacer une bague autour de la boussole indiquant l'orientation du drone tactique. Cette bague représente la consigne. Sur ce cadran, figure également une autre bague indiquant l'orientation réelle du drone tactique. Cette première interaction n'est disponible qu'en mode A3. La deuxième possibilité consiste, à partir de la vue cartographique, à tirer un élastique à partir du symbole représentant le drone. Cette interaction repose sur l'usage de la souris par un geste de glisser-déposer. Cette deuxième interaction est disponible aussi bien en mode A1 qu'en mode A3, c'est pourquoi, les étiquettes de contextes (1) et (3) sont associées à la tâche *Modifier cap via élastique*.

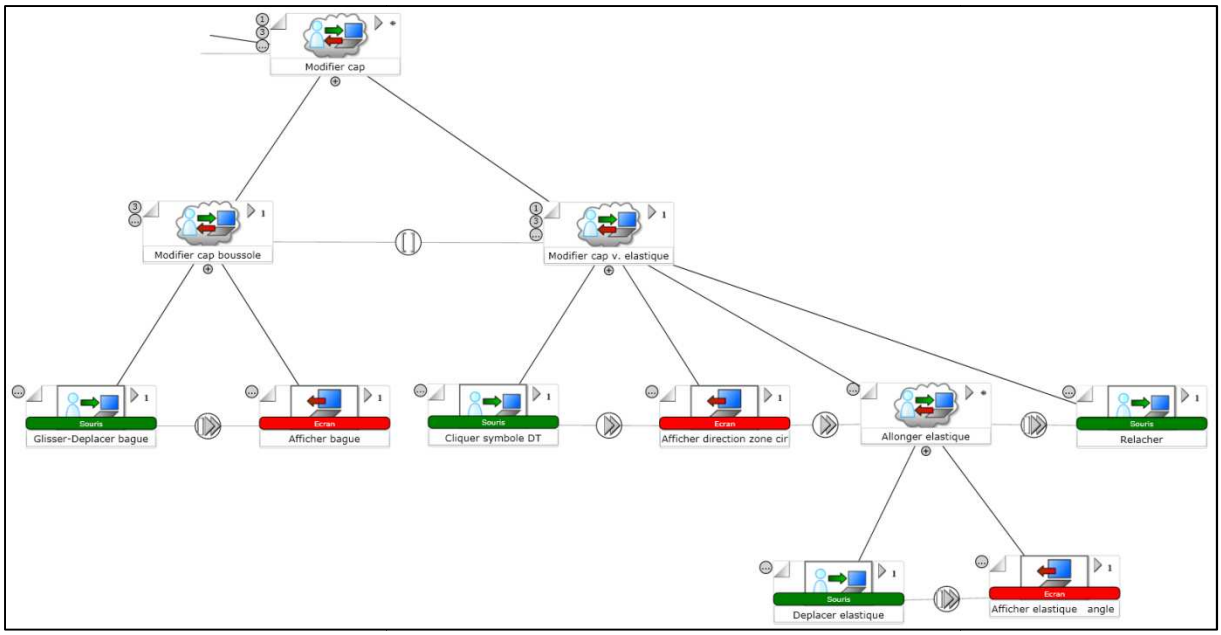


Figure 161 : Tâche : modifier les paramètres de vol (détail de la partie verte de la Figure 158).

5.2.4.3 Contrôler la trajectoire du drone tactique

La tâche *Contrôler trajectoire DT* correspond à l'interface disponible sur le second écran de l'opérateur Vecteur et permet l'interaction avec la vue cartographique pour le contrôle du drone tactique. L'arbre de tâches complet pour cette tâche est présenté à la Figure 162. Nous détaillons dans la suite chacune des parties colorées de cet arbre de tâches, dans le même ordre que dans les sections précédentes (rouge, bleue, puis verte).

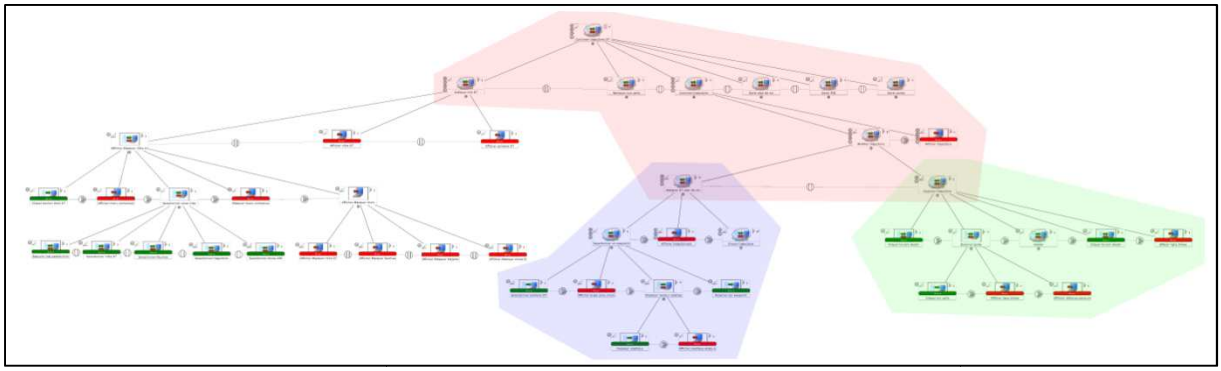


Figure 162 : Tâche : contrôler la trajectoire du drone tactique (Contrôler trajectoire DT).

La portion rouge de l'arbre montrée à la Figure 163 (dans une version reformatée pour des raisons de place) représente la tâche Contrôler trajectoire DT. Elle se décompose en plusieurs sous-tâches principales : *Manipuler la vue cartographie* (qui est décrit en détail dans la section suivante), *Contrôler la trajectoire du DT*, *Gérer le plan de vol* et *Indiquer infos DT* (pour afficher les informations sur les drones contrôlés dans le coin supérieur gauche de l'écran). Toutes ces tâches sont réalisables quelque soit le mode opératoire (A1, A2 et A3).

La tâche de contrôle de trajectoire du drone tactique (*Contrôler trajectoire*) se décompose en plusieurs sous-tâches, dont deux correspondent à la modification de la trajectoire d'un drone tactique. La première consiste à assigner à un drone un plan de vol (*Assigner DT plan de vol*), et la

deuxième à dessiner une trajectoire (*Dessiner trajectoire*). Ces deux tâches correspondent respectivement aux parties bleue et verte de la Figure 162.

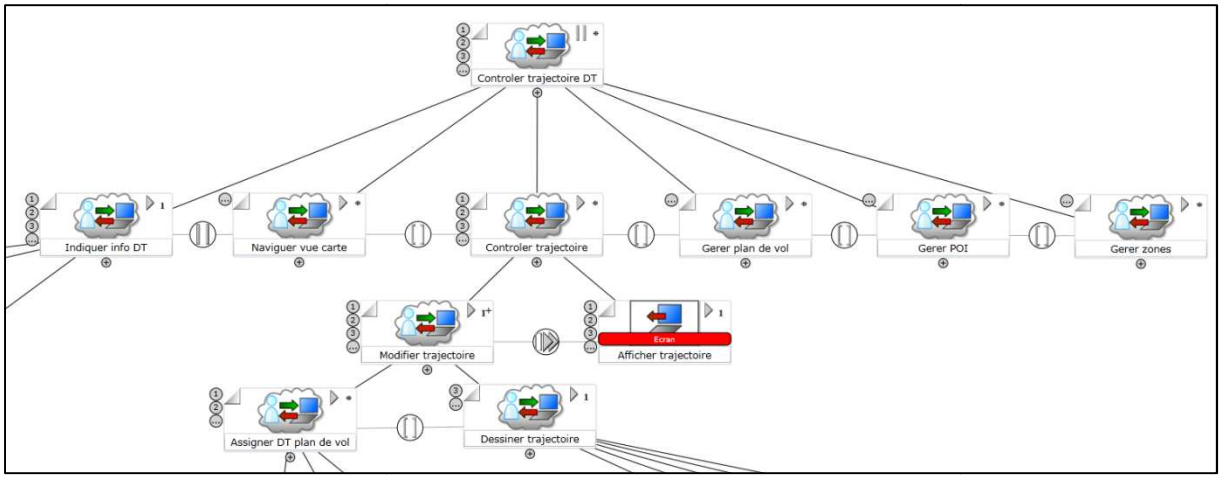


Figure 163 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie rouge de la Figure 162).

Le contrôle de la trajectoire par assignation d'un plan de vol (Figure 164) est disponible pour les modes opératoires A1 et A2 (décorations de contexte (1) et (2)). Pour assigner un plan de vol à un drone, l'opérateur doit d'abord cliquer sur le symbole du DT, puis déplacer le vecteur élastique qui apparaît et le relâcher sur un *waypoint* existant. Dès lors, le système peut proposer une ou plusieurs trajectoires pour rejoindre le *waypoint* ainsi sélectionné (pour les modes opératoires : A1 et A3 ; décorations 1 et 2). L'interaction se termine par la sélection d'une des trajectoires et par la validation de ce choix. Cette dernière n'est disponible que pour le mode A1 (décoration (1)).

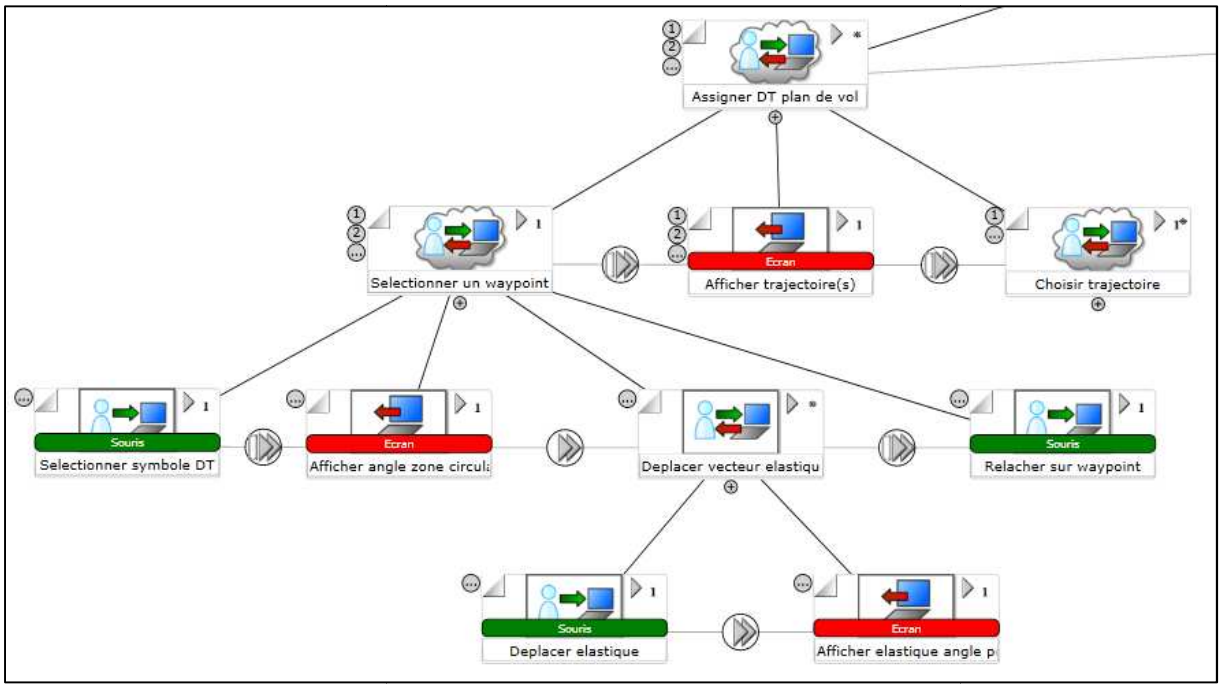


Figure 164 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie bleue de la Figure 162).

Le second cas correspond à la tâche *Dessiner trajectoire* (Figure 165). Elle consiste à dessiner une trajectoire sous la forme d'une ligne brisée. Cette interaction, à l'aide de la souris, consiste à cliquer

sur la carte et à produire une série de points. Pour activer cette tâche, il faut au préalable cliquer sous le bouton pour activer l'outil dessin. Celle-ci n'est disponible que pour le mode A3 (décoration (3)). Outre l'affichage de la ligne, le système indique la distance et le temps de parcours estimé.

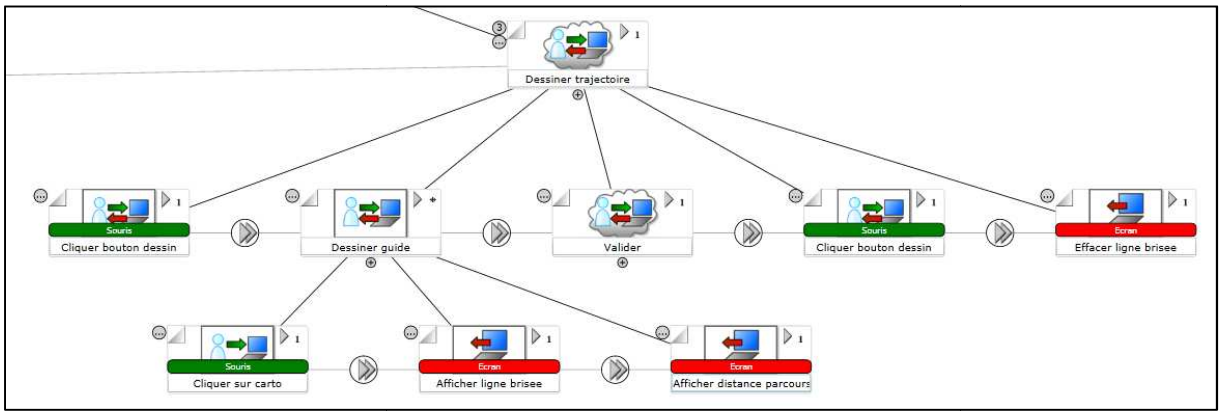


Figure 165 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie verte de la Figure 162).

5.2.4.4 Manipuler la cartographie

La tâche *Manipuler la cartographie* se décompose en un ensemble de tâches articulatoires [Coutaz 1998] permettant la navigation au sein de la carte. L'arbre complet pour cette tâche est montré à la Figure 166. Les trois grandes sous-tâches permettent de déplacer la carte (zone rouge et bleu), de zoomer en avant ou en arrière (zone verte) et de mesurer des distances (zone jaune).

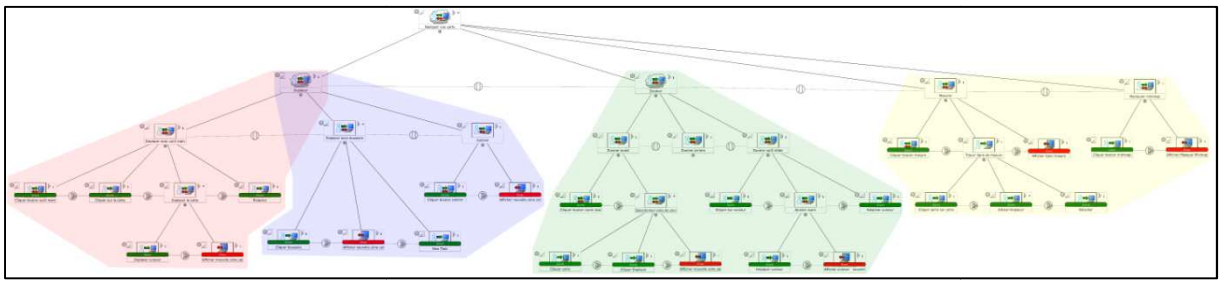


Figure 166 : Tâche : manipuler la cartographie (Naviguer vue carte).

Les parties rouge et bleu de la Figure 166 correspondent au déplacement de la carte. Celui-ci est possible au travers de plusieurs interactions :

1. Déplacer la carte dans une direction donnée à la souris à l'aide de l'outil main (devant être au préalable activé en cliquant sur le bouton correspondant). Cette interaction repose sur le principe du cliquer-glisser/déplacer-relâcher. Cette interaction concrète est décrite dans l'arbre de tâches de la Figure 167.

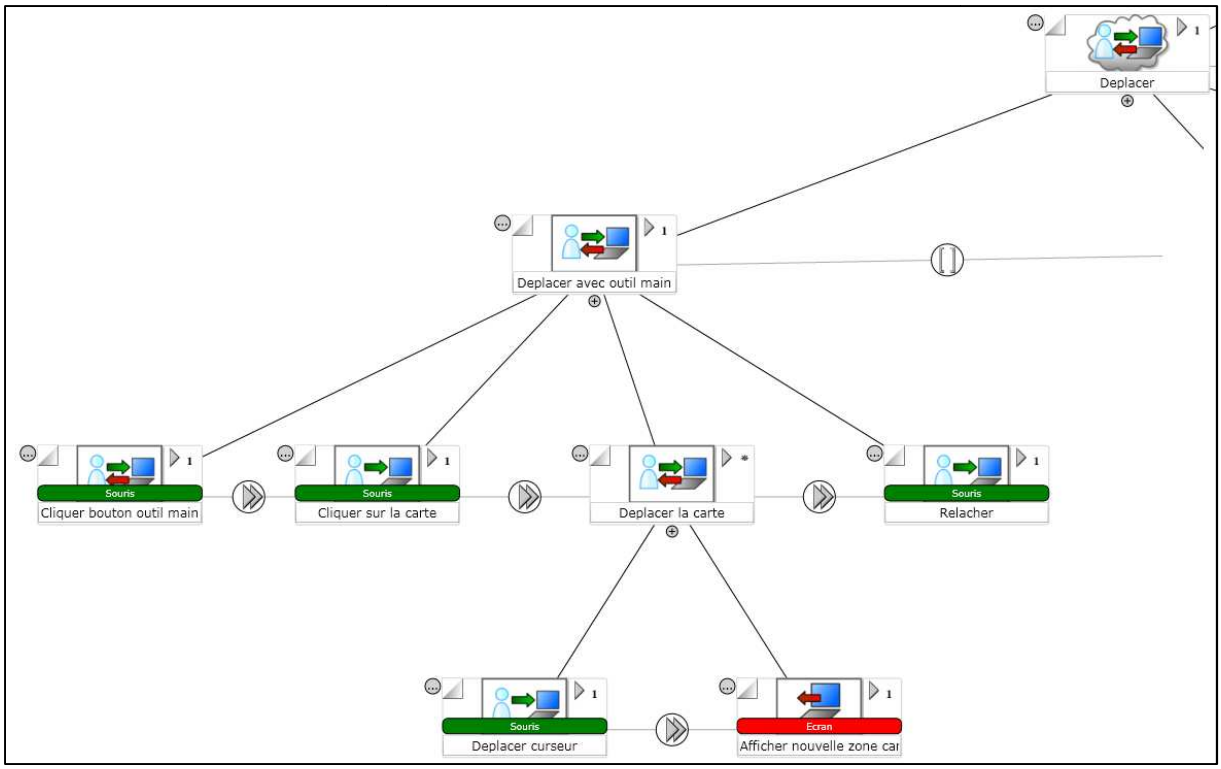


Figure 167 : Tâche : manipuler la cartographie (détail de la partie rouge de la Figure 166).

2. Déplacer la carte dans une direction à l'aide d'une boussole (Figure 168).
3. Recentrer la carte sur la position initiale en cliquant sur le bouton correspondant (Figure 168).

