HAL
open science

Approche spatiale pour la caractérisation du contexte d'exécution d'un système d'information ubiquitaire

Mathieu Petit

To cite this version:

Mathieu Petit. Approche spatiale pour la caractérisation du contexte d'exécution d'un système d'information ubiquitaire. Informatique mobile. Arts et Métiers ParisTech, 2010. Français. NNT : 2010ENAM0016 . pastel-00511919

HAL Id: pastel-00511919

https://pastel.hal.science/pastel-00511919v1

Submitted on 26 Aug 2010

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.

2010-ENAM-0016

École doctorale n° 432 : Sciences et Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Informatique "

présentée et soutenue publiquement par

Mathieu Petit

le 14 juin 2010

Approche spatiale pour la caractérisation du contexte d'exécution d'un système d'information ubiquitaire

Directeur de thèse : **Christophe CLARAMUNT**
Co-encadrement de la thèse : **Cyril RAY**

Jury

Mme. Mireille DUCASSÉ, Professeur, IRISA, INSA de Rennes, Rennes	Président
M. Jérôme GENSEL, Professeur, LIG, Université Pierre Mendès France, Grenoble	Rapporteur
Mme. Florence SEDÈS, Professeur, IRIT, Université Paul Sabatier, Toulouse	Rapporteur
Mme. Bénédicte BUCHER, Ingénieur Habilité à Diriger des Recherches, COGIT, IGN, Paris	Examinateur
M. Christophe CLARAMUNT, Professeur, IRENav, Ecole Navale, Brest	Examinateur
M. Cyril RAY, Maître de Conférences , IRENav, Ecole Navale, Brest	Examinateur

T H È S E

Arts et Métiers ParisTech - Centre de Paris
Institut de Recherche de l'Ecole Navale

Remerciements

En premier lieu, je remercie les membres du jury d'avoir participé à ma soutenance et de s'être intéressés à mon travail. Je pense plus particulièrement à Jérôme Gensel et Florence Sedès qui ont accepté d'être rapporteurs de ce mémoire et dont les retours résument au mieux le travail que j'ai réalisé.

Cyril Ray et Christophe Claramunt m'ont aidé à faire le tri dans mes idées et m'ont conseillé tout au long de ma thèse. Entre autres choses, j'attribue à Cyril l'idée de la "spatialisation" de l'espace d'exécution d'un système (issue de ses lectures de M.F. Goodchild notamment). À son bureau et avec quatre magnets et une règle, Christophe a régulièrement pris le temps de m'expliquer le monde des SI mobiles et les défis scientifiques auxquels il fait face. Tous deux m'ont aidé dans la rédaction d'articles en relisant et en annotant mes brouillons version après version. En dehors de mes encadrants, je remercie les nombreuses personnes qui m'ont aidé à faire progresser ma recherche. En particulier, Michela Bertolotto, Gaëlle Calvary, Gavin McArdle et Tryphon Tournesol ont chacun à leur niveau influencé mon travail. Marie Coz, attachée scientifique du groupe SIG, m'a beaucoup aidé lors de la préparation de ce mémoire.

L'Ecole Navale et le ministère de la défense ont assuré l'intendance pendant mes quatre années à Brest. En échange, j'ai eu le plaisir de dispenser des cours aux élèves de l'école et de participer aux activités pédagogiques de la direction de l'enseignement et du département informatique. Respectueusement, merci aux élèves et aux cadres !

Je remercie chaleureusement l'ensemble du personnel de l'institut de recherche de l'Ecole Navale : les stars du centre de ressources en informatique, l'excellent service d'étude et fabrication, le toujours efficace secrétariat, l'irréprochable personnel d'entretien et les nombreux chercheurs des équipes *mécanique et énergétique en environnement naval*, *acoustique sous-marine* et *carcinologie tropicale*. Au sein de l'équipe *systèmes d'information géographique*, je remercie les permanents (Éric, Thomas, Rémi, Joseph et Michel, en plus de Cyril et Christophe) pour l'animation de l'activité de recherche et les nombreux échanges que nous avons eus. J'ai partagé mon bureau successivement avec Valérie et Meriam, que je remercie de m'avoir supporté au jour le jour et d'avoir toujours été là pour m'aider à procrastiner sans remords. Pendant mes premières années, la rigueur scientifique de Valérie a été exemplaire pour moi. Ensuite, avec Meriam, je me suis beaucoup amusé à animer le sous-sous-sous-groupe SIG/IHM, en enseignement comme en recherche.

Mes collègues doctorants m'ont motivé et m'ont fait confiance pour les représenter au conseil de l'institut. Je remercie plus particulièrement (et dans le désordre) David (collection-

neur de fourmis), Géraldine (graffiti artist), Jean-Marie (légufrulabélophile), Laurent E. (sauver ou périr), Marie-Laure (besoin caaaafé, viiiiite !), Thierry (foncièrement méchant), Thomas (la barbe, c'est un état d'esprit), Ariane (modèle de longévité), Antoine (guitariste vu aux USA), Shaopei (king of Guangzhou), Laurent D. (au fond c'est quoi l'électromagnétisme ?), Rachel (l'amie d'Herman), Pascal (dont j'ai oublié le nom) ...

À Grenoble, mes nouveaux collègues de l'équipe IIHM et de l'ENSIMAG m'ont accueilli sans réserve. Je remercie en particulier Gaëlle et Audrey, victimes involontaires de mon retard d'écriture, pour leur patience et pour leurs encouragements.

Parmi ceux qui m'ont accompagné et soutenu figurent mes amis d'Alsace. Notamment, Théophane, Isa, Marie, Vincent, Jim, Eugénie, Dominique, Lise, Johan, je vous dédie respectivement les pages 113, 47, 23, 61, 163, 181, 137, 89 et 197. Très chers, merci d'avoir su m'extraire de mon univers de régions mobiles quand il le fallait. Je remercie aussi mes amis du cercle lorrain : compagnons de voyages et hôtes habituels. Merci entre autres à Florian (Florent ?), Céline, Nicolas, Benoît, Ania et Marion. À l'association Point d'Appui, j'ai retrouvé avec plaisir Daniel, Brigitte, Jean-Marie, Marlène, Viviane et tous les membres le temps de réunions, manifestations ou passages en Bolivie.

Enfin, je remercie tout particulièrement mes parents, Marie-Hélène et Jean-Paul de m'avoir encouragé dans mes (longues) études et plus spécialement au cours de ma thèse. Ma sœur et mon frère, Claire et Sylvain, et leurs conjoints Damien et Emmanuelle m'ont aussi beaucoup aidé. Grâce à eux, j'ai appris l'intérêt du béton précontraint, la vraie nature des "œufs de cigogne" alsaciens, de nouvelles recettes de tartes flambées et quelques éléments de thermodynamique. Merci à ma belle famille pour son accueil. Merci à ma merveille (nom d'emprunt ;-)) pour tout le positif qu'elle m'inspire.

Table des matières

Table des matières **i**

Liste des figures **v**

Liste des tableaux **ix**

Introduction **1**

Contexte de la thèse 1

Problématique scientifique 2

Objectifs et plan du mémoire 4

I Systèmes d'information mobiles **7**

I.1 Évolution des systèmes d'information distribués 8

I.2 Systèmes d'information distribués et mobilité 9

I.2.1 Principes et origines de la mobilité des systèmes 10

I.2.2 Infrastructures réseau pour la transmission d'informations 11

I.2.3 Plates-formes clientes pour la mobilité 13

I.2.4 Enjeux de la mobilité des systèmes 15

I.3 Solutions pour la mise en œuvre d'un système mobile 17

I.3.1 Conception à bas niveau : distribution, réseau et communication . . . 17

I.3.2 Conception à haut niveau : Plate-forme cliente 20

I.4 Conclusion 30

II Systèmes d'information sensibles au contexte **33**

II.1 Nature contextuelle des systèmes mobiles 34

II.2 Notion de contexte appliqué aux systèmes d'information 36

II.2.1 Définitions de la notion de contexte 37
II.2.2 Dimensions du contexte 38
II.3 Mesure des dimensions du contexte 43
II.3.1 Mesure du contexte d'usage 43
II.3.2 Mesure du contexte d'exécution 46
II.3.3 Intérêt d'une mesure pour l'adaptation du système 47
II.4 Conception et implémentation de systèmes contextuels 49
II.4.1 Approche conceptuelle : familles de modèles contextuels 49
II.4.2 Approche pour l'implémentation : architectures d'exécution 52
II.5 Utilisation des informations contextuelles 58
II.5.1 Typologies des formes d'exploitation du contexte 59
II.5.2 Initiative et responsabilité de l'adaptation contextuelle 62
II.5.3 Déclenchement du processus d'adaptation 64
II.5.4 Objets de l'adaptation à l'exécution 65
II.6 Conclusion 68

III Modélisation spatiale et temporelle du contexte d'exécution 71
III.1 Illustration d'un système distribué et mobile à l'exécution 73
III.2 Décomposition de l'exécution d'un système mobile et distribué 75
III.2.1 Décomposition temporelle 76
III.2.2 Décomposition physique 79
III.2.3 Décomposition géographique 81
III.2.4 Décomposition contextuelle 90
III.3 Description conceptuelle de l'environnement d'un système mobile et distribué 94
III.3.1 Description physique 95
III.3.2 Description géographique 97
III.3.3 Analyse spatiale 102
III.3.4 Description conceptuelle des contextes d'exécution 105
III.4 Conclusion 108

IV Intégration de l'environnement spatial dans la modélisation d'un système interactif 109
IV.1 Modèle de conception pour la mise en œuvre des contextes d'exécution 110
IV.1.1 Cadre de conception unifié : une approche pour le développement de systèmes interactifs 110

IV.1.2 Cadre de conception étendu : proposition pour l'intégration de la modélisation géographique . . . 112
IV.2 Étapes de corrélation avec les contextes d'exécution . . . 114
IV.2.1 Primitives du cadre de conception unifié : scénario nominal, modèle des tâches et des informations . . . 115
IV.2.2 Recherche d'équivalences entre contextes d'exécution . . . 118
IV.2.3 Arbre des tâches situé . . . 128
IV.3 Conclusion . . . 135

V Utilisation des contextes d'exécution pour la personnalisation 137
V.1 Mesure implicite des préférences des utilisateurs . . . 141
V.1.1 Création d'un profil par l'apport d'expérience . . . 141
V.1.2 Méthode hybride pour la construction de profils dans un système mobile et distribué . . . 144
V.2 Propositions pour la construction de profils et la mesure de scores . . . 147
V.2.1 Création d'un profil utilisateur . . . 149
V.2.2 Perspective multi-utilisateurs et regroupement de profils . . . 156
V.3 Conclusion . . . 166

VI Illustration de la démarche de modélisation 167
VI.1 Cas d'étude : système de suivi de régate . . . 168
VI.1.1 Contexte du système de suivi de régate . . . 169
VI.1.2 Descriptions géographique et contextuelle de l'exécution . . . 170
VI.1.3 Analyse des tâches et construction d'un arbre situé . . . 175
VI.1.4 Maquettage pour l'implémentation du tiers client . . . 179
VI.2 Cas d'étude : système d'information de campus . . . 181
VI.2.1 Contexte du système d'information campus . . . 181
VI.2.2 Architecture distribuée au sein du campus . . . 182
VI.2.3 Perspective multi-utilisateurs et adaptations . . . 185
VI.3 Conclusion . . . 195

Conclusion 197
Perspectives de recherche . . . 199
Bilan et généralisation . . . 201

Annexes 205

A Détails des calculs de profil de groupe **207**

A.1 Calcul du score moyen instantané : cas d'étude 1 207

A.2 Calcul du score moyen instantané : cas d'étude 2 207

A.3 Calcul du score de groupe : cas d'étude 1 208

B Mesure de qualité de service **209**

C Notation matricielle du contexte d'exécution **211**

D Liste de publications **213**

D.1 Articles de revues 213

D.2 Actes de conférences 213

D.3 Autres communications 214

Bibliographie **215**

Liste des figures

1 Organisation et positionnement du mémoire par rapport aux problématiques adressées 5

I.1 Principe de distribution d'un système d'information 8

I.2 Principe de la mobilité dans un système d'information distribué 10

I.3 Trois types de mobilité selon 11

I.4 Exemples de clients pour les systèmes d'information mobiles 14

I.5 Solutions pour l'architecture et le support des infrastructures mobiles 18

I.6 Exemples de clients de systèmes d'interaction conçus pour la mobilité 22

I.7 Exemples d'interfaces de saisie et de présentation sur des dispositifs mobiles : AppLens (), ZoneZoom (), VirHKey () et EdgeWrite () 25

I.8 Premiers prototypes de systèmes ubiquitaires 26

II.1 Principaux domaines de recherche qui contribuent à l'émergence de l'informatique ubiquitaire (d'après) 36

II.2 Synthèse des propositions pour le dimensionnement du contexte 42

II.3 Variation de l'utilité en fonction du contexte d'exécution 49

II.4 Exemples de modèles de contexte (1) 50

II.5 Exemples de modèles de contexte (2) 51

II.6 Architectures pour la perception du contexte à l'exécution 55

II.7 Synthèse des propositions typologiques de l'exploitation du contexte 61

II.8 Partage des responsabilités dans le déroulement d'une adaptation (d'après) . 64

II.9 processus d'exécution d'une adaptation dans le cas des systèmes adaptables/adaptatifs (d'après) 65

II.10 Solutions alternatives pour l'adaptation d'interfaces graphiques () 66

II.11 Adaptations et exemples de dialogues pour un même système par la sélection de modèles de plate-forme, de tâche et de présentation appropriés () 67

III.1 Corrélation entre la conception et l'exécution d'un système mobile et distribué 72
III.2 Exemple d'une distribution d'un système d'information et de son impact géographique 74
III.3 Exemple de cycle de vie du système 78
III.4 Evolution des régions du système de suivi cotier entre t_0 et t_2 87
III.5 Graphe des relations entre régions et notation matricielle du contexte d'exécution dans le cas du système de suivi côtier 91
III.6 Espaces accessibles pouvant être occupés par les régions et espaces de mobilité pouvant être occupés par les composants 101
III.7 Illustration des règles d'exclusion systématique des relations entre régions de l'environnement 104
III.8 Illustration des règles d'inclusion systématique des relations entre régions de l'environnement 104
III.9 Superposition des espaces accessibles et des espaces de mobilité pour la recherche des règles d'inclusion et d'exclusion systématiques 106
III.10 Représentation graphique de quelques contextes d'exécution de l'ensemble $STATES$ 107

IV.1 Cadre de conception et de mise en œuvre d'un système d'information sensible au contexte 111
IV.2 Cadre de conception étendu 113
IV.3 Exemple d'arbre des tâches annoté 117
IV.4 Exemple d'équivalence entre contextes d'exécution par la propriété "utilisateur isolé" 120
IV.5 Exemple d'équivalence entre contextes d'exécution par la propriété "Région d'intérêt indéfinie" 121
IV.6 Exemple d'équivalence entre contextes d'exécution par la propriété "Région d'intérêt distante" 122
IV.7 Exemple d'équivalence entre contextes d'exécution par la propriété "Contextes locaux identiques" 123
IV.8 Exemple d'équivalence entre contextes d'exécution par la propriété "Relation non significative" 124
IV.9 Exemple de construction d'une chaîne d'équivalence par association de binômes de contextes d'exécution 126
IV.10 Groupes de contextes équivalents constituant l'ensemble $CHAINS$ dans l'exemple du système de suivi côtier 127
IV.11 Exemple simplifié de décomposition des tâches et des contextes d'exécution . 128
IV.12 Construction d'une représentation d'un arbre de tâches annoté mis en contexte 131

IV.13 Transformations par factorisation de contraintes pour la construction d'un arbre contextualisé 133

IV.14 Exemple d'arbre des tâches situé 135

V.1 Intégration des mesures du contexte d'usage pour l'individualisation du client 138

V.2 Exemples d'arbres situés pour lesquels l'individualisation peut être appliquée 140

V.3 Exemple de tâche réalisable dans le cas du système de suivi côtier à t_3 par l'utilisateur sur son composant Cu_2 146

V.4 Exemple d'affichages cartographiques au niveau de l'interface du client Cu_2 pour la tâche "S'informer sur la côte" réalisable à t_3 147

V.5 Étapes de la construction d'un profil de groupe à un instant de l'exécution . . 148

V.6 Actions implicites de l'utilisateur mesurées entre t_2 et t_3 150

V.7 Parcours de deux utilisateurs du système entre t_4 et t_9 158

VI.1 Infrastructure du système de suivi en temps réel 169

VI.2 Exemple de répartition des composants, des régions d'exécution et de la région d'intérêt à t_x 171

VI.3 Espaces accessibles et espaces de mobilité du système de suivi de régate . . . 173

VI.4 Ensemble $STATES$ des contextes mesurables à l'exécution 174

VI.5 Chaînes des contextes d'exécution équivalents du système de suivi régate . . . 175

VI.6 Arbre des tâches annoté du système de suivi de régate 176

VI.7 Arbre des tâches situé du système de suivi de régate 178

VI.8 Comportement du système modifié selon le contexte d'exécution rencontré . . 180

VI.9 Interface utilisateur du système d'information campus 182

VI.10 Exemple de répartition des composants et des régions d'exécution du système d'information campus 183

VI.11 Contextes d'exécution mesurables dans le cas du système de visite de campus 184

VI.12 Arbre des tâches situé du système de visite de campus 184

VI.13 Exemples de variations du contenu et de l'interface présentés en fonction du contexte d'exécution. 185

VI.14 Instantané à t_1 d'une visite effectuée par les trois utilisateurs entre t_0 et t_5 . . 186

VI.15 Primitives captées pour la construction des scores personnels 187

VI.16 Comparaison des variations des scores de groupe et des valeurs moyennes instantanées 190

VI.17 Comparaison des variations entre les scores personnels et les scores multi-utilisateurs 191

VI.18 Vue d'ensemble de la chaîne de génération de scores et d'adaptations 193
VI.19 Développements pour les tests du système de navigation campus 194
VI.20 Exemples d'interfaces générées à t_5 194

B.1 Proportion des tâches réalisables en fonction du contexte d'exécution rencontré 209

C.1 Exemple de contexte d'exécution d'un système distribué : empreinte spatiale des régions et notation matricielle des relations entre régions 211

Liste des tableaux

I.1 Technologies de transmission pour l'infrastructure réseau de systèmes mobiles () 13

I.2 Domaines d'utilisation des plates-formes clientes selon les types de mobilité (d'après) 15

I.3 Solutions apportées par les différents cadres de conception de systèmes mobiles aux contraintes de développement des systèmes distribués 20

I.4 Synthèse des solutions proposées pour la conception de systèmes d'information à clients mobiles 28

II.1 Exemples de *capteurs physiques* et d'instruments de mesure () 44

II.2 Synthèse des qualités et défauts des familles de modèles de contexte par rapport aux propriétés désirables de la sensibilité au contexte dans les systèmes ubiquitaires 53

II.3 Typologie des utilisations du contexte pour 59

II.4 Comparaison des systèmes adaptables et adaptatifs (d'après) 63

IV.1 Exemple d'entretien entre concepteurs et utilisateurs 115

IV.2 Exemple de scénario nominal 116

IV.3 Motivation des propriétés d'équivalence fonctionnelles 119

V.1 Méthodes pour l'hybridation des techniques de mesure de la dimension utilisateur (d'après) 143

V.2 Scores individuels des éléments Sem et TaP entre t_4 et t_9 158

V.3 Scores des éléments "TaP" et "Sem" entre t_4 et t_9 165

V.4 Profils de groupe 165

VI.1 Extraits du scénario nominal établi pour le système de suivi de régate 176

VI.2 Scénario alternatif présentant le cas d'un utilisateur isolé 179

VI.3 Scores des éléments "c1" à "c4" calculés à t_1. 188

VI.4 Scores par groupe des éléments “c1” à “c4” calculés à chaque instant entre t_0 et t_5 (ordre des éléments par scores décroissants : 1er, 2ème, 3ème et 4ème) . 190

VI.5 Synthèse des performances comparées des scores de groupe par rapport aux valeurs moyennes instantanées 192

Introduction

Contexte de la thèse

La mobilité est profondément inscrite dans la nature humaine. Des temps les plus éloignés, où les tribus nomades parcouraient à pied les continents, à l'époque moderne, où le nomadisme s'envisage en fonction de moyens de transport et où les distances traversées augmentent sans que le temps du parcours ne s'accroisse, la mobilité des personnes est quotidienne, pour le travail, pour les loisirs, etc. Le point de départ de cette étude considère que l'évolution de la mobilité des personnes est en train de croiser celle des développements récents du monde informatique. Un tel phénomène se prolongera inéluctablement par un changement de nature de la mobilité des utilisateurs d'une part, et de l'utilisation des systèmes informatiques d'autre part.

Les changements de comportements humains provoqués par l'apparition d'une informatique portable, et l'évolution de la mobilité des personnes, posent des problématiques comportementales, sociologiques ou philosophiques qui dépassent le cadre de cette étude. De notre point de vue, l'émergence des systèmes mobiles, résultats du croisement de la mobilité et de l'informatique, est étudiée dans une perspective d'ingénierie pour la conception et le développement des systèmes numériques.

Plus récemment, la convergence de la mobilité des utilisateurs et de l'informatique constitue la toile de fond du développement des systèmes d'information mobiles. Le thème de la mobilité des systèmes d'information est désormais au coeur des problématiques de développement et de recherche des acteurs majeurs de l'informatique, tant du point de vue du matériel que de celui du logiciel. Depuis l'adoption massive de la téléphonie cellulaire, l'engouement du grand public pour les systèmes mobiles ne se dément pas et oriente industries et sociétés de services vers la recherche d'innovations.

Cette forte volonté d'innovation dans les systèmes d'information et de communication est-elle suffisante pour répondre aux enjeux de l'utilisation en mobilité ? La plupart du temps, les solutions proposées sont, soit développées pour la mobilité et se dédient à des usages exclusifs, soit consistent en des plates-formes génériques qui sacrifient en partie leur mobilité pour se rapprocher de l'informatique "station de travail". La conception de ces différentes solutions se concentre sur les défis techniques posés par la mobilité (p. ex. développement d'interfaces réduites en encombrement, problématique de l'alimentation énergétique ou de la

connectivité réseau, etc.), et passe à côté de ce qui constitue la différence fondamentale de l'informatique mobile par rapport à l'informatique fixe : les utilisateurs se déplacent dans un espace géographique qui constitue leur environnement. Cet environnement est par nature variable, imprévisible et propre à contraindre les utilisateurs de systèmes mobiles. Dans la suite, cette étude initie des pistes de réflexion centrées sur la description géographique du système et propose une méthodologie conceptuelle pour remettre les notions d'environnement et de variabilité des conditions d'exécution au coeur d'une démarche de développement de systèmes mobiles.

Problématique scientifique

Cette étude porte sur les systèmes mobiles et distribués, définis comme un ensemble de composants physiques et logiciels répartis, et concourant à la réalisation d'un service au profit d'un ou de plusieurs utilisateur(s) mobile(s). Plus spécifiquement, ces systèmes sont abordés sous l'angle conceptuel par l'analyse de l'impact de l'environnement du système sur son fonctionnement et sur les besoins de ses utilisateurs. Cet environnement d'exécution est difficile à caractériser au moment de la conception des systèmes mobiles. Sa description complète passerait en effet par la détermination de l'ensemble des situations que l'utilisateur pourrait rencontrer lors de sa mobilité. Compte tenu de la grande complexité de l'espace qui entoure le système et de la quantité des éléments qui peuvent impacter la tâche des utilisateurs, décrire exhaustivement l'environnement d'un système d'information mobile est une tâche considérable qu'à notre connaissance, aucune approche conceptuelle ne saurait prétendre résoudre.

À défaut de pouvoir tout prévoir à la conception, il est nécessaire d'admettre que le système mobile devra faire face à des situations complexes et difficiles à appréhender lors de son exécution. De telles situations nouvelles et changeantes peuvent impacter son fonctionnement, le dégrader, le rendre inapproprié, etc. Pour mieux répondre à ces changements, il reste à imaginer des méthodes d'ingénierie adéquates, propres à implémenter des systèmes réactifs et sachant s'adapter à différentes situations d'exécution. Dans cette optique, cette étude considère le cas de l'informatique distribuée comme point de départ méthodologique et adresse un ensemble de questions relatives à la mobilité de ces systèmes, à leur conception et à leur exécution :

- **Quelle définition donner au concept de mobilité dans le cas des systèmes d'information ? En quoi la nature mobile des systèmes distribués impacte-t-elle leur conception ?**
  Les systèmes d'information mobiles sont en général caractérisés par leur capacité à accompagner un utilisateur dans ses déplacements. Dans un système distribué, cette mobilité concerne les composants en interaction avec l'utilisateur, notamment sa plate-forme cliente. Plus généralement, aborder la mobilité au niveau de l'ensemble des composants permet d'envisager de nouvelles situations à l'exécution, où la dynamicité du système s'ajoute à celle de l'environnement d'utilisation.

- **Comment appréhender l'environnement d'un système mobile ? Quelles sont les dimensions et les éléments qui caractérisent une situation d'exécution ?**

La notion d'environnement dans le développement des systèmes d'information est étudiée par le domaine de recherche de l'informatique contextuelle. La plupart des définitions du contexte et des dimensions qu'il recouvre privilégient la mesure de l'environnement de l'utilisateur pour adapter l'utilisation des systèmes mobiles. Dans un système distribué, mobile pour l'ensemble de sa plate-forme, les conditions d'exécution et le contexte changent localement pour chacun des composants. L'impact global de ces variations locales doit être mesuré à la conception, et nécessite un nouveau dimensionnement du contexte qui dépasse le niveau des utilisateurs pour inclure l'ensemble du système.

– **Est-il possible lors de la conception de catégoriser les situations d'exécution que le système rencontrera ?**
La description des variables du contexte d'un système mobile donné dimensionne l'ensemble des états qu'il peut prendre à l'exécution. Pour implémenter un système contextuel, les concepteurs devraient prévoir des comportements appropriés en réaction à certaines de ces situations, et en éluder d'autres. En experts du domaine d'application, ils estiment la vraisemblance d'une situation et l'utilité d'y réagir par une adaptation. Établir la liste des situations d'exécution et la pertinence des adaptations lorsque chaque composant génère un contexte propre devient une tâche complexe. Une telle démarche implique le développement de méthodes semi-automatiques pour fournir aux concepteurs les situations d'exécution les plus probables et les plus utiles, et leur permettre de concentrer leurs efforts sur les réactions à apporter à ces situations.

– **Comment rendre compte de ces situations d'exécution dans une démarche de conception ? Quel impact provoque leur intégration pour la modélisation du système ?**
Après avoir déterminé les dimensions du contexte et présélectionné un ensemble de situations d'exécution d'un système mobile et distribué, se pose la question de la capacité d'une démarche de conception à intégrer ces situations. En particulier, dans le cas des systèmes à la fois mobiles et distribués, l'un des principaux défis consiste à faire cohabiter dans une même démarche de conception des situations provoquées par des variations du contexte du système ou de l'infrastructure réseau, avec des situations correspondant à des variations des propriétés de l'utilisateur et de son environnement.

– **Parmi les situations prévues à la conception, comment offrir une latitude d'exécution suffisante pour améliorer la valeur du système ?**
Les situations contextuelles caractérisées à la conception ne représentent en définitive qu'une vue très partielle de la réalité du contexte d'exécution. Quand il se retrouve face à une situation contextuelle nouvelle, l'utilisateur reste le meilleur juge de la nécessité d'une adaptation du système. Les possibilités offertes en termes de personnalisation selon des usages ou des préférences viennent compléter à un grain plus fin les adaptations d'exécution prévues dès la conception. La personnalisation dans les systèmes mobiles et distribués passe par l'étude de nouvelles méthodes, implicites ou explicites, de mesure du contexte utilisateur.

Ces problématiques de recherche partagent un questionnement sur l'influence de la mobilité spatiale et temporelle dans la détermination de situations rencontrées à l'exécution. L'étude approfondie des verrous scientifiques soulevés par chacune des questions dépasse largement le cadre d'une seule thèse. Pour répondre à la plupart d'entre elles, des travaux originaux et de qualité ont déjà été menés. Ils seront présentés en bibliographie de cette étude et dès que nécessaire dans la suite du mémoire.

En se reposant sur l'existant, l'originalité de cette étude est à rechercher dans l'enchaînement et l'adaptation des solutions existantes, ainsi que dans l'approche "géographique" de la notion de contexte. Ces contributions visent à simplifier la conception et l'utilisation de systèmes mobiles et distribués. Cette étude ne propose pas de définition propre de la notion de contexte, mais réorganise différents éléments de définition existants pour correspondre au mieux à la description de l'environnement d'un système mobile. Elle ne propose non plus pas de nouvelles mesures implicites des préférences des utilisateurs, mais justifie l'application de méthodes existantes dans le cas de la mobilité.

Objectifs et plan du mémoire

Un ensemble de domaines scientifiques fédèrent les travaux détaillés dans ce manuscrit. La décomposition régionale du contexte est associée aux domaines des *sciences de l'information géographique* et de *l'informatique contextuelle*. Les descriptions des systèmes mobiles et distribués sont issus de travaux en *informatique répartie*. L'intégration de ces descriptions du contexte et de la plate-forme dans une démarche conceptuelle constitue une proposition de *modélisation des systèmes d'information*. L'utilisateur placé au centre de la démarche conceptuelle implique d'analyser cette contribution sous l'angle de l'*interaction Humain-machine*. Enfin, l'utilisation à l'exécution des préférences d'utilisateurs pour la recommandation de contenu est associée aux travaux en *informatique participative* et plus spécifiquement aux recherches du domaine des *systèmes de recommandation multi-utilisateurs*.

Ce mémoire aborde les problématiques scientifiques précédemment énoncées sous forme de questions ouvertes en trois parties décomposées en six chapitres, et qui s'articulent autour du thème central de l'apport d'une modélisation de l'environnement géographique pour la conception de systèmes mobiles et distribués (Fig. 1).

L'étude de modèles et de solutions existantes dans la littérature constitue un état de l'art des systèmes mobiles. Cette première partie est développée dans deux chapitres :

- le chapitre "**Systèmes d'information mobiles**" traite de la notion de système mobile et distribué et souligne le besoin de méthodes de développement centrées sur l'analyse des espaces de mobilité de l'ensemble des composants du système. Après une introduction aux systèmes distribués, ce chapitre détaille les différents types de mobilité et les moyens pour les mettre en oeuvre, tant du point de vue de l'infrastructure réseau que du point de vue de la plate-forme cliente. Dans un second temps, infrastructures réseau et clients mobiles sont étudiés d'un point de vue conceptuel et différentes méthodologies pour l'analyse et l'implémentation de systèmes mobiles sont détaillées ;

– le chapitre "**Systèmes d'information sensibles au contexte**" établit la nature contextuelle des systèmes d'information mobiles et la nécessité pour un modèle conceptuel de prendre en compte l'environnement d'usage et d'exécution dans la démarche de modélisation du système. Ce chapitre étudie différentes propositions de décomposition des dimensions de l'environnement au sein de modèles du contexte. Ces dimensions du contexte forment la base d'une proposition de classification du contexte en composantes d'*usage* et d'*exécution*. Dans une seconde partie, ce chapitre détaille quelques méthodes pour la modélisation et la mesure de ces deux composantes du contexte, notamment au sein de systèmes mobiles.