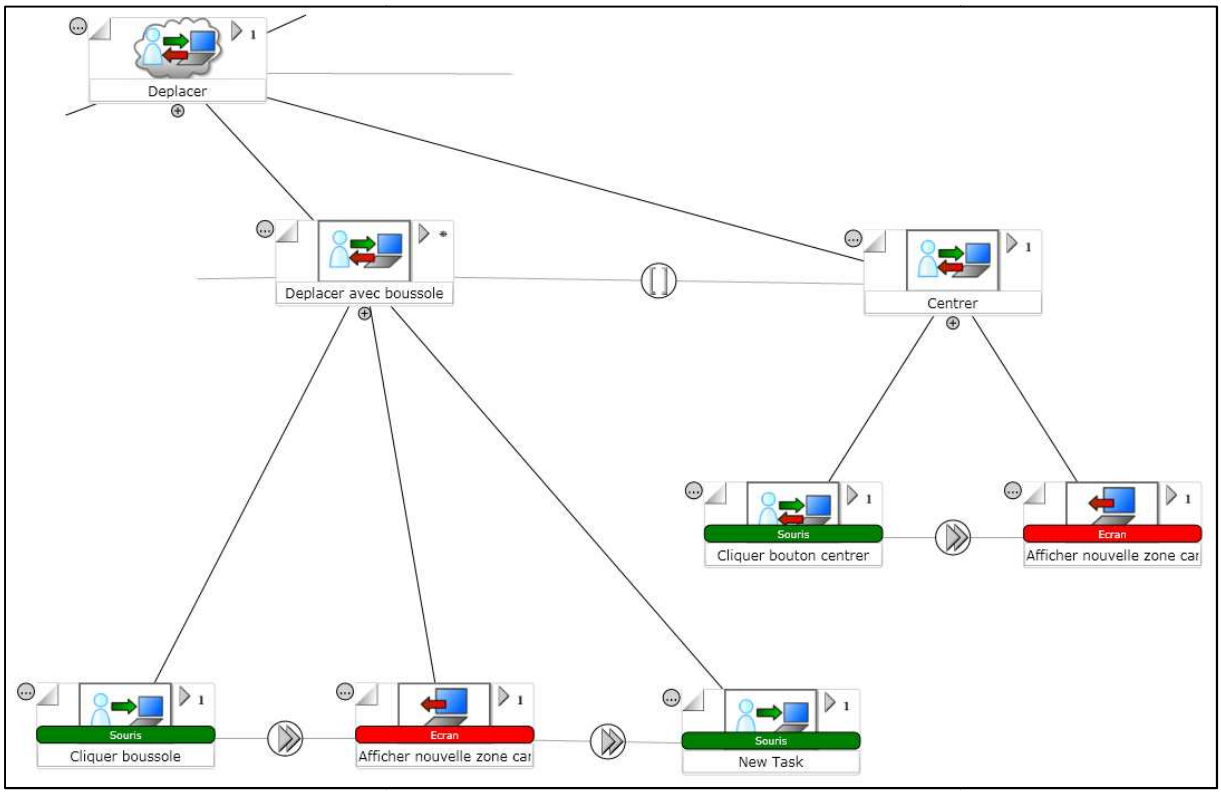


Figure 168 : Tâche : manipuler la cartographie (détail de la partie bleue de la Figure 166).

La partie verte de la Figure 166 correspond à la tâche de modification du niveau zoom sur la carte. Elle est détaillée à la Figure 169. Pour zoomer sur la carte, plusieurs interactions sont possibles :

1. Zoomer en avant sur une zone choisie et délimitée par deux points sélectionnés à la souris. Il faut au préalable cliquer sur le bouton zoom avant.
2. Zoomer en arrière par un simple clic sur la carte. De même, il faut cliquer sur le bouton activant cette fonctionnalité.
3. Contrôler le zoom à l'aide d'une glissière (*slider*).

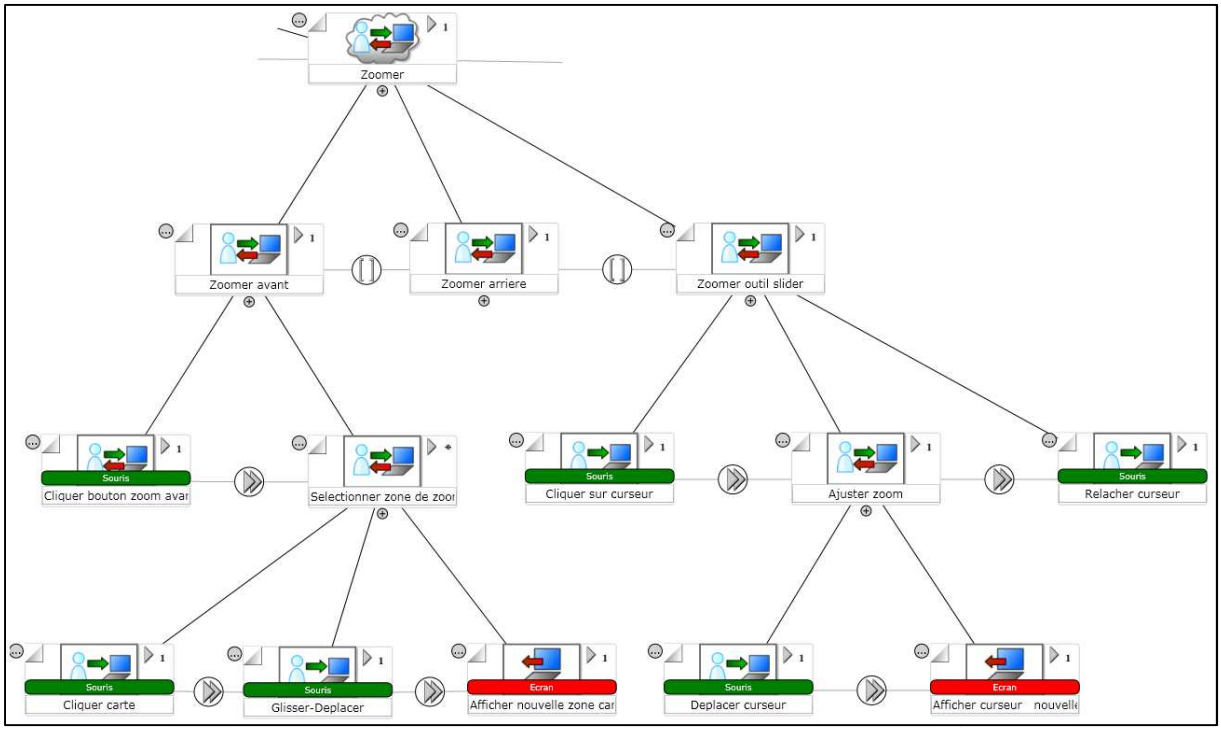


Figure 169 : Tâche : manipuler la cartographie (détail de la partie verte de la Figure 166).

La partie jaune de la de la Figure 166, détaillée à la Figure 170, correspond à une tâche de mesure et à une tâche de manipulation de la *mini-map.* La tâche *Mesurer* consiste à cliquer à deux endroits de la carte visible pour déterminer la distance réelle séparant les deux points. Une fois que les deux points sont sélectionnés (technique glisser/déplacer), une ligne et la distance sont affichées à l'écran. De même, cet outil est activé suite à un clic sur le bouton correspondant à l'outil de mesure.

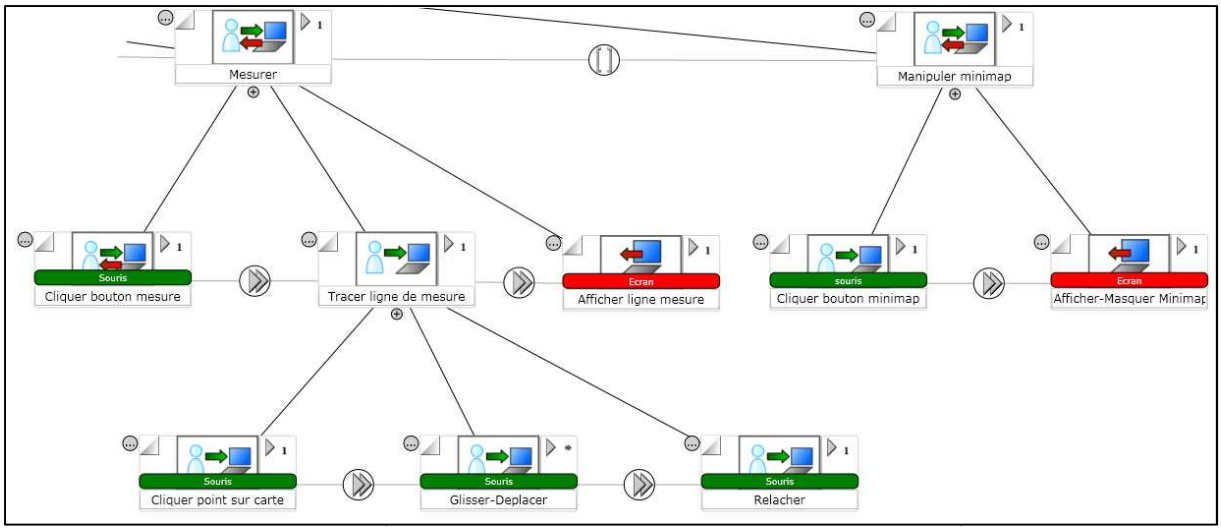


Figure 170 : Tâche : manipuler la cartographie (détail de la partie jaune de la Figure 166).

5.2.4.5 Gérer un plan de vol

La tâche *Gérer plan de vol* (Figure 171) permet soit de créer un nouveau plan de vol soit de modifier un plan de vol existant. La sous-tâche *Créer plan de vol* repose sur l'usage de la souris, par manipulation directe, pour positionner une série de *waypoint* par des clics. A chaque clic, la trace en cours de construction est affichée à l'écran. Cette tâche se termine par un double-clic. Pour réaliser cette tâche, il faut au préalable cliquer sur le bouton outil de création de plan de vol. Enfin, à l'issue de la création du plan de vol, le système affiche un plan de coupe du plan de vol, sur lequel peut être modifié l'altitude des *waypoint.*

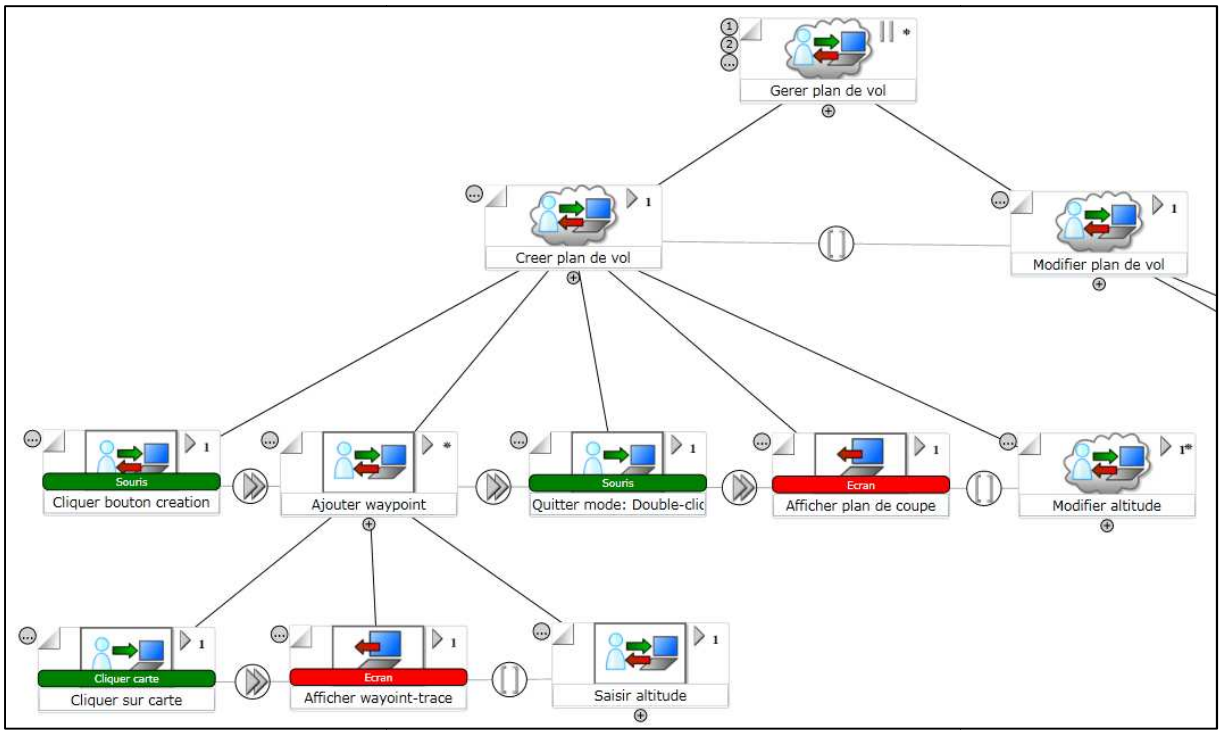


Figure 171 : Tâche : gérer un plan de vol.

Pour modifier un plan de vol existant (par exemple pour déplacer un *waypoint)* existant, il faut au préalable sélectionner le plan de vol en cliquant dessus avec la souris. La tâche *Editeur un plan de vol* est détaillée à la Figure 172. Elle consiste en deux sous-tâches : *Déplacer un waypoint* et *Modifier un waypoint*.

Le déplacement d'un *waypoint* repose sur une interaction à la souris selon la technique du glisser-déposer. Elle nécessite également l'appui sur la touche Shift pour autoriser le déplacement.

La modification d'un *waypoint* requiert d'abord sa sélection. Ensuite, l'opérateur Vecteur peut modifier les propriétés du *waypoint* sélectionné ou modifier son altitude au sein d'une vue en coupe.

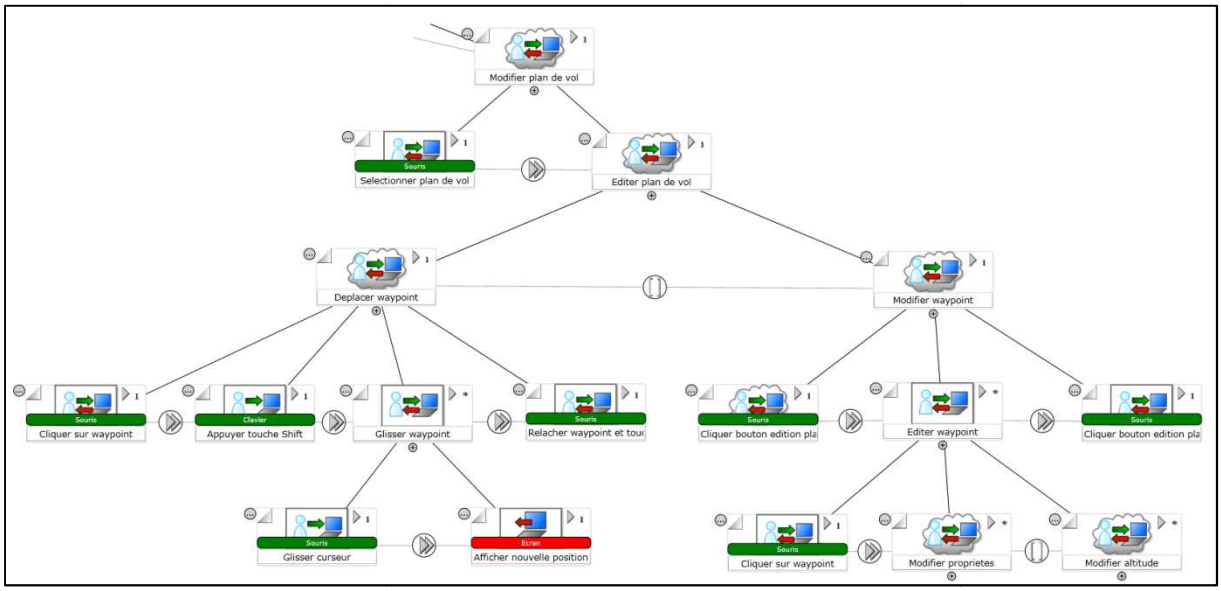


Figure 172 : Tâche : éditer un plan de vol.

La modification des propriétés d'un *waypoint* est décrite au sein de l'arbre de tâches de la Figure 173. Les propriétés modifiables incluent le nom, la longitude, la latitude et l'altitude.

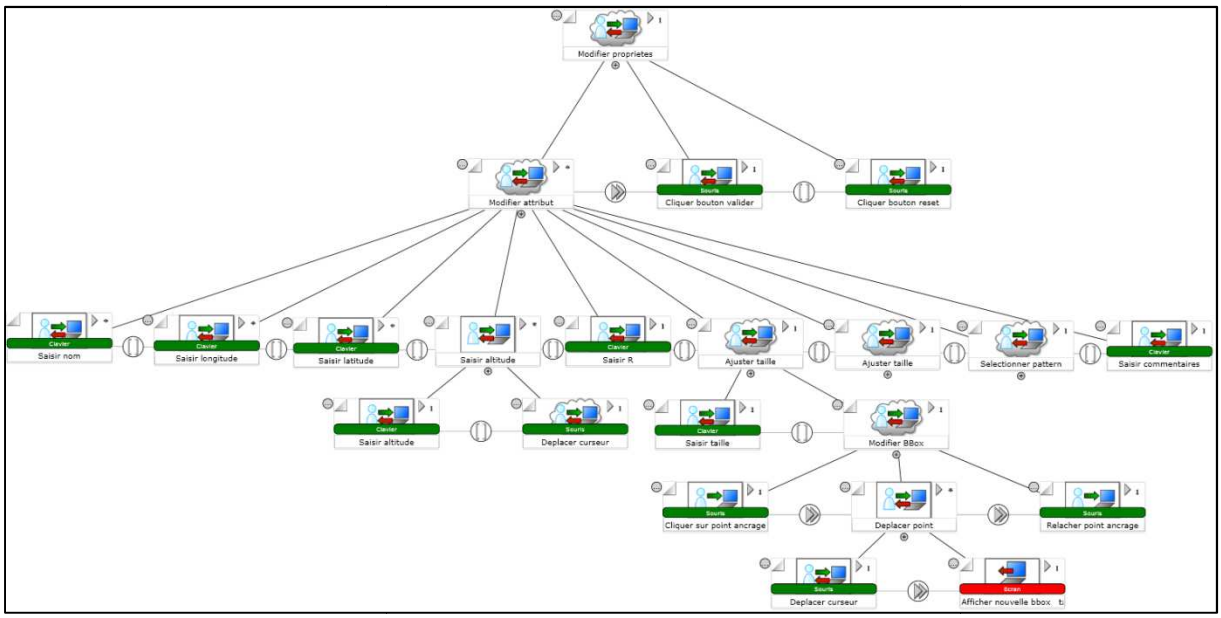


Figure 173 : Tâche : modifier les propriétés d'un *waypoint*.

La modification de l'altitude au sein d'une vue en coupe est décrite au sein de l'arbre de tâches de la Figure 174. Après ouverture du plan de coupe par un clic sur le bouton correspondant, elle consiste à déplacer les *waypoint* à la souris par des glisser-déposer verticaux.

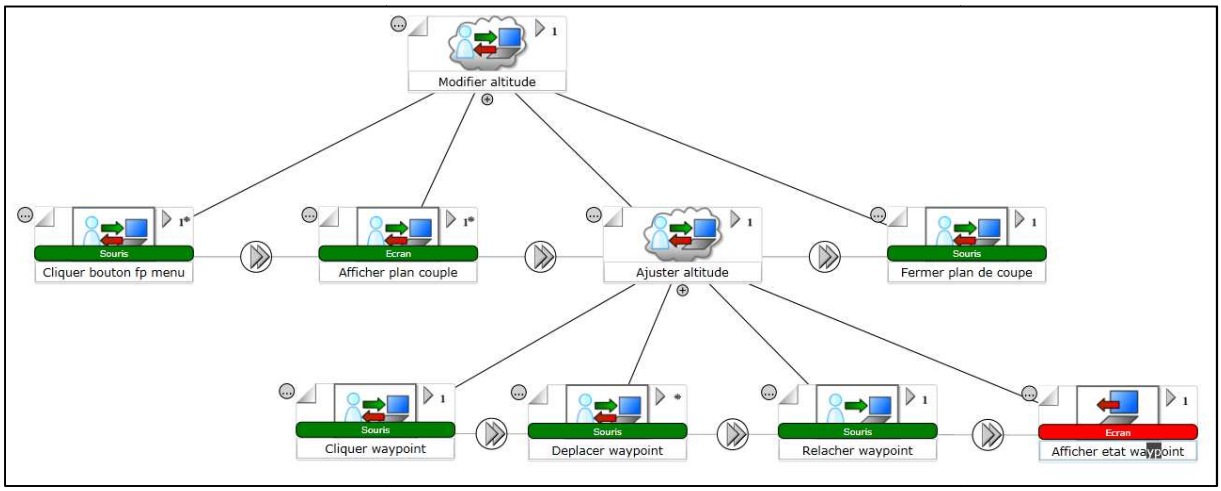


Figure 174 : Tâche : modifier l'altitude.

5.2.4.6 Gérer les zones

La tâche *Gérer des zones* permet de créer et de modifier des zones de vol rectangulaires, circulaires ou polygonales. Elle est décrite complètement par l'arbre de tâches de la Figure 175.

Pour *Gérer des zones*, il faut au préalable cliquer sur le bouton outil de création de zones. Il est alors possible d'ajouter une zone (Figure 176) ou de modifier une zone existante (Figure 177). Dans tous les deux cas, l'interaction se réalise avec la souris.

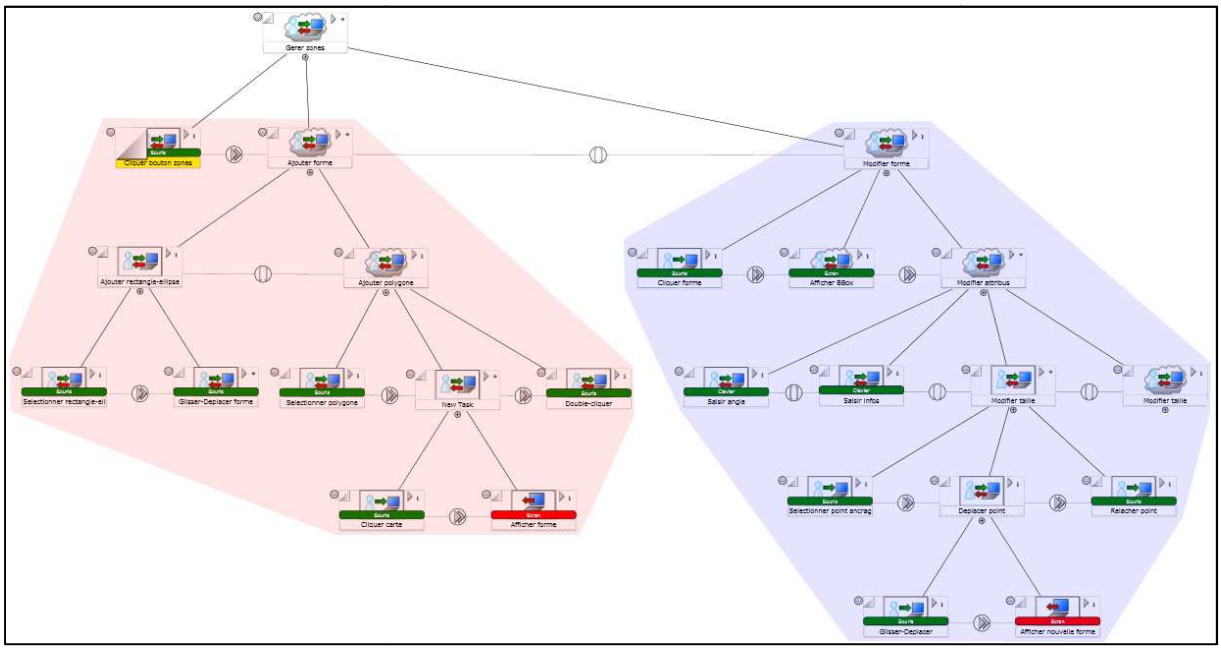


Figure 175 : Tâche : gérer les zones (Gérer zone).

La Figure 176 détaille la partie rouge de la Figure 175, c'est-à-dire le clic sur le bouton de gestion des zones, et la tâche d'ajout de forme (*Ajouter Forme*). L'interaction est semblable à celle d'un éditeur

de dessin vectoriel. En effet, l'ajout d'une forme rectangulaire ou circulaire consiste à définir deux points par la technique du glisser/déplacer. Pour ajouter une forme polygonale, il s'agit simplement de définir une série de points à la souris.

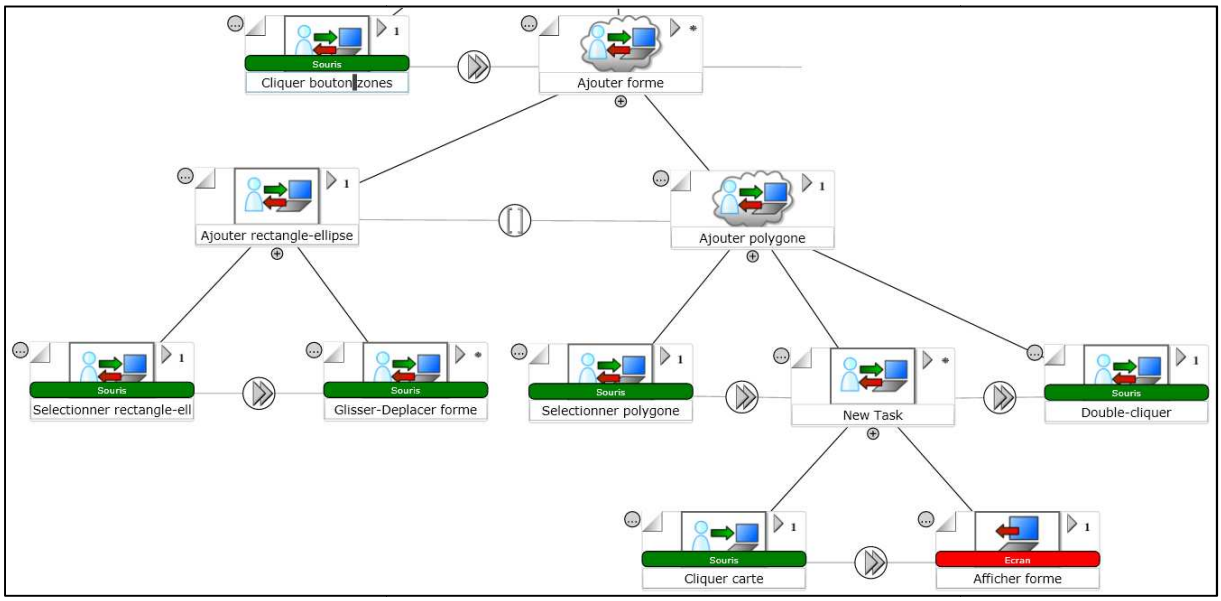


Figure 176 : Tâche : gérer les zones (détail de la partie rouge de la Figure 175).

La Figure 177 détaille la partie bleue de la Figure 175 : la tâche de modification d'une forme existante (*Modifier Forme*). Pour cela, l'utilisateur doit au préalable cliquer sur la forme. Il est alors possible de modifier les propriétés au clavier dans le panneau situé sur le bord gauche ou à la souris, par la technique du glisser/déplacer, en déplaçant une poignée (point d'ancrage) de la boîte englobante (bbox).

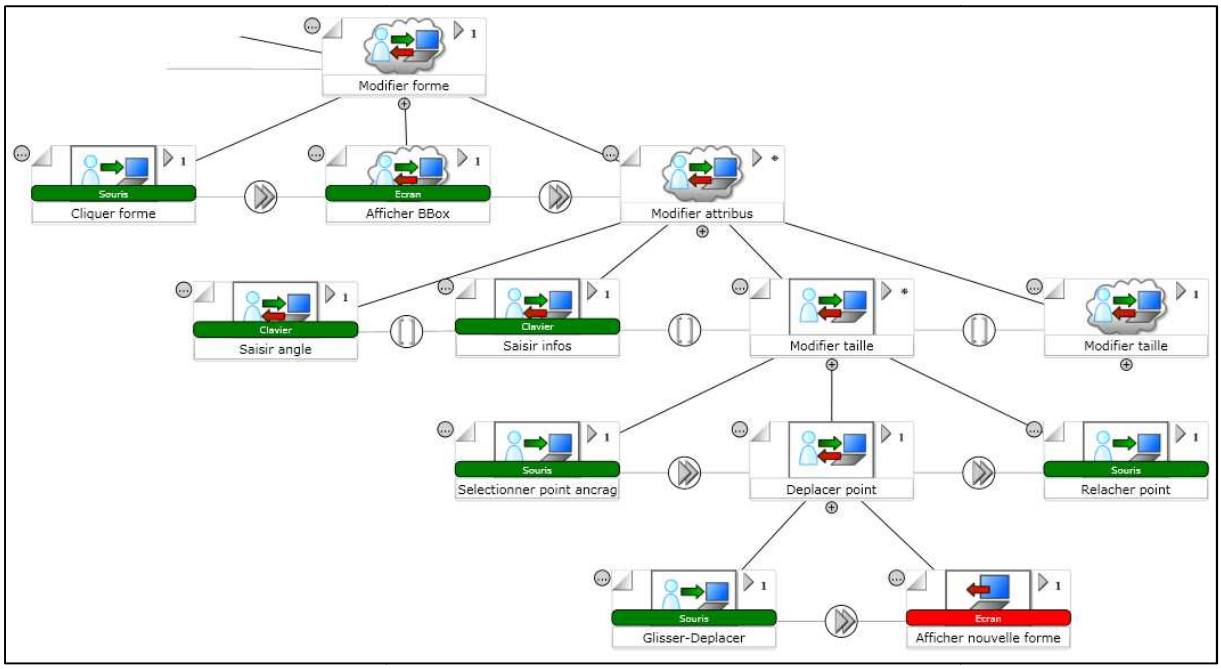


Figure 177 : Tâche : gérer les zones (détail de la partie bleue de la Figure 175).

5.2.5 Spécification COMM de l'interaction pour l'opérateur Charge Utile

Dans cette partie, consacrée aux tâches de l'opérateur Charge Utile, nous décrivons uniquement les tâches relatives au contrôle de la vue 3D. En effet, le contrôle de la vue cartographie par l'opérateur CU est semblable à celui de l'opérateur Vecteur, sauf que l'opérateur CU ne peut interagir avec les plans de vol et les trajectoires des drones.

Le contrôle de la vue 3D par l'opérateur CU est décrit dans la forme complète par l'arbre de tâches de la Figure 178, dont les parties rouge et bleue sont détaillées respectivement au sein de la Figure 179 et de la Figure 180.

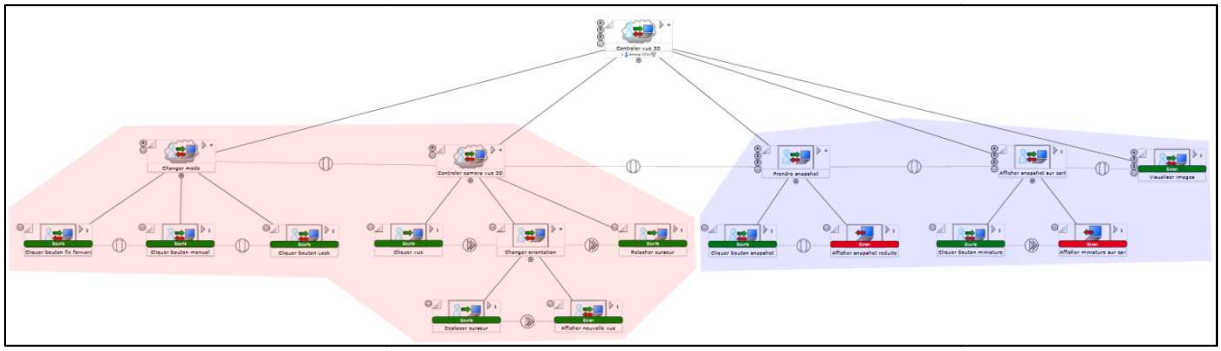


Figure 178 : Tâche : gérer la vue 3D.

Les deux premières tâches de contrôle de la vue 3D par l'opérateur Charge Utile consiste d'une part à changer le mode d'utilisation du contrôle en cliquant sur l'un des trois boutons : *Fix Forward*, *Manual*, *Lock* et d'autre part à contrôler l'orientation de la vue 3D avec la souris par la technique du glisser-deposer.

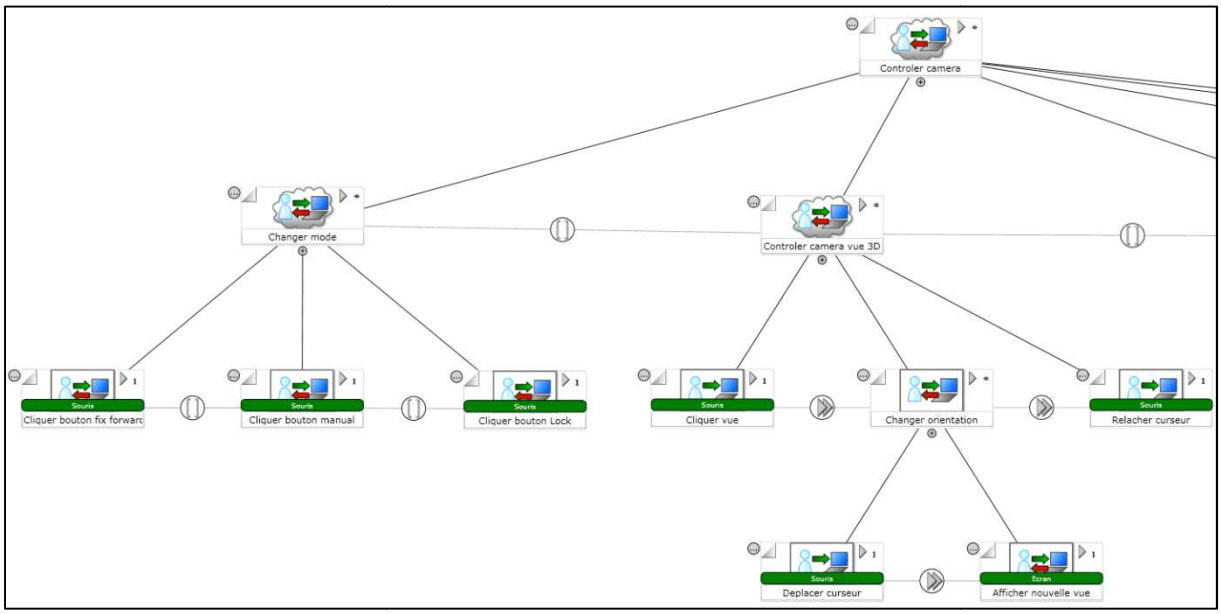


Figure 179 : Tâche : gérer la vue 3D (détail de la partie rouge de la Figure 178).

Les autres interactions avec la vue 3D concerne la prise d'instantanés (*Snapshot*) par un clic sur le bouton *Snapshot*, et l'affichage des instantanés déjà pris par un clic sur le bouton *Miniature*.

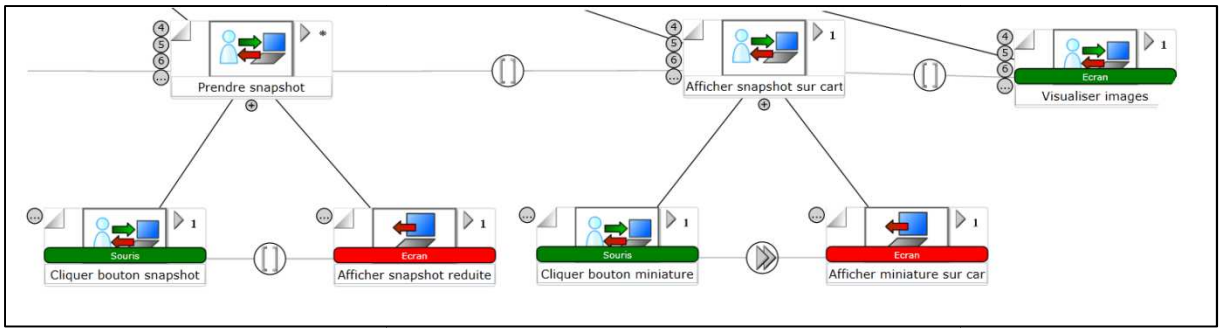


Figure 180 : Tâche : gérer la vue 3D (détail de la partie bleue de la Figure 178).

5.2.6 Spécification COMM de l'interaction collaborative de partage de la caméra

Les principales tâches des opérateurs VE et CU pour une station ont été décrites précédemment. Nous présentons maintenant la tâche de partage du contrôle d'une caméra (faisant partie de la charge utile d'un drone) entre deux stations. Le partage du contrôle de charge utile entre deux stations peut être utile dans le cas où les opérateurs d'une station ont la nécessité de réaliser des observations prioritaires sur une zone qu'ils ne peuvent couvrir avec leur propre drone.

La tâche Partager la caméra (Figure 181) introduit la possibilité de partager la caméra de la Charge utile entre deux stations. Cette tâche est collaborative et implique deux opérateurs avec le rôle métier CU et deux rôles interactifs (*Préteur*, et *Récepteur*). Pour procéder au partage de la caméra, un opérateur CU doit *Proposer le partage de la caméra.* Ce faisant, il prend alors le rôle interactif <*Préteur*>. Pour cela, il clique sur le bouton partage, ce qui fait apparaitre une liste de sélection permettant de choisir une station. Une fois la station sélectionnée, le <*Préteur*> doit *Cliquer sur le bouton partager* pour transmettre sa proposition.

L'opérateur CU de la station sélectionnée prend alors le rôle de <*Récepteur*> et voit alors apparaitre l'écran de confirmation du partage de la caméra (tache *Afficher écran confirmation*). Il peut alors décider d'accepter le partage soit de le refuser.

S'il décide d'accepter, le drone passe sous le contrôle de l'opérateur jouant le rôle interactif <*Récepteur*>. Cette interaction non décrite ici est identique celle décrit plus haut pour le contrôle de caméra d'un drone initialement contrôlé par une station. L'opérateur jouant le rôle interactif <*Récepteur*> contrôle le drone jusqu'à ce que l'opérateur <*Préteur*> lui propose d'annuler le partage. Ceci débouche pour les deux rôles sur une confirmation de la fin du partage que les deux opérateurs doivent accepter pour mettre fin au partage.

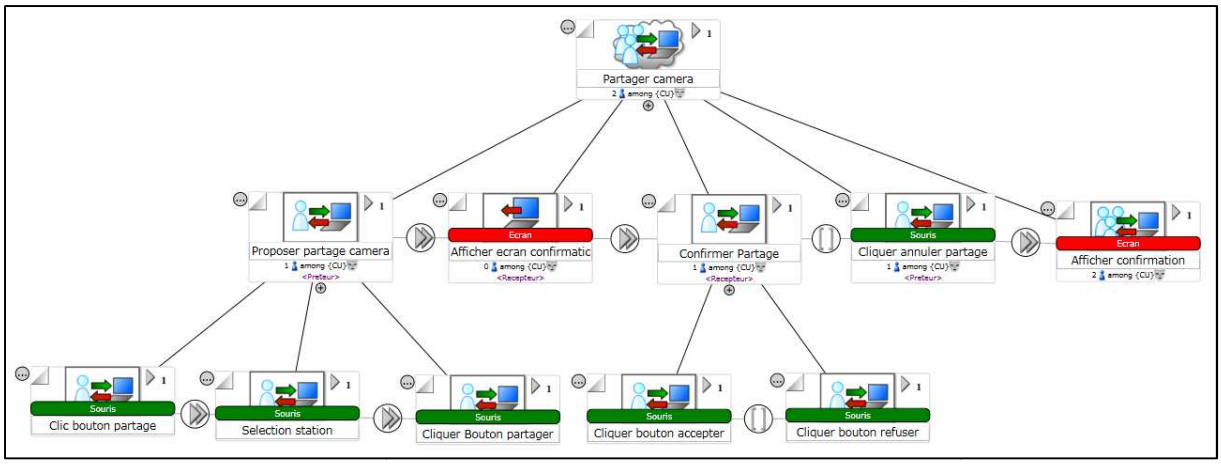


Figure 181 : Tâche : partage d'une caméra entre deux stations.

5.3 Conclusion

L'utilisation de la notation COMM pour la spécification de l'interaction des deux prototypes de poste de commande de drones montre en particulier la capacité de passage à l'échelle de la notation ainsi que la robustesse de l'éditeur e-COMM pour gérer une spécification complète correspondant à un arbre de tâches de 12 niveaux de profondeur et comportant un peu plus de 180 tâches.

L'interaction pour l'opérateur Vecteur et une partie de l'interaction pour l'opérateur Charge Utile a été spécifiée avec la notation COMM. Celle-ci nous a permis de décrire l'interaction aussi bien au niveau abstrait que concret notamment les commandes complexes à la souris sur des artefacts graphiques. Ces spécifications ont été exploitées par les partenaires du projet et en particulier les collègues de Bertin qui ont commenté et modifié les spécifications.