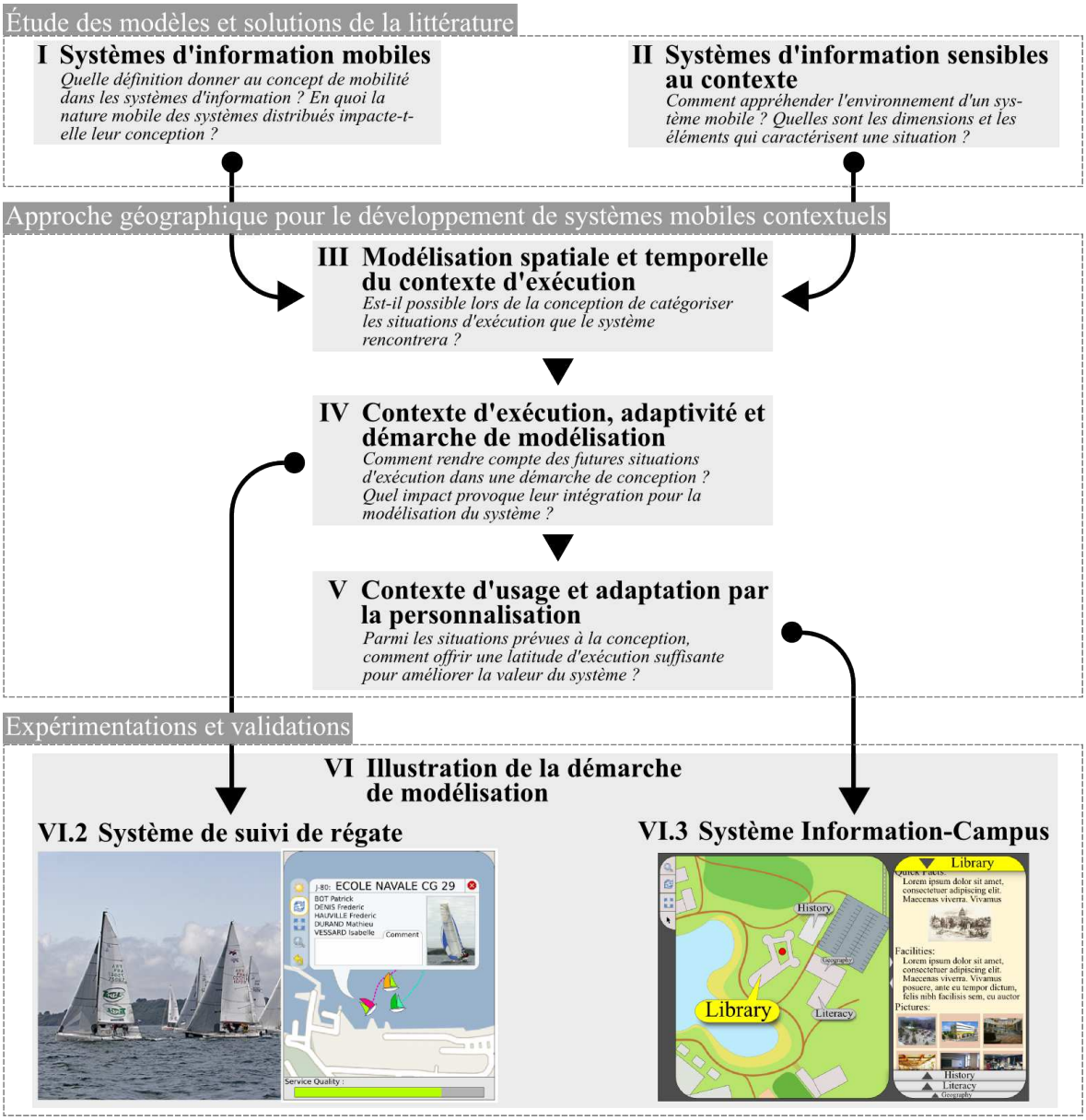


Figure 1 – Organisation et positionnement du mémoire par rapport aux problématiques adressées

Fondée sur l'étude bibliographique, la seconde partie de ce mémoire applique les principes des systèmes mobiles contextuels dans un cadre de modélisation. Celui-ci repose sur

une approche géographique de la mise en oeuvre de systèmes mobiles. Cette seconde partie constitue le coeur des travaux de thèse et est développée dans trois chapitres :

- le chapitre "**Modélisation spatiale et temporelle du contexte d'exécution**" analyse l'exécution d'un système d'information mobile et distribué dans son espace géographique, et construit sa représentation au cours du temps selon un ensemble de régions d'intérêt. Dépendantes physiquement ou sémantiquement des composants du système, ces régions sont mobiles dans une portion limitée de l'espace géographique. L'étude des relations entre régions et des limites de leur déplacement détermine l'ensemble des configurations topologiques pouvant apparaître à l'exécution. Chaque configuration reflète une organisation du système mesurant le *contexte d'exécution* à un moment donné ;

- dans le chapitre "**Contexte d'exécution, adaptivité et démarche de modélisation**", les étapes de modélisation géographique du contexte d'exécution sont associées à une démarche de conception de système interactif centrée sur l'analyse scénarisée des besoins et des tâches des utilisateurs. Les configurations du contexte d'exécution constituent une entrée complémentaire de la démarche pour spatialiser l'organisation des tâches et compléter les besoins des utilisateurs en regard des capacités offertes par le système ;

- le chapitre "**Contexte d'usage et adaptation par la personnalisation**" se concentre sur les utilisations possibles du modèle de l'environnement géographique pour la définition des dimensions du contexte d'usage. Dans le modèle proposé, l'approche géographique offre de nouveaux critères de regroupement pour la collaboration inter-utilisateurs. Plus particulièrement, la dimension contextuelle de l'utilisateur est mesurée de façon implicite au sein de profils par l'étude des comportements de groupes partageant une proximité contextuelle d'exécution. Les profils de groupe générés bénéficient des déplacements des composants du système pour diffuser préférences et habitudes dans tout l'espace du système.

Dans une troisième partie, le chapitre "**Illustration de la démarche de modélisation**" expérimente deux cas d'étude. Le cas d'étude du ***système de suivi de régate*** concerne la mise en place d'un outil de suivi en temps réel de compétition nautique à destination des spectateurs de l'évènement. Les premières étapes de la conception sont détaillées, depuis la modélisation de l'espace du système et des régions d'intérêt, jusqu'à l'intégration des différentes situations d'exécution au sein du modèle des tâches des utilisateurs. Le second cas d'étude du ***système d'information-campus*** met en place un service de documentation sur les bâtiments d'un campus universitaire. Ce système, construit selon l'approche des régions d'intérêt, illustre les principes de mesure implicite des profils personnels, de leur fusion et de l'adaptation des interfaces et données présentées au client.

Le chapitre de **conclusion** reprend l'ensemble des problématiques scientifiques adressées dans cette étude. Pour chacune d'elle, un bilan de l'apport et des inconvénients de l'approche géographique pour la modélisation des systèmes mobiles est présenté. En extension, différentes voies d'amélioration possibles des modèles et des développements sont détaillées.

CHAPITRE I

Systèmes d'information mobiles

Sommaire

I.1 Évolution des systèmes d'information distribués 8
I.2 Systèmes d'information distribués et mobilité 9
I.2.1 Principes et origines de la mobilité des systèmes 10
I.2.2 Infrastructures réseau pour la transmission d'informations 11
I.2.3 Plates-formes clientes pour la mobilité 13
I.2.4 Enjeux de la mobilité des systèmes 15
I.3 Solutions pour la mise en œuvre d'un système mobile 17
I.3.1 Conception à bas niveau : distribution, réseau et communication . . 17
I.3.2 Conception à haut niveau : Plate-forme cliente 20
I.4 Conclusion 30

"Les avancées technologiques les plus fondamentales sont celles qui savent s'effacer. Elles sont les fils qui tissent notre quotidien, jusqu'à en devenir indiscernables" (). Cette vision de technologies présentes mais non intrusives, qui accompagnent les usages du quotidien et les déplacements des utilisateurs, a guidé vingt années de recherches en informatique ubiquitaire (). L'ubiquité informatique adresse les problématiques de l'utilisation dans un environnement changeant par l'approche des systèmes contextuels, et de l'utilisation pervasive, par l'approche des systèmes mobiles.

Ce chapitre introduit la notion de systèmes mobiles comme une première approche du développement de systèmes ubiquitaires. En introduction, et dans une perspective historique, l'émergence des systèmes mobiles se situe comme une conséquence du développement de systèmes distribués. Plus particulièrement, les méthodes de conception de systèmes mobiles reposent sur un ensemble d'outils qui dépassent le cadre du développement des systèmes distribués et qui incluent des méthodes pour prendre en compte la dynamique du système à l'exécution. Les avancées technologiques des infrastructures réseau et des plates-formes clientes qui ont rendu possible le développement de l'informatique mobile sont analysées et comparées. L'énoncé des enjeux et des perspectives offertes par la mobilité en tant que nouveau paradigme d'interaction avec les systèmes d'information guidera le développement de la suite du chapitre.

Dans un second temps, un état de l'art des différentes solutions pour l'implémentation de systèmes mobiles est proposé et les travaux fondateurs pour le développement d'infra-

structures client-serveur supportant la mobilité sont présentés. Ces travaux sont repris dans différentes approches conceptuelles et solutions prototypées pour l'interaction en mobilité avec la plate-forme cliente. La conclusion établit un bilan des apports des systèmes mobiles dans la définition de l'informatique ubiquitaire, en résume les enjeux majeurs et les futurs développements.

I.1 Évolution des systèmes d'information distribués

Depuis les premières communications inter-systèmes entre l'université de Californie Los Angeles (UCLA) et l'Institut de Recherche Stanford (SRI) à la fin des années soixante, les systèmes informatiques se confrontent au monde de la distribution et des réseaux. L'avènement d'Internet, réseau de dimension mondiale, et de ses façades "grand public" du Mail et du Web a démocratisé l'utilisation de systèmes d'information répartis à l'échelle de la planète (). Pour adresser cette distribution du point de vue de l'architecture logicielle, les systèmes d'information sont décomposés en briques fonctionnelles élémentaires, qui constituent les **composants du système**. En général, une partie métier est déployée sur des **composants serveurs** qui garantissent l'accès distant du système aux utilisateurs. Ces derniers interagissent avec le système par le biais d'un **composant client** qui propose une interface d'accès et de modification des informations enregistrées au niveau de la partie métier (). Cette distribution permet à un utilisateur d'accéder à un même système d'information depuis différents espaces physiques (Fig. I.1) : l'*espace des ressources* matérialise les services qui supportent la partie métier du système, les espaces domestique ou professionnel sont des exemples d'*espaces locaux d'utilisation* qui matérialisent les clients du système et leurs aires d'accès au système pour l'utilisateur (p. ex. pièce, bureau, etc.) (;).

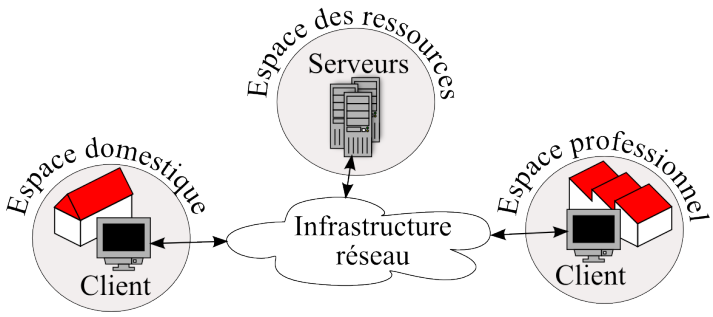


Figure I.1 – Principe de distribution d'un système d'information

L'infrastructure réseau lie les composants entre eux et garantit en particulier la communication entre les clients et les serveurs. Ces architectures dites en *tiers répartis* sont désormais largement adoptées et Internet s'établit comme le support réseau privilégié pour la communication entre tiers (). Du point de vue des utilisateurs, tout client accédant à l'infrastructure réseau peut échanger avec la partie métier. De même, la partie métier, liée à l'infrastructure réseau, capitalise ses ressources au niveau du serveur et rend possible l'échange d'informations avec les clients. Ce lien réseau, inhérent aux systèmes distribués, constitue leur principal atout et leur plus grande faiblesse. et notent que cette prépondérance du réseau au centre de l'infrastructure biaise la modélisation des systèmes distribués et idéalise leur fonctionne-

ment. En particulier, l'infrastructure réseau est présupposée 1) fiable, 2) instantanée dans ses transmissions, 3) de largeur de bande suffisante, 4) homogène, 5) de topologie constante, 6) sûre et 7) gratuite d'accès. Ces présupposés de conception assujettissent l'exécution des systèmes distribués aux aléas d'architecture, d'environnement et d'utilisation. Par exemple, si l'infrastructure réseau est inaccessible, tiers clients comme tiers métiers deviennent isolés et le fonctionnement réparti du système d'information est compromis. D'autre part, les utilisateurs sont contraints dans leurs usages du système au sein d'espaces d'utilisation limités (bureau, salon, etc.) dans lesquels le client accède à l'infrastructure réseau au moyen d'une liaison filaire, c'est-à-dire par un branchement physique tangible entre la plate-forme cliente et le reste de l'infrastructure réseau. Du point de vue de l'utilisateur, il y a discontinuité du système lors de son passage entre deux espaces d'utilisation non connectés. Par exemple, dans le cas du système présenté en figure I.1, l'utilisateur perd l'accès au service lorsqu'il se rend de son domicile à son travail.

Plusieurs approches pour réduire l'impact de ces présupposés conceptuels ont été proposées, chacune s'attachant à la résolution d'un ou plusieurs problèmes en particulier. Par exemple les architectures "pair à pair" proposent une solution aux problèmes de la robustesse de l'infrastructure réseau et des variations dans la topologie des composants (). Les clients légers (*thin clients*) agissent uniquement comme interface de manipulation du système et délèguent l'intégralité de la logique métier et des données au niveau du serveur pour économiser la bande passante et les coûts de communication réseau (). Le langage de description d'infrastructure Darwin permet de concevoir des modèles distribués reconfigurables, qui supportent une topologie réseau dynamique (). Plus généralement, l'exécution d'un système distribué est hautement contextuelle, non seulement du point de vue de la plate-forme cliente, mais aussi pour tous les composants distribués et pour l'infrastructure elle-même. Cette exécution doit donner lieu à une analyse conceptuelle qui intègre la topologie du réseau, l'environnement, le trafic, les coûts, etc. comme des paramètres variant lors de l'exécution. La suite de ce chapitre s'attache plus particulièrement au développement de la mobilité appliquée aux systèmes d'information comme solutions conceptuelles aux biais des infrastructures distribuées. L'émergence de la mobilité comme un cas particulier de systèmes distribués a remis au centre du processus de conception les biais d'architecture et d'infrastructure recensés par . Ceux-ci sont considérés comme des caractéristiques intrinsèques de la mobilité et non plus comme des cas d'erreur de la distribution des systèmes. Les systèmes d'information développés deviennent conscients de leur mobilité ("Mobile-aware" selon) et des limites techniques que celle-ci implique, et y répondent par des adaptations au niveau du transport d'information comme au niveau du dialogue et de l'interaction avec les utilisateurs.

I.2 Systèmes d'information distribués et mobilité

Le nomadisme et la mobilité quotidienne croisent la voie de la technologie, et cette tendance se manifeste par de nouveaux comportements d'utilisateurs. Accompagnés dans leurs déplacements par un ensemble de dispositifs mobiles, téléphones, ordinateurs, assistants personnels, etc., ces utilisateurs veulent accéder à leurs informations personnelles et à différents services d'information quels que soient l'endroit, le moment et les conditions d'utilisation. L'informatique mobile, issue de l'évolution des systèmes distribués, se présente comme un cadre de définition de ces nouveaux usages des systèmes d'information.

I.2.1 Principes et origines de la mobilité des systèmes

Un système distribué devient mobile s'il permet la transition sans interruption d'accès entre ses espaces d'utilisation (). Cette mobilité émerge depuis le milieu des années 90, à la suite des avancées des Nouvelles Technologies de l'Information et de la Communication (NTIC). En particulier, deux principes, apparus avec les NTICs, différencient un système d'information statique et son équivalent mobile :

- l'infrastructure physique du réseau est accessible par l'intermédiaire de liaisons non filaires dans l'espace de mobilité ;
- le client devient mobile, et est utilisé tout au long du parcours de l'utilisateur dans les différents espaces d'utilisation.

généralisent leur étude de l'impact des téléphones portables sur l'utilisation des systèmes d'information, et montrent comment ces deux changements d'infrastructure permettent la concrétisation du principe de mobilité au sein des systèmes d'information et alimentent la tendance au nomadisme de leurs utilisateurs.

Dans le système décrit en introduction (Fig. I.1), un utilisateur se rendant à son lieu de travail effectue un passage d'un espace d'utilisation domestique à un espace d'utilisation professionnel. La transition s'effectuant hors de tout espace d'utilisation, l'utilisateur n'accède plus au service pendant sa période de mobilité. Le pendant mobile de ce système distribué offre à l'utilisateur un chemin d'accès continu entre les deux espaces d'utilisation, par le biais d'un *espace de mobilité* intermédiaire (Fig. I.2).

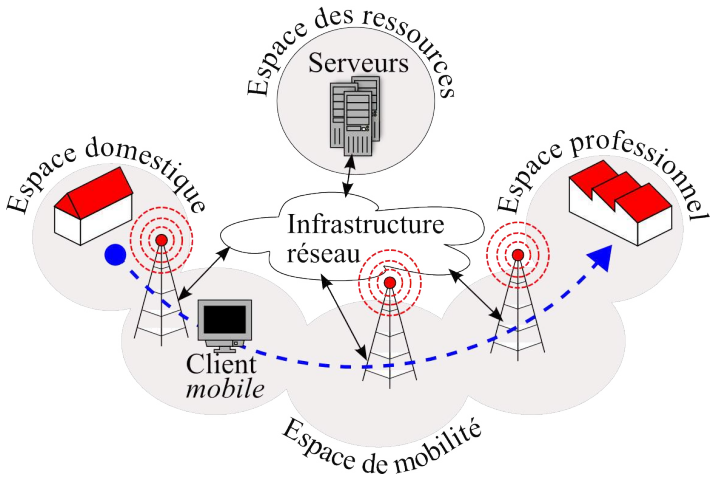


Figure I.2 – Principe de la mobilité dans un système d'information distribué

Durant la transition, l'utilisateur est mobile 1) dans l'espace géographique, le long du trajet vers son bureau ; 2) dans le temps, entre ses moments de départ et d'arrivée ; et 3) dans l'espace des conditions d'environnement caractérisant les influences extérieures au système. reprennent l'espace, le temps et le contexte comme les trois dimensions orthogonales du concept de mobilité dans les relations entres personnes et systèmes. Les auteurs argumentent en faveur d'une mobilité définie non seulement comme la relation d'utilisateurs à un espace géographique (), mais aussi comme un ensemble de déplacements d'objets (p. ex. un ordinateur mobile accompagnant l'utilisateur), de contenus symboliques (p. ex. sons, images,

textes), et d'espaces (p. ex. passage d'un espace géographique à un "cyberespace" dans lequel les distances se mesurent en liens entre des contenus symboliques). Ces trois dimensions de la mobilité des utilisateurs ont été largement développées avec l'apparition de nouveaux moyens et techniques de communication et d'interaction. En particulier, ces déplacements prennent sens dans les systèmes d'information mobiles et sont contraints par les dimensions orthogonales mises en jeu lors de la mobilité () :

- la dimension spatiale référence les lieux d'utilisation des clients mobiles accédant aux services du système. Le client représente la partie de l'utilisateur perceptible par le système ;
- la dimension temporelle représente le moment de l'utilisation et les interactions passées entre le client et le serveur ;
- la dimension contextuelle perçoit l'environnement du système et mesure ses changements pour modifier le comportement du système et le rendre plus approprié.

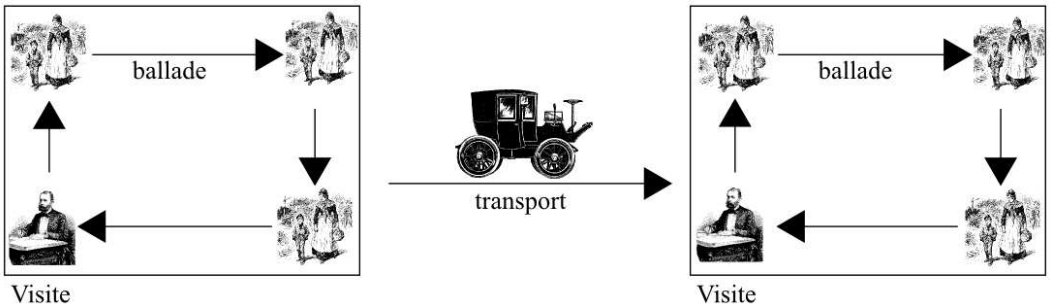


Figure I.3 – Trois types de mobilité selon

catégorisent le nomadisme des utilisateurs selon qu'ils effectuent un *transport*, une *visite* ou une *balade*. Lors d'un *transport*, l'utilisateur emprunte un moyen de locomotion pour se déplacer d'un endroit à un autre. Il est passager ou conducteur et subit des contraintes d'interaction liées à la mobilité du véhicule ou à la charge cognitive engendrée par la conduite. Le système mobile utilisé repose éventuellement sur les ressources du véhicule. Quand il se *balade*, l'utilisateur est mobile lors de son accès au système d'information, mais contrairement au transport, il n'emprunte pas de véhicule. En *visite*, l'utilisateur accède à un système mobile en différents endroits, entre lesquels il se déplace éventuellement en continuant d'accéder au service. Balade et transport permettent de rejoindre deux endroits de visite en gardant accès au système d'information. Ils s'effectuent dans des espaces de mobilité, tandis que les endroits de visite sont associés à des espaces d'utilisation locaux. Dans l'exemple de système mobile représenté en figure I.2, l'utilisateur se déplace entre l'endroit de visite "Domicile" et l'endroit de visite "Professionnel", respectivement deux espaces d'utilisation locaux, par une balade ou un transport au sein d'un espace de mobilité, selon qu'il se rend à pied ou en voiture au travail.

I.2.2 Infrastructures réseau pour la transmission d'informations

Le premier principe pour le développement de systèmes d'information mobiles rend compte de la nécessité de transmettre des données par des liaisons sans fil entre les différents composants du système. Dans le cas des systèmes informatisés, la nature bi-directionnelle des communications, les distances à couvrir, et les volumes importants des données à transmettre

ont soulevé des problèmes techniques limitant l'adoption initiale d'infrastructures de support pour la mobilité. La première infrastructure raisonnablement accessible et disponible pour le développement de systèmes mobiles est apparue avec les réseaux cellulaires et la téléphonie mobile. L'adoption massive par le grand public de ce nouveau moyen de communication a démontré l'intérêt des systèmes mobiles et de nombreuses solutions d'infrastructures de transmission de données ont depuis été développées. classent ces moyens de communication non filaires selon qu'ils reposent sur une *infrastructure fixe* ou une *infrastructure dynamique*.

Parmi les infrastructures fixes, les réseaux cellulaires ont été développés à des échelles nationales, voire internationales. En France et en Europe on retrouve le GSM (Global System of Mobile communications) et son successeur de troisième génération, l'UMTS (Universal Mobile Telecommunication System) comme réseau support de la téléphonie et de l'Internet mobile. Les réseaux satellitaires comme Iridium, Inmarsat ou Globalstar garantissent un accès à l'infrastructure réseau quelle que soit la localisation de l'utilisateur, mais aux dépens de l'encombrement du client et des coûts de communication. Ces réseaux sont utilisés particulièrement en milieu maritime pour communiquer la position et la route des navires de transport de marchandises (). Les systèmes GPS (Global Positioning System) et le futur Galileo, permettent à un client à terre de déterminer sa position et son altitude en se situant par rapport à des satellites localisés (). La transmission est dirigée depuis le satellite vers le client et n'offre pas l'infrastructure bi-directionnelle nécessaire à un système d'information. Toutefois, la localisation est devenue prépondérante pour la formalisation de la dimension contextuelle de la mobilité. La localisation, par l'intermédiaire, par exemple, du système GPS, est considérée comme un constituant essentiel des systèmes mobiles. Enfin, les infrastructures par "point d'accès" Wi-Fi ou Wi-Max établissent des réseaux WLAN (Wireless Local Area Network) dans lesquels l'accès au système est garanti dans une cellule de communication à proximité du point d'accès. Le passage entre plusieurs cellules par des techniques d'itinérance d'adressage (p. ex. le roaming IP) permet un accès continu au support réseau lors des déplacements des composants mobiles du système.

Les infrastructures dynamiques établissent des communications directes entre les constituants mobiles du système d'information, sans passer par une infrastructure fixe. Les réseaux *ad-hoc* considèrent les composants du système comme les nœuds du réseau et établissent des protocoles de routage de l'un à l'autre (). L'architecture est dynamique et se modifie en fonction de la capacité des composants à communiquer entre eux. Les réseaux radio à portée limitée comme BlueTooth ou le Wi-Fi en mode ad-hoc permettent un échange d'informations entre composants très proches, dans une pièce, ou sur un même bureau. Ces liaisons WPAN (Wireless Personal Area Network) structurent en général le tiers client. Par exemple, certains clients mobiles légers reposent sur des liaisons de type BlueTooth ou infrarouge pour communiquer avec une station de travail et accéder aux services de cette dernière. De la même manière, un transport Wi-Fi peut constituer un support de réseau mobile local à un espace d'utilisation domestique, pour communiquer entre une station cliente et ses périphériques numériques (télévision, chaîne Hi-fi)().

Ces infrastructures réseau offrent une palette d'outils matériels pour la mise en place d'architectures mobiles quelle que soit l'échelle (Tab. I.1). De plus, tous ces supports physiques permettent la transmission de paquets IP (Internet Protocol). L'interopérabilité de ces réseaux, garantie par cet adressage commun, permet à un composant de système d'information mobile de transiter de façon transparente d'un niveau de mobilité personnel à un niveau

Tableau I.1 – Technologies de transmission pour l'infrastructure réseau de systèmes mobiles ()

Type	Couverture	Portée	Débit	Exemple
Satellite	monde	$1000km$	$100Kb.s \rightarrow 2Mb.s$	Iridium, Inmarsat
Cellulaire	pays	$100m \rightarrow 30km$	$10Kb.s \rightarrow 1000Kb.s$	GSM, GPRS, UMTS
WLAN	bâtiment	$10m \rightarrow 10km$	$10Mb.s \rightarrow 100Mb.s$	Wi-Fi, Wi-Max
WPAN	pièce	$1m \rightarrow 10m$	$100Kb.s \rightarrow 1000Mb.s$	BlueTooth, Wi-Fi, IrDA, Zigbee

global sans rupture dans l'accès au reste du système. Les moyens de transmission sont en constante évolution, et l'introduction de l'adressage IP dans sa version 6 promet d'améliorer encore la continuité du support réseau au niveau des plates-formes mobiles (). Essentiels pour la mobilité, les routeurs et infrastructures IPV6 supporteront nativement la mobilité de réseau (Network Mobility, NEMO [1]) et le contrôle de la qualité de service (Quality of Service, QoS [2]). Toutefois, ces technologies sont encore marginales et leur mise en place ne se fait que très progressivement. Même si l'interopérabilité des différents réseaux s'améliore et favorise un accès universel au support de transmission, les questions des coûts d'accès, de la portée ou du débit variant selon le support, restent ouvertes et impliquent souvent pour le système de savoir s'adapter aux conditions de transmission.

I.2.3 Plates-formes clientes pour la mobilité

L'adoption massive de la téléphonie mobile a apporté une preuve de l'engouement du grand public pour les systèmes mobiles. Du milieu des années 90 à aujourd'hui, près de la moitié de la population mondiale s'est équipée et dans les pays développés on compte désormais neuf téléphones portables pour dix habitants (). Ce succès a été un préalable à l'émergence des systèmes d'information mobiles. Face aux attentes nouvelles des utilisateurs découvrant les possibilités de l'information en mobilité, les constructeurs et les opérateurs poussent la recherche et l'innovation et développent des clients aptes à de nouveaux usages. En particulier, la miniaturisation des composants électroniques et les améliorations en termes de consommation et de stockage d'énergie transforment les terminaux mobiles en instruments multimédia. Ces avancées technologiques permettent le passage d'une utilisation de clients mobiles mono-usage (p. ex. téléphone GSM pour téléphoner, pager pour envoyer des messages ou récepteur GPS pour se repérer) à des clients mobiles multi-usages (p. ex. smartphone de type Iphone pour téléphoner, se repérer, envoyer des messages, etc.) (). Ces clients mobiles se rapportent historiquement à différentes familles d'appareils (Fig. I.4) :

- **Les téléphones portables** reposent sur les réseaux de communication cellulaires et proposent un ensemble de services entres plusieurs utilisateurs : conversation orale, messagerie instantanée, partage de photos et vidéos. Les faibles ressources énergétiques nécessaires au fonctionnement de ces systèmes offrent aux téléphones portables une longue autonomie.
- **Les ordinateurs portables** ont été les premiers clients mobiles. Ils offrent des capacités matérielles similaires aux ordinateurs de bureau et peuvent exécuter les mêmes

1. http ://tools.ietf.org/html/rfc3963
2. http ://tools.ietf.org/html/rfc2990

outils logiciels. Par rapport aux autres plate-formes mobiles, leur encombrement et leur faible autonomie limitent toutefois leur mobilité.

- **Les assistants personnels**, outils de gestion d'agenda et de carnet d'adresses, sont des ordinateurs de poche d'une autonomie moyenne. A leur origine, ces clients disposaient de capacités réseau très limitées (type WPAN), voire inexistantes.
- **Les systèmes embarqués** sont associés à un système de transport. Par exemple, les systèmes de navigation pour l'automobile utilisent les ressources énergétiques du véhicule et sont moins contraints par les besoins de miniaturisation et d'autonomie. Avec les assistant personnels, cette catégorie est aujourd'hui en très forte progression sur le marché grand public des dispositifs nomades.
- **Les ordinateurs portés** (Wearable computers) suivent leurs utilisateurs dans leur mobilité. Ils perçoivent leur environnement et s'y adaptent pour proposer une interaction naturelle à l'utilisateur, sans le perturber dans sa tâche (). Les ordinateurs portés représentent un futur possible pour l'informatique embarquée, dans lequel le système et ses services font partie d'un environnement technologique "tranquille" ("Calm technology",) auquel l'utilisateur accède naturellement, en harmonie avec son activité.

Figure I.4 – Exemples de clients pour les systèmes d'information mobiles

Aujourd'hui, les frontières entre ces différentes familles de clients mobiles tendent à s'amenuiser et leurs domaines d'utilisation s'étendent pour couvrir la palette des types de mobilité décrits par (Tab. I.2) (). Les téléphones portables se sont enrichis de claviers (physiques ou virtuels) et d'appareils photo pour devenir des unités multimédia autonomes. imaginent des systèmes mobiles reposant sur des unités communicantes de la taille de téléphones portables et qui peuvent se connecter aux interfaces d'interaction disponibles aux endroits de visite, tout en proposant des interfaces appropriées lors des transports ou des balades. Les ordinateurs portables voient leur taille se réduire, et leurs interfaces prennent mieux en compte la mobilité (p. ex. utilisation de tablettes tactiles). Les assistants personnels augmentent en capacité de calcul et de stockage pour devenir des outils de traitement comparables aux ordinateurs de bureau. Enfin, ces appareils intègrent souvent des moyens de mesure de leur environnement d'exécution. Les plates-formes mobiles accueillent des récepteurs GPS pour la localisation, les ordinateurs portables mesurent la lumière ambiante pour adapter la

luminosité de l'écran ou le rétro-éclairage du clavier ; les capteurs inertiels fournissent des informations sur le placement et les mouvements de l'appareil dans l'espace.

Tableau I.2 – Domaines d'utilisation des plates-formes clientes selon les types de mobilité (d'après)

	Transport	Balade	Visite
Ordinateur porté		✓	
Système embarqué	✓		
Assistant personnel	✓	✓	
Ordinateur portable	✓		✓
Téléphone portable	✓	✓	
Ordinateur de bureau (fixe)			✓

Cette convergence entre les mondes de la téléphonie, de l'assistance à la navigation, et de l'informatique embarquée se retrouve dans les outils logiciels disponibles pour ces clients. Les grands acteurs de l'industrie logicielle tentent d'unifier le développement d'outils pour cette variété de plates-formes et chacun propose une solution logicielle pour la mobilité. Par exemple, Apple [3] et Microsoft [4] proposent des versions simplifiées de leurs systèmes d'exploitation optimisées pour la mobilité. Google développe une boîte à outils logicielle pour exploiter ses services de recherche et de communication via Internet sur une plate-forme mobile [5]. Ces nouveaux outils d'exploitation des clients mobiles s'accompagnent de librairies de programmation qui abstraient le niveau physique de la plate-forme cliente et permettent le développement rapide de logiciels adaptés à une catégorie entière d'appareils ().