Néanmoins toutes les capacités de la notation COMM n'ont pas été illustrées par la spécification du poste de commande drones :

- Premièrement, pour décrire l'interaction, nous avons distingué l'interaction liée aux postes des deux opérateurs Vecteur et Charge Utile au sein de deux séries d'arbres de tâches distincts puisque le système est faiblement couplé et peu visqueux : l'interaction d'un utilisateur avec le système n'a pas un impact immédiat sur l'interaction de l'autre. En effet les conséquences sont visibles à plus long terme et peuvent être anticipées par les opérateurs. Par exemple lorsque l'opérateur Vecteur dirige un drone dans une direction qui empêcherait l'opérateur Charge Utile d'effectuer des observations, celui-ci peut anticiper cette action et parvenir par la discussion à un accord avec l'opérateur Vecteur pour rectifier la trajectoire du drone ou pour différer dans le temps les observations.
- Deuxièmement, l'éditeur d'arbre de tâches e-COMM ne dispose pas encore de la possibilité de décrire des *Template*. Or la description de l'interaction du poste de commande aurait pu être plus concise en utilisant des *Template*. En effet, de nombreuses tâches d'interaction concrètes sont raffinées par un glisser-déposer à la souris ou par une interaction identique sur une série d'objets (tels que la définition de consignes).

L'usage de la notation COMM et de l'éditeur e-COMM pour spécifier un système réel nous a permis d'enrichir la notation. En effet, nous rappelons que c'est la version préliminaire de la notation COMM

(non présentée dans ce manuscrit) qui a été initialement utilisée pour spécifier le premier prototype. En particulier, l'utilisation de COMM pour spécifier le premier prototype a permis d'identifier le besoin de décrire des contextes d'utilisation, autrement que par des annotations informelles. Nous avons alors enrichi la notation pour permettre d'associer un contexte d'utilisation à une tâche par le biais d'étiquettes numérotées.

Par la suite, le second prototype a été spécifié directement avec la notation COMM au travers de l'éditeur e-COMM, ce qui montre son opérationnalisation dans un projet réel et de grande envergure. L'utilisation de l'éditeur e-COMM pour décrire les spécifications a cependant souligné l'absence de lien (informatisé) entre les différents arbres de tâches qui ont été spécifiés. Ce problème n'a pas été identifié lors de l'expérimentation avec les étudiants (section 4) car la spécification correspondait à un arbre unique. Le passage à l'échelle étudié dans le cas du projet PEA FH/PA a souligné ce problème, dès lors que la spécification est conséquente et nécessite de décomposer l'arbre de tâches en plusieurs arbres. L'éditeur permet de créer plusieurs arbres de tâches pour décomposer une spécification d'un projet : ainsi un arbre de tâches peut avoir comme racine une tâche feuille d'un autre arbre. Néanmoins il s'agit alors de deux tâches distinctes et aucun lien n'est maintenu par l'éditeur. Une amélioration substantielle de notre éditeur consisterait à détacher la représentation de l'arbre de son modèle. Cela permettrait de proposer un système de navigation dans l'arbre qui permette à l'utilisateur d'afficher (et d'éditer) que la portion de l'arbre qui l'intéresse, et non la totalité de l'arbre. Le coût de développement étant élevé, cette modification n'a pas encore été réalisée et constitue une des extensions pratiques à nos travaux.

6 Conclusion : évaluation de COMM et e-COMM

Dans ce chapitre, nous avons présenté les évaluations que nous avons conduites concernant la notation COMM et l'éditeur e-COMM. Nous avons adopté une approche d'évaluation à la fois conceptuelle en étudiant la notation au regard des critères de Green [Green 2000] et Olsen [Olsen 2007], et expérimentale en utilisant la notation pour spécifier des systèmes. Nous avons présenté trois évaluations expérimentales ou expériences d'utilisation de la notation et de l'éditeur :

1. la spécification de l'interaction de l'étude de cas WCCM (Warcraft Collaboratif, Coopératif et Multimodal) que nous avons élaborée comme experts de la notation,
2. la spécification par 16 ingénieurs-concepteurs de l'interaction d'un système collaboratif d'édition de posters,
3. la spécification par les membres du PEA FH/PA de l'interaction du prototype de poste de commande de drones.

Ces trois expériences d'utilisation nous ont permis de montrer les points suivants :

- La notation permet de décrire des interactions multiutilisateurs coopératives et collaboratives, ainsi que des interactions concrètes multimodales, illustrées dans la spécification COMM de l'étude de cas WCCM. Nous avons montré au Chapitre 2 que les notations existantes ne permettaient pas de décrire de manière satisfaisante cette étude de cas. Nous illustrons ainsi l'apport de la notation à l'état de l'art des notations existantes.

- La notation COMM peut être utilisée par des utilisateurs de la population cible. La notation est donc compréhensible et utilisable par les concepteurs d'interfaces utilisateur.
- La notation COMM peut être utilisée dans le cadre d'un projet d'envergure. La notation passe à l'échelle de projets complexes.

Les résultats de ces évaluations expérimentales menées tout au long de nos travaux ont permis de faire évoluer la notation vers sa version finale mais aussi d'améliorer l'éditeur. Néanmoins des modifications de l'éditeur e-COMM restent à faire et constituent des extensions pratiques à nos travaux.

Enfin, en début de chapitre, nous avons souligné la difficulté d'évaluation d'une notation de spécification de l'interaction. Aussi, dans le temps imparti d'une thèse, nous sommes conscients que nos évaluations sont partielles. D'autres évaluations doivent être menées. Ceci constitue des extensions à court terme à nos travaux que nous décrivons dans le chapitre suivant.

Chapitre 6 : Conclusion et perspectives

Les travaux présentés dans ce mémoire concernent la conception des systèmes interactifs multiutilisateurs et multimodaux, c'est-à-dire des systèmes permettant à plusieurs utilisateurs d'interagir ensemble simultanément ou les uns après les autres, à l'aide de techniques d'interaction variées, parfois concurrentes. Ce travail a été motivé par le constat qu'il existe un nombre croissant de systèmes de ce type mais que leur conception est souvent *ad hoc*, du fait du manque d'outils adaptés aux besoins spécifiques liés au couplage des deux facettes de l'interaction multiutilisateur et multimodale. Aussi, les contributions de ce travail de thèse visent à combler cette lacune, en particulier par la proposition d'une nouvelle notation de spécification, nommée COMM (COllaborative et MultiModale), couvrant conjointement ces deux formes d'interaction, multiutilisateur et multimodale. Pour cela, nous avons étudié en détail les notations issues du domaine du TCAO et celles issues du domaine de la multimodalité. De plus, la notation COMM est instrumentée grâce au développement d'un éditeur logiciel nommé e-COMM.

Pour conclure ce manuscrit, nous résumons nos contributions conceptuelles et pratiques en soulignant les points forts. Puis, nous présentons une analyse critique de ce travail pour dégager des perspectives à moyen et long termes.

1 Résumé des contributions

1.1 Analyse détaillée des notations existantes

La notation COMM et l'éditeur, e-COMM sont le fruit d'une réflexion menée suite à une analyse approfondie de notations existantes (Chapitre 2) issues des domaines du TCAO et de la multimodalité. Afin d'analyser et comparer les différentes notations, nous avons adopté une démarche basée sur l'élaboration d'une grille d'analyse couvrant quatre catégories de caractéristiques : (1) des caractéristiques générales, communes à toute notation de spécification ; (2) celles relevant de l'interaction homme-machine ; (3) celles spécifiques à l'interaction multiutilisateur ; et (4) celles spécifiques à l'interaction multimodale.

Afin de remplir cette grille, nous avons construit trois exemples décrivant trois types d'interaction multiutilisateur et multimodale, inspirés d'une application existante tirée de la littérature du domaine [Tse 2007] : WCCM (Warcraft Collaboratif Coopératif et Multimodal). L'intérêt de cette approche est double. D'une part, cela permet la comparaison entre les notations sur une même base concrète et permet ainsi de mettre explicitement en lumière les similarités, les différences et les lacunes. D'autre part, l'approche est très pédagogique en facilitant la compréhension de l'analyse par un discours concret appliqué à une application unique.

Nous avons effectué une analyse conséquente la plus exhaustive possible, de 25 notations, dont 6 issues du domaine du TCAO, en les comparant sur la base de 18 caractéristiques. L'analyse fine de cette grille, présentée en conclusion du Chapitre 2, permet de mettre en évidence les atouts et les lacunes des notations existantes et permet ainsi de jeter les bases de la notation COMM.

Cet état de l'art et la grille sur lequel il s'appuie constituent une contribution originale car nous avons considéré de façon conjointe deux aspects de l'interaction, l'interaction multiutilisateur et multimodale. Outre son utilité dans notre démarche de travail, cet état de l'art et la grille peuvent servir de base, voire de point de référence, à d'autres travaux.

1.2 Notation COMM

Les lacunes des notations existantes identifiées à l'issue de notre analyse et présentées en synthèse du Chapitre 2, nous ont conduits à proposer une nouvelle notation, nommée COMM. Le Chapitre 3 détaille les aspects qui la caractérisent. Celle-ci constitue une extension de la notation d'arbre de tâches CTT. Les concepts clés de la notation COMM incluent :

- Le concept de tâches multiutilisateurs qui permet de décrire l'interaction multiutilisateur de manière plus souple et plus concise que les concepts de tâches coopératives et collaboratives présents dans d'autres notations ;
- Le concept de rôle interactif permettant de décrire des associations utilisateurs-tâches. Ce type de description offre une vision de l'interaction centrée sur l'utilisateur. En effet ce type d'association est complémentaire à celui d'association classique rôles-tâches (au sens de rôle métier), ce dernier offrant avant tout une vision de l'aspect organisationnel de l'activité de groupe. Ainsi, le concept de rôle interactif permet de spécifier aisément les cas d'interaction collaborative ;
- Le concept de tâche modale et les opérateurs issus des relations temporelles de Allen pour décrire finement l'interaction multimodale ;
- Un mécanisme de *Template* permettant de gagner en concision.

Pour présenter la notation COMM, le Chapitre 3 adopte la notation CTT comme point de départ pour proposer des extensions et affinements successifs. Outre l'aspect pédagogique, cette démarche de présentation permet le développement d'une démonstration sur l'apport des différents concepts de la notation COMM, d'argumenter et de justifier l'intérêt de ces concepts, et, enfin, de souligner leur articulation cohérente au sein de celle-ci. De plus, cette démarche permet de montrer que la notation répond point par point aux lacunes identifiées lors de l'analyse de l'existant (Chapitre 2). Enfin, la présentation de la notation COMM s'appuie sur les mêmes exemples utilisés pour l'analyse des notations existantes : ces exemples permettent d'étayer et d'illustrer la démonstration, mais aussi de mettre en évidence les apports des nouveaux concepts au regard des notations existantes.

1.3 Editeur e-COMM

La notation COMM est instrumentée par un éditeur logiciel, nommé e-COMM, présenté au Chapitre 4. L'usage d'une notation est facilité s'il existe des outils idoines, en particulier des outils logiciels [Caffiau 2010]. De plus, de tels outils contribuent à populariser les notations comme c'est le cas pour la notation CTT [Mori 2002]. Ce constat nous a donc conduits à concevoir et développer un éditeur sous la forme d'une application web riche basée sur la technologie *SilverLight* de Microsoft.

Le Chapitre 4 détaille la conception et le développement de l'éditeur e-COMM. Pour la conception de l'éditeur, nous avons d'abord étudié et parfois testé (si disponibles) les éditeurs existants adossés à des notations, tels que CTTE, K-MADe ou Euterpe (GTA). Un point important identifié lors de cette analyse et qui est central dans notre rationnel de conception est de favoriser la manipulation directe

des concepts de la notation lors de l'édition de la spécification. De plus, les problèmes relatifs aux différences entre systèmes d'exploitation et d'installation d'applications logicielles, ainsi que l'existence de technologies fiables autour du web sont autant de raisons qui nous ont convaincus de développer une application web.

e-COMM est une application complète dédiée à l'édition d'arbres de tâches et qui couvre l'ensemble des concepts de la notation COMM. Ainsi, e-COMM est publiquement accessible sur internet et utilisable directement au sein d'un navigateur internet. Le développement complet et abouti de l'éditeur ainsi que les différents formats d'exportation des spécifications permettent son utilisation effective au sein d'un cycle de développement de systèmes interactifs. La réalisation de l'éditeur e-COMM constitue une contribution logicielle conséquente et aboutie qui est incontournable pour promouvoir l'utilisation de la notation COMM.

1.4 Evaluation de la notation COMM et de son éditeur

L'évaluation d'une notation est un exercice complexe mais aussi difficilement totalement réalisable dans le temps imparti d'une thèse. En effet, il est difficile de saisir toutes les qualités objectives d'une notation et de déterminer pourquoi une notation est plus populaire, et donc utilisée, qu'une autre tant les facteurs subjectifs sont aussi nombreux. La très grande variété des facteurs objectifs ou subjectifs explique sans doute que très peu d'évaluations ont été conduites et documentées [Coutrix 2009]. Néanmoins la littérature révèle différentes formes d'évaluation : (1) des démonstrations/évaluations conceptuelles comme celle de la notation K-MAD et son éditeur K-MADe dans [Caffiau 2010] ou (2) des évaluations expérimentales comme celle de la notation de conception de l'interaction multimodale dans [Serrano 2010] ou celle de K-MAD/K-MADe dans [Caffiau 2010]. Notons qu'aucune évaluation des notations spécifiques aux systèmes interactifs multiutilisateurs et multimodaux n'a été effectuée et documentée.

Face à cette difficulté d'évaluation, notre approche a consisté à mener deux types d'évaluation complémentaires : (1) la première que nous qualifions de conceptuelle car cette forme d'évaluation n'a pas été faite par l'expérience mais par une approche analytique en étudiant la notation COMM au regard des critères de la littérature (inspection de la notation, sans l'appliquer pour spécifier un système), (2) la deuxième expérimentale en nous appuyant sur trois expériences.

La première évaluation expérimentale a consisté à considérer la même application WCCM (Warcraft Collaboratif Coopératif et Multimodal) et trois types d'interaction multiutilisateur et multimodale pour étudier et comparer les notations existantes et COMM. Ainsi dans le Chapitre 2, nous modélisons WCCM avec les notations existantes et identifions leurs limites en terme de pouvoir d'expression qui ont mené à définir COMM. De plus au Chapitre 5 consacré à l'évaluation de nos contributions, nous modélisons complètement WCCCM selon COMM avec l'éditeur e-COMM, soulignant ainsi le pouvoir d'expression et de concision de COMM au regard des notations existantes et le fait que l'éditeur e-COMM couvre l'ensemble des concepts de COMM.

De façon complémentaire à cette évaluation centrée sur la comparaison avec les notations existantes, nous avons mené deux autres formes d'évaluation expérimentale.

- La première évaluation a consisté en une expérience d'utilisation de la notation COMM et de l'éditeur e-COMM sur la base d'un scénario, avec 16 étudiants de Master 2 Professionnel

Génie Informatique : concevoir un outil de prototypage de posters, outil collaboratif et multimodal pour une société fictive d'édition de posters. Le but de cette expérience était de mesurer la facilité d'apprentissage des concepts de la notation et d'obtenir des retours sur l'usage de l'éditeur. La synthèse des observations et des questionnaires a mis en évidence l'assimilation des concepts de la notation et la facilité d'utilisation de l'éditeur. Toutefois, il convient de nuancer ces résultats car le concept de rôle interactif n'a pas été employé dans les différentes réalisations et les utilisateurs ont relevé l'absence de fonctions élémentaires comme le copier-coller dans e-COMM.

- La deuxième évaluation empirique a consisté à exploiter COMM/e-COMM comme support pour la phase de spécification dans un projet d'envergure dans le cadre d'un PEA (Programme d'Etude Amont), intitulé *Facteurs Humains et Partage d'Autorité* (FH-PA), à l'initiative de la DGA (Direction Générale des Armements), en partenariat avec les sociétés Bertin Technologies, EADS, Sagem et PY Automation, sur la conception d'une nouvelle génération de postes de commande de drones (projet achevé en Mars 2011). Ainsi, les solutions d'IHM produites au cours de ce projet ont été entièrement décrites avec la notation COMM, ce qui représente au final plus de 180 tâches avec des arbres de plus de 12 niveaux de profondeur. De cette expérience, nous avons conclu que l'éditeur a très nettement facilité la production des arbres de tâches et que la notation a bien joué son rôle d'outil de communication entre les différents partenaires. Cette étude a aussi montré que la notation COMM permet le passage à l'échelle en considérant un projet d'envergure.

Nous identifions pour nos travaux de multiples perspectives. Là encore, la dualité, notation et instrumentation de la notation apparaît dans nos propositions. Ces dernières s'organisent en deux parties : les extensions ou prolongements à moyen terme qui concernent les limites identifiées de nos résultats et les prolongements à plus long terme. Nous présentons dans la section suivante les extensions à moyen terme, avant d'introduire les perspectives à long terme.

2 Limitations et extensions

2.1 Enrichir la notation

L'évaluation de nos contributions, présentée au Chapitre 5, nous a permis d'identifier plusieurs limites. En particulier, l'utilisation de la notation dans le cadre du PEA FH-PA, au cours de la première itération, nous a fait prendre conscience que la notion de contexte d'utilisation, soulignée dans [Dix 2008], était absente de la notation. Ce constat nous a donc amené à ajouter à COMM la notion de contexte, dans sa version actuelle. Cependant, la description d'un contexte reste non structurée et optionnelle. Aussi, il conviendrait de mieux structurer la notion de contexte au sein de COMM, en intégrant des résultats issus, par exemple, de la littérature sur le contexte en IHM [Rey 2005]. Nous avons également observé que la notation ne prévoit pas de décrire explicitement la distinction entre interaction co-localisée et interaction distante. Il convient d'étudier la nécessité et les formes de représentation de la co-localisation de l'interaction (relations spatiales), afin de proposer une manière de la représenter au sein de COMM.

Plus généralement, ce travail s'est concentré sur la description de l'interaction en termes de tâches. Cependant, la description d'un système interactif ne peut se réduire à un arbre de tâches pour

décrire son interface homme-machine et son comportement. En effet, l'activité de conception et de spécification de l'interaction nécessite des représentations complémentaires telles que des maquettes graphiques ou des diagrammes de classes pour représenter les objets du domaine. Les objets manipulés par les tâches sont soulignés dans [Dix 2008] qui met en avant trois éléments principaux d'une tâche « Tâches = Données + Action + Contexte ». La spécification des objets représentés et des objets du domaine est d'autant plus indispensable que ces objets sont utilisés au sein des arbres de tâches. Aussi, une perspective à moyen terme est d'enrichir la notation avec des représentations complémentaires à la représentation d'arbre de tâches, qui s'intègrent harmonieusement à la version actuelle de la notation COMM, avec le même souci de cohérence qui a guidé son élaboration.

Enfin, il convient de mener une réflexion sur son usage puisque notre évaluation expérimentale a montré que certains concepts avancés (rôle interactif, opérateurs de Allen) posent des problèmes de compréhension aux utilisateurs novices. Dans un premier temps, il conviendra d'approfondir les évaluations expérimentales pour identifier la source de cette difficulté (les évaluations constituant une de nos perspectives à court terme décrite à la section 2.3).

2.2 Améliorer l'éditeur

L'amélioration de l'éditeur va de pair avec l'enrichissement de la notation. Comme souligné dans [Caffiau 2010], l'éditeur se doit d'être le banc d'essai, « Iron Bird », de la sémantique de la notation. Par exemple, notre perspective d'intégration de représentations complémentaires implique de développer de nouvelles fonctionnalités d'édition et de navigation.

Indépendamment de ces évolutions, la version actuelle de l'éditeur e-COMM souffre de l'absence de certaines fonctionnalités génériques des interfaces graphiques. Ces limitations ont été identifiées lors des deux évaluations empiriques expérimentales mais ne constituent néanmoins pas un frein à l'utilisation de la notation. Les améliorations à développer à court terme incluent :

- L'ajout de fonctionnalités telles que le copier-coller de portions d'arbre, le faire-défaire, ou des outils de navigation.
- L'intégration du mécanisme de *Template*. En effet, la description de l'IHM du poste de commande de drones dans le cadre du PEA FH-PA aurait pu gagner en concision avec l'usage des *Template* car de nombreuses tâches d'interaction sont identiques, comme la tâche de glisser-déposer à la souris.
- L'amélioration de l'import et export de projets au format XML et la génération graphique au format PDF.