I.2.4 Enjeux de la mobilité des systèmes

L'accès sans fil à un réseau et le développement des plates-formes clientes ont été les changements technologiques nécessaires pour permettre à l'utilisateur d'être mobile tout en conservant sa capacité à réaliser sa tâche et atteindre ses buts associés à l'utilisation d'un système d'information. Progressivement, au gré des évolutions de ces deux domaines, l'utilisateur mobile accèdera à ses données et outils de travail ou de loisir n'importe où et n'importe quand (). expriment ce changement de contexte d'utilisation entre l'environnement confiné du bureau, connu et sûr, et le monde sans frontière tangible de l'utilisation en mobilité comme le principal défi de la mise en œuvre des systèmes d'information mobiles. Les difficultés rencontrées sont d'abord d'ordre technique : gestion de l'énergie (), localisation et suivi des utilisateurs (), adaptation des méthodologies de conception (), gestion des interactions avec le client (). Le développement et l'adoption des systèmes mobiles passent ensuite par un changement de la représentation que s'en font les utilisateurs. Plus particulièrement, montrent qu'un fossé sépare toujours les attentes des utilisateurs nomades vis-à vis des systèmes d'information et la réalité matérielle et logicielle de ce qui est à leur disposition.

retiennent les problématiques de la **communication**, de la **mobilité** et de la **portabilité** comme axes de recherches pour le développement de systèmes mobiles :

3. OS X iPhone : http ://developer.apple.com/iphone/
4. Windows Mobile : http ://www.microsoft.com/windowsmobile/en-us/default.mspx
5. Android : http ://www.android.com

- l'étude des moyens de communication porte sur le développement d'une infrastructure réseau en support de la mobilité ;
- la dimension de la mobilité pose le problème du suivi et de l'adressage des composants mobiles ;
- la portabilité adresse les nouveaux enjeux de la plate-forme de l'utilisateur, avec en particulier les styles d'interaction et les capacités de traitement en mobilité.

En partant du même point de vue technique, résume les particularités des systèmes mobiles selon quatre contraintes nouvelles : 1) les plates-formes mobiles disposent de **peu de ressources** par rapport à leurs équivalents statiques contemporains ; 2) **la mobilité est "risquée"** pour les utilisateurs et les plates-formes qui sont vulnérables aux contraintes extérieures (p. ex. perturbation par un voisinage bruyant, casse ou panne de la plate-forme cliente, etc.) ; 3) les capacités d'interconnexion d'éléments mobiles varient en *performance et fiabilité* ; et 4) les plates-formes mobiles reposent sur une *source d'énergie finie*.

Ces caractéristiques, indépendantes des progrès techniques, montrent que la mobilité dans les systèmes d'information est *particulière*, et illustrent le besoin de méthodes de conception appropriées consacrant une place importante à l'adaptation du système. appellent trois changements méthodologiques pour une conception plus efficace des systèmes d'information mobiles :

- la mesure des besoins de l'utilisateur doit être si possible corrélée aux conditions d'utilisation du système et cette mise en relation doit intervenir en amont du processus de conception ;
- les performances et l'organisation de l'ensemble du système doivent être mesurées et caractérisées en termes de capacités techniques au sein du processus de conception ;
- la méthodologie de développement doit être capable d'intégrer des changements technologiques imprévus, et des modalités d'interaction nouvelles au niveau du client.

Dans leurs définitions des enjeux de la mobilité, , et reviennent chacun implicitement sur les biais des systèmes distribués, et montrent qu'adresser les enjeux de la mobilité par des méthodes de conception appropriées permet dans une large mesure d'éviter les biais de la distribution. Bien qu'étant apparus comme une évolution des systèmes distribués, les systèmes mobiles présentent des conditions d'exécution plus libres, conséquence en particulier de la variabilité de leur environnement, et leurs méthodes de conception doivent adresser tout ou partie des pré-supposés conceptuels des systèmes distribués. Dans cette optique, les méthodes de conception de systèmes mobiles contribuent au développement des systèmes distribués. Construit comme un système mobile dont les composants sont fixes à un moment donné, un système distribué non mobile échapperait aux biais rapportés dans . En plus de cette unification des démarches de modélisation des systèmes distribués et mobiles, ; puis plus récemment rappellent que la mobilité de la plate-forme utilisateur engendre de nouvelles utilisations, pour lesquelles le milieu d'exécution et les limitations techniques ne peuvent être ignorées. Les auteurs voient dans le couplage de l'informatique mobile et de l'informatique sensible au contexte une approche pour l'étude de ces nouvelles utilisations et une solution d'adressage de la problématique posée par un environnement d'exécution dynamique.

I.3 Solutions pour la mise en œuvre d'un système mobile

L'interaction utilisateur/plate-forme, l'organisation du système mobile et la souplesse de son développement constituent les étapes de conception et d'implémentation des systèmes mobiles. En réponse à ces enjeux méthodologiques, un ensemble de travaux de recherche adresse la problématique de la conception de systèmes d'information mobiles. Cette section détaille les recherches menées aux niveaux de l'infrastructure de communication et de la plate-forme cliente, et présente quelques méthodologies existantes pour la conception, l'implémentation et le déploiement d'applications mobiles.

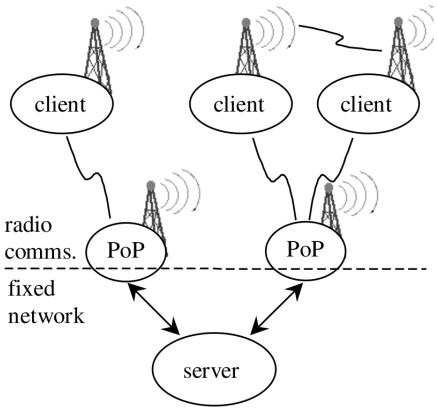
I.3.1 Conception à bas niveau : distribution, réseau et communication

La mise en place d'une infrastructure de communication constitue une première étape du développement de systèmes d'information mobiles. De nombreux travaux de recherche proposent des infrastructures originales, servant de cadre de démonstration aux principes et enjeux de la mobilité. Ils offrent des exemples d'abstraction des couches réseau ou de l'accès aux données, et fournissent un cadre de développement pour la formalisation de modèles de systèmes d'information mobiles (). Dans la suite, quelques exemples d'infrastructures pour la mobilité des systèmes sont détaillés. Ces solutions d'infrastructures sont positionnées par rapport aux biais des systèmes distribués qu'elles adressent [6].

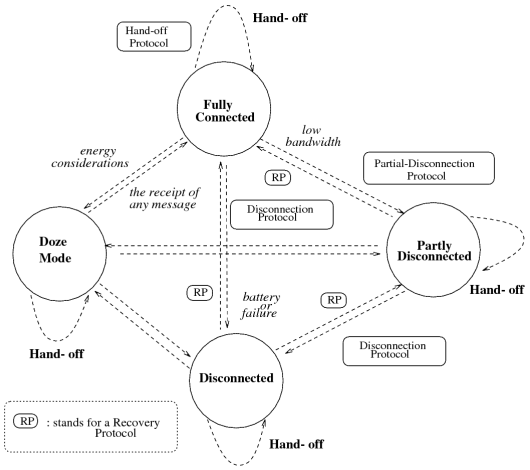
proposent un cadre de conception qui distingue différents modes de connexion au support réseau : le client peut être connecté, partiellement déconnecté, déconnecté, ou placé dans un mode de repos pour économiser de l'énergie (Fig. I.5b). À l'exécution, le client mobile passe d'un mode à un autre en fonction de l'état du support et de l'activité de l'utilisateur. Le cadre de conception proposé traite ces transitions par des protocoles de changement d'état. Par exemple, le protocole de déconnexion partielle réduit au maximum les communications entre client et serveur en activant des caches de données au niveau du client. Dans leur cadre de conception, les auteurs considèrent les changements d'état comme des opérations normales qui ne doivent pas générer d'erreurs de fonctionnement pour le client. Leur cadre de conception ne précise toutefois pas à quels moments ni comment les transitions sont détectées. Par exemple, le protocole de déconnexion doit être exécuté avant la coupure réseau effective pour avoir le temps de transmettre des caches de données depuis le serveur. Les auteurs ne précisent pas comment prévoir une déconnexion avant que celle-ci ne soit opérée.

proposent une démarche conceptuelle pour les systèmes mobiles reposant sur la notion de "Points de présence" à l'interface entre l'utilisateur et l'infrastructure réseau filaire (Fig. I.5a). Ces composants matériels, qui étendent le modèle client-serveur, fonctionnent comme des proxies réseau et répliquent une partie des données et des traitements du serveur dans un espace proche des utilisateurs. Lors d'une utilisation collaborative, les points de présence groupent des ensembles d'utilisateurs du système et évitent le recours systématique aux traitements du serveur. Le support réseau est économisé et la proximité des utilisateurs aux

6. Ces exemples sont considérés comme représentatifs de différentes propositions d'infrastructures pour les systèmes mobiles. Le lecteur intéressé se réfèrera à pour une présentation plus exhaustive de différents systèmes mobiles et distribués et des infrastructures de communication associées.



(a) Architecture intermédiaire des "Points de présence" ()



(b) Mode de connexion au support réseau ()

Figure I.5 – Solutions pour l'architecture et le support des infrastructures mobiles

points de présence permet une meilleure réactivité, en particulier au niveau des temps de réponse de l'interface d'accès sur le client.

Dans leur approche conceptuelle, reprennent l'idée d'élément proxy et l'appliquent au cas des systèmes de commerce électronique mobiles, pour lesquels la sécurité des connexions est primordiale. En aval, leurs "Intergiciels mobiles" (*Mobile middleware*) assurent la permanence et la sécurité de la communication vers les clients. En amont, ils agissent comme les briques de connexion aux réseaux mobiles. Ils s'adaptent aux technologies de transfert réseau à disposition et assurent la gestion des interruptions de transaction lors du passage d'un réseau mobile à un autre. Cette infrastructure constitue le point de départ de leur proposition conceptuelle. Elle pose un cadre de développement unifié pour implémenter des services de e-commerce adaptés à la fois à un usage statique (en visite) et à un usage mobile (en balade ou en transport). Les auteurs précisent cependant que les coûts non négligeables de mise en œuvre de l'intergiciel mobile, et la difficulté de caractérisation des capacités du client, restreignent l'adoption de leur plate-forme.

Dans le domaine maritime, le projet Share-Loc propose une infrastructure mobile pour le suivi de bateaux en milieu côtier (). Ce système repose sur le réseau GSM pour transmettre les positions de navires de plaisance alentour vers les téléphones portables des marins. Le cadre de conception proposé repose sur des technologies ouvertes pour l'accès aux données de positionnement. Les téléphones cellulaires sont des clients WAP qui interrogent à intervalles réguliers un service Web pour connaître l'évolution de la situation maritime. Pour économiser sur les coûts de connexion au réseau GSM, une passerelle entre le service Web et le client WAP encode et compresse les données XML en des fichiers binaires de taille moindre. Tant que les navigateurs restent dans la zone côtière de couverture du réseau GSM, l'accès au service Web est assuré. Le passage d'une cellule à l'autre sans déconnexion ("roaming") y assure la continuité du service. Par contre, le modèle ne propose pas d'adaptation particulière en cas de sortie de la zone de couverture.

propose de ré-attribuer les rôles des clients et des serveurs des systèmes distribués pour mieux les préparer à la mobilité. En particulier, le client peut s'attribuer provisoirement une partie des tâches du serveur et limiter sa dépendance à l'infrastructure réseau pour les exécuter. Inversement, si le client est limité en ressources, le serveur peut soulager le client et prendre le rôle de générateur d'interface. Cette répartition dynamique des rôles, essentiellement logicielle, évite l'installation d'un proxy à l'interface entre les réseaux mobile et statique.

En sur-couche du modèle proposé par , fournit la boîte à outils de développement "Odyssey" pour gérer l'attribution des rôles de client et de serveur. Les outils mobiles, ici un lecteur multimédia et une application de cartographie, utilisent les librairies d'Odyssey pour accéder aux ressources du réseau. La boîte à outils se charge de dégrader ou d'améliorer la qualité des ressources en fonction de l'état du réseau (). L'auteur introduit la notion de *fidélité* comme mesure de la dégradation de qualité entre les données présentées au client et la copie de référence du serveur. La décision de limiter la qualité des données transmises est partagée entre un superviseur de l'état réseau fourni par Odyssey et l'utilisateur par l'intermédiaire de l'application cliente. Ce partage des responsabilités permet à une application de revenir sur une politique bas niveau de transmission réseau. Le système mobile devient conscient de l'application cliente ("application aware" selon l'auteur) et permet de contrôler la qualité de transmission en mobilité au niveau des logiciels du client.

Grille de lecture croisée : niveau de l'infrastructure mobile

Le tableau I.3 reprend les différents travaux présentés et résume leur contribution pour atténuer ou solutionner les problèmes de conception et de fonctionnement des systèmes distribués énoncés par . Du point de vue de la fiabilité, les différents travaux s'accordent sur la nécessité de prévoir les pannes des composants ou du support. Ils proposent de répliquer les données et les procédures de traitement du serveur, soit au niveau de composants matériels intermédiaires (;), soit directement au niveau du client (;).

Le problème des délais d'accès au support réseau est devenu moins fédérateur ces dernières années. Dans le cas des systèmes d'information, les progrès techniques permettent de négliger en partie la latence dans les transmissions. précise cependant que certaines applications comme les interfaces utilisateur distribuées doivent rester très réactives et que la problématique des délais du support reste entière dans ces cas. Les délais d'accès restent aussi un thème d'innovation et de recherche important pour la communication d'informations en temps réel, dans le cas par exemple de la mise en place de réseaux de capteurs.

Deux approches proposent de limiter la consommation de bande passante. D'une part, les données transmises sur le support réseau peuvent être réduites par une compression destructive ou non (;). D'autre part, propose de mieux répartir la charge des communications au niveau des composants proches de l'utilisateur pour ne transmettre depuis le ou les serveur(s) distant(s) que le minimum des informations nécessaires au fonctionnement du client.

Du point de vue de l'homogénéité des composants, la diversité des plate-formes clientes est généralement constatée. gèrent cette variété de clients par l'utilisation de formats de

Tableau I.3 – Solutions apportées par les différents cadres de conception de systèmes mobiles aux contraintes de développement des systèmes distribués

fiabilité	cache de données sur le client	cache de données sur les points de présence	connexion Middleware ↔ client garantie	-	échange des rôles entre clients et serveurs
délais	mode partiellement connecté	points de présence proches du client	-	-	-
bande passante	-	répartition par sous-réseau	pari de l'augmentation	compression des données	qualité de service adaptée
non-homogénéité	constat de la diversité des clients	-	pari de la convergence	formats ouverts, facilement utilisables	données adaptées par client ciblé
topologie dynamique	notion de proxies de transaction	points de présence en réseau pair à pair	transition statique ↔ mobile	réseau GSM acceptant le roaming	considérée comme une perspective
sécurité	-	limitation admise	encodage des échanges	transmission de données binaires	-
coûts	-	-	limitation admise	réduction des volumes de transmission	-

données ouverts. choisit d'adapter la qualité des informations en fonction des capacités d'affichage et de traitement du client.

La topologie dynamique fait partie de la définition des systèmes mobiles : au minimum, l'utilisateur et la plate-forme cliente sont en mouvement. Leurs déplacements impliquent des reconfigurations régulières de la topologie du réseau. ; utilisent leurs composants intermédiaires pour gérer la topologie dynamique, et notamment le passage d'un réseau fixe à un réseau sans fil. choisissent d'utiliser un support de transmission gérant naturellement la mobilité du client par des techniques d'itinérance (roaming ()). proposent la notion de proxies de transaction au niveau du client, qui permettent de répéter les transmissions qui auraient été perdues lors d'un changement de support réseau.

La sécurité est un aspect essentiel des communications dans le cadre d'applications marchandes ; recommandent l'encodage des échanges entre clients et serveurs. Pour , la sécurité des échanges est une conséquence de la réduction des volumes échangés et le passage de données de type XML à des échanges binaires, encodés et non lisibles.

Enfin, la thématique des coûts a été largement éludée. Le support réseau est toujours considéré comme gratuit d'accès et les coûts de mise en place de l'infrastructure n'ont eu que peu d'impact dans les solutions proposées. , qui utilisent le réseau payant GSM comme infrastructure réseau, cherchent à limiter les coûts d'utilisation du support en compressant les données et en transmettant les informations uniquement si nécessaire.

I.3.2 Conception à haut niveau : Plate-forme cliente

Les plates-formes clientes des systèmes d'information mobiles posent de nouveaux défis conceptuels, en particulier dans le domaine de l'ingénierie des interactions Homme-machine. Même si leurs capacités suivent une croissance exponentielle, les clients mobiles doivent concilier les particularités de leurs interfaces aux nouveaux besoins des utilisateurs pour l'inter-

action avec des données riches quel que soit l'endroit où ils se trouvent, et quel que soit l'environnement d'utilisation. Selon , les plate-formes mobiles représentent quatre défis nouveaux pour la conception et le développement logiciel :

- **Conception pour la mobilité** : les utilisateurs mobiles ne disposent pas d'un environnement de travail cohérent. Les concepteurs doivent tenir compte en particulier du fait que l'utilisateur est seul avec sa plate-forme pour l'interaction avec le système (pas de bureau, d'imprimante, de notes papier, etc.)
- **Conception pour une population hétérogène** : les plates-formes clientes mobiles s'éloignent des ordinateurs conventionnels et sont utilisées par des populations hétéroclites, dont les comportements et les attentes vis-à-vis du système diffèrent sensiblement. De plus en plus, la conception de système mobiles passe par une meilleure appréhension du ou des public(s) concernés, et notamment de leurs différentes approches de la mobilité.
- **Conception pour des interfaces réduites** : la mobilité impose que le client soit portable, ce qui limite ses interfaces et complique l'interaction avec l'utilisateur. Prendre en compte ces limites à la conception définit le cadre des capacités interactives du système.
- **Conception pour un contexte dynamique** : la plate-forme mobile se dote des moyens de mesure de l'environnement dans lequel elle s'exécute. Cette mesure du contexte complexifie la conception des systèmes mobiles mais permet une granularité plus fine de la tâche de l'utilisateur.

Dans la suite de la section, ces orientations conceptuelles pour la plate-forme cliente de systèmes mobiles sont définies. Pour chacune, les travaux fondateurs et différents prototypes sont présentés et expliqués. Les approches de validation de ces systèmes mobiles, et les perspectives de recherche sont détaillées ensuite. Enfin, une grille de lecture positionne les différents travaux présentés selon l'orientation des recherches présentées et selon la problématique qu'ils adressent.

Conception pour la mobilité

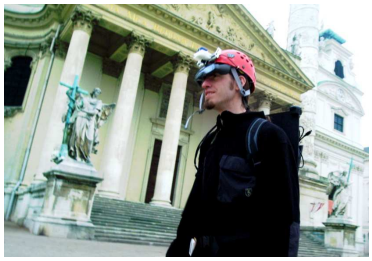
Les systèmes d'information mobiles sont envisagés comme un paradigme d'utilisation entièrement différent de l'approche fixe au moyen d'un ordinateur de bureau. Les utilisateurs se déplacent, passent du temps en extérieur, alternent leurs occupations, se rencontrent et collaborent (;). Selon ces nouveaux usages, associés aux évolutions techniques des réseaux et de l'électronique mobile, appellent un changement de paradigme d'interaction. Par exemple, montrent que malgré les nombreuses expérimentations dans le domaine des guides touristiques numériques, la plate-forme mobile, en général de type "assistant personnel", limite la mobilité de l'utilisateur et perturbe son activité (p. ex. un utilisateur en *ballade* s'arrête pour consulter son assistant personnel). Les auteurs montrent que parfois ces limites poussent les utilisateurs à se passer des outils proposés. Pour , le passage à une informatique mobile devra nécessairement dépasser le cadre connu de la métaphore du bureau et des interactions WIMP [7] pour proposer des nouvelles formes d'interaction. Pour les auteurs, la mobilité implique plus qu'un simple transfert de nos habitudes d'interaction avec les outils informatiques.

7. Window, Icon, Menu, Pointer : les quatre éléments d'interface de la métaphore du bureau.

Plusieurs initiatives de recherche rejoignent ce point de vue et conçoivent des plates-formes mobiles adaptées à certaines utilisations plutôt que d'adapter les usages à une plate-forme existante. Dans le domaine des assistants touristiques, les projets Camelio et EdFest proposent des *interactions naturelles* en mobilité et explorent notamment les utilisations du papier comme élément d'interface (;). EdFest enrichit des plans touristiques conventionnels par des informations localisées. L'utilisateur manipule une carte papier, un stylo interactif, et écoute les informations retournées par le système (Fig. I.6a).

(a) Interface naturelle : EdFest ()

(b) Réalité augmentée ()



(c) Ordinateur porté : visite de Vienne ()

Figure I.6 – Exemples de clients de systèmes d'interaction conçus pour la mobilité

La *réalité augmentée* représente une étape supplémentaire par rapport aux interfaces naturelles. Les objets de l'environnement servent non seulement à la saisie d'informations, mais sont aussi augmentés par des informations numériques. Par exemple, décrivent une visite de musée où les œuvres sont augmentées d'informations multimédia. reprend l'idée de la réalité augmentée en mobilité et envisage la collaboration multi-utilisateurs autour d'espaces d'affichage communs (Fig. I.6b). Dans les deux cas, l'utilisateur interagit avec un ordinateur portable qui se comporte comme une fenêtre affichant en temps réel les objets augmentés.

Selon , l'*informatique portée* repose en partie sur les principes de la réalité augmentée et propose des moyens d'interaction non intrusifs pour les utilisateurs. Pour , l'informatique portée immerge l'utilisateur dans une scène augmentée. S'il agit naturellement dans son environnement, il porte des lunettes de vision qui lui proposent des informations complémentaires sur l'objet qu'il regarde (Fig. I.6c). présentent un système de guidage multimodal pour accompagner les déplacements des personnes déficientes visuelles. Le système contrôle le parcours des utilisateurs et signale les chemins d'évitement d'obstacles par des vibrations au niveau de la ceinture, et par un retour auditif. Contrairement aux précédents exemples de réalité augmentée, dans ces exemples d'informatique portée l'utilisateur interagit avec le système tout en gardant les mains libres et il n'est pas interrompu dans son déplacement.

Par rapport à ces nouvelles modalités d'interaction et de dialogue utilisateur-système, il n'existe pas à notre connaissance de grille d'évaluation consensuelle. Chaque projet propose sa grille d'évaluation et effectue ses propres tests d'utilisabilité, en général en laboratoire . Quelques travaux recherchent des standards d'évaluation de la mobilité : dans le cas des systèmes portés, proposent des mesures pour l'évaluation par questionnaire et par observation des utilisateurs lors de tâches de déplacements assistées. proposent des mesures de laboratoire pour garantir la reproductibilité des scénarios d'utilisation. résume 180 méthodes empiriques d'évaluation de systèmes mobiles proposées entre 2000 et 2002 dans les principales

conférences du domaine. L'auteur conclut que les mesures de la qualité d'interaction et de la capacité d'apprentissage des utilisateurs sont encore manquantes et que, souvent, les mesures de l'efficacité dans l'exécution de tâches particulières sont généralisées à l'utilisabilité globale du système mobile. En termes d'évaluation, rejoignent les points de vue de et préconisent d'évaluer les interactions avec une plate-forme mobile en conditions réelles d'utilisation plutôt qu'en laboratoire. Pour combiner les approches de laboratoire et l'évaluation *in-situ*, propose une solution par "cheval de Troie" où certaines fonctionnalités du système ou du dispositif d'interaction à évaluer cachent des outils d'évaluation. L'auteur illustre ce principe avec l'outil "E-Skiing" embarqué sur un assistant personnel et servant à l'auto-analyse des performances d'un skieur. La caméra portée par le skieur pour l'analyse video de ses descentes est détournée de son rôle et sert en fait de dispositif de capture des actions d'interaction avec l'assistant personnel mobile en vue de son évaluation ergonomique.

La variété des solutions présentées en termes de saisie et de présentation d'information dénote une exploration encore large des moyens d'interaction avec une plate-forme mobile. Une solution d'interface aussi établie que le triplet clavier/souris/écran des ordinateurs de bureau ne s'est pas encore dégagée. La vision ubiquitaire de , qui envisage les interfaces mobiles comme intégrées à l'environnement, guide toujours la conception de nouvelles interfaces pour les plates-formes mobiles.

Conception pour une population hétérogène

Les technologies mobiles élargissent considérablement le champ d'application des systèmes d'information. Les nouvelles méthodes d'interaction avec un système numérique s'adressent à une population d'utilisateurs aux profils très hétérogènes en âge, expérience de la mobilité, intentions, ou besoins spécifiques (). Les expérimentations de systèmes mobiles ont souligné les attentes de cette population vis-à-vis des systèmes et de leur nomadisme. Dans leur étude des utilisations du système Active Campus de l'université UC San Diego (), montrent qu'une population à priori sensibilisée à la mobilité des systèmes d'information (en l'occurrence, un panel de 35 étudiants du département d'informatique entre 19 et 26 ans) pouvait éprouver des difficultés dans l'adoption d'un système d'information mobile. Dans le cas de la conception des systèmes mobiles, les auteurs suggèrent que la caractérisation du milieu social et culturel d'utilisation du système est aussi importante que l'étude des utilisateurs, et que ces facteurs sont déterminants dans l'adoption du système.

En termes d'ingénierie et de conception des systèmes mobiles pour une population hétérogène, la population des utilisateurs est en général subdivisée en catégories (p. ex. les novices et les experts, les utilisateurs en *ballade*, en *transport* ou en *visite*, etc.). L'enjeu conceptuel essentiel est de discriminer des groupes suffisamment larges pour ne pas avoir à spécialiser inutilement le système, et suffisamment particuliers pour permettre une adaptation optimale à une catégorie d'utilisateurs. Pour entrevoir cette limite de discrimination, définissent la notion d'*accès universel*, qui rejoint l'idée de *"design pour tous"* proposée par . Selon les auteurs, l'accès aux systèmes d'information n'est plus une tâche d'utilisateur unique, mais s'effectue au sein de communautés hétérogènes qui disposent d'espaces communs d'interaction. Cette nouvelle notion d'utilisation par tous des systèmes d'information mobiles pose le problème conceptuel de l'accès universel. Celui-ci implique l'utilisabilité et l'accessibilité de l'information pour chacun, de n'importe quel endroit et à n'importe quel mo-

ment. Les auteurs relèvent des défis pour l'application des systèmes mobiles à une population hétérogène, mais n'établissent toutefois pas de cadre de référence pour le développement et l'implémentation de systèmes concrets. recommandent d'appliquer la conception pour tous dans le développement de systèmes mobiles et citent en exemple qui proposent d'inclure dans une approche de conception centrée utilisateur un échantillon représentatif de la population des utilisateurs, incluant les catégories généralement non représentées lors du développement des systèmes mobiles (pensent notamment aux personnes âgées ou déficientes visuelles).

proposent de mesurer la portabilité d'un système à différents types d'utilisateurs. Les auteurs reprennent la notion de "conception pour tous" dans le cadre de l'évaluation du système et proposent de mesurer son efficacité en le testant sur des populations hétérogènes. Plus l'efficacité dans la réalisation des tâches en mobilité est constante d'un groupe à un autre, plus le système est dit "conçu pour tous". Les auteurs illustrent une mesure de l'utilisabilité dans le cas de l'accès à un site Internet sur un assistant personnel. Trois groupes d'utilisateurs sont discriminés ("novices", "experts" et "seniors") et effectuent les mêmes tâches selon quatre scénarios d'utilisation. L'étude des taux de réussite et des temps de réalisation montre qu'il n'existe pas de différences statistiques notables entre les catégories d'utilisateurs. Dans ce cas précis le logiciel de navigation Internet, dans son utilisation sur PDA, peut être considéré comme "universellement" accessible.

Le design pour tous appliqué aux systèmes mobiles semble relativement jeune, et les recherches menées dépassent rarement l'application de principes généraux d'ergonomie et de design des systèmes interactifs au développement d'applications mobiles. En ce sens, le développement de systèmes mobiles hétérogènes reste une démarche de ré-ingénierie des méthodes de développement du génie logiciel. Les principes et enjeux sociaux dans l'adoption des systèmes mobiles mesurés par , associés à l'idée d'accès universel de , dessinent des perspectives prometteuses pour l'établissement de cadres de conception dédiés au développement de "systèmes d'information mobiles pour tous".

Conception pour des interfaces réduites

La plate-forme cliente, pour rester mobile, est contrainte à des dimensions et des capacités inférieures à celles des ordinateurs de bureau. Les moyens de présentation de l'information (;), les moyens de saisie d'information (; ;), l'autonomie ou encore les ressources (;) sont régulièrement décrits comme des facteurs limitant l'utilisation de plates-formes mobiles. Les démarches de conception pour des interfaces réduites suivent des approches pragmatiques, dirigées par les problèmes concrets d'interaction en mobilité avec des dispositifs existants, au risque parfois d'entrer en contradiction avec les principes de la conception pour une population hétérogène.

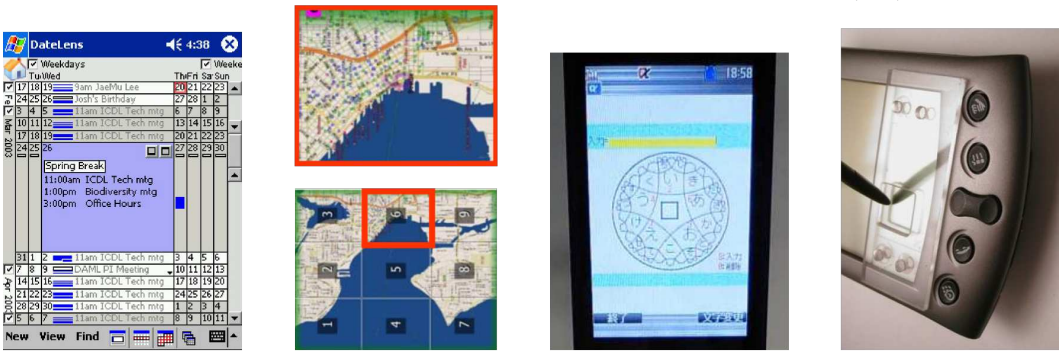
Les difficultés, essentiellement techniques, de l'interaction Humain-machine avec des interfaces réduites, sont le point de départ de nombreuses propositions de recherche. dressent un panorama de cet espace solution et proposent un classement des modalités d'entrée et de sortie pour les dispositifs mobiles (Fig. I.7). Les entrées d'informations s'effectuent soit par :

- des *saisies* sur des claviers complets, réduits, ou à bouton unique (; ;) ;
- des *actions de pointage* sur des Widgets (;) ;
- la *reconnaissance de gestes* de l'utilisateur (;) ;

– l'*oralisation* en langage naturel ou pseudo-naturel (;).

En complément, la sortie du système est effectuée soit par :

– *présentation visuelle* d'informations avec plusieurs techniques de multiplexage pour optimiser l'affichage sur des périphériques de petite taille (; ;) ;
– *retour auditif* par la synthèse vocale ou des indications sonores (;) ;
– *interaction kinesthésique* par retour d'efforts ou par vibrations (;).



(d) AppLens (d) ZoneZoom (d) VirHKey (d) EdgeWrite

Figure I.7 – Exemples d'interfaces de saisie et de présentation sur des dispositifs mobiles : AppLens (), ZoneZoom (), VirHKey () et EdgeWrite ()

L'évaluation des solutions proposées aux différents problèmes des interfaces réduites découle essentiellement d'une adaptation des techniques d'observation et de mesure ergonomiques des interfaces comme GOMS (Goals, Operators, Methods, and Selection rules) ou KLM (Keystroke Level Model) (;). montrent plus particulièrement que très peu d'outils d'évaluation sont dédiés aux interactions en mobilité. La saisie de texte fait toutefois exception et dispose de mesures statistiques pertinentes pour comparer l'efficacité des méthodes de saisie textuelle en mobilité (;).

L'approche pragmatique de composition avec le matériel et les technologies existantes permettent le développement de systèmes mobiles fonctionnels et, éventuellement, leur exploitation commerciale. Cette approche trouve toutefois ses limites lorsqu'on l'appréhende sous l'angle de la mobilité. Par exemple, étudient le comportement de touristes et leur utilisation de guides au format papier, montrant la complexité et la variété de leurs tâches en mobilité, et exprimant la difficulté à effectuer ces tâches avec un système numérique mobile à interface réduite. Comme cela est pressenti par et , pour , la mobilité numérique est spéciale et devrait être adressée en dépassant le cadre des outils et techniques d'interaction existants.

Conception pour l'ubiquité

La notion de système informatique contextuel a été associée dès son origine à la mobilité des utilisateurs. Le projet ActiveBadge, premier prototype de système contextuel, a été porté par une problématique de mobilité des utilisateurs : en suivant les employés au sein des bâtiments du centre de recherche Olliveti, ActiveBadge pouvait rediriger les appels vers le téléphone le plus proche de la personne appelée ()(Fig. I.8a). Le projet ParcTab propose des clients mobiles d'intérieur qui peuvent communiquer par liaison infrarouge entre eux et avec les autres équipements électroniques à proximité () (Fig. I.8b). Comme ActiveBadge, ce prototype connaît la localisation des utilisateurs et utilise leur mobilité d'une pièce à une autre

pour proposer des adaptations du système. Par exemple, les ParcTab supportent le travail collaboratif pour des utilisateurs d'une même pièce (limite technique due à la transmission par infrarouge sur un réseau de type WPAN), ou alors servent de télécommandes adaptées aux matériels disponibles dans la pièce (téléviseur, radio, magnétoscope). Avec Parctab, le client mobile s'adjoint la contextualisation de l'exécution du système, qui est différente selon le lieu d'utilisation. Cette capacité des systèmes d'information à s'acquitter de la mobilité des utilisateurs tout en percevant et en interprétant leur environnement à l'exécution est appelée *informatique ubiquitaire* (). Elle repose sur les notions de *contexte* et de *sensibilité au contexte*. Selon , un système sensible au contexte peut répondre aux questions : "Où est l'utilisateur ?, "Qui accompagne l'utilisateur ?" et "Quelles ressources sont disponibles à proximité de l'utilisateur ?". Lorsque le système, à deux moments distincts de l'exécution, est amené par la mobilité de l'utilisateur à apporter des réponses différentes à ces questions, il adapte son comportement aux nouvelles informations dont il dispose pour fournir à l'utilisateur une *bonne information au bon endroit et au bon moment* ().

(a) 1992 : ActiveBadge () (b) 1995 : ParcTab () (c) 1996 : Cyberguide ()

Figure I.8 – Premiers prototypes de systèmes ubiquitaires

Ces bases de l'informatique contextuelle, établies au début des années quatre-vingt dix, ont depuis été largement développées et ont fait l'objet de nombreux projets de recherche. et proposent deux états de l'art des expérimentations en informatique mobile et contextuelle. Parmi les travaux cités, "Cyberguide" utilise les variations de localisation de l'utilisateur, en intérieur comme en extérieur pour fournir aux utilisateurs d'un guide touristique numérique des informations relatives à leur position (Fig. I.8c)(). Le système enregistre aussi les parcours passés des utilisateurs et leur suggère des visites en fonction des sites auxquels ils se sont précédemment attachés. Pour Cyberguide, le contexte se compose des informations de localisations courante et passées. Certains travaux appréhendent l'activité de l'utilisateur comme une dimension supplémentaire du contexte, en complément de la localisation. "Conference Assistant" est un prototype d'ordinateur porté qui utilise un profil des préférences des utilisateurs (dans ce cas des chercheurs en congrès) pour leur proposer d'assister à des séminaires selon leur localisation et leur intérêt scientifique (). "Office Assistant" est un agent numérique qui gère l'agenda d'un employé de bureau en fonction de la personne se présentant à la porte, de l'activité du propriétaire du bureau, de l'opportunité du rendez-vous, etc. (). Ce dernier projet dépasse le niveau de la localisation pour proposer des interprétations de la mobilité, notamment de celle du propriétaire du bureau (déductions contextuelles de la forme : "S'il n'est pas au bureau, alors son activité ne peut pas être ...", etc.).

L'évaluation de l'ubiquité des systèmes mobiles est critique pour la validation des systèmes ubiquitaires. Par l'observation des utilisateurs dans leurs usages, elle doit faire

ressortir les dimensions contextuelles qu'il peut être utile d'intégrer au système d'information mobile (). Toutefois, les projets d'informatique ubiquitaire se concentrent souvent sur la définition et la mesure du contexte, avant d'étudier l'utilité et l'utilisabilité des adaptations contextuelles provoquées par le système (). pose les limites des méthodes d'évaluation de systèmes interactifs par rapport aux systèmes ubiquitaires. Plus encore que dans le cas de la conception pour un public hétérogène, les tests de laboratoire n'apportent plus la richesse expérimentale nécessaire à l'évaluation. Comme l'attention de l'utilisateur par rapport à sa tâche n'est pas garantie, l'analyse de son efficacité ne constitue plus une mesure pertinente d'utilisabilité. Dans sa thèse, imagine des mesures d'utilisabilité dédiées aux systèmes ubiquitaires qu'il applique à l'évaluation de systèmes mobiles existants. L'auteur caractérise les risques potentiels de la sensibilité au contexte pour l'utilisabilité et propose un guide de conception pour faciliter l'implémentation de systèmes mobiles contextuels garantissant un bon niveau d'utilisabilité et d'ergonomie.

précisent dans leur état de l'art que la large majorité des projets d'informatique mobile contextuelle considèrent uniquement la position de l'utilisateur comme caractéristique dynamique. Bien qu'en général les projets présentent des définitions du contexte beaucoup plus larges que le simple suivi de la localisation des utilisateurs, ces autres dimensions du contexte restent souvent inexploitées voire ignorées. Aujourd'hui, cette tendance ne s'est pas démentie, et le domaine d'application des services localisés (*LBS : Location-Based Services*) fait de la localité contextuelle son thème d'étude. Il définit des "services accessibles depuis des clients mobiles via un réseau non filaire, capables de localiser ses clients et d'utiliser cette localisation pour adapter le contenu présenté au niveau du client" (). La localisation est le moteur de l'adaptation de la plate-forme cliente, au détriment des autres dimensions contextuelles. Selon , cette prépondérance de la localisation s'explique par la facilité technique de sa mesure, l'immédiateté de ses applications à la mobilité, mais aussi, à l'opposé, par l'absence de moyens de capture d'autres dimensions du contexte. En prenant la mesure des évolutions technologiques et notamment de l'intégration de capteurs à la plate-forme mobile, et imaginent une mobilité ubiquitaire accordant plus d'importance à l'environnement d'interaction et aux facteurs humains pour la compréhension du contexte.

Grille de lecture croisée : niveau de la plate-forme cliente mobile

Pour positionner ces travaux les uns par rapport aux autres, les différentes dimensions pour la conception de systèmes mobiles de détaillées précédemment sont croisées avec les approches de l'ingénierie du logiciel proposées par [8] :

- les travaux de **compréhension** apportent un éclairage sur les enjeux et les problématiques de la plate-forme mobile ;
- les travaux d'**ingénierie** proposent des solutions originales pour l'utilisation et l'interaction avec une plate-forme mobile ;
- les travaux de **ré-ingénierie** adaptent des solutions existantes à une utilisation sur une plate-forme mobile ;

8. Les travaux présentés sont considérés comme représentatifs de différentes solutions pour la conception de plate-forme mobiles. Le lecteur intéressé se référera à pour une classification exhaustive selon les critères de des systèmes interactifs mobiles proposés entre 2000 et 2002.

– les travaux d'**évaluation** quantifient les apports d'une nouvelle solution d'ingénierie par rapport aux solutions existantes ;
– les travaux de **description** proposent différentes orientations et caractéristiques désirables pour les futures plates-formes mobiles.

Tableau I.4 – Synthèse des solutions proposées pour la conception de systèmes d'information à clients mobiles

		Orientation de la méthodologie de conception			
		Mobilité	**Population hétérogène**	**Interfaces réduites**	**Contexte dynamique**
Nature des travaux de recherche	**Compréhension**	; ; ; ;	; ;	; ; ; ; ;	; ; ; ;
	Ingénierie	; ; ; ; ;	-	; ; ;	; ; ;
	Ré-ingénierie	-	;	; ; ; ; ; ; ; ; ; ;	-
	Évaluation	; ;		;	; ;
	Description	; ;	;	;	; ; ;

Le tableau I.4 reprend l'ensemble des travaux et expérimentations introduits dans ce chapitre et les ordonne selon leur approche de l'ingénierie du logiciel et pour chaque orientation conceptuelle : Mobilité, Population hétérogène, Interface réduites et Contexte dynamique. Dans la plupart des cas, les travaux recensés se concentrent sur les aspects technologiques de la mobilité, soit en les adressant par l'approche des interfaces réduites, soit en considérant l'utilisateur dans sa mobilité.

Dans le cas des interfaces réduites, une large majorité des travaux propose des solutions de ré-ingénierie des outils et méthodes existantes à l'origine pour l'informatique non mobile. En particulier, ces travaux proposent d'adapter la métaphore du bureau à une utilisation mobile, même si, du fait des interfaces réduites, les interactions WIMP perdent de leur efficacité et de leur intérêt. Par exemple, malgré de nombreux prototypes, la saisie de texte sur une version de clavier numérique et mobile est toujours beaucoup moins efficace que la saisie sur clavier physique d'un ordinateur de bureau. Une autre conséquence de cette reproduction de la métaphore du bureau est la production d'une interaction mobilisatrice pour l'utilisateur. En particulier, aucun des prototypes proposés ne semble approprié pour une utilisation en mode de mobilité "Balade". Les utilisateurs interrompent leurs tâches courantes et leurs déplacements pour interagir de façon tactile et visuelle avec les interfaces proposées. Malgré ces restrictions d'utilisation en mobilité, la conception d'outils et méthodes pour les interfaces réduites reste un vecteur d'innovation majeur dans le développement de systèmes mobiles : d'une part, les plates-formes sont disponibles sur le marché à des prix de plus en plus abordables, et d'autre part, les outils et solutions mis en place ne s'arrêtent pas à des cas d'application particuliers, mais proposent des solutions universelles de ré-ingénierie pour les systèmes d'information existants.

La conception pour les usages en mobilité prend le point de vue opposé et propose des solutions non intrusives pour une approche de la mobilité de type "Balade", quitte à restreindre leur domaine d'utilisation au détriment des usages en "Transport" et surtout en "Visite". Les travaux d'ingénierie présentés dans ce domaine, tous relativement récents, se concentrent sur une tâche ou un usage particulier d'un système d'information mobile. Souvent, les domaines d'application sont liés à des activités touristiques ou de repérage et d'orientation en mobilité. Les solutions proposées ne relèvent pas de méthodes de conception

génériques, mais adressent des cas d'application bien particuliers. La jeunesse du domaine explique aussi que l'on ne trouve que très peu de travaux de ré-ingénierie et que les solutions étudiées n'ont pas encore solidement établi leur efficacité. Les prototypes présentés demandent aussi des moyens d'infrastructure plus importants au niveau de la plate-forme d'interaction. Par exemple, chaque système d'information cité s'accompagne d'un prototype matériel dédié, ce qui limite le développement à large échelle de cette approche des systèmes mobiles. Ces solutions, propres à un cadre d'application, se dédient aussi à la mobilité de type "Balade". Les prototypes présentés perdent en partie leur efficacité et leur intérêt lors d'un usage en "Transport" et en "Visite", pour lesquels les plates-formes mobiles à interfaces réduites semblent d'un intérêt plus immédiat.

La conception pour la prise en compte d'un contexte dynamique en mobilité est transverse aux deux approches d'ingénierie, qui admettent chacune les notions de contexte dynamique et de sensibilité au contexte comme fondamentales dans l'interaction avec les systèmes mobiles. Les bornes de ce domaine, établies au début des années 90, ne sont aujourd'hui que partiellement étudiées. La sensibilité au contexte pour les systèmes mobiles se résume souvent au positionnement de l'utilisateur et au filtrage/recommandation de contenu en fonction de sa localisation, et éventuellement de ses déplacements passés. Cette restriction des systèmes contextuels à la notion de services localisés a surtout été la conséquence de l'apparition de technologies de localisation à grande échelle et facilement transportables comme le GPS. La vision d'une informatique mobile "ubiquitaire" reste une perspective prometteuse, mais qui manquait toujours de base technologique solide pour réellement se développer. Aujourd'hui, cette situation change rapidement. Les dernières plates-formes mobiles, téléphones ou assistants personnels, ressentent l'entourage des utilisateurs et intègrent la localisation, l'orientation, le son, l'image, la giration, l'accélération, la connectivité réseau, etc. Ces plates-formes offrent les moyens technologiques de mesure d'un contexte riche nécessaires au développement des systèmes ubiquitaires.

La problématique de l'hétérogénéité des utilisateurs en mobilité est aussi traitée en général dans des approches de ré-ingénierie des méthodes existantes pour les systèmes non mobiles. Plus encore que dans les autres approches de conception, la prise en compte récente de cette problématique reste marquée. À notre connaissance, il n'existe pas de méthode d'ingénierie dédiée à la modélisation de systèmes mobiles pour tous qui ne soit pas une version "sans fil" de méthodes génériques. Cela n'implique pas forcément que les solutions de ré-ingénierie comme le "design pour tous" ne soient pas applicables ni efficaces dans le cas de systèmes mobiles. Toutefois, les mesures d'évaluation manquent et les nouveaux enjeux de la mobilité des systèmes laissent à penser que la conception pour tous bénéficierait favorablement d'un éclairage par la mobilité.

La compréhension des enjeux pour la conception de clients mobiles semble bien établie depuis le début des années 90. À l'exception notable de la conception pour une population hétérogène, les travaux cités font référence et sont abondamment repris dans la communauté de l'ingénierie des systèmes mobiles. L'évaluation des prototypes proposés présente encore de nombreux défis, dont notamment la mise en place de grilles de critères et de méthodes d'analyse de terrain qui garantissent, comme en laboratoire, la reproductibilité des résultats d'une évaluation. Mis à part qui appliquent des mesures d'évaluation génériques, et qui propose des critères adaptés à la mesure de l'utilité de la perception contextuelle, les autres travaux présentés recensent les objectifs et les besoins pour l'évaluation des systèmes mo-

biles plus qu'ils n'envisagent de solutions. Signes d'un domaine de recherche encore jeune, les analyses des enjeux de la mobilité des systèmes sont remises en perspective. En particulier, les écrits de Weiser sur les systèmes ubiquitaires établissent de futures convergences entre la mobilité, le contexte dynamique et les interfaces réduites. Il apparaît notamment que pour répondre aux enjeux de la mobilité des systèmes du point de vue de la plate-forme cliente, un effort de compréhension des implications sociales de l'utilisation en mobilité permettrait de mieux borner ce domaine d'utilisation. D'autre part, l'approche contextuelle pour la mesure des variations d'un environnement dynamique mériterait d'être enrichie de dimensions supplémentaires, complétant la localisation des individus et l'historique de leur mobilité.

I.4 Conclusion

Dans ce chapitre, les principales définitions et les enjeux fondamentaux de la mobilité dans les systèmes d'information ont été présentés. Après une mise en perspective historique par rapport aux systèmes d'information distribués et au modèle client-serveur, la notion de mobilité est apparue, portée par le nomadisme quotidien des utilisateurs et les avancées technologiques en termes d'infrastructure réseau et de portabilité des plates-formes clientes. L'objectif des systèmes mobiles est de fournir des services et des informations à différents endroits, moments et conditions d'utilisation. Pour cela, les solutions particulières des dimensions du transport d'information et de la qualité d'usage sur la plate-forme cliente ont été détaillées. Elles adressent la mobilité des utilisateurs selon les trois typologies de déplacement : la visite, la balade et le transport.

Au niveau de l'infrastructure réseau, des solutions qui adressent les principaux défis de la distribution des systèmes (fiabilité, délai, bande passante, non-homogénéité, topologie dynamique, sécurité, et coûts) ont été présentées. Certaines dimensions comme l'économie des coûts ou de la bande passante bénéficient aujourd'hui des avancées des NTIC et ne sont plus d'un intérêt direct en matière de recherche. En revanche, si la mobilité est prise du point de vue global du système, les problématiques de la non-homogénéité des composants et de la topologie dynamique de l'infrastructure de transport conservent un fort potentiel scientifique. La mobilité et l'hétérogénéité généralisées à l'ensemble du système et de ses composants dénotent un besoin de cadres de référence adaptés à la conception de systèmes mobiles pour répondre à ces nouveaux enjeux.

Des premiers prototypes ubiquitaires des années 90, aux récents développements des méthodes d'interaction avec les plates-formes à interfaces réduites, un panorama des solutions de recherche pour la mobilité du client a été présenté. Ces clients adressent la thématique de l'utilisation nomade d'outils et systèmes d'information selon les orientations de la mobilité, d'un public hétérogène, des interfaces réduites et du contexte dynamique. Les développements proposés dans chacune des orientations sont en général très liés aux avancées technologiques, et peuvent parfois sembler se contredire. En particulier, les orientations de la conception pour la mobilité et pour les interfaces réduites prennent respectivement le parti d'un nomadisme de type "Balade" ou de types "Visite" et "Transport". Proposer des solutions mobiles qui s'adaptent avec continuité du service et de façon transparente pour l'utilisateur aux trois typologies de la mobilité reste l'un des enjeux pour la recherche et le développement de nouveaux clients mobiles. Du point de vue des utilisateurs également, la variété est de mise et les

méthodes de développement appropriées aux systèmes mobiles manquent. Enfin, la dimension transverse de l'évaluation des clients mobiles devra développer des grilles d'évaluation et des critères qualitatifs et quantitatifs pour estimer la fonctionnalité mobile d'un système d'information et comparer différentes solutions entre elles.

L'étude des systèmes mobiles se poursuit par l'exploration de la dimension des contextes d'utilisation et d'exécution et par l'analyse des conséquences de leur prise en compte dans le développement de systèmes d'information mobiles. Plus particulièrement, l'informatique contextuelle est pressentie comme une solution conceptuelle possible pour prendre en charge 1) la mobilité généralisée au système, 2) la variété des utilisateurs et des composants, et 3) la continuité du service entre différents types de nomadisme au sein de démarches de modélisation unifiées.

CHAPITRE II

Systèmes d'information sensibles au contexte

Sommaire

II.1 Nature contextuelle des systèmes mobiles 34
II.2 Notion de contexte appliqué aux systèmes d'information 36
II.2.1 Définitions de la notion de contexte 37
II.2.2 Dimensions du contexte 38
II.3 Mesure des dimensions du contexte 43
II.3.1 Mesure du contexte d'usage 43
II.3.2 Mesure du contexte d'exécution 46
II.3.3 Intérêt d'une mesure pour l'adaptation du système 47
II.4 Conception et implémentation de systèmes contextuels 49
II.4.1 Approche conceptuelle : familles de modèles contextuels 49
II.4.2 Approche pour l'implémentation : architectures d'exécution 52
II.5 Utilisation des informations contextuelles 58
II.5.1 Typologies des formes d'exploitation du contexte 59
II.5.2 Initiative et responsabilité de l'adaptation contextuelle 62
II.5.3 Déclenchement du processus d'adaptation 64
II.5.4 Objets de l'adaptation à l'exécution 65
II.6 Conclusion 68

Après l'installation durable des ordinateurs et des systèmes d'information à domicile, l'informatique devient de plus en plus présente au quotidien. L'explosion de la téléphonie mobile, et l'arrivée sur le marché des ordinateurs portables, d'assistants numériques et de toutes sortes de plates-formes clientes portables constituent des indicateurs forts des utilisations futures des systèmes informatiques impliquant plus de mobilité et de variété de situations d'usage. Après la présentation au chapitre précédent des enjeux de la mobilité informatique pour l'accompagnement d'utilisateurs nomades et le développement de systèmes, ce chapitre se concentre sur la variété des situations d'utilisation impliquée par la mobilité. Ces situations sont caractérisées dans les systèmes d'information par des approches contextuelles, dans lesquelles le système se donne les moyens de percevoir son environnement pour adapter sa réaction.

De tels systèmes contextuels dépassent le cadre strictement déterministe des outils logiciels développés jusqu'alors : la réponse produite par le système n'est plus seulement une fonction des entrées des utilisateurs ou d'autres systèmes. Dans un système contextuel, ce modèle de la "boîte noire" logicielle est enrichi et considère des entrées auxiliaires mesurant différentes dimensions du contexte ou de l'environnement pour influencer la sortie produite. Pour des entrées identiques, cette sortie n'est plus universelle. Le système passe d'un déterminisme global à un déterminisme pour un contexte donné.

L'abstraction du contexte d'utilisation a constitué l'une des forces du développement logiciel. En particulier, un système aux réactions identiques quelles que soient les situations favorise son appropriation par les utilisateurs. L'informatique de bureau des années 80 a mis en œuvre des logiciels aux comportements universels (traitements de texte, jeux, etc.) pour lesquels les situations d'utilisation sont globalement les mêmes et qui permettent à un utilisateur de changer de plate-forme et d'environnement sans pour autant revoir ses habitudes de manipulation des logiciels. Le développement d'usages informatiques nomades apparaît en contradiction avec cette universalité des situations et illustre certaines limites du principe de la "boîte noire" logicielle. Dans une première introduction au contexte, ce chapitre motive l'apparition de l'informatique contextuelle en parallèle de l'informatique mobile pour doter les systèmes de réactions plus appropriées qu'universelles. Dans la suite, les notions clés de "contexte" et de "dimensions du contexte" sont détaillées. Des modèles conceptuels pour leur implémentation au sein de systèmes adaptatifs sont proposés et analysés. Enfin, l'adaptation contextuelle et la personnalisation dénotent la capacité d'un système à exploiter les mesures des dimensions du contexte. Différentes formes de mesures contextuelles et d'adaptations sont proposées et illustrées par le développement de systèmes mobiles et/ou ubiquitaires.

II.1 Nature contextuelle des systèmes mobiles

Les systèmes mobiles ont bénéficié des progrès techniques des infrastructures de transmission d'informations et des avancées dans la portabilité des plates-formes clientes. En pleine émergence depuis une quinzaine d'années, ces systèmes nous accompagnent désormais au quotidien, sur nos lieux de travail, dans nos déplacements, à domicile. La recherche en informatique et en systèmes d'information a proposé et expérimenté des solutions matérielles et logicielles qui ont apporté des preuves confirmant à la fois la faisabilité et l'intérêt des utilisateurs pour les applications nomades.

Les solutions techniques disponibles aujourd'hui permettent d'appréhender la mobilité à grande échelle, auprès d'utilisateurs nombreux et variés, et dans des situations d'utilisation qui n'ont plus rien à voir avec l'informatique fixe. Celle-ci a représenté un standard pour l'accès aux systèmes d'information qui a largement impacté nos habitudes et nos attentes vis-à-vis des systèmes mobiles. Les ordinateurs de bureau représentent un *monde connu*. Les accès au réseau sont permanents, rapides et sécurisés. Les moyens d'interaction, l'écran, le clavier, la souris ou les interfaces WIMP se retrouvent universellement. Sur le modèle de l'architecture de Von Neuman, les ordinateurs eux-mêmes disposent de ressources que l'on peut aujourd'hui considérer comme largement satisfaisantes (p. ex. énergie, capacité de stockage et de calcul, etc.) (). Ce modèle d'ordinateur standard a offert un cadre de référence pour le développement de systèmes distribués, de logiciels client-serveur ou de services Web.

Il est toujours implicitement considéré comme le modèle type des plates-formes clientes de nombreuses méthodes de développement. Avec la mobilité, ce cadre de référence disparaît et l'informatique nomade relève pour l'essentiel d'un *monde inconnu*. Les conditions d'utilisation, les interfaces et moyens de présentation, les déplacements des utilisateurs, le cadre de travail, etc., toutes ces dimensions figées et connues dans le monde fixe deviennent variables et inconnues avec la mobilité. Le modèle de l'ordinateur de bureau tend à se déporter hors du cadre de travail usuel. Les évolutions technologiques accompagnant ces nouveaux usages brouillent les frontières entre les domaines de la téléphonie, de l'informatique de bureau, et des systèmes et services mobile ou nomade.

Répondant à un engouement nouveau pour les dispositifs mobiles, cette utilisation nomade est un thème aujourd'hui repris par l'industrie informatique, qui l'a largement placé comme moteur de ses développements scientifiques et techniques des systèmes et des plates-formes. Des solutions pour la mobilité proposent l'accès continu à des systèmes d'information variant entre différents espaces d'utilisation (*mobiles* ou *locaux*), entre différentes échelles (*personnelle*, *locale* ou *globale*), et quelle que soit la typologie de déplacement adoptée (*balade*, *transport* ou *visite*)(; ;). Malgré ces progrès indéniables dans le matériel client et les infrastructures de transport d'information, les solutions proposées sur le marché pensent trop souvent la mobilité et le développement de systèmes mobiles comme une question de ré-ingénierie des systèmes existants et de leurs moyens de présentation et de saisie d'information. Le modèle proposé par le *monde connu* de l'informatique de bureau reste sous-jacent aux démarches d'ingénierie pour les plates-formes mobiles. Par exemple, les récents développements de logiciels pour la consultation de sites Web sur des dispositifs mobiles proposent différentes techniques pour adapter l'affichage et la navigation dans une page Web à l'utilisation d'un stylet et d'un écran de dimension réduite (; ;).

Ces outils cherchent à offrir à l'utilisateur les moyens de retrouver en mobilité les mêmes résultats qu'à son bureau. Autrement dit, la page Web consultée doit être autant accessible et si possible avec des moyens d'interaction relativement similaires (p. ex. couple écran/pointeur). Souvent, l'analyse du sens et des causes de la consultation de systèmes numériques en mobilité, difficile à appréhender, est occultée au profit d'un questionnement sur la faisabilité technique du transfert des usages du *monde connu* de l'informatique fixe vers le *monde inconnu* de l'informatique mobile. L'analyse de l'utilisabilité des plates-formes nomades et des services associés, notamment de consultation Web, montrent qu'ils restent moins pratiques et moins faciles d'accès que leurs cousins de bureau (;). Si les utilisateurs mobiles acceptent ces situations dégradées pour la consultation de pages Web comme de tout autre service mobile, c'est qu'il existe des causes dans leur environnement qui les poussent à une utilisation nomade. Par exemple l'utilisation d'un système d'information mobile est souvent motivée par des attentes précises et immédiates en relation avec le lieu d'activité de l'utilisateur : recherche d'un restaurant, informations sur un endroit de visite, etc. (;).

Comprendre les comportements nomades d'utilisateurs et les besoins d'une utilisation mobile de systèmes d'information représente à la fois le point d'entrée et le principal défi du développement de l'informatique ubiquitaire. Du point de vue de , l'analyse des intentions des utilisateurs mobiles par la perception de leur environnement doit permettre au système mobile de suggérer des réponses qui soient en adéquation avec leurs besoins. Appliquée à la navigation Web, une approche ubiquitaire d'un système mobile peut s'appuyer sur des mesures des préférences de l'utilisateur, du moment et de la localisation d'utilisation, des

capacités de la plate-forme, pour constituer un modèle contextuel et l'appliquer à l'adaptation des informations présentées et des moyens d'interaction avec ces informations (; ;). Si le modèle de référence reste le *monde connu*, celui-ci est contextualisé et bénéficie d'un éclairage par la situation propre à faciliter son adaptation au monde et aux usages mobiles.

Même si cet aspect est masqué par les défis techniques et scientifiques de l'interaction en mobilité, le questionnement sur le sens des usages et les buts des utilisateurs est aussi central pour le développement de l'informatique mobile qu'il ne l'est dans le cas de l'informatique ubiquitaire. Les travaux fondateurs de l'informatique contextuelle et mobile considèrent le nomadisme et la mobilité des systèmes d'information comme une étape dans l'émergence de l'informatique ubiquitaire. résume les thématiques centrales de la recherche en informatique ubiquitaire, et place l'émergence de celle-ci comme une conséquence de l'évolution des systèmes distribués vers plus de mobilité, et des systèmes mobiles vers plus de contextualité (Fig. II.1).

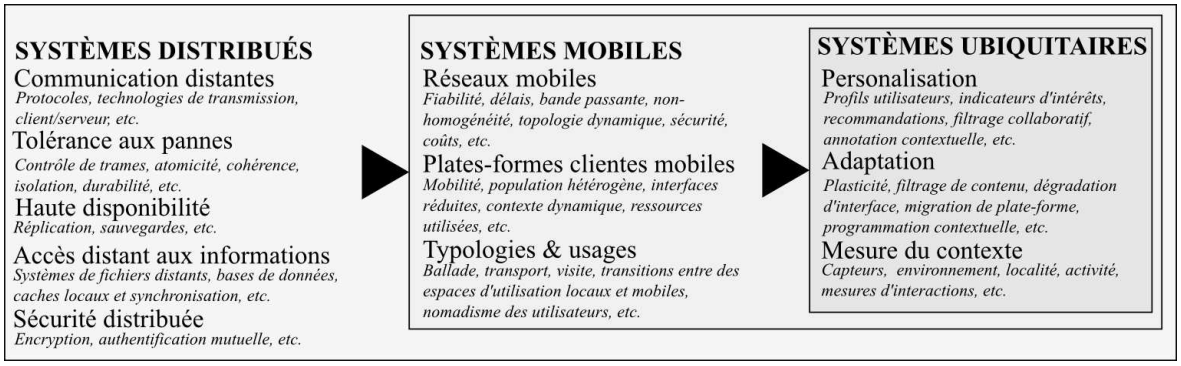


Figure II.1 – Principaux domaines de recherche qui contribuent à l'émergence de l'informatique ubiquitaire (d'après)

II.2 Notion de contexte appliqué aux systèmes d'information

L'établissement de la notion d'ubiquité dans les systèmes mobiles a été posée comme consécutive à la prise en compte de l'environnement d'exécution et d'utilisation du système . Les systèmes nomades et ubiquitaires effectuent la liaison entre l'informatique mobile, l'informatique contextuelle et leurs communautés de recherche respectives. Cette section présente une perspective historique sur l'apparition de l'informatique contextuelle et précise quelques tentatives de définitions empiriques de la notion de contexte de la fin des années 90. En l'absence de consensus clair sur la définition du contexte, son utilisation en informatique contextuelle passe par la définition pragmatique de classes de mesures : les dimensions du contexte. Détaillées et classées, elles laissent transparaître une division du contexte en deux sous-ensembles d'usage et d'exécution.

II.2.1 Définitions de la notion de contexte

Le contexte aide à comprendre les causes des comportements des utilisateurs mobiles et leurs attentes vis-à-vis du système mobile. Dans un système contextuel, l'initiative d'adaptation se partage entre le système et l'utilisateur, l'objectif étant pour le système d'utiliser le contexte pour réduire les actions d'interaction avec l'utilisateur nécessaires pour lui apporter une réponse appropriée. Dans cette perspective, la notion de contexte apparaît comme centrale dans une démarche d'adaptation logicielle ; elle fait par ailleurs l'objet de nombreuses définitions et interprétations.

L'une des premières tentatives de formulation de la notion de contexte est donnée par . Selon les auteurs, **les systèmes distribués contextuels s'adaptent au lieu d'utilisation, à l'ensemble des utilisateurs présents, aux composants matériels à proximité et aux changements de ces dimensions**. Définir le contexte à un moment donné revient à répondre aux questions "Qui est l'utilisateur ?" , "Avec qui se trouve-t-il ?" et "Quelles sont les ressources à proximité ?". Parmi les mesures de l'environnement qui permettent d'inférer les réponses à ces questions, les auteurs répertorient la luminosité, le niveau de bruit, la connectivité réseau, les coûts de communication, la bande passante, la situation sociale. Selon , **le contexte regroupe la localisation, les identités des utilisateurs et personnes alentours, le temps, la saison, la température**, etc. énumère l'état émotionnel de l'utilisateur, son point d'attention, sa position et son orientation, les objets et les personnes de son entourage, et le moment d'utilisation du système (date/heure) comme dimensions caractéristiques du contexte. Ces premières définitions sont insufflées par le niveau expérimental (ActiveBadge, Parctab, StickeMap, etc.) et ne proposent pas de réelle abstraction par rapport à ces prototypes, ni de classification claire de la notion de contexte et des dimensions qu'il implique. Quelques définitions plus larges ont été énoncées. Pour , qui appliquent la sensibilité contextuelle dans le cas d'un logiciel de prise de note pour l'archéologie, **le contexte se définit de façon plus générale comme l'environnement, l'identité, la position, et le temps relatifs à l'utilisateur**. voit le contexte comme le sous-ensemble mesurable d'un environnement porté à l'intérêt des utilisateurs.

Dans la première étude extensive de la notion de contexte, veulent donner un sens opérationnel au contexte pour le développement d'applications ubiquitaires et mobiles. Ils abstraient les travaux précédents de leurs liens aux expérimentations, et, à la suite de , définissent le contexte comme "**toute information qui peut être utilisée pour caractériser une entité. Une entité est une personne, un objet ou un endroit considérés comme pertinents pour l'interaction entre un utilisateur et une application, y compris l'application et l'utilisateur**". Les auteurs dérivent ensuite selon différents degrés de granularité cette définition et montrent comment passer de la notion de contexte au développement d'applications contextuelles qui savent s'adapter à l'exécution. Dans une version augmentée de leur définition, qui deviendra la référence de la notion de contexte, précisent les dimensions habituelles du contexte : **la localisation, l'identité et le statut des personnes, groupes de personnes, composants informatiques ou autres objets**.

Malgré son statut de "référence", cette définition du contexte est toujours largement débattue (). Certains auteurs la trouvent trop vague et générale pour être adaptée dans un processus de conception (;). Pour d'autres, le contexte est essentiellement subjectif et est défini par l'interprétation d'une situation faite par l'utilisateur lui-même. En éliminant cette

subjectivité du regard de l'utilisateur, et en favorisant à la place l'interprétation d'une situation par les concepteurs, les tentatives de caractérisation contextuelles de l'environnement iraient dès le départ à l'encontre même de la notion de contexte (;)[1].