2.3 Poursuivre l'évaluation

L'évaluation d'une notation, comme nous l'avons souligné, est une tâche complexe. Les résultats obtenus nous ont permis de nous conforter dans nos choix et de mettre en évidence des limites de la notation et de l'éditeur. Néanmoins il convient de mener d'autres évaluations en particulier expérimentales et sur la durée incluant une phase d'apprentissage de la notation.

Dans un premier temps, nous planifions en mai 2011 de mener de nouvelles expériences utilisateurs, dans le cadre du cours sur les collecticiels du Master 2 Professionnel Génie Informatique de Grenoble

(18 étudiants). Nous proposerons aux étudiants de décrire avec l'éditeur e-COMM une application multiutilisateur et multimodale, possiblement WCCM, afin de vérifier la compréhensibilité de la notation par les étudiants. De plus, nous prévoyons une période d'apprentissage de la notation importante dans le cadre du cours. Lors de l'expérimentation, il s'agira notamment de vérifier si les résultats obtenus lors de la première campagne d'évaluation sont confortés et si les concepts avancés de COMM seront cette fois totalement assimilés.

Parmi les autres formes d'évaluation, nous envisageons d'interroger les utilisateurs de l'éditeur e-COMM, qui sont enregistrés sur le site internet de la notation. Il s'agit de proposer à ces utilisateurs un questionnaire en ligne afin d'identifier les qualités et les défauts de l'éditeur, des manques de la notation et des attentes en termes de fonctionnalités.

Enfin, des enseignants en IHM[7] envisagent d'utiliser l'éditeur e-COMM dans le cadre de leur enseignement : ces expériences fourniront des retours complémentaires informels sur la notation et l'éditeur.

3 Perspectives à long terme : ouverture

Au sein de ce mémoire, nous avons identifié les enjeux de la réalisation de systèmes multiutilisateurs et multimodaux. Or, nos contributions traitent uniquement de la conception et spécification de l'interaction. Aussi, nous identifions deux pistes de recherche pour continuer et étendre nos travaux sur le plus long terme :

- La première consiste à intégrer finement la spécification de l'interaction au sein du processus de conception logicielle, c'est-à-dire d'adapter une méthode de conception existante afin qu'elle s'enrichisse de l'usage de la notation d'arbre de tâches COMM. Cette piste d'intégration de l'utilisation d'arbres de tâches COMM au sein du processus de conception logicielle pourrait ainsi conduire plus facilement à l'adoption des arbres de tâches par le monde industriel.
- La deuxième consiste à dépasser largement le cadre de la phase de conception et de spécification, pour traiter également les autres étapes du cycle de vie logiciel. Ainsi la génération de code applicatif ou de squelette [Sottet 2007], ainsi que l'intégration des spécifications au sein d'outils de développement peuvent constituer des extensions pertinentes pour la phase de développement. De même, la génération de cas de tests de l'interface utilisateur à partir des spécifications de l'interaction [Jambon 1999] [Madani 2009] est une perspective intéressante, visant à garantir une cohérence entre spécification de l'interaction et interface homme-machine finale.

[7] Cette perspective est le résultat d'échanges avec des enseignants en IHM lors de la démonstration de l'éditeur e-COMM à la conférence IHM 2010, du 20 au 23 septembre 2010 à Luxembourg.

Table des figures

Figure 1 : Objectif de nos travaux au sein d'un cycle de vie en V : phase de spécification. ---------- 18
Figure 2 : Démarche de travail : trois grandes étapes. ---------- 19
Figure 3 : Trois fonctions d'un collecticiel. ---------- 26
Figure 4 : Ontologie pour la description des tâches (illustration issu de [Veer 2000]).---------- 28
Figure 5 : Filtrage d'un geste intempestif dans le cadre du respect de la vie privée [Coutaz 1999].--- 30
Figure 6 : Vues (A) de dessus, (B) liée au camion et (C) liée à la lance.---------- 34
Figure 7 : Disposition du pilote de la lance (sujet) et du conducteur du camion (Magicien d'oz) pour l'expérimentation.---------- 35
Figure 8 : Perception de la dépendance du pilote selon la vue utilisée. ---------- 36
Figure 9 : Interface utilisateur du jeu « les 4 magiciens » (ici en mode à deux joueurs). ---------- 38
Figure 10 : Perception du couplage en fonction du mode d'utilisation. ---------- 39
Figure 11 : Met ça là, illustration tirée de [Bolt 1980]. ---------- 47
Figure 12 : Utilisation complémentaire de deux modalités : (wiimote, pointage) et (ballon, seuil de pression) [Juras 2008]. ---------- 48
Figure 13 : Poste de commande militaire, illustration issue de [Myers 2002]. ---------- 49
Figure 14 : Simulateur de conduite, illustration issue de [Sezgin 2009].---------- 49
Figure 15 : « Wiisoft » et Wiimote utilisés de manière redondante pour parcourir un diaporama [Serrano 2009].---------- 50
Figure 16 : Environnement collaboratif d'édition musicale, illustration issue de [Fiebrink 2009]. ----- 52
Figure 17 : Utilisation de Warcraft 3 collaboratif et multimodal, illustration issue de [Tse 2007].----- 52
Figure 18 : Les Sim's collaboratif et multimodal, illustration issue de [Tse 2007]. ---------- 53
Figure 19 : Interface collaborative pour l'édition de plans, illustration issue de [Tuddenham 2009].- 54
Figure 20 : Poste de commandement collaboratif pour la gestion de situations d'urgence, illustration issue de [Rauschert 2002]. ---------- 55
Figure 21 : Système de visite réelle et virtuelle, illustration issue de [Brown 2005]. ---------- 56
Figure 22 : Notations de spécification organisées selon leur domaine.---------- 79
Figure 23 : Diagramme de cas d'utilisation pour les activités (a) collaboratives, (b) coopératives de WCCM. ---------- 81
Figure 24 : Diagramme de séquence pour la création d'un fantassin.---------- 82
Figure 25 : Diagramme de communication pour la création d'un fantassin.---------- 83
Figure 26 : Diagramme d'activité pour la gestion d'un baraquement. ---------- 84
Figure 27 : Diagramme de classe UML.---------- 85
Figure 28 : Diagramme d'objet UML. ---------- 85
Figure 29 : Diagramme d'état-transition UML.---------- 86
Figure 30 : Diagramme d'interaction de WCCM.---------- 86
Figure 31 : Diagramme de classe UML-G.---------- 89
Figure 32 : Activité collaborative de création d'un soldat avec BPMN. ---------- 90
Figure 33 : Arbre de tâches cooperatives CTT. ---------- 92
Figure 34 : Arbres de tâches individuelles CTT pour le Sergent (A) et pour le Chef (B).---------- 93
Figure 35 : Arbre de tâches K-MAD de WCCM.---------- 94
Figure 36 : Modèle des concepts de WCCM avec RML. ---------- 96
Figure 37 : Modèle des rôles et utilisateurs de WCCM avec RML.---------- 96

Figure 38 : Modèle de tâche TaskMODL de WCCM. ---------- 97
Figure 39 : Modèle de l'interface graphique de WCCM. ---------- 98
Figure 40 : Modélisation du dialogue avec DiaMODL entre un utilisateur et le système pour le déplacement d'un fantassin dans WCCM.---------- 99
Figure 41 : Arbre de tâches coopératives de l'application WCCM avec CTML.---------- 101
Figure 42 : Description du site web d'une agence de voyage avec WebTaskModel [Bomsdorf 2007]. ---------- 102
Figure 43 : Modèle SpIeLan de l'interface de l'application WCCM (avec les *Canonical Abstract Prototypes*). ---------- 104
Figure 44 : Description de la tâche déplacer un groupe avec SpieLan.---------- 104
Figure 45 : Modélisation du comportement d'un soldat avec IOG. ---------- 106
Figure 46 : Description des rôles et agents avec GTA.---------- 107
Figure 47 : Modèle de tâche GTA de WCCM. ---------- 108
Figure 48 : Diagramme d'activité GTA. ---------- 108
Figure 49 : Modèles de rôles et des utilisateurs MABTA.---------- 110
Figure 50 : Modèle de tâche de groupe MABTA. ---------- 111
Figure 51 : Décomposition hiérarchique en sous-tâches individuelles. ---------- 112
Figure 52 : Sociogramme CIAN de l'application WCCM. ---------- 113
Figure 53 : Diagramme d'Inter-Action CIAN de l'application WCCM.---------- 114
Figure 54 : Description détaillée de la tâche collaborative de gestion des bâtiments.---------- 116
Figure 55 : Diagramme de structure organisationnelle.---------- 117
Figure 56 : Diagramme de collaboration la tâche de gestion des troupes.---------- 118
Figure 57 : Diagramme de flot de travail CUA. ---------- 119
Figure 58 : Description du scénario collaboratif « créer un soldat » selon CUA.---------- 120
Figure 59 : Modèle de tâches CUA pour le scénario « créer un soldat ».---------- 121
Figure 60 : Gestion des soldats modélisée avec Orchestra.---------- 122
Figure 61 : Création d'un fantassin modélisée avec Orchestra.---------- 123
Figure 62 : Configuration d'entrées (ICOM) pour WCCM. ---------- 127
Figure 63 : La création d'un soldat dans un baraquement selon la notation ICO.---------- 128
Figure 64 : Modélisation de l'interaction d'un utilisateur avec la table de WCCM selon la notation ICO. ---------- 129
Figure 65 : Assemblage de composants ICARE pour la tâche multimodale de déplacement d'un soldat. ---------- 130
Figure 66 : Interaction multimodale pour déplacer un groupe modélisée avec Dynamo-AID.-------- 132
Figure 67 : Contexte d'utilisation de la table augmentée de WCCM selon Dynamo-AID. ---------- 132
Figure 68 : Déplacer un soldat avec CIS. ---------- 134
Figure 69 : Description ASUR de l'interaction autour de la table de WCCM. ---------- 136
Figure 70 : Description ASUR de l'interaction avec un bâtiment Hôtel de ville pour la création d'un Paysan dans WCCM. ---------- 137
Figure 71 : Modèle IRVO pour la tâche de création d'un fantassin dans WCCM.---------- 139
Figure 72 : Diagramme d'interaction mixte (MIM) pour l'interaction avec un soldat dans WCCM. - 140
Figure 73 : Positionnement de la notation COMM dans le cycle de vie en V. ---------- 154
Figure 74 : Eléments et structure de la notation initiale. ---------- 158
Figure 75 : Limite de la description de la coopération avec un rôle joué par plusieurs utilisateurs.- 159

Figure 76 : Introduction des tâches abstraites pour décrire une coopération avec un rôle joué par plusieurs utilisateurs. ---------- 160
Figure 77 : Tâches collaboratives de gestion d'un baraquement. ---------- 160
Figure 78 : Utilisation de tâche coopérative pour palier aux limites de la modélisation de la collaboration. ---------- 161
Figure 79 : Introduction de tâche assignée à un rôle au sein d'une collaboration.---------- 161
Figure 80 : Utilisation de tâche coopérative pour palier aux limites de la modélisation de la collaboration. ---------- 162
Figure 81 : Eléments de la notation COMM après l'introduction de la tâche multiutilisateur.-------- 164
Figure 82 : Factorisation de la distribution des rôles dans un arbre de tâche. ---------- 164
Figure 83 : Association des rôles aux tâches à chaque étape du raffinement en sous-tâches.-------- 165
Figure 84 : Tâche gérer les soldats avec la spécification des rôles et du nombre d'utilisateur. ------- 165
Figure 85 : Eléments de la notation COMM après l'introduction des rôles et du nombre d'utilisateurs pour les tâches. ---------- 166
Figure 86 : Utilisation de rôle interactif pour décrire la tâche de gestion d'un baraquement.-------- 167
Figure 87 : Utilisation conjointe de rôle métier et de rôle interactif pour la tâche de gestion d'un baraquement. ---------- 168
Figure 88 : Eléments de la notation COMM après introduction du concept de rôle interactif. ------- 168
Figure 89 : Typage des tâches non-feuille dans CTT.---------- 169
Figure 90 : Modélisation des tâches abstraites et des tâches concrètes.---------- 173
Figure 91 : Eléments de la notation COMM après la mis en cohérence des différents type de tâches. ---------- 174
Figure 92 : Répétition parallèle de la tâche déplacer un soldat. ---------- 176
Figure 93 : Elément de la notation COMM après la refonte des opérateurs unaires d'itération. ---- 176
Figure 94 : Utilisation d'une tâche modale pour déplacer un soldat.---------- 180
Figure 95 : Utilisation du concept de tâche modale pour la tâche de gestion d'un baraquement. -- 181
Figure 96 : Utilisation des tâches modales pour la tâche de gestion d'un baraquement. ---------- 182
Figure 97 : Utilisation de tâche modale de type interaction de groupe pour exprimer la redondance. ---------- 182
Figure 98 : Tâches modales en entrée et en sortie pour la sélection d'un soldat.---------- 183
Figure 99 : Eléments de la notation COMM après l'adoption du concept de tâche modale. --------- 183
Figure 100 : Utilisation des opérateurs CTT pour exprimer l'usage complémentaire, redondant ou équivalent de modalité.---------- 185
Figure 101 : Les 5 relations temporelles de Allen.---------- 185
Figure 102 : Les 13 opérateurs issus des relations temporelles de Allen. ---------- 186
Figure 103 : Spécifier l'interaction multimodale avec un opérateur de Allen. ---------- 186
Figure 104 : Eléments de la notation COMM après adoption des opérateurs issus des relations temporelles de Allen. ---------- 187
Figure 105 : Utilisation d'une tâche d'action pour spécifier la sélection d'un soldat. ---------- 188
Figure 106 : Eléments de la notation COMM après l'adoption des tâches d'action et d'action de groupe. ---------- 189
Figure 107 : Division horizontale d'un arbre de tâches.---------- 191
Figure 108 : Définition (1) et utilisation (2) d'un template dans un diagramme de classe UML. ----- 192
Figure 109 : Définition (1) et utilisation (2) de template sur les arbres de tâches.---------- 192
Figure 110 : Définition (1) de contextes et utilisation (2) au sein d'un arbre de tâches.---------- 194

Figure 111 : Utilisation de pré-condition au sein d'un arbre de tâche COMM. ---------- 195
Figure 112 : Eléments de la version définitive de la notation COMM.---------- 197
Figure 113 : Interface de CTTE (http://giove.isti.cnr.it/tools/ctte/). ---------- 201
Figure 114 : Interface de K-MADe (http://kmade.sourceforge.net/). ---------- 202
Figure 115 : Interface de l'éditeur d'arbre de tâches intégré dans TOUCHE Case (http://www.penichet.net/).---------- 203
Figure 116 : Interface d'EUTERPE (http://www.cs.vu.nl/~gerrit/gta/euterpe.html). ---------- 204
Figure 117 : Editeur d'arbre de tâches de Dynamo-Aid, illustration tirée de [Clerckx 2005].---------- 205
Figure 118 : Editeur d'arbre de tâche CTML, illustration issue de [Wurdel 2008]. ---------- 206
Figure 119 : Déploiement de notre solution lors de l'exécution. ---------- 209
Figure 120 : Portail e-COMM (www.e-comm.fr.nf).---------- 211
Figure 121 : Barre d'accès rapide lorsque l'utilisateur est connecté au portail e-COMM. ---------- 212
Figure 122 : Page "Notation COMM" du portail e-COMM.---------- 213
Figure 123 : Page "Editeur e-COMM" du portail e-COMM. ---------- 214
Figure 124 : Page "Editeur e-COMM : vos projets" du portail e-COMM (requiert une identification). ---------- 215
Figure 125 : Page de gestion d'un projet du portail e-COMM. ---------- 215
Figure 126 : Un projet ouvert e-COMM. ---------- 216
Figure 127 : Spécification d'un rôle avec e-COMM.---------- 217
Figure 128 : Spécification du contexte d'utilisation avec e-COMM. ---------- 217
Figure 129 : Interface de l'éditeur d'arbre de tâches e-COMM suite à la création d'un nouvel arbre de tâches.---------- 218
Figure 130 : Création de deux sous-tâches. ---------- 219
Figure 131 : Spécification d'un opérateur CTT entre deux sous-tâches. ---------- 220
Figure 132 : Spécification d'un opérateur de Allen entre deux sous-tâches modales. ---------- 221
Figure 133 : Sélectionner une tâche.---------- 221
Figure 134 : Utilisation du lasso pour sélectionner plusieurs tâches.---------- 222
Figure 135 : Apparition des points d'insertion lors du déplacement d'une tâche. ---------- 223
Figure 136 : Déplacement d'une tâche dans un arbre en utilisant un point d'insertion. ---------- 224
Figure 137 : Rose des vents pour le déplacement et glissière pour le réglage du facteur de zoom.- 225
Figure 138 : Fenêtre modale de propriétés d'une tâche. ---------- 226
Figure 139 : Tâche (1) lorsqu'aucun contexte n'est défini, (2) lorsqu'au moins un contexte est défini. ---------- 227
Figure 140 : Fenêtre modale de sélection des contextes. ---------- 227
Figure 141 : Tâche Jouer à WCCM (Warcraft Collaboratif, Coopératif et Multimodal). ---------- 233
Figure 142 : Tâche Gérer les troupes dans WCCM.---------- 233
Figure 143 : Tâche Déplacer un soldat dans WCCM.---------- 234
Figure 144 : Tâches Former et Déplacer un groupe dans WCCM. ---------- 234
Figure 145 : Tâche Gérer les bâtiments dans WCCM. ---------- 235
Figure 146 : Tâche Gérer un baraquement dans WCCM. ---------- 235
Figure 147 : Mission de surveillance d'un convoi. ---------- 244
Figure 148 : Maquette : système de cartographie de l'opérateur Vecteur.---------- 246
Figure 149 : Maquette : système de contrôle de vol de l'opérateur Vecteur.---------- 247
Figure 150 : Maquette : système de cartographie de l'opérateur Charge Utile.---------- 248
Figure 151 : Maquette : système de visualisation 3D de l'opérateur Charge Utile. ---------- 249

Figure 152 : Arbre de tâches pour la gestion d'un drone tactique (DT). ---------- 250
Figure 153 : Tâche : indiquer l'état du drone tactique (Indiquer état DT). ---------- 250
Figure 154 : Tâche : indiquer l'état du drone tactique (détail de la partie rouge de la Figure 153). - 251
Figure 155 : Tâche : indiquer l'état du drone tactique (détail de la partie bleue de la Figure 153). - 251
Figure 156 : Tâche : indiquer l'état du drone tactique (détail de la partie verte de la Figure 153). -- 252
Figure 157 : Tâche : indiquer l'état du drone tactique (détail de la partie jaune de la Figure 153). - 252
Figure 158 : Modifier les paramètres de vol. ---------- 253
Figure 159 : Tache : modifier les paramètres de vol (détail de la partie rouge de la Figure 158). ---- 253
Figure 160 : Tâche : modifier les paramètres de vol (détail de la partie bleue de la Figure 158). ---- 254
Figure 161 : Tâche : modifier les paramètres de vol (détail de la partie verte de la Figure 158). ----- 255
Figure 162 : Tâche : contrôler la trajectoire du drone tactique (Contrôler trajectoire DT). ---------- 255
Figure 163 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie rouge de la Figure 162). ---------- 256
Figure 164 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie bleue de la Figure 162). ---------- 256
Figure 165 : Tâche : contrôler la trajectoire du drone tactique (détail de la partie verte de la Figure 162). ---------- 257
Figure 166 : Tâche : manipuler la cartographie (Naviguer vue carte). ---------- 257
Figure 167 : Tâche : manipuler la cartographie (détail de la partie rouge de la Figure 166). ---------- 258
Figure 168 : Tâche : manipuler la cartographie (détail de la partie bleue de la Figure 166). ---------- 258
Figure 169 : Tâche : manipuler la cartographie (détail de la partie verte de la Figure 166). ---------- 259
Figure 170 : Tâche : manipuler la cartographie (détail de la partie jaune de la Figure 166). ---------- 260
Figure 171 : Tâche : gérer un plan de vol. ---------- 260
Figure 172 : Tâche : éditer un plan de vol. ---------- 261
Figure 173 : Tâche : modifier les propriétés d'un *waypoint*. ---------- 261
Figure 174 : Tâche : modifier l'altitude. ---------- 262
Figure 175 : Tâche : gérer les zones (Gérer zone). ---------- 262
Figure 176 : Tâche : gérer les zones (détail de la partie rouge de la Figure 175). ---------- 263
Figure 177 : Tâche : gérer les zones (détail de la partie bleue de la Figure 175). ---------- 263
Figure 178 : Tâche : gérer la vue 3D. ---------- 264
Figure 179 : Tâche : gérer la vue 3D (détail de la partie rouge de la Figure 178). ---------- 264
Figure 180 : Tâche : gérer la vue 3D (détail de la partie bleue de la Figure 178). ---------- 265
Figure 181 : Tâche : partage d'une caméra entre deux stations. ---------- 266
Figure 182 : Spécification complète d'un étudiant n'ayant utilisé ni tâche modale, ni opérateur de Allen. ---------- 299
Figure 183 : Extrait de spécification d'un étudiant ayant utilisé des tâches modales pour décrire la création d'un « post ». ---------- 300
Figure 184 : Aperçu d'une spécification complète d'un étudiant. ---------- 300
Figure 185 : Détail de la zone rouge de la spécification complète de la Figure 184. ---------- 301
Figure 186 : Détail de la zone verte de la spécification complète de la Figure 184. ---------- 301

Publications

Revue internationale avec comité de lecture

A. Benoit, L. Bonnaud, A. Caplier, I. Damousis, F. Jourde, J-Y L. Lawson, L. Nigay, M. Serrano, D. Tzovaras Multimodal signal processing and interaction for a driving simulator : component-based architecture, Journal on Multimodal User Interface, Volume 1, Issue 1, March 2007, Springer-Verlag Berlin Heidelberg, pp. 49-58.