Malgré ces contradictions apparentes dans l'appréhension de la nature du contexte, l'informatique contextuelle a su adopter des démarches pragmatiques pour proposer différents modèles de classification des dimensions du contexte et des formalismes pour leur intégration dans des méthodologies de conception de systèmes d'information. La description des dimensions du contexte, leur intégration au sein d'un modèle de contexte lors de la conception, et la prise en compte de ce modèle à l'implémentation, constituent les trois grandes étapes du développement de systèmes sensibles au contexte. Le point de départ communément adopté passe par une description de l'objet de l'adaptation. Répondre aux questions, "Qui s'adapte ou qui adapte ?", "Pourquoi adapter ?", "Comment et quand adapter ?", "Que faut-il adapter ?", "À quoi adapter ?", pousse vers la définition de taxonomies opérationnelles de l'utilisation du contexte pour l'adaptation sans pour autant nécessiter de définition cadre ().

II.2.2 Dimensions du contexte

La définition de la notion de contexte, sans frontières bien établies, encourage des interprétations variées des dimensions de l'environnement mesurées par le contexte et de la façon de les organiser au sein d'un modèle conceptuel.

et considèrent que les dimensions jouent le rôle de couleurs primaires du contexte. Les auteurs proposent d'utiliser les mesures effectuées dans ces différentes dimensions pour construire un niveau supérieur d'objets contextuels, appelé "contexte secondaire". Par exemple, relever l'occupation d'une pièce, le nombre de personnes dans la pièce, leurs positions relatives, etc., permet de déterminer si une réunion a lieu, s'il s'agit plutôt d'une conférence, etc. Cette idée de répartition entre plusieurs niveaux de contexte est reprise par ; et leurs notions de capteurs physiques, virtuels ou logiques. Les *capteurs physiques* et *capteurs logiques* sont assimilables respectivement aux notions de contextes primaire et secondaire. Les *capteurs virtuels* composent des objets contextuels à partir d'éléments logiciels. Par exemple, pour déterminer la localisation d'un utilisateur, il est possible de mesurer sa position (capture physique pour ou contexte primaire pour), ou alors, d'étudier son agenda électronique ou son programme de déplacement pour estimer sa position courante (capture virtuelle pour).

Dans une définition contextuelle et ubiquitaire, le sens du mot "capteur" se détache de la notion de dispositif physique ou d'instrument de mesure associée à sa définition dans les domaines de l'électrotechnique, l'électronique et de la mécanique. Dans la suite du chapitre, le terme "capteur" désignera indifféremment un capteur physique, logiciel ou virtuel. Son rôle est de produire une mesure du contexte soit 1) de nature physique et réalisée par un capteur matériel (p. ex. Mesure de température réalisée par un thermomètre), soit 2) de nature logique et inférée à partir de mesures physiques ou logiques (p. ex. Mesure d'une destination par inférence à partir des mesures physiques de position et de vitesse), ou enfin

1. L'utilité même de donner une définition au contexte a été longuement débattue. Le lecteur intéressé se référera à pour une reprise détaillée des différents arguments.

3) de nature virtuelle (p. ex. mesure de l'âge de l'utilisateur réalisée par la consultation d'un profil logiciel)

La description des dimensions des contextes primaires et secondaires, mesurées ou inférées dans un système sensible au contexte, a fait l'objet de nombreuses propositions, qui parfois se recoupent. Cette section revient sur quelques définitions des dimensions du contexte et établit entre elles des relations d'équivalences et de dépendance.

Dimensions *interne* et *externe*

D'après , la distinction entre les dimensions externes et internes figurent parmi les décompositions souvent retrouvées dans la littérature (; ;) :

- la *dimension externe* fait référence aux objets contextuels pouvant être mesurés par des dispositifs matériels, ou alors pouvant être inférés à partir de plusieurs mesures de l'environnement. distingue ces deux approches en *contexte primaire*, avec des mesures de niveau physique comme le temps, la localisation, la luminosité, etc., et en *contexte secondaire*, avec des notions inférées (p. ex. la saison peut être inférée à partir des trois mesures du contexte primaire précédemment citées) ;
- la *dimension interne* est perçue explicitement ou implicitement auprès de l'utilisateur. Elle relève des buts, des tâches, du contexte de travail, de l'état social, de l'état émotionnel, etc. des utilisateurs. Beaucoup moins exploitée que la dimension externe, les objets mesurés dans cette dimension sont la plupart du temps inférés, et appartiennent dans ce cas au contexte secondaire [2]. Par exemple, l'intérêt immédiat de l'utilisateur peut être inféré à partir des mesures de l'historique de son activité (documents ouverts, pages Web consultées, etc.) ().

Dimensions de *localité*, *statut* et *temporalité*

rejoignent le point de vue de et considèrent que les objets du contexte secondaire se construisent selon des informations primaires de *localité*, de *temporalité* et de *statut*. La localité peut être absolue, avec des mesures de position comme dans les projets ActiveBadge ou ParcTab () ou alors relative aux entités et composants environnants, avec des indications de proximité. Les auteurs citent l'exemple du sol intelligent, captant la pression des pas des utilisateurs pour les situer dans une pièce (). La temporalité caractérise des objets contextuels dont l'existence ou l'importance dans la perception du contexte est liée à un moment de l'exécution. Comme pour la localité, la temporalité peut être absolue (p. ex. mesure de dates, d'intervalles temporels, etc.) ou alors logique, par la succession ordonnée d'évènements temporels (p. ex. mesure de l'historique de navigation d'un navigateur Web). Le statut mesure l'environnement physique des entités appartenant au système d'information (utilisateurs, objets, plates-formes clientes, composants distribués), et revient à la dimension externe décrite par .

2. Le contexte primaire est exploitable, mais nécessite une instrumentation plus importante. Par exemple, les mesures par capteur physique des rythmes biologiques ou la reconnaissance vidéo des expressions faciales constituent des éléments pour la perception d'un état émotionnel, d'un niveau de stress, etc.

Dimensions des *appareils* et de *l'identité*

Statut, localité et temporalité sont aussi repris par et comme catégories d'objets du contexte. Les auteurs proposent une quatrième classe, les "*appareils*", qui mesurent les caractéristiques de la plate-forme cliente. Cette dimension supplémentaire est appropriée dans le cas de la mobilité où le système, pour être universellement accessible, doit prendre en compte une panoplie de nouvelles plates-formes d'interaction, aux capacités et aux conceptions très hétérogènes. considèrent la notion d'*identité* comme une dimension supplémentaire du contexte. Celle-ci se définit comme la capacité à mesurer par un identifiant unique chaque entité du système d'information.

Dimensions de l'*utilisateur*, de la *plateforme* et de l'*environnement*

propose la notion de *contexte d'usage*, composé du triplet "Utilisateur, Plate-forme, Environnement" (). *L'utilisateur* est un représentant du public cible du système, décrit par ses capacités physiques et cognitives. *La plate-forme* décrit la structure sous-jacente à l'interaction entre l'utilisateur et le système dans le déroulement de sa tâche. *L'environnement* décrit l'ensemble des informations périphériques à la tâche de l'utilisateur, mais susceptibles de l'influencer (). Selon cette approche, une variation d'une dimension du contexte d'usage appelle une adaptation de la présentation de l'information, de l'organisation du dialogue, voire une restructuration de la tâche de l'utilisateur. Un système sachant migrer d'un triplet d'usage à un autre et réarranger les interactions avec l'utilisateur dans le respect de ses possibilités d'utilisation est dit "plastique" (). Selon l'auteur, par analogie aux sciences physiques, la *plasticité informatique* désigne la capacité d'une interface à s'adapter aux déformations matérielles et environnementales mesurées par le contexte d'usage.

Dimensions des entités mesurées par le contexte

Dans une approche originale, propose la notion de groupes d'entités pour classer les objets du monde réel sur lesquels porte la mesure du contexte, plutôt que de chercher à caractériser des groupes de mesures particulières. Selon l'auteur, les utilisateurs du système appartiennent à la classe des *entités utilisateurs* ; les autres êtres vivants sont associés à la classe des *entités sociales* ; les *entités physiques* regroupent les objets inertes n'ayant pas de capacités informatiques ; la classe des *entités système* répertorie les composants ayant des capacités de stockage et de traitement informatique. Pour l'auteur, le système construit des propriétés contextuelles à partir de mesures directes ou inférées de l'environnement, qualifiées d'*observables*. Ces observables ont valeur d'attribut pour les membres des différents groupes d'entités. Les catégories d'entités proposées par s'appliquent à la description des situations d'usage et à l'adaptation de l'interaction humain-machine.

Dimensions de l'*exécution* et de l'*infrastructure*

Dans les précédentes propositions, les dimensions présentées décrivent les éléments d'un contexte centré sur les utilisateurs, que appelé *contexte d'usage*. , dans une approche centrée

sur les caractéristiques distribuées et mobiles des systèmes d'information nomades, proposent des mesures complémentaires au contexte d'usage, qui pourraient composer les dimensions d'un "*contexte d'exécution*".

La *dimension de l'infrastructure* caractérise les impacts de la communication entre les composants distribués sur l'activité fonctionnelle du système et sur l'interaction et le dialogue avec le ou les utilisateur(s). Selon , la qualité de l'interaction et l'utilisation du système ne dépendent plus uniquement de la plate-forme cliente. Dans un système mobile, la transmission des informations entre l'ensemble des composants du système impacte l'utilité générale du système au niveau du client. Ces modifications de comportement liées au niveau du transport d'information doivent être mesurées et signalées jusqu'au niveau de l'utilisateur. Les auteurs citent l'exemple d'un outil collaboratif distribué, où l'ensemble des utilisateurs présents doivent être avertis de la connexion/déconnexion d'un client, et où le système adapte les outils proposés au nombre des utilisateurs (). postule encore que les propriétés particulières de l'infrastructure, comme la topologie du système, doivent être intégrées en parallèle de l'ingénierie de la plate-forme cliente lors de la conception d'un système mobile.

La *dimension du système* mesure les variations dans les rôles des différents composants du système. Avec la distribution, les fonctionnalités d'un système se partagent entre plusieurs composants, dont chacun subit les effets d'un contexte particulier, qui éventuellement impacte son comportement et sa réponse à des requêtes en provenance d'autres composants. Selon , le contexte du système caractérise un ensemble de réponses fonctionnelles particulières à une répartition des rôles des composants à un moment donné. Ces niveaux de réponses fonctionnelles s'élargissent considérablement dans les systèmes mobiles où cette dimension du contexte système est couplée à une infrastructure dynamique.

Synthèse et proposition de décomposition

Dans l'ensemble des propositions de classification, deux tendances transparaissent et pourraient agir comme des sur-classes aux dimensions du contexte. D'une part, le **contexte d'usage** rassemble les dimensions qui décrivent des propriétés liées à l'utilisateur, et qui concernent en priorité la définition de propriétés sur des entités utilisateurs, sociales et physiques de son entourage. D'autre part, le **contexte d'exécution** propose de modéliser le comportement dynamique et mobile d'un système d'information, et considère la distribution, l'organisation, les relations des composants à l'échelle globale du système. Cette dernière approche trouve son origine dans les travaux fondateurs sur la mobilité décrits par et , et qui citent la dynamicité de l'infrastructure et des communications à l'exécution comme un enjeu conceptuel. Ce contexte d'exécution comprend les dimensions qui définissent plus particulièrement des propriétés des entités système.

Les différentes définitions des dimensions du contexte se croisent, et posent des vocabulaires distincts sur des définitions similaires. Parmi les propositions relevant du contexte d'usage, et proposent un modèle à trois dimensions $\{\text{Environnement}, \text{Utilisateur}, \text{Plate} - \text{forme}\}$ qui englobe la plupart des dimensions décrites dans les autres travaux. En particulier, le triplet de localité, temporalité et statut de ; ; détaille essentiellement la dimension de l'environnement. De la même manière, les dimensions

externe et interne de fédèrent les mesures des dimensions "Environnement" et "Utilisateurs" de . La dimension "Plate-forme" est similaire à la notion d'appareil proposée par .

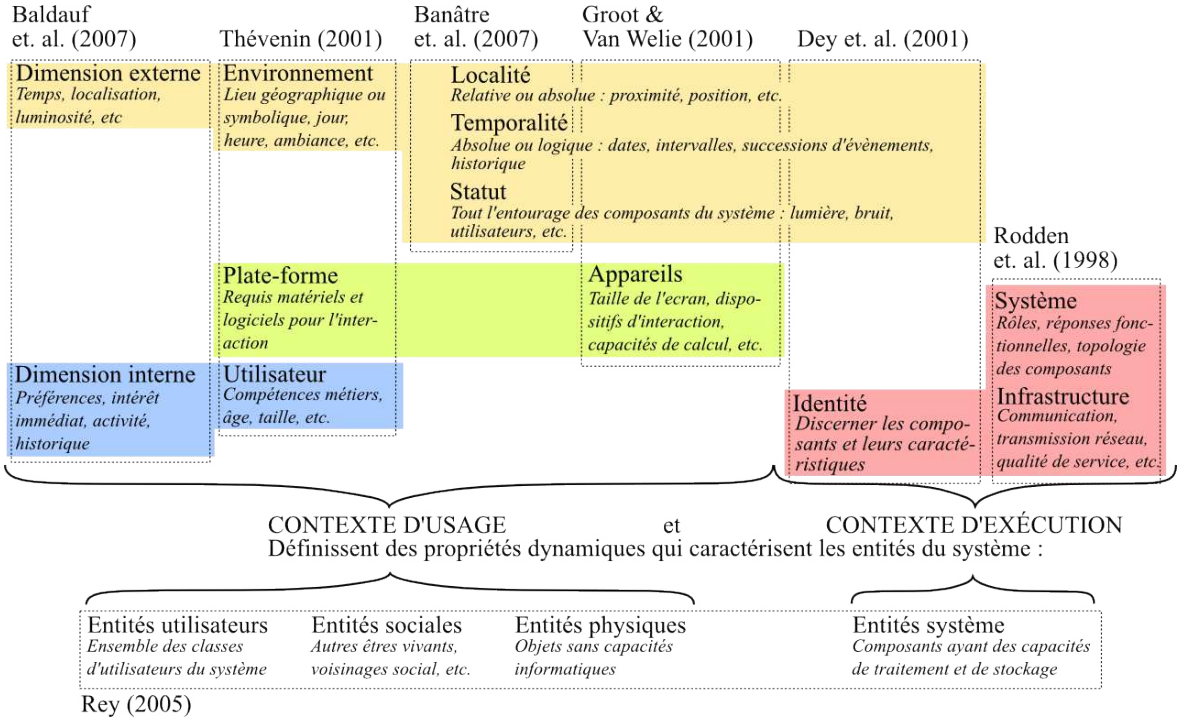


Figure II.2 – Synthèse des propositions pour le dimensionnement du contexte

Du point de vue du contexte d'exécution, l'approche proposée par pour une prise en compte globale du système selon l'identité de ses composants est à rapprocher des efforts de caractérisation d'une infrastructure et d'un système dynamique proposés par . La figure II.2 résume ces différentes propositions et souligne les similarités entre les dimensions du contexte proposées. Les regroupements effectués ne sont pas absolus, et reflètent plutôt les tendances générales des différentes définitions. En considérant des cas limites, il est souvent possible de comprendre la définition d'une dimension comme pouvant inclure des mesures issues d'une autre dimension. Par exemple, ne limite pas explicitement la plate-forme au niveau des caractéristiques du client de l'utilisateur. Cette dimension pourrait aussi inclure en partie le contexte d'exécution en considérant l'ensemble des composants comme un pré-requis matériel pour l'interaction. De la même manière, la dimension du statut de n'a pas de frontières bien définies, et pourrait, sans toutefois s'y dédier, inclure des mesures du niveau de l'utilisateur, de l'infrastructure, etc.

Dans l'objectif du développement de systèmes mobiles et distribués, il semble toutefois plus naturel de conserver une distinction claire entre l'espace d'utilisation, incluant la plate-forme cliente, l'environnement de l'utilisateur, et l'utilisateur lui-même, et l'espace du système, incluant l'ensemble de ses composants et l'infrastructure de communication. Dans cette perspective, l'état de l'environnement est caractérisé à un instant donné t par l'association d'un contexte d'usage à un contexte d'exécution :

$$\underbrace{\{\text{Environnement, Utilisateur, Plate-forme}\}}_{\text{Contexte d'usage à } t} \cup \underbrace{\{\text{Système, Infrastructure}\}}_{\text{Contexte d'exécution à } t}$$

Cette division du contexte en deux sous-ensembles constitue la base de l'approche pour l'analyse et la conception de systèmes mobile et distribués détaillée dans la suite.

II.3 Mesure des dimensions du contexte

La perception des dimensions du contexte propose différentes techniques pour permettre aux capteurs (au sens de l'informatique contextuelle) de produire des mesures quantitatives ou qualitatives. Les contextes d'usage et d'exécution partagent l'environnement du système en deux groupes de mesures. Pour chaque groupe, la perception du contexte repose sur des techniques appropriées. Dans le cas du contexte d'usage, les mesures définissent les alentours de l'utilisateur et les dimensions du triplet {Plate-forme, Utilisateur, Environnement}. Dans le cas du contexte d'exécution, la mesure de l'environnement porte sur l'ensemble de l'infrastructure du système distribué selon les dimensions {Système, Infrastructure}. Cette section détaille quelques méthodes de perception des dimensions du contexte et illustre leur implémentation dans des prototypes de systèmes d'information contextuels.

II.3.1 Mesure du contexte d'usage

L'acquisition des mesures du contexte d'usage peut s'effectuer de deux manières, éventuellement complémentaires :

- l'**approche explicite** considère l'utilisateur comme une source d'information qui pourrait être assimilé à un instrument de mesure et fournir des informations qualitatives et/ou quantitatives sur les dimensions de son environnement proche. Dans ce cas, l'acquisition des mesures du contexte repose sur la perception *explicite* par l'utilisateur.
- l'**approche implicite**, plus naturelle dans la définition historique du contexte, repose sur des moyens extérieurs à l'utilisateur pour mesurer l'environnement. Dans ce cas, la capture est *implicite* et repose sur la perception de mesures instrumentées ou inférées par l'étude implicite des comportements des utilisateurs.

L'approche explicite lève toute ambiguïté sur la perception d'une mesure. L'utilisateur associe naturellement les valeurs qui lui conviennent aux dimensions du contexte d'usage et le système est libéré de la tâche d'interprétation des mesures. Par exemple, le choix explicite d'un utilisateur en faveur d'un mode "jour" ou "nuit" mesure une information d'environnement qui permet sans ambiguïté d'activer le passage d'une adaptation "forte luminosité" à une adaptation "faible luminosité" (ou vice-versa). Une capture implicite équivalente, effectuée à partir d'instruments de mesure, devrait être confrontée à un modèle de l'utilisateur pour savoir à quelle intensité lumineuse ambiante correspond le passage d'un mode jour à un mode nuit. L'approche implicite a les qualités de ses défauts : malgré certains risques de mauvaises interprétations des mesures physiques (p. ex. passage à une luminosité faible alors que du point de vue de l'utilisateur, il fait toujours jour), elle libère l'utilisateur de l'appréciation des mesures du contexte.

En l'absence d'expérimentations démontrant clairement l'avantage d'une approche de mesure par rapport à l'autre, d'après offre le choix de la méthode de mesure du contexte à

l'utilisateur en lui fournissant une interface de contrôle de l'adaptation. Cette "*extra-IHM*" appliquée aux systèmes adaptatifs doit permettre à l'utilisateur de :

1. **choisir la source des mesures** du contexte (approches implicite, explicite ou mixte) ;
2. **comprendre les adaptations** lors de changements de contexte ;
3. **contrôler l'adaptation** en modifiant les éléments adaptés ou les déclencheurs contextuels.

Mesure de la dimension "Environnement d'Interaction"

La mesure implicite de l'environnement d'interaction comme dimension du contexte d'usage s'effectue à bas niveau par la perception instrumentée au moyen de capteurs physiques. Le tableau II.1 illustre quelques objets du contexte perçus par des capteurs physiques selon des informations en provenance d'instruments de mesure.

Tableau II.1 – Exemples de *capteurs physiques* et d'instruments de mesure ()

Objet du Contexte	**Moyens de perception**
Lumière	Photodiodes, capteurs colorimétriques, capteurs ultraviolets ou infrarouges, etc.
Environnement visuel	Tout type de caméras
Environnement audio	Tout type de microphones
Déplacement, accélération	Centrale inertielle, capteur de giration, contacteurs au mercure, détecteurs optiques de mouvement, etc.
Localisation	En extérieur : GPS, GSM ; en intérieur : système Active Badge, Triangulation WiFi, Détecteur RFID, etc.
Toucher	écrans tactiles, capteurs de pression, etc.
Température	Tout type de thermomètres
Temporalité	Horloge, chronomètre, cycles de processeur, etc.

L'approche explicite de la mesure du contexte d'environnement fait l'objet de recherches en sciences de l'information géographique, notamment pour l'identification d'une localisation physique issue d'une description donnée par l'utilisateur. Par exemple, et proposent des langages pour l'interprétation de descriptions verbales d'itinéraires ou de paysages. Associées à un modèle numérique de terrain, elle doit permettre la localisation de l'utilisateur. Dans ces systèmes, l'utilisateur fournit explicitement une description qualitative de son contexte d'environnement que le système interprète implicitement pour fournir une mesure quantitative de localisation.

Mesure de la dimension "Plate-forme d'Interaction"

Les mesures explicites d'objets de la dimension plate-forme d'interaction permettent la personnalisation des systèmes interactifs. En général, l'utilisateur se voit offrir la possibilité de régler des paramètres d'affichage, des préférences de comportements interactifs, des caractéristiques des interfaces de saisie (p. ex. rapidité du clic, vitesse de répétition de frappe clavier, etc.). Les mesures explicites portant sur la plate-forme cliente et sur les logiciels interactifs qu'elle embarque sont la plupart du temps non ambiguës et ne nécessitent

généralement pas d'interprétation sémantique particulière de la part du système avant de déclencher l'adaptation.

Les mesures implicites portent sur la détermination des capacités de la plate-forme cliente et se répartissent en trois catégories : mesure des moyens de saisie d'informations, mesure des moyens de présentation d'information, et mesure des capacités internes de la plate-forme. L'ensemble de ces propriétés détermine un profil de la plate-forme, variant éventuellement au cours de l'exécution, et auquel le système peut s'adapter. Les contraintes du niveau de la plate-forme sont issues de deux sources de mesures implicites :

- les plates-formes clientes sont décrites à la conception du système et leur caractéristiques enregistrées dans des **fichiers de configuration** que le système contextuel vient consulter lors de l'exécution ;
- le système d'exploitation de la plate-forme cliente gère le fonctionnement des périphériques d'interaction et le système contextuel exploite des **librairies d'interface vers le système d'exploitation** pour mesurer les caractéristiques des périphériques et les capacités de la plate-forme.

Les descriptions textuelles par fichier de configuration donnent une idée du dimensionnement d'une plate-forme, qui pourra éventuellement être associée à une catégorie de plate-forme (p. ex. forte capacité contre faible capacité). Ils permettent une *adaptation initiale* du logiciel à la plate-forme au moment du démarrage du système. La mesure via le système d'exploitation permet de détecter des changements des caractéristiques du système lors de l'exécution et une *adaptation à la volée* du système.

Mesure de la dimension "Utilisateurs"

La mesure de la dimension utilisateurs consiste à *générer un profil utilisateur* (User profiling ()) qui contient l'ensemble des propriétés contextuelles en relation avec l'utilisateur et pouvant être utiles pour adapter le comportement du système. Lorsqu'ils sont dynamiques à l'exécution, ces profils entretiennent l'identification, la mesure et la mise à jour des informations sur les utilisateurs (). Les méthodes de construction des profils se repartissent selon les approches de mesures du contexte d'usage :

- le **renseignement explicite** propose à l'utilisateur de saisir lui-même son profil lors de l'exécution. Cette approche est souvent utilisée pour la personnalisation d'*hypermédia adaptatifs* (Adaptive hypermedia ()) dont les sites Web sont les représentants les plus connus. Par exemple, un utilisateur renseignant son âge, ses préférences de loisirs, son adresse, ses coordonnées de paiement, etc., lors de son inscription sur un site de commerce en ligne fournit un profil personnel qui permet au site Web d'adapter le contenu présenté à ses intentions d'achat et de simplifier la procédure de livraison/facturation ;
- le **renseignement implicite** repose sur le monitoring des activités des utilisateurs lors de leurs interactions avec le système pour inférer les mesures du profil. Elle dispense l'utilisateur de toute intervention pour guider le système dans ses adaptations.
- les **profils pré-déterminés** distinguent différentes catégories d'utilisateur. Ces profils sont complétés par les concepteurs qui stéréotypent des sous-ensembles d'individus (p. ex. "novice", "expert", etc.) et proposent aux utilisateurs de choisir parmi ces profils celui qui leur semble le plus approprié. L'approche mixte est souvent utilisée pour

discriminer plusieurs niveaux de fonctionnalités dans les outils techniques comme les anti-virus, les firewalls ou les logiciels d'administration de serveur.

II.3.2 Mesure du contexte d'exécution

A notre connaissance, les mesures du contexte d'exécution sont implicites. Les variations dans l'infrastructure et le système restent pour une large mesure hors du contrôle de l'utilisateur et ne peuvent être mesurées que par les différents composants du système avant d'être rapportées à l'utilisateur sous la forme d'adaptations rendant le système plus approprié aux ressources à sa disposition.

Mesure de la dimension "Infrastructure"

La dimension "Infrastructure" du contexte d'exécution enregistre les variations dans le support réseau et dans l'organisation topologique des composants du système. Les mesures du contexte caractéristiques de cette dimension sont induites par la mobilité et la distribution des composants du système. Les mesures contextuelles qu'il est possible d'opérer sur l'infrastructure concernent les attributs caractéristiques des communications réseau présentées au chapitre I.2.2. Par exemple, le débit d'information entre les composants pour , la latence des communications pour , la perte de connexion pour , la sécurité des transmissions pour , figurent parmi les mesures impactant potentiellement la valeur d'utilité du système. D'après ces auteurs, une modification de ces objets du contexte d'exécution appelle une réaction d'adaptation appropriée à l'intention du client.

Mesure de la dimension "Système"

La dimension "Système" du contexte d'exécution reprend les mesures associées à la dimension "Plate-forme" du contexte d'usage au niveau de l'ensemble des composants distribués et mobiles du système d'information. généralise les mesures de consommation d'une plate-forme et propose trois catégories de mesures valables pour chacun des composants ("Hôtes" pour l'auteur) du système disposant de ressources d'exécution (c-à-d. les entités). La catégorie de mesure sur le logiciel détermine l'empreinte mémoire d'un processus exécuté sur un hôte (taille sur disque, taille de l'allocation à l'exécution, taille des objets chargés, etc.). La catégorie de mesure des performances fournit des indications temporelles sur l'exécution d'un processus sur un hôte (temps de chargement, temps d'exécution) et sur la migration d'un processus d'un hôte vers un autre (temps de sérialisation, de transport et de rechargement des objets ; classes et méthodes migrées). Enfin, la catégorie de mesure des ressources rapporte les capacités de traitement et de stockage d'un hôte. Contrairement à , qui associent les mesures sur le support réseau dans la dimension "Infrastructure", associe ces mesures à la catégorie des ressources de la dimension "Système".

II.3.3 Intérêt d'une mesure pour l'adaptation du système

Lors de la conception d'un système mobile, pour chaque dimension contextuelle se pose la question de l'intérêt d'une mesure et de sa pertinence pour le processus d'adaptation. À l'image de ou , la majorité des systèmes contextuels constituent un ensemble de mesures sur l'environnement et s'en servent pour l'amélioration des possibilités d'utilisation du système. Pour ces systèmes, une mesure de l'environnement est utile si elle permet de répondre par l'affirmative à la question : "Cette mesure permet-elle l'amélioration ou le maintien de la *qualité d'usage* du système ?". Pour et , cette évaluation de la qualité d'usage du système reste un critère nécessaire mais non suffisant pour qualifier la pertinence d'une mesure en vue d'une adaptation contextuelle. Selon , la qualité d'usage d'un système doit être pondérée par le contexte d'utilisation et par les buts de l'utilisateur pour juger de la **valeur du système d'information** par son adéquation aux besoins des utilisateurs dans une situation donnée. considère que la qualité d'usage, adossée à un ensemble de propriétés centrées sur l'utilisateur comme l'observabilité de ses objectifs (), l'universalité ou l'utilité du système (), etc., définit cette notion de valeur.

La *valeur* d'un système d'information donne un cap à l'adaptation et à la mesure du contexte. En particulier, la perception d'un objet du contexte n'est jugée pertinente que si elle sert à la préservation ou à l'amélioration de la valeur d'un système lors des adaptations d'une situation contextuelle à une autre. Dans la suite de cette section, les notions de *valeur interactive* et de *valeur utilitaire* sont définies. Elles décomposent la valeur d'un système selon deux dimensions :

$$Valeur(système) = \{ValeurUtilitaire(système), ValeurInteractive(système)\}$$

Les valeurs utilitaires et interactives sont appliquées respectivement aux contextes d'usage et d'exécution comme critères de pertinence pour choisir les mesures contextuelles à implémenter par rapport aux processus d'adaptation.

Valeur interactive et contexte d'usage

La *valeur interactive* évalue la qualité de l'interaction humain-machine d'un système contextuel, c'est à dire son utilisabilité en regard d'une situation donnée. Selon la norme ISO 9241-11 (Guidance on Usability) [3], l'utilisabilité est définie comme le "*degré selon lequel un produit peut être utilisé par des utilisateurs identifiés pour atteindre des buts définis avec efficacité, efficience et satisfaction dans un contexte d'usage spécifié*". D'après cette définition, le contexte d'usage, articulé autour de la mesure des trois dimensions caractéristiques de l'*utilisateur*, de la *plate-forme d'interaction* et de l'*environnement d'interaction*, influence l'utilisabilité et la valeur interactive du système. Le choix des mesures des différentes dimensions du contexte d'usage s'effectue selon leur impact sur la valeur interactive du système. Par exemple, dans le cas d'un système utilisé en intérieur sur un poste de travail fixe, la probabilité est forte pour que les mesures d'inertie ou de déplacement s'avèrent inutiles pour l'adaptation et n'améliorent pas la valeur interactive du système.

3. http ://www.iso.org/iso/catalogue_detail.htm ?csnumber=16883

La valeur interactive d'un système peut être mesurée qualitativement ou quantitativement par l'analyse expérimentale de leur efficacité dans l'atteinte de leurs objectifs et/ou par l'application de référentiels pour l'évaluation de la satisfaction des utilisateurs (p. ex. ; ; ;). La prise en compte de ces référentiels dans la démarche de modélisation des systèmes d'information adaptatifs permet de poser les frontières du contexte d'usage, en jaugeant l'utilité des mesures et des capteurs pour les processus d'adaptation dès la conception, et en ne conservant que les capteurs et mesures susceptibles d'aider la conservation ou l'amélioration de la valeur interactive du système.

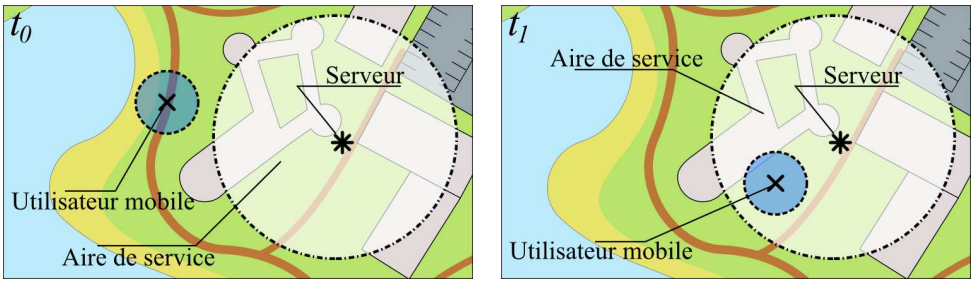
Valeur utilitaire et contexte d'exécution

L'*utilité* d'un système d'information décrite par est complémentaire de la valeur interactive dans la description de la valeur générale du système. Elle apprécie la capacité du système à accompagner l'utilisateur dans la réalisation de ses objectifs par la mise en œuvre d'outils et de fonctionnalités appropriés. Par exemple, un système de navigation mobile par GPS peut avoir une très grande valeur interactive (c-à-d. être utilisable, universellement accessible, etc.) sans pour autant être d'une quelconque utilité, si, par exemple, le fond de carte disponible ne correspond pas à l'endroit où l'utilisateur cherche à se repérer. Dans ce cas, la *valeur utilitaire* du système est faible : les utilisateurs n'atteignent que partiellement ou pas du tout leurs objectifs avec les outils à disposition.

Dans le cadre d'une utilisation mobile au sein d'un système distribué, la valeur utilitaire d'un système ne dépend pas uniquement de l'adéquation des objectifs des utilisateurs aux moyens offerts par le système. Du point de vue du système, ces moyens sont variables et liés à son contexte d'exécution (). L'utilité du système accompagne les variations du contexte d'exécution et offre une satisfaction maximale aux besoins des utilisateurs lorsque l'infrastructure et l'ensemble des composants constituant le système fonctionnent dans leur configuration optimale. Au-delà de ces conditions idéales de fonctionnement, la valeur d'utilité du système décroît. Par exemple, un système accédé via un service localisé rayonnant autour d'un point d'accès à une faible valeur d'utilité pour un utilisateur situé en dehors de son aire de service (Fig. II.3a). La valeur d'utilité augmente avec l'accès de l'utilisateur à l'aire de service et aux capacités de traitement et de stockage d'informations du service localisé (Fig. II.3a). A notre connaissance, l'évaluation de la valeur utilitaire est souvent qualitative et dépend de l'appréciation des fonctionnalités du système par des utilisateurs dans une situation donnée. Une proposition de mesure quantitative de la valeur utilitaire d'un système est présentée au chapitre VI.1.4. Basée sur l'analyse des tâches des utilisateurs, elle exprime la proportion des sous-ensembles fonctionnels qui lui sont offerts à un moment donné de l'exécution.

Les mesures du contexte d'exécution (dimension "Système" et "Infrastructure") sont jugées pertinentes si elles déterminent la valeur d'utilité du système à un moment donné et permettent de comprendre pourquoi celle-ci n'est pas maximale le cas échéant. Les adaptations qui découlent de cette analyse servent à améliorer l'utilité du système, à guider l'utilisateur vers plus d'utilité et à prévenir l'utilisateur d'une variation de la qualité fonctionnelle du système. Dans l'exemple du service localisé présenté en figure II.3, une mesure du contexte d'exécution selon la dimension "Infrastructure" permettrait à la plate-forme cliente de détecter la perte de connexion avec le serveur et de réagir à cette dégradation de la valeur

d'utilité par l'activation de fonctionnalités alternatives ou par le guidage de l'utilisateur pour son retour dans l'aire de service.



(a) "Valeur d'utilité faible" (b) "Valeur d'utilité forte"

Figure II.3 – Variation de l'utilité en fonction du contexte d'exécution

II.4 Conception et implémentation de systèmes contextuels

L'intégration des dimensions du contexte décrites précédemment au sein d'une démarche de conception passe par l'association de ces dimensions au sein d'un *modèle du contexte*. Cette seconde étape de la démarche de développement de systèmes sensibles au contexte définit un modèle informatique pour enregistrer les objets du contexte sous la forme de données numériques pouvant être traitées et prises en compte par un système d'information.

II.4.1 Approche conceptuelle : familles de modèles contextuels

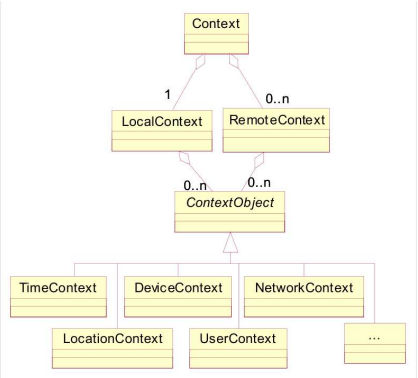
Les modèles de contextes proposés définissent un ensemble de six familles, dont les caractéristiques distinguent différents styles d'architectures numériques pour la capture du contexte, des plus simples associations par paires "clés-valeurs" aux plus complexes "modèles ontologiques" (; ;).

Les *modèles à paires clés-valeurs* proposent une structure simple pour la modélisation d'informations contextuelles. Ils associent la valeur d'un objet du contexte, comme la localisation, à une variable du système d'information. Très utilisée, cette structure est simple à implémenter et à maintenir, mais n'est pas structurellement assez riche pour inférer des relations entre les multiples dimensions du contexte. Les modèles de contexte "Context toolkit" () , "FAWIS" (), "HIPS/HyperAudio" (), "Capeus" () ou encore le modèle sous-tendu au projet "ActiveBadge" (), sont des exemples issus de cette famille.

Les *modèles logiques* appliquent en général les formalismes mathématiques de la logique des propositions et/ou de la logique des prédicats à la description des dimensions du contexte de leurs relations (). Ils formalisent la notion de contexte et définissent ses dimensions par des *propositions* articulées logiquement par un ensemble de *règles*. Les propositions contextuelles sont ajoutées ou supprimées au modèle logique par la perception de nouveaux *faits* issus soit des objets du contexte mesurés, soit d'autres propositions du modèle. Les modèles "CASS"

(), "SCOPES" (), "EXPDOC" (), sont issus de cette famille, initiée par la proposition d'un formalisme de description du contexte de .

Les *modèles orientés objet*, portent au niveau de la description du contexte les notions de classe, d'héritage, d'interface, etc. La plupart des approches existantes considèrent les mesures du contexte comme des objets auxquels le système accède par des interfaces spécifiques. Les détails des méthodes de perception du contexte sont encapsulés dans les objets eux-mêmes et cachés du reste du système. Les modèles objet du contexte s'intègrent facilement dans le développement de systèmes d'information basés sur une conception orientée objet. "Active Object Model" (), les pseudo-objets "Cues" de , "Hydrogen" (Fig. II.4a) (), ou encore les modèles objet appliqués au jeu de Go proposés par constituent quelques exemples de modèles issus de cette famille.



(a) Modèle objet : classes de description du contexte du modèle "Hydrogen" ()