Conférences internationales avec comité de lecture

F. Jourde, Y. Laurillau, L. Nigay, COMM Notation for Spécifuing Collaborative and MultiModal Interactive Systems, Proceedings of the international conference on Engineering Interactive Computing Systems (EICS'10), Berlin, Germany, June, 2010, ACM Press, New York, NY, USA, pp. 125-134.

F. Jourde, Y. Laurillau, A. Moran L, Nigay, Towards Specifuing Multimodal Collaborative User Interfaces : A Comparison of Collaboration Notations, Proceedings of the international conference on Design, Specification and Verification of Interactive (DSV-IS'08), Kinston, Canada, July 2008, LNCS, Volume 5136, Springer-Verlag Berlin Heidelberg, pp. 281-286.

F. Jourde, L. Nigay and I. Parissis Formal test of interactive systems: ICARE-Lutess Conference Proceedings of ICSSEA'2006, 19th International Conference on Software & Systems Engineering and their Applications: Service & System globalization, Paris, France, December , 2006, 8 pages.

Conférences nationales avec comité de lecture

F. Jourde, Y. laurillau, L. Nigay, e-COMM, un éditeur pour spécifier l'interaction multimodale et multiutilisateur, In Actes de la 22ème Conférence francophone sur l'Interaction Homme-Machine (IHM 2010), Luxembourg, Septembre 2010, ACM Press, New York, NY, USA, pp. 225-228.

F. Jourde, Y. Laurillau, L. Nigay, Collecticiels : Neuf Degrés de Couplage, In Actes de la 21ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Grenoble, France, Octobre 2009, ACM Press, New York, NY, USA, pp. 265-272.

A. Leal, J. Bouchet, S. Langlois, F. Jourde, Conception de l'Interaction Homme-Machine et Partage d'Autorité: Application aux Systèmes de Drones, In Actes de la 21ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Grenoble, France, Octobre 2009, ACM Press New York, NY, USA, pp. 283-290.

F. Jourde, Y. Laurillau, L. Nigay. A. Moran, Conception de Systèmes Collaboratifs Multimodaux : analyse Comparative de Notations, In Actes de la 20ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Metz, France, Septembre, 2008, ACM Press, New York, NY, USA, pp. 121- 128.

Références

[Allen 1983] J. F. Allen, Maintaining Knowledge about Temporal Intervals. Communications of the ACM, Volume 26, Issue 11, November 1983, pp. 832-843.

[Appert 2007] C. Appert, Modélisation, Évaluation et Génération de Techniques d'Interaction, Thèse de doctorat de l'université Paris-Sud. 21 Mai 2007, Orsay, France.

[Bangor 2008] A. Bangor, P. T. Kortum, J. T. Miller, An Empirical Evaluation of the System Usability Scale, International Journal of Human-Computer Interaction, Volume 24, Issue 6, 2008, pp. 574-594.

[Baron 2006] M. Baron, V. Lucquiaud, D. Autard, D. L. Scapin, K-MADe : un environnement pour le noyau du modèle de description de l'activité, Proceedings of the 18th Internationnal Conference of the Association Francophone D'Interaction Homme-Machine (IHM'06), Montreal, Canada, 2006, ACM Press New York, NY, USA, pp. 287-288.

[Bass 1992] L. Bass, A metamodel for runtime architecture of an interactive system: the UIMS workshop tool developers, ACM SIGCHI Bulletin, Volume 24, Issue 1, 1992, pp. 32-37.

[Bastien 1992] J. M. C. Bastien, D.L. Scapin, A validation of ergonomic criteria for the evaluation of human-computer interfaces, International Journal of Human-Computer Interaction, Volume 4, Issue 2, 1992, pp. 183-196 .

[Beaudoin 2004] M. Beaudoin-Lafont, Designing interaction not interfaces, Proceedings of the working conference on Advanced visual interfaces (AVI'04), Gallipoli, Italy, 25-28 May, 2004, ACM Press New York, NY, USA, pp. 15-22.

[Bailly 2009] G. Bailly, Technique de menus : Caractérisation, Conception et Evaluation, thèse de doctorat de l'université Joseph Fourier, 26 mai 2009, 288 pages.

[Bergh 2006] J. Van den Bergh, K. Luyten, K. Coninx, High-Level Modeling of Multi-user Interactive Applications, Proceedings of the 5th international conference on Task models and diagrams for users interface design (TAMODIA'06), Springer-Verlag Berlin, Heidelberg, 2007, pp. 153-168.

[Bergh 2007] J. Van den Bergh, B. Bruynooghe, J. Moons, S. Huypens, K. Handekyn, K. Coninx, Model-Driven creation of staged participatory multimedia events on TV, Proceedings of the 5th European conference on Interactive TV: a shared experience (EuroITV'07), 2007, Springer-Verlag Berlin, Heidelberg, pp. 21-30.

[Bernsen 1994] N. O. Bernsen, Modality Theory In Support Of Multimodal Interface Design, Proceedings of Intelligent Multi-Media Multi-Modal Systems, 1994, pp. 37-44.

[BMPN] http://www.bpmn.org/

[Bolt 1980] R. A. Bolt, "Put that there": Voice and Gesture at the Graphics Interface: Proceedings of the 7th annual conference on Computer graphics and interactive techniques (SIGGRAPH '80), Volume 14, Issue 3, ACM Press New York, NY, USA, July 1980, pp. 262-270.

[Bomsdorf 2007] B. Bomsdorf, The WebTaskModel Approach to Web Process Modelling, Proceedings of the 6th international conference on Task models and diagrams for user interface design (TAMODIA'07), LNCS, Volume 4849, Springer-Verlag Berlin, Heidelberg, 2007, pp. 240-253.

[Bomsdorf 2009] B. Bomsdorf, D. Sinnig, Model-Based Specification and Validation of User Interface Requirements, Proceedings of the 13th International Conference on Human-Computer Interaction Part I New Trends (HCII 2009), LNCS, Volume 5610, Springer-Verlag Berlin, Heidelberg, 2009, pp. 587-596.

[Bouchet 2006] J. Bouchet, Ingénieurie de l'interaction multimodale en entrée : Approche à composant ICARE, Thèse de doctorat de l'université Joseph Fourier, Grenoble, 07 décembre 2006, 297 pages.

[Brown 2005] B. Brown, M. Chalmers, M. Bell, M. Hall, I. MacColl, P. Rudman, Sharing the square: Collaborative Leisure in the City Streets, Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work (ECSCW'05), Paris, France, 18-22 September, 2005, Springer-Verlag New York, NY, USA, pp. 427-447.

[Caffiau 2009] S. Caffiau, P. Girard, L. Guittet, D. L. Scapin, Hierarchical Structure: A Step for Jointly Designing Interactive Software Dialog and Task Model, Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques, LNCS, Volume 5611, 2009, Springer-Verlag Berlin, Heidelberg, pp. 664-673.

[Caffiau 2010] S. Caffiau, D. L. Scapin, P. Girard, M. Baron, F. Jambon, Increasing the expressive power of task analysis : Systematic comparison and empirical assessment of tool-supported task models, Journal Interacting with Computers, Volume 22, Issue 6, November, 2010, Elsevier Science Inc. New York, NY, USA, pp. 569-593.

[Carr 1994] D. A. Carr, Specification of interface interaction objects, Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence (CHI(94), 1994, ACM, New York, NY, USA, pp. 372-378.

[Chalon 2005] R. Chalon, B. David, IRVO: an Interaction Model for Designing Collaborative Mixed Reality Systems, Proceedings of the 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA, 22-27 June, 2005, pp. 1-10.

[Charfi 2009] S. Charfi, Conception et Evaluation des Systèmes Interactifs Mixtes selon une Approche Centrée Utilisateur, Thèse de l'université de Toulouse, 08 juillet 2009, 326 pages.

[Clerkx 2005] T. Clerckx, F. Winters, K. Coninx, Tool Support for Designing Context Sensitive User Interfaces using a Model-Based Approach, Proceedings of the 4th international workshop on Task models and diagrams (TAMODIA'05), Gdansk, Poland, 26–27 September, 2005, ACM New York, NY, USA, pp. 11-18.

[Clerkx 2008] T. Clerckx, C. Vandervelpen, K. Coninx, Task-Based Design and Runtime Support for Multimodal User Interface Distribution, Engineering Interactive Systems: EIS 2007 Joint Working Conferences, EHCI 2007, DSV-IS 2007, HCSE 2007, Salamanca, Spain, March 22-24, 2007, Springer-Verlag Berlin, Heidelberg, pp. 89-105.

[Constantine 2003] L. Constantine, Canonical Abstract Prototypes for Abstract Visual and Interaction Design, Proceedings, 10th International Workshop in Interactive Systems: Design, Specification, and Verification (DSV-IS 2003), 2003, LNCS, Volume 2844, Springer-Verlag Berlin Heidelberg, pp. 1-15.

[Coutaz 1994] J. Coutaz, L. Nigay, Les propriétés "CARE" dans les interfaces multimodales, Actes des sixièmes journées sur l'ingénierie des interfaces Homme-Machine (IHM'94). Lille, 8-9 décembre, 1994, pp. 7-14.

[Coutaz 1995] J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. M. Young, Four easy pieces for assessing the usability of multimodal interaction: the care properties, IFIP Conference Proceedings of Interantional Conference on Human-Computer Interaction (INTECTACT'95), 27-29 June, 1995, Chapman & Hall, pp. 115-120.

[Coutaz 1998] J. Coutaz, Interfaces Homme-Machine : "le Futur ne Manque pas d'Avenir", In conférence invitée, Proceedings of ERGO-IA, 1998, pp. 43-55.

[Coutaz 1999] J. Coutaz, F. Bérard, E. Carraux, W. Astier, J. L. Crowley, CoMedi: Using Computer Vision to Support Awareness and Privacy in Mediaspaces, Actes (extension) de la conférence ACM Conference on Human Factors and Computing Systems (CHI'99), 1999, ACM Press New York, NY, USA, pp. 13-14.

[Coutrix 2009] C. Coutrix, Interfaces de Réalité Mixte : Conception et Prototypage, Thèse de doctorat de l'université Joseph Fourier, 7 mai 2009, 392 pages.

[Dam 2007] A. van Dam, Post-WIMP User Interfaces, Communications of the ACM, Volume 40, Issue 2, February 1997, pp. 63-67.

[David 2006] B. David, R. Chalon, O. Delotte, G. Masserey, M. Imbert, ORCHESTRA: Formalism to Express Mobile Cooperative Applications, Proceedings of the 12th international workshop Researchers International Working Group (CRIWG 2006), 2006, LNCS, Volume 4154, Springer-Verlag Berlin Heidelberg, pp. 163-178.

[David 2007] B. David, R. Chalon, O. Delotte, G. Masserey, ORCHESTRA : Formalism to Express Static and Dynamic Model of Mobile Collaborative Activities and Associated Patterns, Proceedings of the 12th international conference on Human-computer interaction: interaction design and usability Part I (HCII 2007), LNCS 4550, 2007, Springer-Verlag Berlin, Heidelberg, pp. 1082-1091.

[David 2009] B. David, R. Chalon, Orchestration Modeling of Interactive Systems, Proceedings of the 13th International Conference on Human-Computer Interaction. Part I: New Trends (HCII 2009), 2009, LNCS 5610, Springer-Verlag Berlin, Heidelberg, pp. 796-805.

[Dillenbourg 2006] P. Dillenbourg, M. Baker, A. Blaye, C. O'Malley, The evolution of research on collaborative learning. Learning in Humans and Machine: Towards an interdisciplinary learning science, 1996, Elsevier, pp. 189-211.

[Dix 2008] A. Dix, Tasks=Data+Action+Context: Automated Task Assistance through Data-Oriented Analysis, Proceedings of Engineering Interactive Systems (HCSE'08 and TAMODIA'08), Pisa, Italy, 2008, LNCS, Volume 5247, Springer-Verlag Berlin, Heidelberg, pp. 1-13.

[Dubois 2001] E. Dubois, Chirurgie Augmentée, un Cas de Réalité Augmentée ; Conception et Réalisation Centrées sur l'Utilisateur. Thèse de doctorat de l'université Joseph Fourier, Grenoble, 2001.

[Dubois 2002a] E. Dubois, P. P. da Silva, P. Gray, Notational Support for the Design of Augmented Reality Systems, Proceedings of Interactive Systems: Design, Specification, and Verification (DSV-IS'02), 2002, LNCS 2545, Springer-Verlag Berlin Heidelberg, pp. 74-88.

[Dubois 2002b] E. Dubois, P. Gray, L. Nigay, ASUR++: A Design Notation for Mobile Mixed Systems, F. Paternò (Ed.): Mobile HCI'02, 2002, LNCS Volume 2411, Springer-Verlag Berlin Heidelberg, pp. 123-139.

[Dragicevic 2001] P. Dragicevic, J. D. Fekete, Input Device Selection and Interaction Configuration with ICON, People and Computers XV Interaction without Frontiers: Joint proceedings of IHM 2001 and HCI 2001 (IHM-HCI '01), 2001, Springer-Verlag Berlin Heidelberg, pp. 543-558.

[Dragicevic 2004] P. Dragicevic, J. D. Fekete, Support for Input Adaptability in the ICon Toolkit, Proceedings of the Sixth International Conference on Multimodal Interfaces (ICMI '04), State College, PA, USA, October 13 - 15, 2004. ACM Press, New York, NY, pp. 212-219.

[Ellis 1991] Ellis, Clarence, Gibbs, Simon et Rein, Gail. Groupware: Some Issues and Experiences. Journal, Communications of the ACM (CACM), Volume 34, Issue 1, 1991, pp. 38-58.

[Fekete 2009] J D. Fekete, N. Elmqvist, Y. Guiard, Motion-Pointing: target selection using Elliptical Motions, Proceedings of the 27th international conference on Human factors in computing systems (CHI'09), Boston, Massachusetts, USA, April 4 - 9, 2009, ACM Press, New York, NY, USA, pp. 289-298.

[Fiebrink 2009] R. Fiebrink, D. Morris, M. R. Morris, Dynamic Mapping of Physical Controls for Tabletop Groupware, Proceedings of the 27th international conference on Human factors in computing systems (CHI'09), Boston, Massachusetts, USA, April 4 - 9, 2009, ACM Press, New York, NY, USA, pp. 471-480.

[Fishkin 2000] K. P. Fishkin, A. Gujar, B. L. Harrison, T. P. Moran, R. Want, Communications of the ACM, Volume 43, Issue 9, 2000, pp. 74-80.

[Francone 2010] J. Francone, G. Bailly, E. lecolinet, N. Mandran, L. Nigay. Wavelet Menus on Handheld Devices: Stacking Metaphor for Novice Mode and Eyes-Free Selection for Expert Mode, Proceedings of the International Conference on Advanced Visual Interfaces (AVi'10), Roma, Italy, May 25-29, 2010, ACM Press, New York, NY, USA, pp. 173-180.

[Grudin 1994] J. Grudin, CSCW: History and Focus, Jounal IEEE Computer, Volume 27, Issue 5, 1994, pp. 19-26.

[Green, 2000] T. R. G. Green, Instructions and descriptions: some cognitive aspects of programming and similar activities, Proceedings of the working conference on Advanced visual interfaces (AVI '00), 2000, ACM New York, NY, USA, pp. 21-28.

[Hartson 1990] H. R. Hartson, A. C. Siochi, D. Hix, The UAN: a user-oriented representation for direct manipulation interface designs, ACM Transactions on Information Systems (TOIS), Volume 8, Issue 3, July 1990, ACM Press, New York, NY, USA, pp. 181-203.

[Jambon 1999] F. Jambon, P. Girard, Y. Boisdron. Dialogue validation from task analysis, Proceedings of the Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'99), Braga, Portugal, 1999, Springer-Verlag Berlin Heidelberg, pp. 205-224.

[Jourde 2008a] F. Jourde, Y. Laurillau, A. Moran, L. Nigay, Towards Specifuing Multimodal Collaborative User Interfaces : A Comparison of Collaboration Notations, Proceedings of the international conference on Design, Specification and Verification of Interactive (DSV-IS'08), Kinston, Canada, July 2008, LNCS, Volume 5136, Springer-Verlag Berlin Heidelberg, pp. 281-286.

[Jourde 2008b] F. Jourde, Y. Laurillau, L. Nigay. A. Moran, Conception de Systèmes Collaboratifs Multimodaux : analyse Comparative de Notations, In Actes de la 20ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Metz, France, 2- 5 Septembre, 2008, ACM Press, New York, NY, USA, pp. 121- 128.

[Jourde 2009] F. Jourde, Y. Laurillau, L. Nigay, Collecticiels : Neuf Degrés de Couplage, In Actes de la 21ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Grenoble, France, Octobre 2009, ACM Press, New York, NY, USA, pp. 265-272.

[Jourde 2010a] F. Jourde, Y. Laurillau, L. Nigay, COMM Notation for Spécifuing Collaborative and MultiModal Interactive Systems, Proceedings of the international conference on Engineering Interactive Computing Systems (EICS'10), Berlin, Germany, 19-23 June, 2010, ACM Press, New York, NY, USA, pp. 125-134.

[Jourde 2010b] F. Jourde, Y. laurillau, L. Nigay, e-COMM, un éditeur pour spécifier l'interaction multimodale et multiutilisateur, In Actes de la 22ème Conférence francophone sur l'Interaction Homme-Machine (IHM 2010), Luxembourg, Septembre 2010, ACM Press, New York, NY, USA, pp. 225-228.

[Juras 2008] D. Juras, L. Nigay, M. Ortega, M. Serrano, Proceedings of the 10th international conference on Multimodal interfaces (ICMI'08), Chania, Crete, Greece, 20-22 October, 2008, ACM Press, New York, NY, USA, pp.193-194.

[Kaiser 2006] E. C. Kaiser, Using redundant speech and handwriting for learning new vocabulary and understanding abbreviations, Proceedings of the 8th international conference on Multimodal interfaces (ICMI'06), 2006, ACM Press, New York, NY, USA, pp. 347-356.

[Koike 2009] H. Koike, W. Nishikawa, K. Fukuchi, Transparent 2-D Markers on an LCD Table top System, Proceedings of the 27th international conference on Human factors in computing systems (CHI'09), Boston, Massachusetts, USA, 4-9 April, 2009, ACM Press, New York, NY, USA, pp. 163-172.

[König 1999] W. A. König, R. Rädle, H. Reiterer, Squidy: a zoomable design environment for natural user interfaces, Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI EA '09), 2009, ACM Press, New York, NY, USA, pp. 4561-4566.