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:cscp  = "http://example.org/CSCPProfileSyntax#"
   xmlns = "http://example.org/SessionProfileSyntax#"
   xmlns:dev = "http://example.org/DeviceProfileSyntax#"
   xmlns:net = "http://example.org/NetworkProfileSyntax#">
   <SessionProfile rdf:ID="Session">
      <cscp:defaults rdf:resource=
         "http://localSessionContext/CSCPProfile/previous#Session"/>
      <device><dev:DeviceProfile>
         <dev:hardware><dev:Hardware>
            <dev:memory>9216</dev:memory>
         </dev:Hardware></dev:hardware>
      </dev:DeviceProfile></device>
   </SessionProfile>
</rdf:RDF>
```

(b) Modèle à langage de description : résultat XML de la modélisation d'un profil avec "CSCP" ()

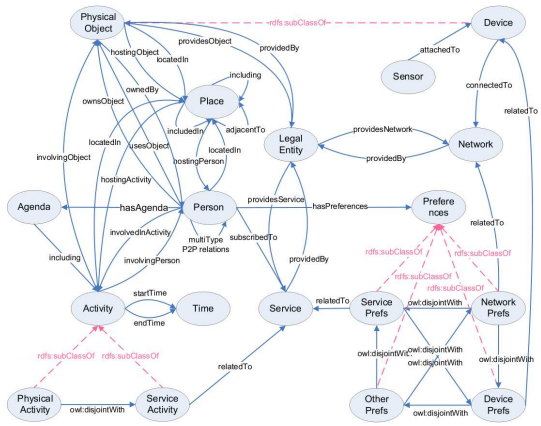
Figure II.4 – Exemples de modèles de contexte (1)

Les *modèles à langage de description* structurent hiérarchiquement les objets du contexte dans un modèle de données à partir de balises, d'attributs et de valeurs. En général, le langage de balisage choisi pour exprimer les dimensions du contexte se définit par rapport à un autre langage de description, communément de la famille XML (eXtended Markup Language). Il est ainsi possible de passer d'un modèle du contexte à un autre en appliquant des règles de transformation de schémas XML (p. ex. par l'utilisation de feuilles de styles XSL). La description d'un profil d'utilisateur est une utilisation courante des modèles langage de description : plusieurs niveaux hiérarchiques distinguent par exemple ses adresses, ses activités ou encore ses préférences d'utilisation. Les modèles "CSCP" (Fig. II.4b) (), "Context-ADDICT" () ou "CC/PP" [4] sont issus de cette famille.

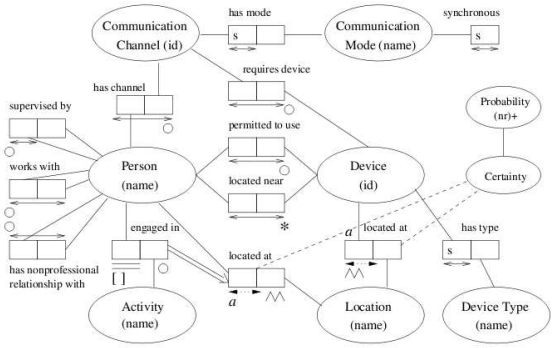
Les *modèles graphiques* proposent différentes sémiologies graphiques pour la représentation des dimensions du contexte, des objets, et de leurs relations. En général, ces modèles facilitent la compréhension de la composition et des relations des éléments du contexte par un langage graphique aisément lisible et modifiable par les concepteurs. Suivant le modèle adopté, l'application de règles de transformation automatiques ou semi-automatiques permet de passer de la représentation graphique à un modèle entité-relation définissant une structure de données informatique. UML (Unified Modeling Language), l'un

4. Composite Capabilities/Preferences Profile : http ://www.w3.org/Mobile/CCPP/

des langages de modélisation graphique les plus connus, est souvent utilisé pour exprimer les relations entre différents objets ou groupes d'objets du contexte (; ;). “Contextual-ORM” (Fig. II.5b)(), “ACTIVITY” () ou “Graphical-CM” () sont d'autres exemples de modèles issus de cette famille.



(a) Modèle Ontologique : “COMMANTO” ()



(b) Modèle Graphique : exemple d'utilisation de la sémiologie “Contextual-ORM” ()

Figure II.5 – Exemples de modèles de contexte (2)

Les *modèles ontologiques* décrivent le contexte comme un ensemble de concepts liés par des relations. Les ontologies ont montré une capacité réelle à modéliser des phénomènes de la vie réelle. Elles proposent des outils de formalisation et de raisonnement, qui, appliqués à la modélisation contextuelle, permettent d'inférer de nouvelles données contextuelles, d'un niveau d'abstraction supérieur, pour réagir à la connaissance d'un contexte dans son ensemble plutôt que de limiter le comportement adaptatif à des actions de mesure et de correction. Les modèles “COMMANTO” (Fig. II.5a) (), “SOUPA” (), “CoDaMos” () ou “SOCAM” () sont issus de cette famille.

présentent comme l'évolution des systèmes d'information depuis le modèle distribué vers l'informatique ubiquitaire, et définissent des propriétés désirables des systèmes ubiquitaires et de leurs méthodes de conception vis-à-vis du modèle de contexte adopté :

1. la **composition distribuée** fournit une mesure de l'adéquation entre le modèle du contexte et la plate-forme distribuée du système ubiquitaire. A priori, dans un système ubiquitaire, le modèle de contexte devrait être adapté à la décentralisation des services, la topologie dynamique des composants, etc ;
2. la **qualité et la richesse de l'information** est un indicateur de la prise en compte par le modèle de contexte des éventuelles pannes des dispositifs de mesures, erreurs d'interprétation ou de mesure, etc. Un système ubiquitaire met en place des indicateurs de la qualité ou fiabilité de la perception du contexte.
3. la **résolution d'ambiguïtés et d'incomplétudes** couvre et résout au niveau du modèle les inconsistances entre des mesures de plusieurs dimensions du contexte, et établit éventuellement des mécanismes d'interpolation pour compléter une perception partielle de l'environnement ;
4. le **niveau de formalisme** dénote l'expressivité atteinte par le modèle dans la description des relations entre plusieurs dimensions du contexte, dans la déduction possible de nou-

velles relations et dans la validation formelle de l'ensemble des propositions établissant des liens entre les objets mesurés du contexte ;

5. la **portabilité aux systèmes existants** mesure la capacité du modèle de contexte à être intégré au sein de systèmes d'information (mobiles, distribués ou autres) initialement insensibles au contexte.

Les auteurs proposent de positionner les familles de modèles de contexte selon leurs aptitudes à répondre à ces attentes. Pour cela, ils analysent le détail d'un ensemble de modèles proposés dans la littérature et considérés comme représentatifs des différentes familles de modèles de contexte, puis les évaluent selon les propriétés désirables des systèmes ubiquitaires.

Le tableau II.2 reprend les arguments développés par pour caractériser l'efficacité d'une famille de modèles dans les réponses apportées aux critères importants pour la mesure du contexte. L'approche adoptée fait clairement ressortir que les modèles plus anciens à base d'associations clés-valeurs restent limités dans leur approche de la caractérisation du contexte, particulièrement lorsqu'il s'agit de passer au niveau de l'interprétation des mesures, de leur corrélation ou de leur validation. A contrario, les modèles objet et ontologiques possèdent en général de bonnes capacités à exprimer des notions complexes à partir de perceptions contextuelles. Dans ces modèles, la composition distribuée des mesures du contexte relève de l'implémentation et reste transparente au moment de la conception. Par exemple, un modèle du contexte objet peut être distribué à l'implémentation par l'appel de méthodes et procédures distantes (p. ex. Java RMI, Corba, etc.). Selon les auteurs, ces dernières approches sont porteuses d'innovations pour le développement de systèmes ubiquitaires. Il convient néanmoins de nuancer l'analyse effectuée. D'une part, les modèles de contexte étudiés ne représentent qu'une partie des solutions existantes. Pour chaque critique des familles de modèles, il existe potentiellement des réponses dans la littérature qui n'ont pas été considérées dans le cadre de l'étude. D'autre part, les cinq critères retenus ne donnent pas d'indications quant à l'effort de conception et de développement nécessaire à la mise en place d'un modèle d'une famille. Par exemple, le succès non démenti des modèles à paires clés-valeurs dans le développement de prototypes de systèmes ubiquitaires tient à la facilité de leur mise en œuvre, indépendamment des défauts de ces modèles. Dans cette perspective, aucune des familles n'est exclue du développement de systèmes ubiquitaires. Le choix de l'implémentation d'un modèle d'une famille particulière doit se faire à la lumière des objectifs et des moyens du système ubiquitaire en cours de conception et de ses utilisations possibles.

II.4.2 Approche pour l'implémentation : architectures d'exécution

Lors de l'implémentation d'un système sensible au contexte, les modèles des différentes familles conceptuelles présentées précédemment peuvent êtres déployés sur plusieurs formes d'architectures de perception du contexte. Leur choix est motivé par l'application cliente et par les limites des dispositifs de capture (p. ex. contraintes de localisation ou mobilité des capteurs) ().

Une première tentative de classement des différentes solutions pour l'implémentation des modèles conceptuels de contexte est proposé par . Les auteurs distinguent trois familles d'architectures pour l'implémentation de modèles qui doivent supporter la coordination entre un ensemble de processus et de composants : les "Context-Widgets", les "Infrastructure de

Tableau II.2 – Synthèse des qualités et défauts des familles de modèles de contexte par rapport aux propriétés désirables de la sensibilité au contexte dans les systèmes ubiquitaires

		Propriétés des modèles de contextes				
		Composition distribuée	**Qualité** et **richesse de l'information**	**Ambiguïtés et incomplétudes**	**Niveau de formalisme**	**Portabilité**
Familles de modèles conceptuels du contexte	**Modèles clés-valeurs**	possible mais manque de structuration	peu appropriés	peu appropriés	assez faibles	assez simples à mettre en œuvre
	Modèles logiques	possible mais formalisme difficile à maintenir	non mesurée en général mais réalisable	non adressé en général	très bon, mais n'empêche pas les erreurs de perception	langages d'inférences souvent non disponibles
	Modèles à langage de description	adressé au cas par cas suivant les modèles, parfois non traité	meta-information aisément ajoutées aux données	résolution déléguée à l'application cliente	correct mais manque de standardisation des modèles XML du contexte	facilement applicable aux systèmes comprenant les langages XML
	Modèles graphiques	adaptation délicate de description structurelle à une plateforme distribuée	notion de "quality labels" et autres descriptions graphiques	adressées au cas par cas suivant les modèles, parfois non traité	relativement faible, outil de structuration essentiellement conceptuel	en général, les propositions sont nouvelles et s'appliquent mal à l'existant
	Modèles orientés objet	distribution transparente par l'ajout de classes et d'interfaces	description qualitative par attributs des instances de classes	utilisation des mesures de qualité pour lever les ambiguïtés	notion d'interfaces et relations. L'encapsulation limite le formalisme	très bon, mais approche objet parfois limitée sur les dispositifs mobiles
	Modèles ontologiques	qualités du modèle objet, notions ontologiques de concept et de fait	adressées au cas par cas suivant les modèles, parfois non traité	meta-informations pour résoudre les ambiguïtés et incomplétudes	très bon : héritage des modèles de raisonnement sur les ontologies	dépendante de la capacité du client à gérer un langage d'ontologie

contexte" et leur propre architecture par "Tableau partagé". Cette décomposition est corrélée par , pour qui un système d'information accède aux mesures du contexte soit 1) par un "Accès direct au niveau des mesures", 2) par l'intermédiaire d'une "Couche d'abstraction logicielle", ou 3) par un "Serveur de contexte" qui maintient un ensemble de connaissances à propos de l'environnement.

Accès direct au niveau des mesures

La capture par "**Accès direct au niveau des mesures**" couple fortement l'application développée au niveau des capteurs, parfois même au niveau physique des dispositifs de mesure. Ces méthodes d'accès bas niveau sont mélangées au niveau de l'interprétation des mesures de l'environnement. Ce mode de capture se prête mal à la distribution de systèmes et à l'interprétation de dimensions contextuelles secondaires, inférées à partir de plusieurs mesures.

Abstraction logicielle et Context-Widget

Une première abstraction du niveau physique est proposée par les architectures à "**Couche d'abstraction logicielle**" ou à "**Context-Widgets**" (Fig. II.6a). Initialement décrites par et , elles reprennent l'idée des interacteurs graphiques qui permettent à une

application cliente de s'abstraire du niveau de la présentation et du dessin de l'interface graphique. Dans le cas des Context-Widget, l'application cliente s'abstrait du niveau de l'acquisition des mesures des dimensions du contexte et propose un ensemble de propriétés de haut niveau pour le développement de clients sensibles au contexte :

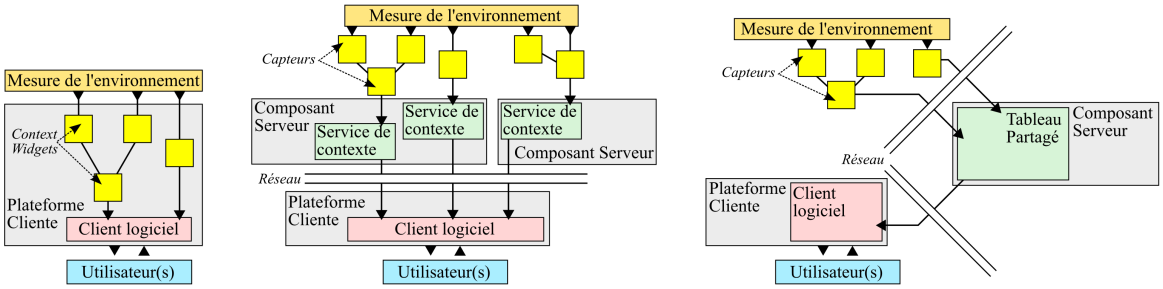
- l'architecture adopte une stricte encapsulation du niveau de la perception du contexte, et définit une sorte de librairie de haut niveau à l'intention des clients, le "Context-Toolkit". Comme pour des boîtes à outils d'interacteurs graphiques, un logiciel client peut être développé et maintenu indépendamment des évolutions du Context-Toolkit ;
- les "Context-Widget" et le "Context-Toolkit" cachent la complexité de la perception de l'environnement à l'application cliente. En particulier quand elle peut être effectuée à partir de captures différentes, la mesure (ou la combinaison de mesures) la plus pertinente est répercutée à l'application cliente ;
- ses composants abstraient les informations contextuelles et n'envoient de nouvelles mesures à l'application cliente que lorsque cela est nécessaire. Par exemple, la mesure des mouvements d'un utilisateur peut être corrélée à la description spatiale de son espace de travail pour signaler à l'application cliente le passage de l'utilisateur d'une pièce vers une autre, mais pas son déplacement au sein d'une même pièce ;
- tout comme leurs inspirateurs graphiques, les Context-Widget sont réutilisables et personnalisables : les mesures de déplacement et de localisation peuvent être utilisées dans un grand nombre d'applications dont certaines ont été présentées sous l'angle de la mobilité des systèmes d'information (cf. Chap. I.2).

Le niveau du contexte secondaire peut être encapsulé dans de nouveaux Widgets, construits à partir de Widgets de niveau inférieur. Par exemple, un Widget "Réunion ?" construit à partir des mesures des Widgets "Présence ?" et "Salle de réunion ?" inférera qu'une réunion débute quand plusieurs personnes sont présentes au même moment dans une salle de réunion. La couche d'abstraction logicielle des Widgets de contexte distingue la mesure des capteurs et le niveau de l'interprétation de ces mesures.

Infrastructures de contexte

Les architectures par "**Infrastructures de contexte**", initiées par , construisent la perception et le traitement des dimensions du contexte par des services accessibles publiquement et de façon sûre. Plus particulièrement, les auteurs décrivent une infrastructure accessible en réseau par n'importe quelle plate-forme ou client logiciel, pour le bénéfice des concepteurs, mais aussi des administrateurs qui maintiennent l'infrastructure de perception et des utilisateurs qui accèdent à cette infrastructure (Fig. II.6b). Une infrastructure de contexte peut être utilisée indépendamment d'un langage de programmation, d'un système d'exploitation ou de ressources matérielles particulières. En reposant sur des formats de données et des protocoles de communication standards et ouverts, elle laisse le client choisir la meilleure manière d'accéder aux mesures du contexte. Contrairement au modèle des Context-Widgets, il n'y a pas de gestionnaire global pour l'écoute des changements de contexte ; chaque client est responsable de l'enregistrement et de la consultation régulière des services de mesure du contexte. En termes de maintenance et d'évolution, les infrastructures de contexte peuvent être modifiées indépendamment des clients qui accèdent à leurs services, sans affecter le fonctionnement des logiciels sensibles au contexte déployés sur le client. Pour cela, les auteurs proposent une double séparation entre d'une part le niveau de perception physique et les

services de l'infrastructure de contexte, et, d'autre part, entre ces services et les clients qui y accèdent. Il est possible de changer un dispositif de mesure sans altérer le fonctionnement des services ou de modifier un capteur ou service sans dégrader l'utilisabilité du client. Enfin, les services de contexte, déployés sur des composants tiers du niveau de l'infrastructure, partagent leurs ressources, leurs capacités de traitement et leurs capacités de perception contextuelle.



(a) "Context-Widget" () (b) "Infrastructure de contexte" () (c) "Tableau partagé" ()

Figure II.6 – Architectures pour la perception du contexte à l'exécution

Serveurs de contexte

Selon , l'architecture des "**Serveurs de contexte**", est à rapprocher des "Infrastructures de contexte" de , et favorise une distribution et une perception du contexte entre plusieurs clients. Les mesures et interprétations des dimensions du contexte sont centralisées par un serveur accédé en réseau. Les clients envoient des requêtes portant sur l'acquisition ou l'enregistrement de nouvelles valeurs d'objets du contexte. Comme dans le cas des infrastructures de contexte, cette approche tire profit de la distribution des rôles d'interprétation et d'utilisation du contexte entre plusieurs composants : les clients bénéficient d'interprétations complexes des dimensions du contexte calculées par un serveur aux capacités appropriées et dédiées à cette tâche. Si elle s'applique particulièrement bien aux systèmes d'information nomades en économisant les ressources des plates-formes mobiles, cette architecture de perception du contexte est sensible aux même biais que les systèmes distribués : pannes d'infrastructure, sécurité délicate, topologie dynamique, etc. (cf. Chap. I.1).

Tableau Partagé

Par rapport aux précédentes architectures qui envisagent la perception du contexte par des approches centrées sur des processus d'enregistrement des perceptions et la répercussion des changements contextuels au niveau des clients, imagine les architectures contextuelles à "**Tableau partagé**" qui gèrent les enregistrements de mesures comme des données inscrites progressivement dans un espace partagé (le tableau), et que les clients peuvent venir consulter (Fig. II.6c). Très simple à mettre en œuvre, cette architecture permet d'ajouter ou de retirer des sources de mesures à la volée. Les clients s'inscrivent à la lecture du tableau partagé en s'abonnant à des catégories de mesures (p. ex. "Localisation", "Déplacement", etc.) et sont régulièrement prévenus de nouveaux enregistrements sur le tableau partagé. Ce modèle est

le moins fortement couplé : clients, mesures du contexte et tableau partagé sont installés sur des composants distincts et fonctionnent en complète indépendance les uns des autres. Entre eux ne circulent que des données brutes de mesures, sans optimisations pour un protocole de communication particulier comme cela peut être le cas dans les serveurs de contexte ou les infrastructures de contexte. Souvent, les échanges pourraient être optimisés pour économiser l'infrastructure. Le tableau partagé peut aussi être un point faible de l'architecture. Si celui-ci tombe en panne, les clients ne reçoivent plus de mise à jour des mesures de contexte.

Synthèse des différentes architectures proposées

À l'exception de l'accès direct aux mesures des capteurs, les différentes architectures proposent un ou plusieurs niveaux d'abstraction entre l'application cliente et la mesure de l'environnement. Cette solution facilite le développement de l'application cliente : une fois les protocoles d'échanges d'information entre les différents niveaux d'abstraction établis, ceux-ci peuvent être développés indépendamment. En particulier, le développement d'un logiciel client exploitant les variations contextuelles peut se focaliser sur les aspects haut niveau de dialogue, d'interaction et de présentation d'information pour l'utilisateur plutôt que sur la perception bas niveau du contexte. Dans le cas des architectures par "Infrastructure du contexte", "Serveurs de contexte" et "Tableau partagé", les applications clientes sont isolées des autres niveaux par un accès réseau. Ce découplage permet l'indépendance de leurs processus à l'exécution, et l'éventuelle re-configuration dynamique de la partie du système chargée de la perception du contexte par l'ajout/remplacement de dispositifs physiques de mesures, la réorganisation de leur structures, etc.

L'ensemble des modèles conceptuels proposés par ; ; peut être implémenté et exécuté sous la forme des architectures décrites par et . Toutefois, selon le modèle conceptuel retenu pour décrire le contexte, un type d'architecture à l'exécution peut s'avérer plus approprié qu'un autre . Par exemple, les modèles orientés objet, les modèles ontologiques ou les modèles à langage de description, qui sont de niveaux supérieur et qui modélisent facilement une composition distribuée, s'implémentent naturellement sous la forme d'architectures par "Serveurs de contexte" ou "infrastructure de contexte". À l'opposé, pour les modèles plus simple par associations clés-valeurs, la composition distribuée, la qualité et richesse d'information, le niveau de formalisme atteint ou la levée des ambiguïtés de mesures ne forment pas des critères déterminants. Ces modèles pourraient être facilement implémenté (voire ajouté à un projet existant) sous la forme de "Context-widgets" ou éventuellement par une prise en compte directe des mesures si la séparation entre les niveaux de perception et d'utilisation des mesures du contexte n'est pas prioritaire. L'architecture par "Tableau partagé" permet d'implémenter indifféremment l'ensemble des modèles conceptuels. La gestion des ambiguïtés et de la qualité des mesures est par contre déléguée au côté client, le tableau se contentant d'un enregistrement passif des mesures captées. Cette solution d'architecture paraît particulièrement adaptée pour le portage d'un système contextuel non distribué vers une version où le client et la mesure du contexte sont isolés par l'infrastructure réseau.

notent enfin que si les approches par "architecture réseau" et "architecture distribuée" de l'acquisition du contexte se qualifient pour l'application de la perception de l'environnement aux systèmes mobiles, celles-ci révèlent plusieurs enjeux :

- La **découverte des ressources pour la mesure du contexte en mobilité** concerne, pour le client, la recherche de nouveaux services de contexte, et pour la partie serveur de contexte, l'enregistrement de nouvelles mesures de l'environnement, et éventuellement de nouveaux capteurs. Par exemple Le modèle "SOCAM" (), implémenté dans une "Infrastructure de contexte", propose un mécanisme de localisation de services de contexte exécuté au niveau du client (*"Service locating service"*). proposent la notion de "Discoverer" ("découvreur"), composant logiciel déployé au niveau de chaque client et qui permet d'enregistrer à la volée de nouvelles sources de mesures par l'ajout de "Context-Widgets" lors de l'exécution.
- Les mesures du contexte peuvent inclure des données à caractère privé sur les utilisateurs ou leur environnement. Les partager en réseau soulève la question de la **sécurité des transactions entre serveur de contexte et plate-forme client**. Pour y répondre, le "Context-Toolkit" propose de donner des droits de lecture sur les mesures des Widgets en fonction de l'utilisateur qui y accède. Les "Mediated Widget" sont des composants qui s'ajoutent dans les réseaux de "Context-Widgets" et qui gèrent le contrôle des droits pour l'accès aux mesures. Le modèle ontologique "SOUPA" () inclut un vocabulaire de contraintes inspiré par le langage de spécification règles OWL (Web Ontology Language) "REI" (). Dans le cas de SOUPA, les contraintes formulées portent sur la définition de différents niveaux d'autorisation en fonction du contexte et de l'utilisateur et sur la définition de règles d'accès aux mesures d'un contexte.
- La **conservation de l'historique des mesures du contexte** constitue un défi supplémentaire pour un système distribué. La plupart des modèles de contexte déployés sur des architectures distribuées enregistrent l'historique du contexte au niveau d'une base de données déployée sur un composant serveur en lien avec les sources de mesure du contexte et consultable par les clients (p. ex. méthode adoptée par les modèles "Context-Toolkit", "SOUPA", "SOCAM", "CASS"). Toutefois, le serveur d'historique de contexte établit un nœud de réseau qui pose problème dans le cas des systèmes mobiles où l'utilisateur comme les sources de mesures peuvent perdre le lien avec ce composant. L'enregistrement de l'historique des mesures serait alors partiel (ou éventuellement différé) et fonction de la qualité de l'infrastructure de communication.

reprennent la plupart des exemples cités en illustration des familles de modèles des architectures à l'exécution et étudient leur efficacité pour relever des enjeux du développement de systèmes sensibles au contexte dans le cadre de l'informatique mobile. Les auteurs concluent que dans la plupart des cas :

1. les politiques d'accès aux informations contextuelles et la gestion des droits sur ces données sont négligées ;
2. les mécanismes de découverte de nouvelles ressources sont en fait rarement implémentés (exception faite du modèle "SOCAM") ;
3. les architectures déployées centralisent les données contextuelles, exposant la sensibilité à l'environnement et la capacité d'adaptation d'un système aux aléas de l'infrastructure.

Ces difficultés des modèles présentés à apporter des réponses aux enjeux de la perception du contexte dans un système distribué et mobile signifient qu'une fois implémentés, ils risquent de tenir leur rôle de façon incomplète. À l'implémentation, la centralisation des mesures fonctionne dans le cas d'un système distribué optimal, insensible aux biais énoncés par . Le chapitre précédent a montré que le développement des systèmes mobiles passait par la résolution

de ces biais et qu'en particulier, l'adaptabilité aux changements d'une infrastructure dynamique est critique en mobilité. Au même titre que lors du développement de systèmes mobiles, les approches conceptuelles et les modèles architecturaux des systèmes sensibles au contexte doivent apporter des solutions aux biais d'infrastructure et s'interroger sur le fonctionnement de la perception contextuelle dans des situations dégradées (p. ex. pannes de dispositifs de mesures, couche de transport réseau inconsistante, mobilité des composants et des clients, etc.).

Quelques initiatives pour améliorer la robustesse des systèmes contextuels mobiles ont été proposées. Par exemple, , dans leur définition du modèle "Hydrogen", suggèrent une démarche de réplication/distribution des données d'historique sur l'ensemble des composants mobiles pour conserver une meilleure qualité globale d'enregistrement et d'accès à un historique du contexte. D'autre part, l'historique des mesures de l'environnement, pris en compte par la plupart des modèles détaillés par , apporte une dimension transverse aux mesures du contexte qui peut être utilisée pour pallier à des inconsistances dans la mesure de l'environnement. Par exemple, l'historique d'une mesure contextuelle peut être utilisé pour éliminer des valeurs aberrantes, pour compléter la perception d'une dimension contextuelle où certains capteurs ne sont plus disponibles ou encore pour prévoir les réactions d'un utilisateur à un environnement donné.

II.5 Utilisation des informations contextuelles

La mesure du contexte ne constitue pas une fin en soi. L'intérêt de la sensibilité du contexte repose sur son utilisation dans les systèmes d'information. Les *systèmes sensibles au contexte* regroupent un ensemble de catégories de systèmes d'information dont le point commun est une certaine utilisation des mesures contextuelles. D'après , le terme "*système sensible au contexte*" () est devenu dans une large mesure un synonyme des notions de "*système situé*" (), de "système adaptatif" (), de "système dirigé par l'environnement" (), ou de "système réactif" (;). Les auteurs classent ces propositions de systèmes sensibles au contexte selon qu'ils utilisent le contexte ou qu'ils s'y adaptent.

- Les *systèmes utilisant le contexte* englobent les définitions et , pour lesquels un système est sensible au contexte s'il est capable de prévoir, détecter, interpréter et répondre à des aspects de l'environnement des utilisateurs et des dispositifs informatiques. définissent la sensibilité au contexte par la capacité d'un système à réagir automatiquement à des variations dans l'environnement de l'utilisateur. Plus spécifiquement, les auteurs limitent l'effet de cette sensibilité au domaine des interfaces humain-machine (). proposent une définition qui met l'accent sur la capacité d'adaptation à l'exécution par la perception en temps réel du contexte.
- Les *systèmes s'adaptant au contexte*, retenus par de nombreux auteurs, restreignent la définition de l'adaptation au contexte à l'objectif de modification du comportement d'une application en fonction de conditions d'utilisation (; ; ; ;). Plus spécifiquement, pour , un système s'adapte au contexte s'il fournit des informations ou prend des décisions en se reposant sur des mesures du contexte. Pour les auteurs, les actions contextuelles comprennent les modifications dans l'organisation de l'interface, la décision de l'exécution d'un programme et le choix des éléments à présenter à l'utilisateur. D'après , les applications dirigées par l'environnement adaptent leur ni-

veau fonctionnel au contexte courant par l'application de guides d'adaptation définis par les utilisateurs.

considèrent que les définitions précédentes, relatives à l'adaptation au contexte, sont une forme particulière des définitions plus générales données aux systèmes utilisant le contexte. Selon les auteurs, "*un système est sensible au contexte s'il utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur (où la pertinence dépend de la tâche de l'utilisateur*". Comme les critères de *valeur interactive* et de *valeur utilitaire* permettent de juger de la pertinence d'une mesure contextuelle par rapport aux attentes des utilisateurs, cette définition de la sensibilité au contexte place en son centre le rôle essentiel de l'utilisateur et de sa tâche pour trier ce qui peut et doit être adapté de ce qui ne le doit pas. Elle ne limite pas la sensibilité au contexte aux seuls domaines des interfaces et des interactions humain-machine, et ne demande pas obligatoirement l'application automatique des adaptations à l'instigation du système.

II.5.1 Typologies des formes d'exploitation du contexte

Dans les sections précédentes, certaines utilisations du contexte ont été présentées comme illustrations des définitions et des concepts liés à la perception des mesures du contexte. À un plus haut niveau, et sous le chapeau commun des "systèmes sensibles au contexte" et de la définition donnée par , différents sous-ensembles de cas d'application du contexte aux systèmes d'information peuvent être discriminés.

Dans une première proposition de classification, organisent les systèmes sensibles au contexte selon deux dimensions : 1) le contexte peut valider la sélection d'information ou l'exécution de commandes et 2) la validation peut être soit automatique, soit manuelle. En combinant la validation et l'utilisation faite du contexte, les auteurs distinguent quatre classes d'utilisation du contexte dans les systèmes d'information (Tab. II.3).

Tableau II.3 – Typologie des utilisations du contexte pour

	Validation des changements	
	manuelle	**automatique**
Sélection d'information	*sélection par proximité*	*re-configuration contextuelle*
Exécution de commande	*commandes contextuelles*	*déclenchement d'actions*

Les systèmes à "*sélection par proximité*" utilisent les mesures contextuelles pour mettre en valeur les informations d'intérêt pour l'utilisateur afin de faciliter leur sélection manuelle. Les systèmes à *re-configuration contextuelle* proposent automatiquement des adaptations de bas niveau aux ressources de l'environnement. Par exemple, le système ParcTab active ou désactive des fonctionnalités en fonction des ressources matérielles environnantes (p. ex. imprimantes, téléviseurs, autres clients ParcTab, etc.)(). Les systèmes à "*commandes contextuelles*" exécutent des commandes "contextualisées" à la demande de l'utilisateur. Toujours avec le système ParcTab, une demande d'impression s'exécutera par défaut sur l'imprimante la plus proche de l'utilisateur. Enfin, les systèmes appliquant automatiquement des commandes en fonction du contexte sont dits à "*déclenchement d'actions*". Ils consistent en un ensemble de services et de fonctionnalités déclenchés à l'initiative du système quand certaines conditions contextuelles sont réunies.

reprend en partie la typologie proposée par : ses définitions de systèmes à "*adaptation contextuelle*" et à "*découverte de ressources contextuelles*" recouvrent respectivement les notions de "*déclenchement d'actions*" et de "*re-configuration contextuelle*". Les auteurs proposent deux types d'utilisation du contexte supplémentaires. Au niveau le plus simple, un système à "*mesure du contexte*" présente une information contextuelle directement à l'utilisateur en complément d'information, et sans chercher à l'utiliser pour modifier d'autres informations ou fonctionnalités (p. ex. un système de carte numérique matérialisant la position courante de l'utilisateur par une indication "vous êtes ici"). L'*augmentation contextuelle* consiste à associer des informations selon différents contextes et à les présenter aux utilisateurs se trouvant dans ces contextes. Cette application de la sensibilité au contexte est souvent rencontrée dans le cas des SIG localisés, dans lesquels les informations présentées sont par exemple relatives à la position de l'utilisateur. Pour les auteurs, les systèmes à augmentation contextuelle permettent à leurs utilisateurs de fournir des nouvelles annotations associées à un contexte pour mettre à jour les informations présentées. excluent de leur typologie des usages du contexte la notion de "commandes contextuelles" de .

proposent une version simplifiée de ces typologies et décomposent l'utilisation du contexte selon trois catégories :

- la *présentation* d'informations et de services à un utilisateur combine la "*sélection par proximité*" et les "*commandes contextuelles*" de ;
- l'*exécution* automatique d'un service reprend les "*déclenchements d'actions*" de et "*l'adaptation contextuelle*" de ;
- l'*annotation* d'informations contextuelles pour une réutilisation ultérieure est à rapprocher de "*l'augmentation contextuelle*" de .

Contrairement à , ne différencient pas la *sélection d'information* de l'*exécution de commandes*, arguant que dans la plupart des cas il est difficile de distinguer ces deux dimensions. Par exemple, un système ordonnant une liste d'imprimantes par rapport à la localisation de l'utilisateur propose à la fois "*sélection par proximité*" pour la présentation de la liste, mais aussi l'exécution de "*commandes contextuelles*" dès que l'utilisateur valide une impression sur l'une des imprimantes de la liste. Les auteurs excluent aussi l'utilisation du contexte pour faciliter l'adaptation aux ressources de l'environnement. Ils associent cette "*re-configuration contextuelle*" à des cas de *présentation* d'information ou d'*exécution* de services. Par exemple, la détection automatique des imprimantes à proximité (c-à-d. une *re-configuration contextuelle*) n'est vue que comme une étape intermédiaire préalable à la présentation d'une liste d'imprimante mise à jour (c-à-d. *présentation* d'informations) ou à l'exécution automatique d'impressions de documents (c-à-d. *exécution* automatique de services).

Plus récemment, distinguent quatre formes d'exploitation du contexte. Construites à partir des propositions de ; ; , elles s'en détachent par une orientation sensible vers le support de la mobilité des systèmes et la distribution des composants. l'**adaptation contextuelle** appartient à la même famille que l'*exécution contextuelle automatique* de et propose une définition similaire à l'*adaptation contextuelle* selon . Pour , elle exploite les mesures du contexte pour s'adapter aux contraintes des environnements d'utilisation et d'exécution. Par exemple, le service mobile de cartographie CRUMPET exploite un modèle contextuel de l'utilisateur pour choisir un ensemble de couleurs de représentation en fonction du passif culturel des utilisateurs ().

L'**annotation contextuelle**, dérivée de l'*annotation* d'information de et de l'*augmentation contextuelle* de , ajoute des mesures de contexte à des objets du système, ou alors étiquette les informations ou les services par des contextes particuliers. Par exemple, l'outil ContextPhoto, repose sur un modèle ontologique du contexte pour annoter automatiquement les photographies par des meta-données inférées des mesures du contexte à la prise de vue (p. ex. des informations temporelles, de saisonnalité, de météo, de localisation, de voisinage personnel, etc.)().

Inspirée par la *sélection par proximité* de et la *présentation* d'information et de services de , la **recherche contextuelle d'information** consiste à filtrer les données pertinentes pour l'utilisateur sur la base de la situation contextuelle mesurée et à les lui présenter de façon appropriée. Par exemple, le projet COMPASS fournit un service mobile de recommandation touristique. Il repose sur différentes sources de données cartographiques et sur une mesure de la dimension environnementale du contexte d'usage pour déterminer les points d'intérêt de l'utilisateur et lui proposer une information filtrée, portant principalement sur ces points d'intérêts ().

Enfin, la **programmation contextuelle**, dernière forme d'exploitation du contexte retenue par , est assimilable aux notions d'*adaptation contextuelle* de et de *déclenchement d'actions* de . Elle consiste à "synchroniser des actions sur la détection de certains contextes". La programmation contextuelle fait partie des première méthodes d'exploitation du contexte. Le projet ParcTab () a initié la programmation contextuelle et propose une grammaire de règles simples de la forme "*Si* (Évènement contextuel) *Alors* {Réaction}". Les évènements détectés sont par exemple l'entrée ou la sortie d'une pièce, l'absence ou la présence d'autres entités (c-à-d. composants du système) dans une pièce, le changement de statut d'une entité, etc. Les réactions produisent des retours d'informations sur la plate-forme cliente mobile Parctab de l'utilisateur. Par exemple, lorsque le café a fini d'être filtré, la cafetière de la salle café du Xerox Parc déclenche un événement contextuel signifiant son changement de statut et qui provoque une sonnerie sur les clients ParcTab de la même pièce pour alerter que le café est prêt .

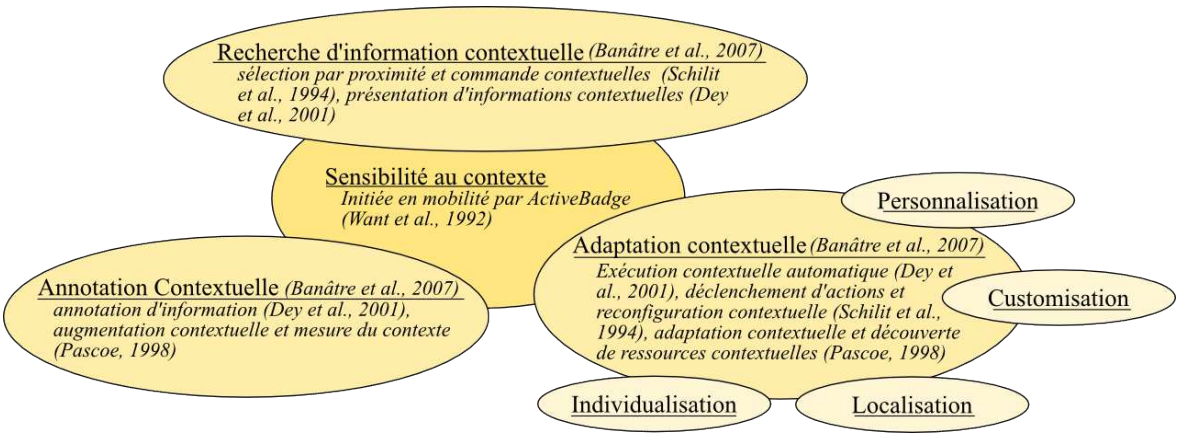


Figure II.7 – Synthèse des propositions typologiques de l'exploitation du contexte

La figure II.7 reprend l'ensemble du vocabulaire posé en définition des différentes typologies d'exploitation du contexte. La sensibilité au contexte s'exprime principalement selon

les catégories retenues par . Celles-ci effectuent la synthèse des différentes formes d'exploitation du contexte explorées depuis la première initiative de système mobile contextuel par le projet ActiveBadge (). La plupart des travaux de recherche et des prototypes proposés dans les domaines de la mobilité et de l'ubiquité des systèmes sont assimilables à une ou plusieurs des catégories d'exploitation contextuelle ; . Toutefois, la définition de la *programmation contextuelle* place cette forme d'exploitation du contexte en décalage par rapport aux autres propositions retenues. En termes de résultats visibles du point de vue de l'utilisateur, il n'est pas évident de distinguer la *programmation contextuelle* de l'*adaptation contextuelle* et l'exemple de la cafetière aurait tout aussi bien pu illustrer cette dernière approche. Plutôt qu'une forme d'exploitation à part entière, la *programmation contextuelle* se pose comme une approche transverse à l'*adaptation contextuelle*, à l'*annotation contextuelle* et à la *recherche d'information contextuelle*. Comme la notion de *commandes contextuelles* pour , la programmation contextuelle apporte une solution d'implémentation, mais ne constitue pas une forme d'exploitation du contexte en soi.

L'exploitation des mesures de l'environnement par l'adaptation contextuelle distingue les systèmes adaptables/adaptatifs de l'ensemble plus général des systèmes sensibles au contexte. Cette forme d'application de la sensibilité à l'environnement peut être détaillée selon différentes approches () :

- la *personnalisation* est l'adaptation contextuelle sur décision du système d'un élément ou d'une fonctionnalité par l'interprétation faite par le système des informations à propos de l'utilisateur (). La personnalisation est relative à un seul utilisateur ;
- l'*individualisation* fournit une définition plus générale de la personnalisation. Elle peut désigner aussi bien l'adaptation à un individu qu' à un groupe de personnes ;
- le *ciblage culturel*[5] est l'adaptation d'un système d'information, en général du niveau de l'interface utilisateur, à une région, une culture, une langue particulières () ;
- la *customisation* est l'adaptation d'un produit ou d'un service sur décision de l'utilisateur (). La customisation peut être suggérée à l'utilisateur lors de l'exécution et utiliser des mesures du contexte, mais le choix de l'adaptation reste à la discrétion de l'utilisateur et le système ne prend pas d'initiative.

La suite de cette section se concentre sur l'*adaptation contextuelle* comme forme d'exploitation du contexte. Différentes solutions pour le déroulement d'un cycle d'adaptation sont détaillées et s'attachent à répondre aux questions "Qui prend la décision de l'adaptation ?", "Quand doit avoir lieu l'adaptation ?", "Que faut-il adapter ?" et "Comment contrôler le résultat d'une adaptation ?" (). Les principes et solutions décrits s'appliqueraient de la même manière à la *recherche d'information contextuelle*.

II.5.2 Initiative et responsabilité de l'adaptation contextuelle

Dans une perspective d'amélioration de la qualité d'usage et d'enrichissement contextuel, le système effectue des adaptations lors de l'exécution pour le bénéfice des utilisateurs. A l'instar de la mesure du contexte d'usage, qui est, selon le cas, effectuée de façon implicite par le système ou alors saisie explicitement par les utilisateurs, l'initiative du déclenchement des processus d'adaptation revient soit à l'utilisateur soit au système. Les systèmes sont dits

5. Dans ses travaux, retient le terme de "localisation" ici remplacé par "ciblage culturel" pour éviter toute confusion de sens.

soit “adaptatifs” soit “adaptables” selon qui, de l’utilisateur ou du système, prend l’initiative de l’adaptation (;). Un *système adaptatif* s’adapte sur l’initiative du client, tandis qu’un *système adaptable* laisse l’utilisateur (éventuellement assisté par le système) prendre l’initiative des adaptations. Le tableau II.4, proposé par , présente les distinctions majeures entre systèmes adaptables et systèmes adaptatifs et les forces et faiblesses de ces deux approches.