[Krahnstoever 2002] N. Krahnstoever, S. Kettebekov, M. Yeasin, R. Sharma, A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays, Proceedings of the 4th IEEE International Conference on Multimodal Interfaces (ICMI'02), 2002, IEEE Computer Society Washington, DC, USA, pp. 349-355.

[Lalanne 2009] D. Lalanne, L. Nigay, P. Palanque, P. Robinson, J. Vanderdonckt, J. F. Ladry, Fusion engines for multimodal input: a survey, Proceedings of the 10th international conference on Multimodal interfaces (ICMI'08), Chania, Crete, Greece, 20-22 October, 2008, ACM Press, New York, NY, USA, pp. 153-160.

[Laurillau 2002] Y. Laurillau, Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe : le modèle et la plate-forme Clover, Thèse de doctorat de l'université Joseph Fourier Grenoble 1, 10 septembre 2002, 232 pages.

[Leal 2009] A. Leal, J. Bouchet, S. Langlois, F. Jourde, Conception de l'Interaction Homme-Machine et Partage d'Autorité: Application aux Systèmes de Drones, In Actes de la 21ème Conférence francophone sur l'Interaction Homme-Machine (IHM'09), Grenoble, France, Octobre 2009, ACM Press New York, NY, USA, pp. 283-290.

[Lim 2004] Y. K. Lim, Multiple Aspect Based Task Analysis (MABTA) for User Requirements Gathering in Hightly-contextualized Interactive System Design, Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA '04), Prague, Czech Republic, 15-16 November, 2004, ACM, New York, NY, USA, pp. 7-15.

[Lucquiaud 2005] V. Lucquiaud, Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs, Proceedings of the 17th French-speaking conference on Human-computer interaction (IHM'05), Toulouse, France, September 27-30 2005, ACM, New York, NY, USA, pp. 83-90.

[Nigay 1996] L. Nigay, J. Coutaz, Espaces conceptuels pour l'interaction multimédia et multimodale. TSI, spécial Multimédia et Collecticiel, AFCET & Hermes Publ., Volume 15, Issue 9, 1996, pp. 1195-1225.

[Madani 2009] L. Madani, I. Parissis, Automatically Testing Interactive Applications Using Extended Task Trees, Journal of Logic and Algebraic Programming, Volume 78, Issue 7, August - Septembre 2009, Elsevier, pp. 454-471.

[Martin 1999] Martin, J. C. TYCOON: six primitive types of cooperation for observing, evaluating and specifying cooperations. Working notes of the AAAI Fall 1999 Symposium on Psychological Models of Communication in Collaborative Systems. 1999. pp. 61-66.

[Mejia 2007] D. A. Mejia, A. L. Moran, J. Favela, Supporting Informal Co-located Collaboration in Hospital Work Proceedings of the 13th international conference on Groupware: design implementation, and use (CRIWG'07), Bariloche, Argentina, 16-20, September, 2007, LNCS , Volume 4715, Springer-Verlag Berlin, Heidelberg, pp. 255-270.

[Molina 2006] A.I. Molina, M.A. Redondo, and M. Ortega, A Conceptual and Methodological Framework for Modeling Interactive Groupware Applications, Process, Proceedings of the 12th international conference on Groupware: design implementation, and use (CRIWG'06), 2006, LNCS, Volume 4154, Springer-Verlag Berlin Heidelberg, pp. 413-420.

[Molina 2007] A.I. Molina, W.J. Giraldo, M.A. Redondo, and M. Ortega, A Proposal of Integration of the GUI Development of Groupware Applications into the Software Development Process, Proceedings of the 13th international conference on Groupware: design implementation, and use (CRIWG'07), 2007, LNCS, Volume 4715, Springer-Verlag Berlin, Heidelberg. pp. 111-126.

[Molina 2008] A.I. Molina, M.A. Redondo, M.Ortega, U. Hoppe, CIAM: A Methodology for the Development of Groupware User Interfaces, Journal of Universal Computer Science, Volume 14, Issue 9, May 2008, 1435-1446.

[Molina 2009] A. I. Molina, M. A. Redondo, M. Ortega, A Review of Notations for Conceptual Modeling of GroupWare Systems, New Trends on Human-Computer Interaction, Chapter 8, 2009, Springer-Verlag London, pp. 75-86.

[Mori 2002] G. Mori, F. Paternò, C. Santoro, CTTE: Support for Developing and Analyzing Task Models for Interactive System Design, IEEE Transactions on Software Engineering, Volume 28, Issue 8, August 2002, IEEE Press Piscataway, NJ, USA, pp. 797-813.

[Myers 1995] B. A. Myers. ''User Interface Software Tools,'' ACM Transactions on Computer-Human Interaction (TOCHI), Volume 2, Issue 1, March 1995, ACM New York, NY, USA, pp. 64-103.

[Myers 2000] B. A. Myers, S. E. Hudson, R. Pausch, Past, Present, and Future of User Interface Software Tools, ACM Transactions on Computer-Human Interaction (TOCHI), Volume 7, Issue 1, Mars 2000, ACM New York, NY, USA, p. 3-28.

[Myers 2002] B. A. Myers, R. Malkin, M. Bett, A. Waibel, B. Bostwick, R. C. Miller, J. Yang, M. Denecke, E. Seemann, J. Zhu, C. H. Peck, D. Kong, J. Nichols, B. Scherlis, Flexi-Modal and MultiMachine User Interface, Proceedings of the 4th IEEE

International Conference on Multimodal Interfaces (ICMI'02), Pittsburgh, Pennsylvania, USA, 14-16 October, 2002, IEEE Computer Society, Washington, DC, USA, pp. 343-348.

[Navarre 2006] D. Navarre, P. Palanque, P. Dragicevic, R. Bastide, An Approach Integrating two Complementary Model-based Environments for the Construction of Multimodal Interactive Applications, Journal Interacting with Computers, Volume 18, Issue 5, September, 2006, Elsevier Science Inc. New York, NY, USA, pp. 910-941.

[Navarre 2009] D. Navarre, P. Palanque, J. F. Ladry, E. Barboni, ICOs: A Model-Based User Interface Description Technique dedicated to Interactive Systems Addressing Usability, Reliability and Scalability, ACM Transactions on Computer-Human Interaction, Volume 16, Issue 4, Article 18, November 2009, 56 pages.

[Nigay 1994] L. Nigay, Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales, Thèse de doctorat de l'université Joseph Fourier Grenoble 1, Janvier 1994, 364 pages.

[Nigay 1996] L. Nigay, J. Coutaz, Espaces conceptuels pour l'interaction multimédia et multimodale, TSI, spéciale Multimédia et collecticiel, AFCET &HERMES, 1996.

[Olsen, 2007] Dan R. Olsen Jr., Evaluating user interface systems research. Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST'07), 2007, ACM New York, NY, USA, pp. 251-258.

[Oviatt 1999] S. Oviatt, Ten myths of multimodal interaction, Communications of the ACM, Volume 42, Issue 11, November, 1999, pp. 74-81.

[Oviatt 2000] S. Oviatt, Taming Recognition Errors with a Multimodal Interface, Communications of the ACM, Volume 43, Issue 9, September, 2000, pp. 45-51.

[Oviatt 2003] S. Oviatt, R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson, L. Carmichael, Toward a theory of organized multimodal integration patterns during human-computer interaction, Proceedings of the 5th international conference on Multimodal interfaces (ICMI'03), Vancouver, British Columbia, Canada, 5-7 November, 2003, ACM New York, NY, USA, pp. 44-51.

[OpenInterface] OpenInterface. European project. IST Framework 6 STREP funded by the European Commission (FP6-35182). www.oi-project.org.

[Palanque 1990] Ph. Palanque, R. Bastide, Petri net objects for the design, validation and prototyping of user-driven interfaces, Proceedings of the third IFIP conference on Human-Computer Interaction (INTERACT'90), Cambridge, UK, 27-31 August, 1990, North-Holland Publishing Co. Amsterdam, The Netherlands, pp. 625-631.

[Parades 2008] M. Paredes, A. I. Molina, M. A. Redondo, M. Ortega, Designing Collaborative User Interfaces for Ubiquitous Applications Using CIAM: The AULA Case Study, Journal of Universal Computer Science, Volume 14, Issue 16, 2008, pp. 2680-2698.

[Paterno 1997] F. Paterno', C. Mancini, S. Meniconi, ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models, Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction (INTERACT'97), 14-18 July, 1997, Chapman & Hall, Ltd. London, UK, pp. 362-369.

[Penichet 2006] V. M. R. Penichet, F. Paternò, J. A. Gallud, M. D. Lozano, Collaborative Social Structures and Task Modelling Integration, Proceedings of the 13th international conference on Interactive systems: Design, specification, and verification(DSV-IS'06), 2006, Springer-Verlag Berlin, Heidelberg, pp. 67-80.

[Penichet 2009a] V. M. R. Penichet, M. D. Lozano, J.A. Gallud, R. Tesoriero, Analysis Models for User Interface Development in Collaborative Systems, Computer-Aided Design of User Interfaces VI, Chapter 17, 2009, Springer-Verlag Berlin, Heidelberg, pp. 189-200.

[Penichet 2009b] V. M. R. Penichet, M. D. Lozano, J. A. Gallud, R. Tesoriero, User Interface Analysis for Groupware Applications in the TOUCHE Process Model, International Journal Advances in Engineering Software, Volume 40, Issue 12, December, 2009, Elsevier Science Ltd. Oxford, UK, pp. 1212-1222.

[Penichet 2010] V. M. R. Penichet, M. D. Lozano, J. A. Gallud, R. Tesoriero, Requirement-based Approach for Groupware Environments Design, Journal of Systems and Software, Volume 83, Issue 8, August, 2010, Elsevier Science Inc. New York, NY, USA, pp.1478-1488.

[Pinelle 2003] D. Pinelle, C. Gutwin, S. Greenberg, Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration, ACM Transactions on Computer-Human Interaction (TOCHI), Volume 10, Issue 4, December, 2003, pp. 281-311.

[Pinelle 2007] D. Pinelle, C. Gutwin, Evaluating teamwork support in tabletop groupware applications using collaboration usability analysis, Personal and Ubiquitous Computing, Volume 12 , Issue 3, pp. 237-254, Springer-Verlag London, UK, 2008

[Rauschert 2002] I. Rauschert, P. Agrawal, R. Sharma, S. Fuhrmann, I. Brewer, A. MacEachren, Designing a human-centered, multimodal GIS interface to support emergency management, Proceedings of the 10th ACM international symposium on Advances in geographic information systems (GIS '02), McLean, Virginia, USA, 8-9 November, 2002, ACM New York, NY, USA, pp. 119-124.

[Reeves 2004] L. M. Reeves , J. Lai , J. A. Larson , S. Oviatt , T. S. Balaji , S. Buisine , P. Collings , P. Cohen , B. Kraal , J. C. Martin , M. McTear , T. Raman , K. M. Stanney , H. Su , Q. Y. Wang, Guidelines for multimodal user interface design, Communications of the ACM - Multimodal interfaces that flex, adapt, and persist, Volume 47, Issue 1, January 2004, pp. 57-59.

[Renevier 2004] P. Renevier, Systèmes Mixtes Collaboratifs sur Supports Mobiles : Conception et Réalisation, Thèse de doctorat de l'université Joseph Fourier, Grenoble 1, Juin 2004, 222 pages.

[Rey 2005] G. Rey, Contexte en Interaction Homme-Machine : le contexteur, Thèse de doctorat de l'université Joseph Fourier, Grenoble 1, 1er août 2005, 186 pages.

[Roseman 1992] Roseman, Mark et Greenberg, Saul. GroupKit: A groupware Toolkit for Building Real-Time Conferencing Applications, Proceedings of the 1992 conference on Computer-supported cooperative work (CSCW'92), 1992, ACM New York, NY, USA, pp. 43-50.

[Rubart 2004] J. Rubart, P. Dawabi, Shared data modeling with UML-G, International Journal of Computer Applications in Technology, Volume 19, Issue 3/4, May, 2004, pp. 231-243.

[Salber 1994] D. Salber, De l'interaction individuelle aux systèmes multiutilisateurs. L'exemple de la Communication Homme-Homme-Médiatisée, Thèse de doctorat de l'université Joseph Fourier, Grenoble 1, Septembre 1995, 303 pages.

[Scapin 1989] D. L. Scapin, C. Pierret-Golbreich, Une méthode analytique de description des tâches. Colloque sur l'ingénierie des Interfaces Homme-Machine, Sophia Antipolis, 1989, pp. 131-148.

[Schneiderman 1997] B. Shneiderman, Direct manipulation for comprehensible, predictable and controllable user interfaces, Proceedings of the 2nd international conference on Intelligent user interfaces (IUI'97), January, 1997, ACM Press New York, NY, USA, pp. 33-39.

[Serrano 2009] M. Serrano, L. Nigay, Temporal aspects of CARE-based multimodal fusion: from a fusion mechanism to composition components and WoZ components, Proceedings of the 2009 international conference on Multimodal interfaces (ICMI-MLMI'09), 2009, ACM Press New York, NY, USA, pp. 177-184.

[Serrano 2010] M. Serrano, Interaction multimodale en entrée : Conception et Prototypage, Thèse de doctoratde l'université de Grenoble, 30 juin 2010, 283 pages.

[Sezgin 2009] T. M. Sezgin, I. Davies, P. Robinson, Multimodal Inference for Driver-Vehicle Interaction, Proceedings of the 2009 international conference on Multimodal interfaces (ICMI-MLMI'09), 2009, ACM Press New York, NY, USA, pp. 193-198.

[Sinnig 2007] D. Sinnig, M. Wurdel, P. Forbrig, P. Chalin, F. Khendek, Practical Extensions for Task Models, Proceedings of the 6th international conference on Task models and diagrams for user interface design (TAMODIA'07), 2007, LNCS, Volume 4849, Springer-Verlag Berlin Heidelberg, pp. 42-55.

[Sottet 2007] J. S. Sottet, G. Calvary, J. Coutaz, J. M. Favre, A language perspective on the development of plastic multimodal user interfaces, Journal on Multimodal User Interfaces, Volume 1, Issue 2, 2007, Springer-Verlag Berlin, Heidelberg, pp. 1-12.

[StarUML] http://staruml.sourceforge.net/en/

[Stuckel 2008] D. Stuckel, C. Gutwin, The Effects of Local Lag on Tightly-Coupled Interaction in Distributed Groupware, Proceedings of the 2008 ACM conference on Computer supported cooperative work (CSCW'08), 2008, ACM Press New York, NY, USA, pp. 447-456.

[Tang 2006] A. Tang, M. Tory, B. Po, P. Neumann, S. Carpendale, Collaborative Coupling over TableTop Displays, Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI'06), Montréal, Québec, Canada, 22-27 April, 2006, ACM Press New York, NY, USA, pp. 1181-1190.

[Tarby 2001] J. C. Tarby, M. F. Barthet, Analyse et modélisation des tâches dans la conception des systèmes d'information : la méthode Diane+, in Analyse et conception de l'IHM interaction homme-machine pour les SI 1, Hermes, Chapitre 4, 2001, pp. 117-144.

[Thevenin 2001] D. Thevenin, Adaptation en interaction homme-machine - le cas de la plasticité, Thèse de doctorat de l'université Joseph Fourrier Grenoble 1, 21 décembre 2001, 234 pages.

[Treatteberg 2002] H. Treatteberg, Model-based User Interface Design, Thesis, Information systems Group, Department of Computer and Information sciences, Faculty of Information technology, Mathematics and Electronical Engineering, Norwegian University Science and Technology, 14 may 2002, 211 pages.

[Treatteberg 2008a] H. Trætteberg, Integrating Dialog Modeling and Domain Modeling - the Case of Diamodl and the Eclipse Modeling Framework. Journal of Universal Computer Science, Volume 14, Issue 19, 2008, pp. 3265-3278.

[Treatteberg 2008b] H. Trætteberg, UI Design without a Task Modeling Language - Using BPMN and Diamodl for Task Modeling and Dialog Design, Proceedings of the 2nd Conference on Human-Centered Software Engineering and 7th International Workshop on Task Models and Diagrams (TAMODIA/HCSE '08), 2008, Springer-Verlag Berlin, Heidelberg, pp. 110-117.

[Tse 2007] E. Tse, S. Greenberg, C. Shen, C. Forlines, Multimodal multiplayer tabletop gaming, Computers in Entertainment (CIE) - Interactive TV, Volume 5, Issue 2, Article 12, April-June 2007, ACM Press New York, NY, USA, 12 pages.

[Tse 2008] E. Tse, S. Greenberg, S. Shen, C. Forlines, R. Kodoma, Exploring True Multi-User Multimodal Interaction over a Digital Table, Proceedings of the 7th ACM conference on Designing interactive systems (DIS'08), Cape Town, South Africa, 25-27 February, 2008, ACM Press New York, NY, USA, pp. 109-118.

[Tuddenham 2009] P. Tuddenham, P. Robinson, Territorial Coordination and Workspace Awareness in Remote Tabletop Collaboration, Proceedings of the 27th international conference on Human factors in computing systems (CHI'09), Boston, Massachusetts, USA, 4-9 April, 2009, ACM Press New York, NY, USA, pp. 2139-2148.

[Turk 2000] M. Turk, G. Robertson, Perceptual user Interfaces (Introduction), Communications of the ACM, Volume 43, Issue 3, March, 2000, pp. 32-34.

[UML] http://www.uml.org/

[UsabilityFirst] www.usabilityfirst.com/

[Veer 2000] G. van der Veer, M. van Welie, Task Based Groupware Design : Putting theory into practice, Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (DIS'00), New York City, New York, United States, 2000, ACM Press New York, NY, USA, pp. 326-337.

[Venema 1999] D. C. Venema, The N-NUAN; a New User Action Notation, Thesis, Faculty of Sciences, division Mathematics and Computer Science, University of Amsterdam, Netherlands, July 1999, 43 pages.

[Vernier 2000] F. Vernier, L. Nigay, A Framework for the Combination and Characterization of Output Modalities, Proceedings of the 7th international conference on Design, specification, and verification of interactive systems (DSV-IS'00), Limerick, Ireland, June, 2000, Springer-Verlag Berlin, Heidelberg, pp. 32-48.

[Viala 2004] J. Viala, E. Dubois, P. D. Gray, GUIDE-ME : environnement graphique de manipulation de la notation ASUR, Proceedings of the 1st French-speaking conference on Mobility and ubiquity computing (UbiMob'004), Nice, France, 1-3 June, 2004, ACM Press New York, NY, USA, pp. 74-77.

[Wurdel 2008a] M. Wurdel, D. Sinnig, P. Forbrig, CTML: Domain and Task Modeling for Collaborative Environments, Journal of Universal Computer Science, Volume 14, Issue 19, 2008, pp. 3188-3201.

[Wurdel 2008b] M. Wurdel, D. Sinnig, P. Forbrig, Task-Based Development Methodology for Collaborative Environments, Proceedings of the 2nd Conference on Human-Centered Software Engineering and 7th International Workshop on Task Models and Diagrams (HCSE-TAMODIA '08), 2008, LNCS, Volume 5247, Springer-Verlag Berlin, Heidelberg, pp. 118–125.

[Wurdel 2008c] M. Wurdel, D. Sinnig, P. Forbrig, Towards a Formal Task-based Specification Frame-work for Collaborative Environments, Computer-Aided Design of User Interfaces VI (CADUI'08) Chapter 20, Albacete, Spain, 2008, pp. 221-232.

Annexes

1 Annexe 1 : Couplage de l'interaction : Expérimentation 1 – camion de pompier

Comme exposé dans le Chapitre 1, nous avons mené une expérimentation avec 16 sujets sur les formes de couplage de l'interaction en modifiant l'interface en sortie (les points de vue) dans un jeu. Le sujet devait contrôler une lance à incendie pour éteindre des feux tout au long d'un parcours. La lance était fixée sur un camion contrôlé par un autre utilisateur. Dans l'expérimentation, la conduite du camion était simulée par un enregistrement préalable de conduite. Les sujets ont rempli un questionnaire à la fin de chaque phase de l'expérimentation.

La partie « Pré-Expérience » du questionnaire détaillé ci-après comporte trois questions posées aux sujets après les essais. La partie « Vue … » comporte plusieurs questions posées aux sujets après l'utilisation de chacune des vues. Le jeu possédant trois vues, chaque sujet a répondu trois fois à cette série de questions. Enfin, la partie « questionnaire global » correspond aux questions posées aux sujets à l'issu de l'utilisation des trois vues.

Questionnaire de l'expérimentation 1 – Camion de pompier

Pré-Expérience

- Avez-vous compris comment utiliser clavier pour orienter la lance ? oui/non
- Avez-vous eu l'impression que la lance réagissait rapidement ? oui/non
- Avez-vous eu l'impression de maîtriser l'orientation de la lance ? oui/non

Vue … (on complète ici le nom de la vue utilisé)

- Pensez-vous avoir été perturbé par les actions du joueur 2 ?
- Indiquer le niveau de gène liés actions du joueur 2 de 1 (quasi nulle) à 10 (très importante) ?
- Avez-vous eu besoin d'intervenir oralement auprès du joueur 2 ?
- Pour quel raison ?
    - Est-ce lié au point de vue ?
    - est-ce lié au contrôle de la lance ?
    - est-ce lié à la difficulté de la réalisation de l'objectif ?
- Pensez-vous que la vue V est adaptée pour viser les feux ? oui/non
- Pouvez-vous noter la jouabilité de la vue V de 1 (injouable) à 10 (parfait) ?
- Pour quel raison ?

Questionnaire global

- Quelle vue vous semble la plus adaptée pour jouer ?
- Pourquoi ?
- Qu'avez-vous pensez du jeu, indépendamment de l'aspect graphique ?

2 Annexe 2 : Couplage de l'interaction : Expérimentation 2 – les 4 magiciens

Nous avons mené une expérimentation avec 24 sujets qui ont joué en binôme à un jeu de tir intitulé « 4 magiciens ». L'objet d'étude expérimentale était les formes de couplage de l'interaction, comme exposé dans le Chapitre 1. Les sujets ont testé trois modes de jeu correspondant à des formes de couplage différentes, notées Collaboratif, Coopératif, et Exclusif. Les sujets ont rempli un questionnaire à la fin de chaque phase de l'expérimentation.

Questionnaire de l'expérimentation 2 – les 4 magiciens

Portion du questionnaire utilisée après l'utilisation d'un mode. Ces questions ont donc été posées trois fois pour chacun des trois modes de jeu (Collaboratif, Coopératif, et Exclusif).

Mode de jeu	☐ Collaboratif ☐ Coopératif ☐ Exclusif						
Niveau		Vague		Score Total		Score individuel	

Question :	réponse
Pouvez-vous évaluer la difficulté de ce mode de jeu de 1 (très difficile) à 10 (très facile) ?	
Pouvez vous évaluer l'intérêt de ce mode de jeu de 1 (sans intérêt) à 10 (très intéressant) ?	
Pouvez-vous évaluer la jouabilité de ce mode de jeu de 1 (injouable) à 10 (très facile à jouer)	
Pouvez-vous évaluer la façon de jouer de l'autre joueur de 1 (il n'a rien fait) à 10 (il a tout fait).	
Pouvez-vous évaluer votre niveau d'expertise dans le maniement de la wiimote pour ce jeu de 1 (très très mauvais) à 10 (dieu de la wiimote) ?	

Remarque ou Commentaire :

Portion du questionnaire utilisée à l'issu de l'utilisation des trois modes de jeu.

Question :	**réponse**
Quel mode de jeu vous semble le plus facile ?	☐ Collaboratif ☐ Coopératif ☐ Exclusif
Quel mode de jeu vous semble le plus intéressant ?	☐ Collaboratif ☐ Coopératif ☐ Exclusif
Quel mode de jeu vous semble le plus jouable ?	☐ Collaboratif ☐ Coopératif ☐ Exclusif
Dans quel mode estimé vous que l'autre joueur a le mieux jouer ?	☐ Collaboratif ☐ Coopératif ☐ Exclusif

Remarque ou Commentaire :

3 Annexe 3 : Notation COMM et éditeur e-COMM : Expérimentation

3.1 Questionnaire fourni aux étudiants

Background

- Age ?
- Sexe ? Féminin / Masculin
- Occupation et Niveau d'études ?

Informations générales

- Comment vous évaluez-vous en terme de compétences sur les arbres de tâches ?
- Avez-vous déjà utilisé des notations de spécification ? Si oui lesquels ? Si oui quelles étaient les principales difficultés rencontrées?
- Avez-vous déjà utilisé des éditeurs pour spécifier un système? Si oui lesquels? Si oui quelles étaient les principales difficultés rencontrées?

Questions dirigées

- Enoncer trois points forts de la notation COMM.
- Enoncer trois points faibles de la notation COMM.
- Est-ce que vous aimeriez utiliser ce type d'éditeur avec ce genre d'interaction pour une autre notation de spécification (UML etc.) ? Pourquoi ?
- Est-ce que vous réutiliseriez l'éditeur e-COMM pour vos projets futurs ? Si possible argumenter pourquoi et pour quels types de systèmes (collecticiel, mobiles, multimodalité, système graphique classique de type écran-clavier-souris ?
- Pensez-vous pertinent de pouvoir sauvegarder des sous-arbres pour les réutiliser dans un autre arbre ou un autre emplacement dans le même arbre ?
- Suggestions d'améliorations de l'éditeur ou de la notation ?

Série de phrases à évaluer

(Sur une échelle de 1 à 5, ou 1 exprime le désaccord, et 5 l'accord total)

- Je pense que j'aimerais utiliser l'éditeur fréquemment
- J'ai trouvé l'éditeur inutilement complexe
- J'ai pensé que l'interaction était facile à utiliser
- Je pense que j'aurais besoin de l'aide d'une personne technique pour pouvoir utiliser pleinement cet éditeur
- Je pense que beaucoup de personnes aimeraient utiliser cet éditeur
- J'ai trouvé l'éditeur encombrant à utiliser
- J'ai eu besoin d'apprendre beaucoup de choses avant de pouvoir me débrouiller avec l'éditeur

3.2 Exemples de spécifications COMM réalisées par les étudiants

Les spécifications COMM ci-dessous sont des extraits des spécifications réalisées par les étudiants avec l'outil e-COMM.

La spécification de la Figure 182 montre un exemple simple d'une spécification : celle-ci utilise les notions de rôles métiers (*Admin, User*), de tâches multiutilisateurs (*Editer le pixter*). Cependant, cette spécification ne décrit pas l'interaction multimodale concrète. En effet, les tâches élémentaires décrites (*Créer un pixster*, *Editer sur le "mur"*) sont des tâches pouvant être décomposées en tâches modales.

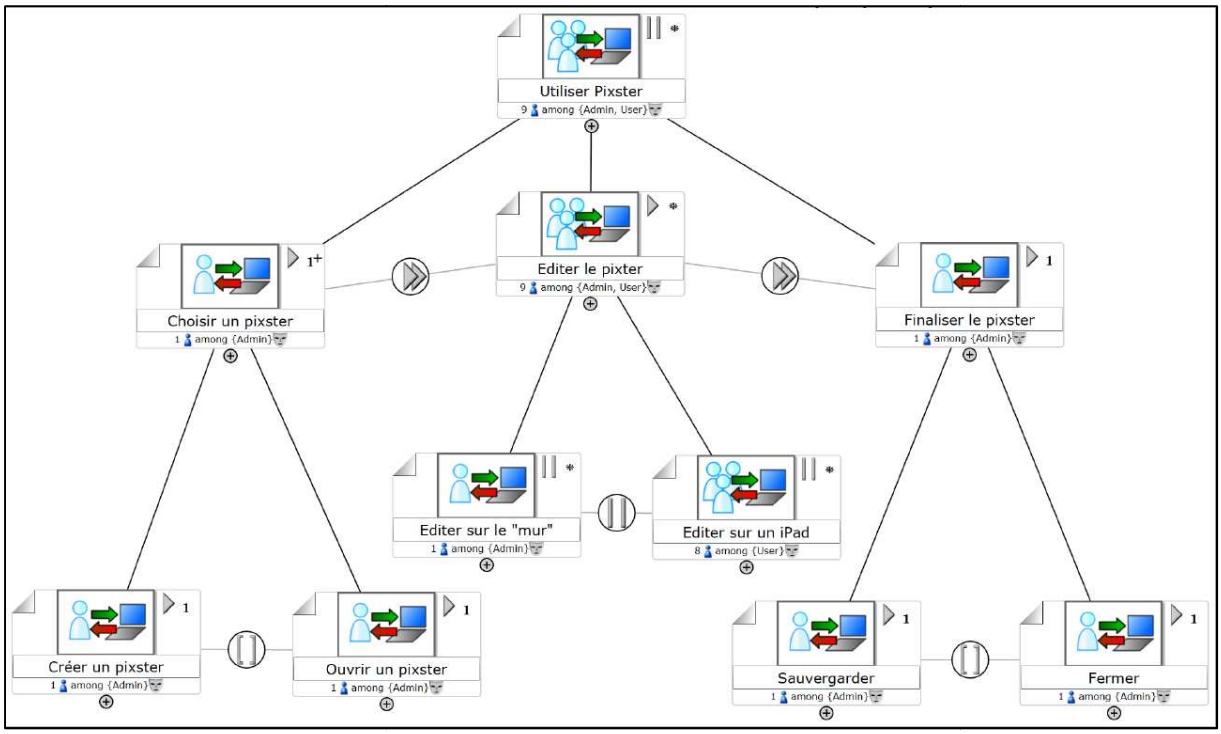


Figure 182 : Spécification complète d'un étudiant n'ayant utilisé ni tâche modale, ni opérateur de Allen.

L'extrait de spécification présenté à la Figure 183 montre au contraire l'utilisation par un étudiant des tâches modales pour décrire l'interaction concrète. Nous pouvons observer que les tâches modales telles que *Create a poste* utilisant un dispositif *Screen* restent d'un niveau d'abstraction élevé et ne décrivent pas l'interaction concrète.

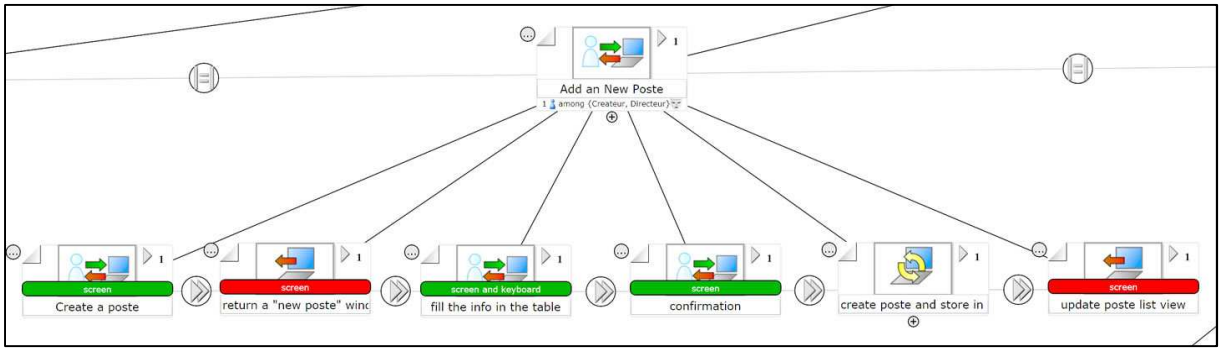


Figure 183 : Extrait de spécification d'un étudiant ayant utilisé des tâches modales pour décrire la création d'un « post »[8].

Nous illustrons à la Figure 184 une spécification complète produit par un étudiant. Les zones rouge et verte de la figure sont respectivement détaillées au sein de la Figure 185 et de Figure 186.

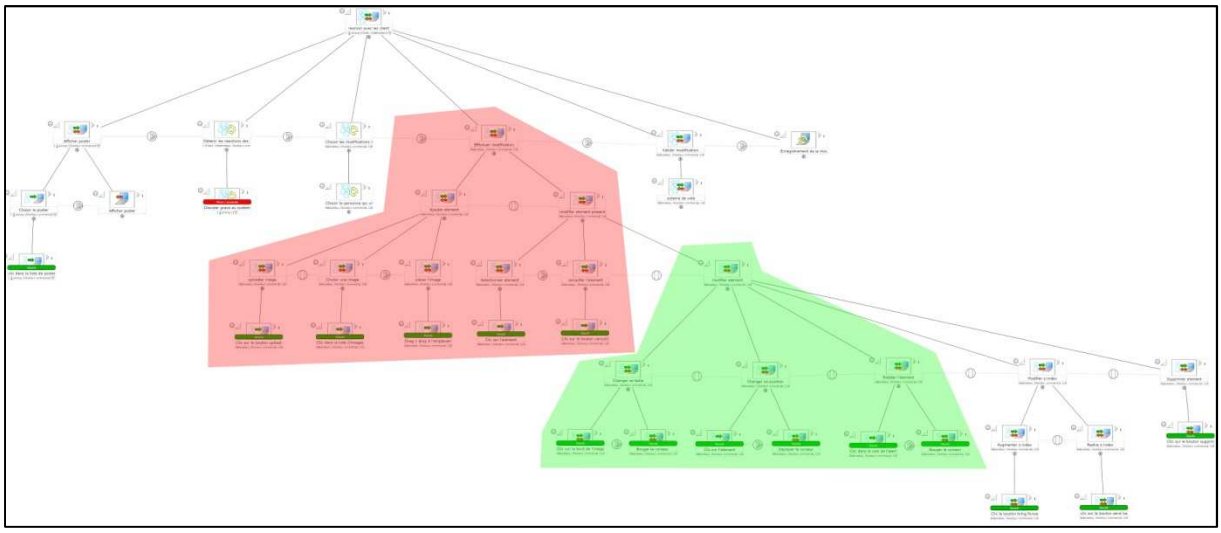


Figure 184 : Aperçu d'une spécification complète d'un étudiant.

Sur le détail de spécification de la Figure 185, nous observons l'utilisation de tâches modales en entrée pour raffiner les tâches interactives. Par exemple, la tâche interactive *Placer l'image* est raffinée par une tâche modale consistant à utiliser le dispositif *Souris* pour réaliser un *Drag and drop*. De la même manière, une tâche modale *Clic sur le bouton verrouillage* en utilisant le dispositif *Souris* est décrite pour raffiner la tâche interactive *Verrouiller l'élément*.

[8] Le terme post est utilisé sur les forums pour désigner une contribution d'un membre, telle que la création d'un sujet de conversation ou la réponse à un sujet de conversation.

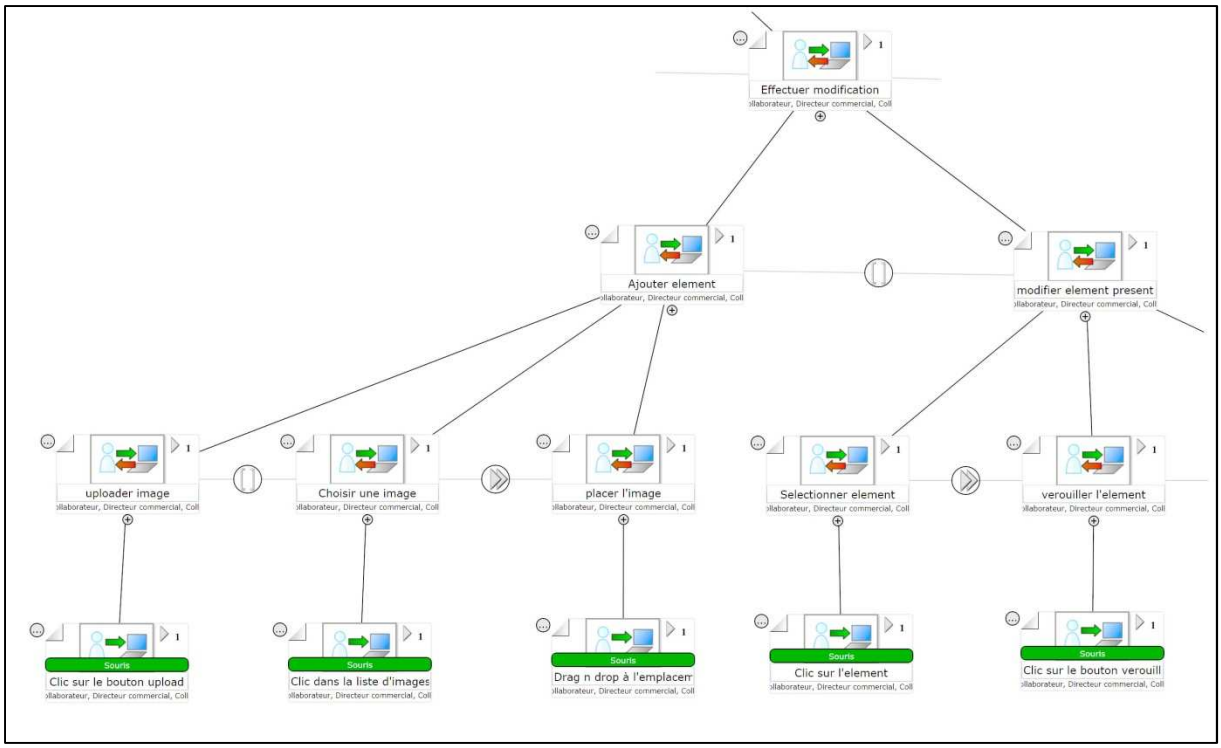


Figure 185 : Détail de la zone rouge de la spécification complète de la Figure 184.

Dans l'extrait de la Figure 186, l'étudiant décrit des interactions sous la forme de plusieurs tâches modales pour réaliser une tâche interactive, tandis que dans la Figure 184, une seule tâche modale est utilisée pour une tâche interactive. Ainsi, la tâche interactive *Changer la position* d'un élément est décrite par deux tâches modales séquentielles *Clic sur l'élément* et *Déplacer le curseur*, toutes deux réalisées au moyen d'un dispositif Souris.

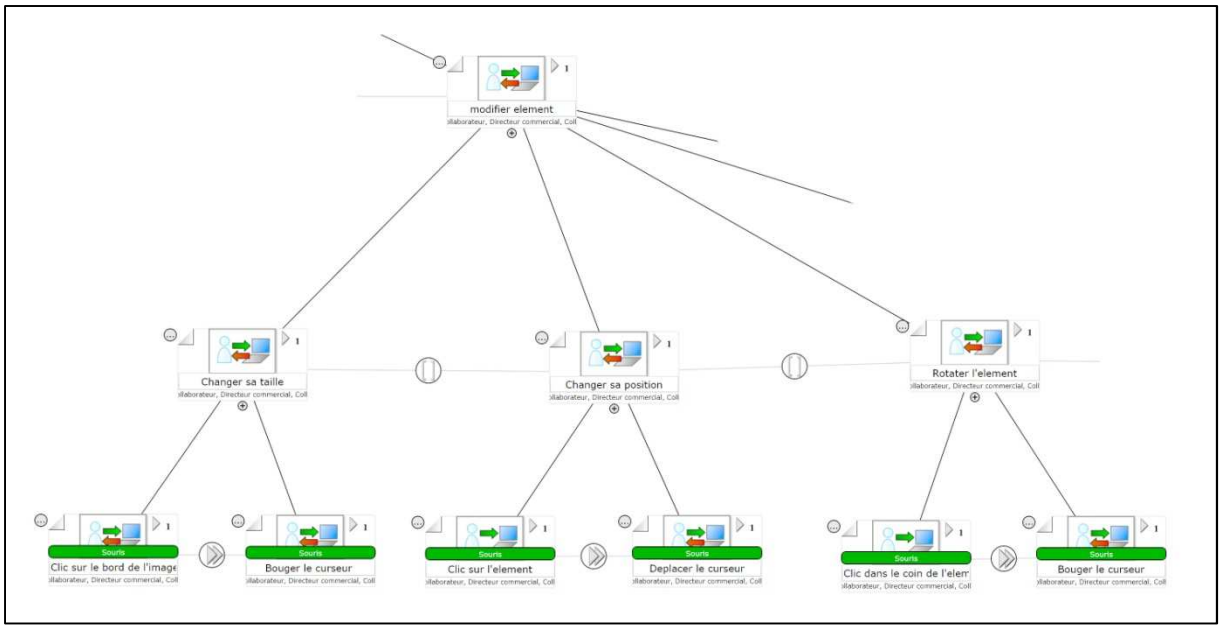


Figure 186 : Détail de la zone verte de la spécification complète de la Figure 184.