d’après propose de classer les systèmes à adaptation contextuelle par niveaux de sophistication croissante. En plus de la distinction binaire des systèmes adaptables/adaptatifs, les auteurs détaillent différentes catégories de systèmes adaptatifs :

- un **système autorégulateur** est capable d’évaluer l’effet des adaptations qu’il déclenche par une mesure de *feedback* ;
- un **système automédiateur** sait planifier une adaptation et prévoir son effet à priori ;
- un **système automodificateur** est capable d’apprendre de nouveaux déclencheurs et de nouvelles réactions.

La différence majeure entre la distinction simple adaptatif/adaptable et ces niveaux de sophistication provient de la capacité des systèmes autorégulateurs, automédiateurs et automodificateurs à construire sur la base d’un système adaptatif des niveaux de régulation, de contrôle et de planification de l’adaptation. Ces capacités supplémentaires sont à la charge du système. D’autres classifications des systèmes adaptatifs pourraient être ajoutées, si, par exemple, l’utilisateur avait la charge de juger la valeur d’une adaptation.

Tableau II.4 – Comparaison des systèmes adaptables et adaptatifs (d’après)

	Système adaptable	**Système adaptatif**
Initiative de l’adaptation	Utilisateur	Système
Forces	l’utilisateur, en pleine connaissance de ses buts et de sa tâche, garde le contrôle sur le système	peu d’efforts ou de connaissances supplémentaires requises de la part des utilisateurs
Faiblesses	demande plus d’attention de la part de l’utilisateur, nécessité d’apprendre comment et à quel moment déclencher l’adaptation	difficultés à percevoir le fonctionnement du système, sentiment de perte de contrôle
Domaine d’application	systèmes personnalisables, systèmes reconfigurables, filtrage semi-automatique d’information, etc.	systèmes d’aide active, customisation d’interface utilisateur, filtrage automatique d’information, etc.

Un partage des responsabilités plus équitable entre l’utilisateur et le système dans l’exécution d’une adaptation a été proposé par . Les auteurs détaillent quatre étapes pour effectuer lors de l’exécution un cycle d’adaptation à partir des mesures du contexte : l’étape d’*initiative* (notée e_1) décide du déclenchement du cycle d’adaptation ; l’étape de *suggestion* (e_2) propose différentes réactions d’adaptation ; l’étape de *décision* (e_3) sélectionne la réaction appropriée parmi les différentes suggestions ; l’étape d’*exécution* (e_4) met en oeuvre la réaction choisie. Chaque étape est validée soit par l’utilisateur, soit par le système et seize partages de responsabilités forment l’ensemble des décompositions possibles des cycles d’adaptation.

adoptent une notation matricielle pour l’écriture de ces décompositions, avec la lecture des différentes étapes en ligne (Fig. II.8($e_1 \rightarrow e_4$)) et le choix de la responsabilité entre le système et l’utilisateur en colonne (Fig. II.8(🯅 & 🯆)). Les auteurs retiennent ensuite les

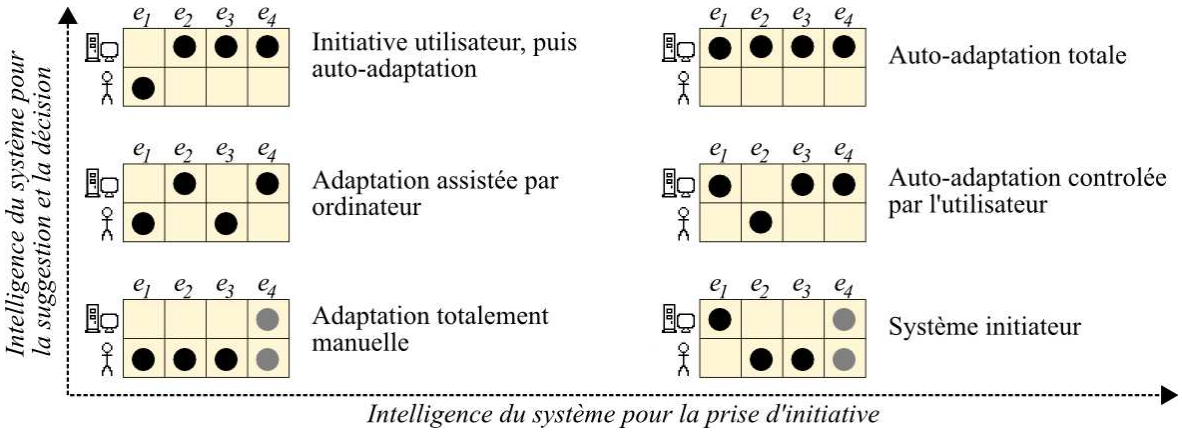


Figure II.8 – Partage des responsabilités dans le déroulement d'une adaptation (d'après)

partages de responsabilités les plus intéressants selon leur étude de l'état de l'art et proposent de discriminer les systèmes adaptables et adaptatifs en six catégories depuis une *adaptation totalement manuelle* où l'utilisateur décide de toutes les étapes, jusqu'à l'*auto-adaptation totale* pour laquelle le système valide toutes les étapes. Ces catégories sont positionnées dans un graphique à deux dimensions où l'ordonnée désigne l'intelligence du système pour les étapes de suggestion et de décision et l'abscisse désigne l'intelligence du système pour l'étape de prise d'initiative (Fig. II.8). D'après les auteurs, cette répartition positionne les systèmes adaptables sur la colonne gauche du graphique (abscisse proche de l'origine), et les système adaptatifs sur la droite. La distinction entre systèmes adaptables et adaptatifs s'effectue essentiellement selon l'acteur qui prend l'initiative de l'adaptation. Plus précisément, l'auto-adaptation totale de correspond à la définition de système adaptatif dans . L'adaptation totalement manuelle, l'adaptation assistée par ordinateur et l'auto-adaptation contrôlée par l'utilisateur correspondent à différents niveaux d'un "système adaptable" pour .

II.5.3 Déclenchement du processus d'adaptation

Que le système soit adaptable ou adaptatif, et quels que soient les partages des responsabilités de l'adaptation entre l'utilisateur et le système, le déclenchement d'une adaptation a lieu à un moment donné et est éventuellement répété à une certaine fréquence. identifient trois moments pour l'adaptation d'un système :

- l'**adaptation avant la première utilisation** est initiée par les utilisateurs ou les administrateurs de la plate-forme cliente lors, par exemple, de l'installation de la partie cliente sur un système où cette dernière est adaptée à un profil de contexte d'utilisation (en particulier, la meilleure adaptation à la plate-forme est souvent recherchée) ;
- l'**adaptation après l'utilisation** attend la relance du système pour effectuer les changements contextuels mesurés ou demandés par les utilisateurs lors de la session précédente ;
- l'**adaptation pendant l'utilisation** consiste en une re-configuration dynamique du système en cours d'exécution. Pour ce dernier moment de l'adaptation, les auteurs distinguent encore l'**adaptation en continu**, qui a lieu tout au long de l'exécution à

intervalles réguliers, de l'**adaptation conditionnelle**, qui a lieu lors de la validation de certaines conditions (p. ex. lors du passage d'un contexte à un autre).

Une distinction similaire est proposée par . Pour les auteurs, la **macro-adaptation** désigne une adaptation occasionnelle, qui a lieu normalement avant ou après l'utilisation du système. À l'opposé, la **micro-adaptation** reprend l'idée de l'adaptation en continu de et désigne une *adaptation pendant l'utilisation*, ayant lieu à des intervalles courts et réguliers.

Le choix du moment de l'adaptation inclut implicitement un questionnement sur les déclencheurs d'une adaptation. Pour , l'adaptation est conditionnelle et est déclenchée soit par un **changement dans le temps** soit par une **différence de situation**. Les changements au cours du temps sont déclenchés par les écarts entre des mesures d'objets du contexte (p. ex. température extérieure, localisation d'un utilisateur, etc.). Une différence de situation n'est par contre pas forcément issue d'une comparaison entre deux mesures temporelles (p. ex. à un moment donné le système peut produire des adaptations différentes selon des profils d'utilisateurs particuliers).

La suite de cette section détaille plus particulièrement la forme d'adaptation au moment ou pendant l'exécution, à la fois lors de déclenchements conditionnels ou dans un cycle d'adaptations d'une fréquence donnée.

II.5.4 Objets de l'adaptation à l'exécution

L'ensemble des systèmes capables de s'adapter au moment de l'exécution construisent une boucle procédurale d'adaptation (Fig. II.9). Un modèle d'**objet de l'adaptation** (p. ex. une interface graphique utilisateur) est transformé en un **objet adapté** (p. ex. interface graphique simplifiée sans zone de saisie de texte) par un ensemble de **processus d'adaptations** (p. ex. sélection entre plusieurs solutions d'interfaces). Ces processus réagissent à des **mesures du contexte** (p. ex. mesure de la capacité de saisie de texte) portant sur une **cible d'adaptation** particulière (p. ex une plate-forme client sans clavier) et organisées selon les dimensions du contexte d'usage et du contexte d'exécution. A l'issue du processus d'adaptation, l'espace existant entre la cible et l'objet adapté qualifie la **qualité de l'adaptation** : plus cet espace est réduit, meilleure est l'adaptation.

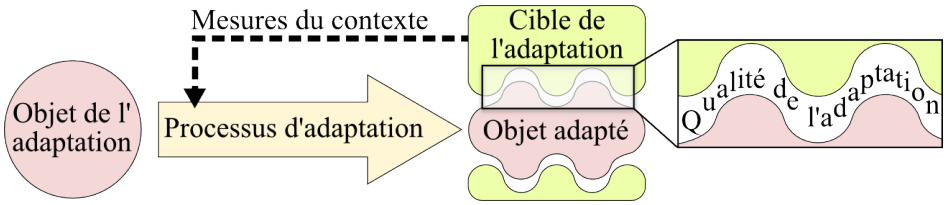
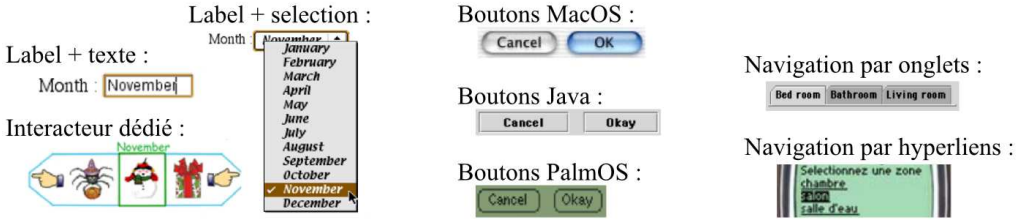


Figure II.9 – processus d'exécution d'une adaptation dans le cas des systèmes adaptables/adaptatifs (d'après)

Les objets de l'adaptation procèdent essentiellement de deux catégories : les *objets de contenu* d'une part, et les *objets de forme* d'autre part (;). Les **objets de contenu** désignent l'ensemble des informations traitées et stockées par le système pouvant être adaptées à une cible particulière. Elles sont présentées à l'utilisateur en réponse à ses requêtes ou alors automatiquement suite à une initiative d'adaptation par le système. L'adaptation d'objet de

contenu considère la requête de l'utilisateur comme une mesure du contexte à part entière, éventuellement associée à une ou plusieurs autres mesures d'usage ou d'exécution qui ensemble concrétisent une exploitation du contexte par "*Recherche d'information contextuelle*" (). Le but du processus d'adaptation est alors de filtrer l'information source (c-à-d. l'objet du contexte) pour produire en sortie un objet adapté contenant les informations requises pour satisfaire l'utilisateur sachant son contexte. Par exemple, proposent un service mobile de cartographie s'adressant aux campeurs dans un parc naturel de Finlande. Les campeurs équipés peuvent rechercher les zones où il est possible de faire du feu à proximité de l'endroit où ils se trouvent en ayant connaissance des conditions météo globales des jours précédents. Dans un tel système l'ensemble des données sources (p. ex. informations sur le parc, zones de camping, zones protégées, zones de pique-nique, etc.) constitue l'objet du contexte. La requête de l'utilisateur ("Où faire du feu ?") est associée à deux mesures du contexte d'usage (c-à-d. dimension "Utilisateur" → mesure de "localisation" et dimension "Environnement" → mesure de "conditions météo"). Fournies en entrée des processus d'adaptation, ces mesures établissent des critères de filtrage de l'objet de l'adaptation pour produire un objet adapté : le système présente à l'utilisateur les zones de pique-nique équipées d'un foyer sûr à proximité.

Les **objets de forme** désignent l'ensemble des moyens de présentation des informations et d'interaction avec l'utilisateur capable de s'adapter à une cible particulière. Contrairement au cas de l'adaptation d'objets de contenu, les objets de forme ne sont pas filtrés, mais plutôt, d'après , *édités*, *altérés*, *remplacés* ou *reconfigurés* pour permettre une meilleure interaction avec les informations et les outils présentés par le système à l'utilisateur. Cette catégorie d'objet est adaptée par un vaste ensemble d'applications contextuelles. Elles peuvent être appréhendées selon une approche centrée sur la mobilité des systèmes ou sur l'interaction humain-machine, ou alors elles se dédient plus spécifiquement à des cas particuliers de systèmes d'information, comme les SIG ou les logiciels de cartographie numérique.



(a) Adaptation des objets d'interaction

(b) Adaptation des objets de présentation

(c) Adaptation du style de navigation

Figure II.10 – Solutions alternatives pour l'adaptation d'interfaces graphiques ()

proposent une approche d'adaptation générale de l'interaction humain-machine où les objets de l'adaptation sont issus des différents niveaux du modèle d'architecture ARCH (*Interaction*, *Présentation*, *Contrôle du dialogue* et *Noyau fonctionnel* ()) :

- l'**adaptation de l'interaction** conserve la nature des interacteurs mais modifie éventuellement leur style (Fig. II.10b) ;
- l'**adaptation de la présentation** propose différentes alternatives à un ensemble d'interacteurs (Fig. II.10a) ou à un style de navigation (Fig. II.10c) ;
- l'**adaptation du contrôleur de dialogue** modifie l'ordre ou les opérations de dialogue interactif pour la réalisation d'une tâche de l'utilisateur (p. ex. passage d'un

dialogue par “sélection d’objet → commande sur ces objets” à un dialogue de la forme “sélection de commande → choix des objets opérants”) ;
– l’**adaptation du noyau fonctionnel** est appliquée en cas de fortes contraintes contextuelles, par exemple lorsque des fonctionnalités du système ne peuvent plus être assurées ou lorsque les données nécessaires à l’exécution de ces fonctionnalités sont indisponibles.

Les auteurs favorisent dans leurs exemples le *remplacement* (au sens de) d’un composant d’interaction ou d’un ensemble fonctionnel par un autre.

adressent plus particulièrement la problématique de l’adaptation d’un système à un ensemble de plates-formes mobiles, d’usages et de styles d’interaction hétérogènes. Les auteurs distinguent trois sous-ensembles d’objets de l’adaptation qui reflètent les catégories de mesures du contexte d’usage :
– les **modèles de plate-forme** décrivent un ensemble de solutions matérielles pour l’exécution du tiers client ;
– Les **modèles de présentation** décrivent différentes solutions pour l’affichage et la saisie d’informations, en particulier les hiérarchies de menus ou les styles des interacteurs ;
– Les **modèles de tâche** proposent différentes organisations de tâches et sous-tâches pouvant se substituer pour favoriser l’accomplissement des buts des utilisateurs.

Les processus d’adaptation sélectionnent les objets parmi ces trois sous-ensembles en connaissance du contexte d’usage, et plus particulièrement des dimensions de la “Plate-Forme” et de l’“Utilisateur”. Pour les auteurs, cette décomposition est appropriée dans le cas de l’informatique mobile où le choix entre différentes solutions de présentation permet l’adaptation à des plates-formes mobiles dont les capacités varient (notamment les moyens de saisie et d’affichage) (Fig. II.11).

(a) Adaptation à un assistant personnel

(b) Adaptation à un télephone portable

Figure II.11 – Adaptations et exemples de dialogues pour un même système par la sélection de modèles de plate-forme, de tâche et de présentation appropriés ()

s’intéresse au domaine de la cartographie adaptative dans le contexte des systèmes mobiles et ubiquitaires. L’auteur dresse la liste exhaustive des objets de l’adaptation liés à la cartographie numérique et donne pour chaque objet des exemples de ce qui peut être adapté. Les auteurs distinguent trois catégories d’objets :
– la catégorie “**Information géographique**” est équivalente aux *objets de contenu*. Ils sont appelés par l’auteur “éléments cartographiques” ;

- la catégorie "**Interface utilisateur**" concentre les objets d'interaction et les objets fonctionnels permettant d'agir sur la carte numérique ;
- la catégorie "**Visualisation**" répertorie les objets déterminant l'aspect visuel et le rendu de la carte (style, agencement, choix des graphiques, textes).

Par exemple, les éléments cartographiques sont adaptés au niveau de leur présentation (p. ex. encodage vectoriel ou raster, aire géographique affichée, quantité d'informations affichées sur la carte, etc.) ; les objets de *style cartographique* adaptent leur orientation, perspective, échelle, niveau de détail, etc. ; les objets fonctionnels (zoomer, orienter, sélectionner, etc.) peuvent être actifs, cachés ou désactivés.

II.6 Conclusion

L'informatique et les systèmes contextuels s'établissent comme une extension naturelle des systèmes mobiles et de l'informatique nomade pour l'émergence d'une véritable informatique ubiquitaire. Depuis les premières propositions de définition jusqu'aux différentes dimensions constituant l'environnement d'interaction et d'exécution d'un système, ce chapitre a introduit et illustré les fondamentaux de la notion de contexte. Face aux difficultés rencontrées pour établir une définition consensuelle du contexte ou de l'environnement dans le cadre des systèmes d'information, plusieurs travaux de recherche se concentrent sur l'analyse et la classification des éléments qui constituent le contexte. Dans cette perspective, il apparaît que **la notion de contexte se décompose d'une part en un *contexte d'usage*, et d'autre part en un *contexte d'exécution***. Les objets du contexte des dimensions "Utilisateur", "Plate-forme" et "Environnement" constituent l'ensemble mesurable du contexte d'usage. Les objets des dimensions "Système" et "Infrastructure" constituent l'ensemble mesurable du contexte d'exécution.

Différents modèles conceptuels pour l'organisation de ces mesures ont été présentés. Des plus évidents modèles à paires clés-valeurs, aux expressifs modèles ontologiques, chaque forme d'organisation des données contextuelles répond plus ou moins complètement aux propriétés désirables de la sensibilité au contexte dans les systèmes ubiquitaires et mobiles. Sans toutefois éclipser les autres propositions, les modèles à langage de description combinent les avantages d'une simplicité de mise en œuvre, d'un bon support de la composition distribuée et d'un niveau de formalisme satisfaisant. Ces modèles semblent particulièrement appropriés pour une mise en œuvre dans un système d'information mobile et distribué. Quelques architectures pour l'implémentation des modèles conceptuels ont été proposées. Qu'elles centralisent la perception et l'utilisation du contexte au niveau du client ou qu'elles répartissent l'information contextuelle entre plusieurs composants, ces architectures ont avant tout pour objectif la collecte et la diffusion des mesures de l'environnement. La plupart reste sensible aux variations de la topologie des composants et de l'infrastructure réseau rencontrée dans les systèmes mobiles. En quelque sorte, il devient nécessaire de concevoir des plates-formes de perception du contexte elles-mêmes capables de s'adapter à un contexte d'exécution dynamique.

À l'exécution, les mesures de l'environnement sont effectuées pour caractériser les contextes d'usage et d'exécution. Dans le cas du contexte d'usage, ces mesures peuvent être explicites, c'est-à-dire fournies par l'utilisateur ou le concepteur, ou alors implicites, c'est-à-dire déterminées par le système à partir de captures instrumentées de l'environnement. Dans

le cas de la dimension "Utilisateur", les mesures implicites sont inférées à partir de l'étude de l'utilisation du système. Leur analyse statistique détermine et ordonne un ensemble de points d'intérêt des utilisateurs vis-à-vis des données présentées ou des fonctionnalités à disposition. Dans la plupart des prototypes présentés, ces indicateurs d'utilisation proviennent essentiellement de l'analyse des interactions utilisateur-système au niveau local de la plate-forme cliente. Dans un système mobile, trajectoires, localisations, déplacements ou temps d'arrêt des utilisateurs constituent une source supplémentaire non négligeable de mesures pour la détermination des points d'intérêt. Dans le cas du contexte d'exécution, les mesures sont systématiquement implicites : les changements de contexte d'exécution s'effectuent à l'insu de l'utilisateur, au bénéfice de ses déplacements ou des déplacements des autres composants mobiles, ou alors suite à une altération dans la qualité du support réseau. Les mesures du contexte d'exécution servent notamment à prévoir les données et les services accessibles dans un contexte particulier et à adapter le système à ces nouvelles conditions d'exécution.

Deux mesures sont proposées pour quantifier la valeur d'une mesure contextuelle suivant la valeur de l'adaptation qu'elle engendre. **La valeur interactive** donne un critère de sélection des mesures du contexte d'usage. **La valeur utilitaire** quantifie l'apport d'une mesure du contexte d'exécution. Cette dernière n'est pas adossée à un référentiel particulier et dépend essentiellement de la variabilité du contexte d'exécution. Des propositions pour qualifier l'utilité d'un système mobile dans un contexte donné restent à imaginer.

Enfin, **la capacité à réagir aux changements des mesures du contexte définit l'ensemble des logiciels sensibles au contexte**, parmi lesquels les systèmes adaptatifs et les systèmes à recherche d'information contextuelle forment les deux grandes approches pour une utilisation du contexte dans le but d'améliorer la valeur du système pour l'utilisateur. Elles diffèrent essentiellement par les objets auxquels s'appliquent les adaptations. Les systèmes adaptatifs, qui adaptent des objets de forme, héritent d'une approche "Interaction humain-machine" pour laquelle le contexte sert à adapter la présentation et l'interaction, sans toucher au contenu informationnel. A l'opposé, les systèmes à recherche d'information contextuelle adaptent des objets de contenu et soutiennent une approche "Intelligence Artificielle" de l'adaptation pour laquelle l'essentiel est de présenter un contenu approprié à une situation. Selon les domaines d'applications, ces distinctions ont tendance à s'amenuiser. Par exemple, les premiers travaux initiés en cartographie dynamique et contextuelle montrent que dans les applications mobiles le fond rejoint la forme () et l'adaptation d'objets de contenu impacte les objets de forme et vice-versa.

CHAPITRE

III Modélisation spatiale et temporelle du contexte d'exécution

Sommaire

III.1 Illustration d'un système distribué et mobile à l'exécution . . . 73
III.2 Décomposition de l'exécution d'un système mobile et distribué 75
III.2.1 Décomposition temporelle 76
III.2.2 Décomposition physique 79
III.2.3 Décomposition géographique 81
III.2.4 Décomposition contextuelle 90
III.3 Description conceptuelle de l'environnement d'un système mobile et distribué 94
III.3.1 Description physique 95
III.3.2 Description géographique 97
III.3.3 Analyse spatiale 102
III.3.4 Description conceptuelle des contextes d'exécution 105
III.4 Conclusion 108

Les avancées technologiques observées depuis le milieu des années 90 ont favorisé l'émergence de nouveaux modes d'utilisation de l'outil informatique. Dans le domaine des systèmes d'information ; Internet, les technologies de transmission sans fil et la démocratisation des systèmes de positionnement permettent l'utilisation nomade de services, l'émergence des systèmes de géo-positionnement, et la localisation de l'utilisateur dans son environnement. La multiplicité des plates-formes d'utilisation a initié la répartition des systèmes en différents tiers et les architectures logicielles sont désormais distribuées. Un système d'information doit dorénavant faire le lien entre une utilisation en mobilité et une plate-forme distribuée évoluant dans un espace géographique. Ces nouvelles contraintes dans l'architecture d'un système et dans son utilisation demandent une description appropriée, qui caractérise cette versatilité dans les conditions d'exécution.

D'après l'étude des principes de la mobilité des systèmes d'information proposée au chapitre I, un système distribué et mobile se constitue d'un ensemble d'*espaces d'utilisations* (p. ex. domestique, professionnel) liés à un ensemble d'*espaces de ressources* par une infrastructure réseau. La mobilité est introduite dans ce système par le biais de clients sans fil qui définissent des *espaces de mobilité* et par la nature dynamique du support réseau. La version étendue de ce modèle de système mobile, mise en perspective lors de l'étude bibliographique, est proposée et détaillée dans la suite de ce chapitre. Ce modèle considère que l'espace de mobilité n'est pas uniquement dédié aux plates-formes clientes, mais que l'ensemble des composants du système peuvent définir leurs propres espaces de mobilité.

Dans cette perspective, les espaces des utilisations et des ressources, souvent non-mobiles dans les systèmes distribués, deviennent un cas particulier des systèmes mobiles, pour lesquels les composants sont considérés fixes à tout instant de l'exécution. Les systèmes mobiles et les systèmes distribués peuvent être modélisés par une approche conceptuelle commune. Cette approche, présentée dans cette étude, est initiée par la description géographique d'espaces de mobilité générés par l'évolution géographique des composants de la plate-forme.

Ce chapitre décrit les bases d'un modèle conceptuel de système mobile et distribué. Son but est d'aider les développeurs et les concepteurs à caractériser les évolutions possibles de la mobilité du système au cours de sa future exécution. De ces évolutions découlent un ensemble de conditions d'exécution, caractéristiques d'autant de **contextes d'exécution**, qui donnent aux concepteurs un aperçu de la variété des situations que le système rencontrera potentiellement une fois implémenté.

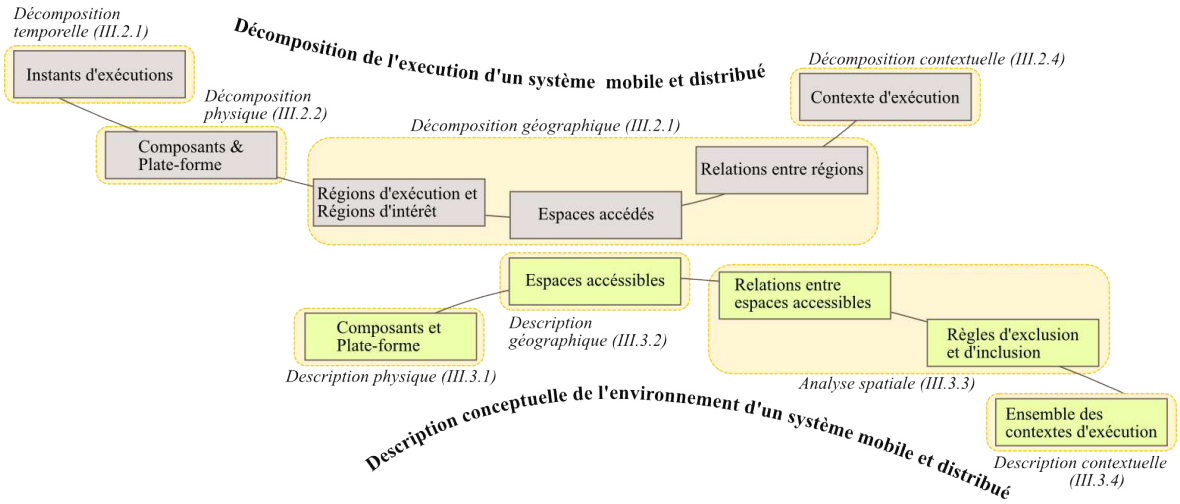


Figure III.1 – Corrélation entre la conception et l'exécution d'un système mobile et distribué

Dans la suite, ce modèle est introduit en deux temps :

– dans un premier temps, le modèle proposé est décrit à l'exécution. Autrement dit, un cas d'exécution en temps discretisé est analysé, et les différentes configurations matérielles, géographiques et contextuelles sont précisés à chaque instant de l'exécution (Fig. III.1 – arc haut). L'analyse de cette exécution permet de formaliser : 1) la fenêtre temporelle d'exécution du système, 2) les composants et la plate-forme, 3) les régions d'exécution et les régions d'intérêt, 4) les espaces accédés par ces régions,

5) les relations entre régions et 6) la notion de contexte d'exécution à un moment donné.
- dans un second temps, l'approche conceptuelle qui mène à l'implémentation d'un système tel que décrit à l'exécution est proposée (Fig. III.1 – arc bas). Les concepteurs décrivent l'ensemble des composants et la plate-forme puis spécifient quel sera l'espace possiblement occupé par les différentes régions (d'intérêt et d'exécution). La relation d'intersection sur ces régions accessibles est définie et appliquée à la génération automatique de l'ensemble des contextes d'exécution. Cet ensemble, une fois épuré des cas géographiquement aberrants, intègre une démarche de modélisation centrée sur l'analyse de la tâche des utilisateurs, présentée au chapitre IV.

Les espaces accédés et accessibles constituent le nœud entre les descriptions de l'exécution et le cadre méthodologique pour la conception. Plus spécifiquement, l'approche conceptuelle qui mène à l'implémentation d'un système tel que décrit à l'exécution repose sur la détermination d'un ensemble d'espaces accessibles. Ces espaces décrivent l'ensemble des positions que pourra occuper une région d'intérêt ou d'exécution. Ils posent les limites de l'espace de fonctionnement "prévu" du système. En pratique, les équipes de conception et d'implémentation doivent identifier les composants du système et étudier leur mobilité pour déterminer son espace accessible. Les relations existantes entre les espaces accessibles "conceptuels" et les espaces accédés "à l'exécution" sont précisées lors de la description des étapes du processus de conception, en seconde partie de chapitre.

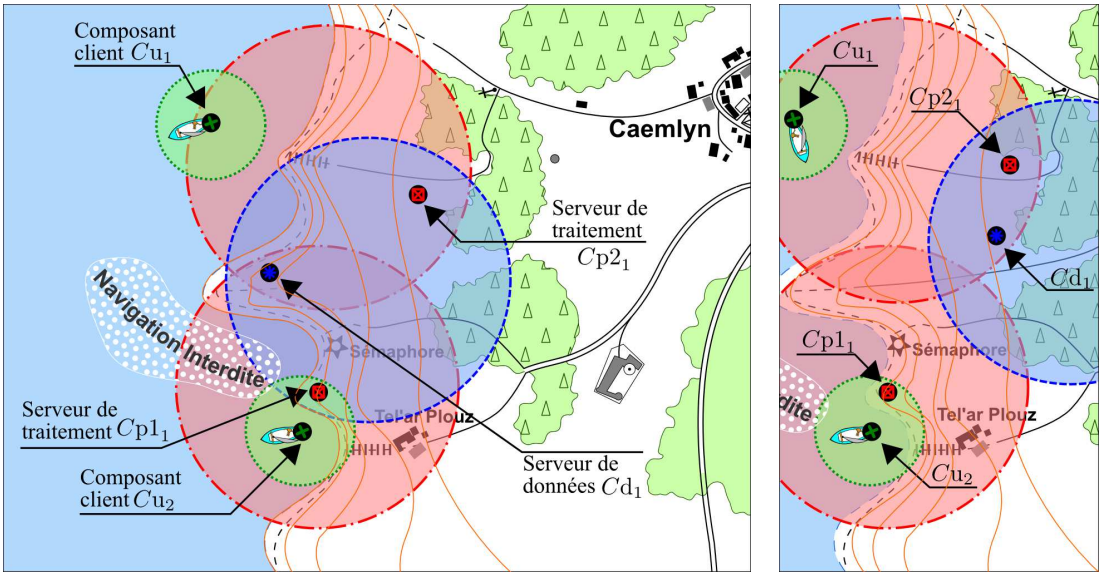
III.1 Illustration d'un système distribué et mobile à l'exécution

Chaque composant matériel de la plate-forme d'un système est situé une position précise dans l'espace à un moment donné. Ces composants possèdent des propriétés particulières en termes de capacités de traitement, de stockage et de transmission d'information. Ces propriétés complètent la description de la localisation des composants et dotent chaque composant d'une couverture spatiale dans laquelle il est capable de diffuser et recevoir des informations. Plus généralement, par l'intermédiaire des composants de sa plate-forme, un système distribué établit sa couverture spatiale dans l'environnement géographique dans lequel il s'exécute.

Dans ce chapitre, un exemple fictif de système distribué constitue le référentiel d'étude de la définition du contexte d'exécution. Ce système s'adresse à des utilisateurs mobiles, ici des marins, et leur fournit un support cartographique qui leur permet de se repérer dans leurs parcours en mer (Fig. III.2). En approchant de la côte, les marins, plaisanciers, pêcheurs, etc., peuvent interroger le système à partir de leur client embarqué. Ils reçoivent des informations sur les conditions locales de mer, sur les dangers potentiels pour la navigation et sur les bateaux dans leur proximité immédiate. Une région de navigation interdite leur est aussi signalée. Au moment où ils en franchissent la frontière, le système réagit en les prévenant qu'il est nécessaire de faire demi-tour. Cette zone de navigation interdite est mobile, elle s'est décalée vers le sud-ouest à t_2 (Fig. III.2b) [1].

1. Ce cas d'une zone interdite mobile peut sembler arbitraire ; dans les faits, des situations similaires sont couramment rencontrées en rade de Brest où les navires civils composent leur route en fonction des sorties

Ce système est distribué sur cinq composants à l'instant t_1 : deux clients (Cu_1 et Cu_2), deux serveurs de traitement ($Cp1_1$ et $Cp2_1$), un serveur de données (Cd_1)[2]. La mobilité du système est rendue possible par l'utilisation de moyens aériens de transport de l'information.



(a) Situation à l'instant t_1 (b) Situation à l'instant t_2

Figure III.2 – Exemple d'une distribution d'un système d'information et de son impact géographique

Dans l'exemple proposé, l'infrastructure du réseau connecte les composants sur un mode pair à pair. Dans ce cas, chaque composant s'entoure d'une région dans laquelle il est capable de diffuser de l'information. Ces régions d'émission découlent directement des capacités sans-fil d'accès au réseau embarquées par les composants (p.ex. cartes réseau Wi-Fi ou Wi-Max). Dans la figure III.2, les couvertures spatiales de ces régions d'émission sont matérialisées par des cercles autour des composants qui les génèrent. Par exemple, les deux composants de traitement diffusent au moins partiellement des données vers le large et les clients peuvent envoyer des informations aux alentours de leurs bateaux. La région de navigation interdite existe pour le système sous forme de données géo-localisées stockées sur le serveur de données. Le déplacement de la couverture spatiale de cette région entre $t1$ et $t2$ représente, du point de vue du système, un ensemble de modifications sur les données de définition de la région.

Une lecture graphique des positions des composants par rapport aux aires de diffusion informe sur les capacités d'échange d'information entre les composants à un moment donné. Dans l'exemple proposé, à t_1, le composant serveur de données Cd_1, qui est dans la région d'émission du premier serveur de traitement $Cp1_1$, peut recevoir des informations en provenance de ce composant. De manière symétrique, ce premier composant de traitement,

des sous-marins et des bâtiments militaires dont l'approche est strictement réglementée. Une autre situation d'espace maritime limité peut être lié à la gestion de situations de crise. Notamment, en cas de marée noire, les nappes dérivantes constituent un exemple de zone de navigation interdite mobile

2. Les notations de la forme Cnom_role$_{indice}$ signifient "*indice*^{ième} $\underline{C}$omposant de rôle *nom_role*". Par exemple $Cp2_1$ peut être lu "premier composant de rôle p2", avec p2 signifiant "$\underline{p}$rocessing 2". Ces conventions de nommage et les natures des rôles des composants sont précisés au paragraphe III.2.2.

lui-même localisé dans la région d'émission du serveur de données, peut recevoir des données de ce dernier. Les communications entre $C\mathrm{p1}_1$ et $C\mathrm{d}_1$ sont dans ce cas bi-directionnelles. D'autres communications bi-directionnelles (symbole "$\leftrightarrow$") ou mono-directionnelles (symboles "$\rightarrow$" ou "$\leftarrow$") peuvent s'établir à t_1 :

$$\text{Client 2} \leftrightarrow \text{Traitement 1} \leftrightarrow \text{Données 1} \leftrightarrow \text{Traitement 2} \rightarrow \text{Client 1}$$

L'utilisation d'infrastructures réseau sans fil permet à chaque composant de devenir mobile et de changer de position au cours du temps. Dans la littérature présentée au chapitre I, de nombreuses expérimentations se focalisent sur la mobilité du client. Le modèle proposé considère chaque composant sur un pied d'égalité. Dans l'exemple du système de suivi côtier, d'une part, les clients sont mobiles et se déplacent en même temps que les bateaux des utilisateurs. D'autre part, le serveur de données est mobile. Ces trois composants voient leur localisation varier. Les intersections entre les composants et les aires d'émission changent au cours du temps. À t_2 (Fig. III.2b), les transmissions d'information via l'infrastructure réseau se scindent en deux chaînes de liaisons :

$$\text{Client 2} \leftrightarrow \text{Traitement 1} \qquad \text{Données 1} \leftrightarrow \text{Traitement 2} \rightarrow \text{Client 1}$$

Dans cet exemple, s'il est admis que le fonctionnement optimal du système repose sur la continuité des communications bi-directionnelles entre les serveurs de données et les serveurs de traitement, alors, le système subit entre t_1 et t_2 une dégradation de ses conditions d'exécution du fait du déplacement du composant serveur de données $C\mathrm{d}_1$. Intuitivement, les situations à t_1 et à t_2 constituent deux exemples distincts de contextes d'exécution, qui, s'il ont été prévus et pris en compte à la conception du système, déclenchent des actions d'adaptation *a minima* au niveau des clients ou, *a maxima*, sur l'ensemble des composants.

III.2 Décomposition de l'exécution d'un système mobile et distribué

Cette section présente un modèle formel générique appliqué à la description de systèmes mobiles et distribués. Par rapport aux descriptions de systèmes étudiés aux chapitres I et II, ce modèle formel est centré sur une décomposition géographique de l'espace d'exécution et sur une mobilité généralisée à l'ensemble des constituants du système. Les relations inter-composants sont exprimées à la lumière de cette décomposition géographique.

Dans la suite, l'ensemble des instants de l'exécution est défini par la discrétisation temporelle de l'intervalle entre l'initialisation et l'arrêt du système. Ces instants de l'exécution étiquettent chaque composant du système pour constituer la plate-forme à un moment donné. Les composants de la plate-forme se dotent de régions d'exécution, qui, associées à des régions d'intérêt, désignent la couverture spatiale du système. L'évolution de cette couverture spatiale tout au long de l'exécution détermine différents espaces accessibles à la mobilité des composants, des régions d'intérêt et des régions d'exécution. La couverture spatiale des espaces accessibles matérialise les frontières d'utilisation du système. Des relations entre régions

sont définies par rapport aux positions relatives et aux intersections des espaces accessibles. L'agrégation de ces relations entre régions forme le contexte d'exécution à un moment donné.

III.2.1 Décomposition temporelle

Pour décrire les évolutions d'un système qui s'exécute dans un environnement variable, une discrétisation du temps d'exécution du système est effectuée. La dynamique de l'environnement, les changements dans la plate-forme d'exécution, et les évolutions consécutives aux niveaux des outils et données offerts aux utilisateurs sont effectués à des moments précis de l'exécution. Dans la suite, t_x, pris sur l'ensemble des réels positifs, désigne un **instant de l'exécution** du système qui fait partie de l'ensemble des instants d'exécution T :

$$T = \{t_0, t_1, t_2, \ldots, t_x, \ldots, t_{end}\}$$

L'indice x d'un instant t_x est un entier naturel positif ($x \in \mathbb{N}_+$). Il permet de désigner le $x^{\text{ième}}$ instant observé de l'exécution du système.

Comme $T \subseteq \mathbb{R}_+$, ce sous-ensemble des réels hérite de la relation d'ordre total sur ses éléments. Par convention, et quels que soient t_a et t_b pris parmi T, si l'indice entier a est inférieur à l'indice b, alors l'instant t_a précède l'instant t_b et la relation $t_a \leq t_b$ est vérifiée dans $\mathbb{R}$. Plus formellement [3] :

$$\forall\ t_a, t_b \in T\ :\ \ (a \leq b) \in \mathbb{N}_+\ \Rightarrow\ (t_a \leq t_b) \in \mathbb{R}_+$$

Instants d'initialisation et d'arrêt

Dans la suite, t_0 représente dans T l'**instant de l'initialisation**, à partir duquel le système est disponible à l'utilisation. Dans le cas d'un système distribué reposant sur un ensemble de composants, l'initialisation correspond à la mise en service du premier composant matériel (c-à-d. mise sous tension, établissement de la connectivité réseau et chargement du logiciel du système). Dans ce cas, t_0 est l'instant auquel au moins l'un des composants matériels supportant le système est disponible à l'utilisation. A l'inverse, t_{end} désigne la borne supérieure de l'intervalle temporel d'exécution du système. Il référence l'**instant de mesure terminal** avant l'arrêt du dernier des composants matériels du système. L'indice *end* de t_{end} vérifie :

$$\forall\ t_x, t_{end} \in T\ :\ \ x \leq end \wedge t_x \leq t_{end}$$

Après l'instant t_{end}, la partie logicielle du système d'information qui s'exécute sur le dernier composant actif est déchargée. Eventuellement, ce dernier composant est déconnecté du réseau et mis hors tension.

Par convention l'instant de l'initialisation est choisi nul (c-à-d. $t_0 = 0$). Les autres instants d'exécution sont construits en référence à t_0. Par exemple, si la description des évolutions du système est échantillonnée à intervalles réguliers selon un pas de mesure α, les

3. Dans la suite, la notation "$\forall\ e_x \in E\ :\ p_y$" est adopté pour exprimer des assertions de quantification universelle. Cette proposition logique se lit "Quel que soit un élément e_x de l'ensemble E, e_x vérifie la propriété p_y". Les assertions de quantification universelle peuvent être *vraies* ou *fausses*

éléments de T prennent pour valeurs $t_0 = 0,\ t_1 = t_0 + \alpha,\ t_2 = t_0 + 2 \times \alpha$, etc. En généralisant, la valeur réelle d'un instant t_x est :

$$\forall\ t_x \in T\ :\ t_x = t_0 + x \times \alpha = x \times \alpha$$

et deux éléments t_a et t_b de T sont liés par la relation :

$$\forall\ t_a, t_b \in T\ :\ t_a = t_b - \alpha \times (b - a)$$

Comme l'arrêt du système n'est pas calé sur le rythme arbitraire des mesures de périodicité α, il n'est pas possible de connaître avec précision l'instant de l'arrêt. Il existe un décalage entre le dernier instant de l'exécution t_{end} qui reste en phase avec t_0 et l'arrêt effectif du logiciel qui a lieu dans l'intervalle $[t_{end}, t_{end} + \alpha[$.

Cycle de mesures contextuelles

Les changements dans l'environnement sont caractérisés au niveau du système lors de mesures effectuées à chaque instant d'exécution avant l'arrêt du système. L'initialisation du système à t_0 marque aussi le moment de la première mesure des conditions d'environnement. Dans la suite de l'exécution, celles-ci s'enchaînent à la même fréquence que les instants d'exécution.

Comme chaque instant d'exécution est lié à une mesure de l'environnement, lorsqu'un changement de situation est mesuré à t_x, alors cet évènement s'est effectivement produit lors de l'intervalle $[t_{x-1}, t_x[$. En conséquence, les systèmes tels qu'implémentés à partir du modèle décrit dans cette étude réagissent en *quasi temps réel* à des modifications de l'environnement. Il est toutefois possible d'approcher la modélisation d'un système temps réel en faisant tendre le pas de mesure α vers 0.

La suite des instant d'exécution constitue une décomposition arbitraire du temps du système qui n'est pas corrélée aux évènements de l'environnement. Les instants exacts des évènements de l'environnement sont repérés relativement à un instant donné. Notamment, lors de la mesure de l'état du système à des intervalles réguliers, les incréments ε_x^+ et ε_x^- sont des petites quantités réelles, qui, ajoutées ou retirées à la valeur d'un instant t_x, permettent de désigner des évènements ayant lieu respectivement juste avant ou juste après t_x. Plus particulièrement, relativement à l'instant $t_x \in T$, l'incrément noté ε_x^- désigne le moment $t_x - \varepsilon_x^-$ exact auquel a lieu un changement dans l'environnement du système. L'incrément noté ε_x^+ désigne le moment $t_x + \varepsilon_x^+$ exact à partir duquel l'ensemble des modifications engendrées après la mesure d'un changement contextuel ont été répercutées au niveau du système d'information. Par exemple, lorsqu'un utilisateur s'éloigne d'une zone de service et perd la connectivité avec une borne d'accès à $t_x - \varepsilon_x^-$, le système (dans ce cas, la plate-forme de l'utilisateur) détecte ce changement à t_x et effectue les adaptations contextuelles (p. ex. limitation des services) dans l'intervalle $[t_x, t_x + \varepsilon_x^+]$. Un ensemble d'**opérations d'adaptation** sont effectuées dans cet intervalle entre la mesure contextuelle et l'état adapté du système. L'intervalle $[t_x - \varepsilon_x^-, t_x + \varepsilon_x^+]$ entre le changement des conditions d'environnement jusqu'à la fin des adaptations est appelé "**cycle d'adaptation**" du système. L'enchainement des cycles d'adaptation décrit un fonctionnement synchrone du système. L'ensemble des composants de

la plate-forme est calé sur un même pas de mesure. Le premier composant du système, qui initie le cycle de d'exécution, constitue la référence initiale du pas de mesure. cette référence est transmisse de proche en proche lors de l'adjonction de nouveaux composants à la plate-forme. En pratique, ce fonctionnement synchrone est difficile à maintenir pour un système réparti. Chaque composant est source d'asynchronisme et il est nécessaire qu'un tiers assume le rôle de référence temporelle et pallie aux décalages des pas de mesure. Dans le cas d'une plate-forme dynamique, ce rôle d'horloge du système doit pouvoir être transmis d'un composant actif à un autre, et chaque composant doit pouvoir se synchroniser à l'horloge. Afin de limiter leur complexité, les systèmes décrits ci-après fonctionnent sur un mode "quasi-synchrone" où le pas de mesure est transmis une fois pour toute aux composants lorsqu'ils se greffent à la plate-forme initiale.

Pour éviter au maximum le chevauchement temporel des cycles d'adaptation, et, de façon plus impérative, pour distinguer les opérations d'adaptation les unes des autres, les valeurs des incréments ε_x^+ et ε_x^- sont nécessairement inférieures à la valeur du pas de mesure α[4]. Dans ce cas, les incréments désignent un instant avant ou après une mesure dans les limites de l'intervalle temporel entre les instants d'exécution qui précèdent et qui suivent cette mesure. Plus formellement, quelles que soient les valeurs des incréments ε_x^- et ε_x^+, $t_x - \varepsilon_x^-$ et $t_x + \varepsilon_x^+$ vérifient :

$$\forall\ t_x \in T - t_0\ :\ t_x - \varepsilon_x^- \in [t_{x-1}, t_x[\qquad \forall\ t_x \in T - t_{end}\ :\ t_x + \varepsilon_x^+ \in [t_x, t_{x+1}[$$

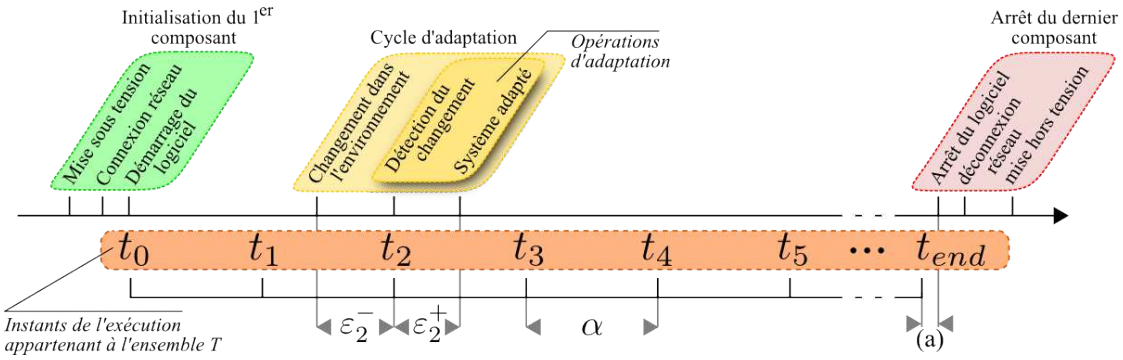


Figure III.3 – Exemple de cycle de vie du système

Cycle de vie du système

La figure III.3 reprend l'ensemble des notions définissant le cycle de vie d'un système adaptatif à un seul composant. L'extinction et l'allumage de celui-ci bornent l'intervalle des instants d'exécution, modulo le décalage entre le dernier instant de mesure et l'arrêt effectif (Fig. III.3(a)). Dans cet exemple, une différence contextuelle apparaît à $t_2 - \varepsilon_2^-$. Elle initie un cycle d'adaptation et est mesurée à t_2. Ces mesures entraînent un ensemble d'opérations d'adaptation entre t_2 et le moment où le système est adapté à $t_2 + \varepsilon_2^+$. Les mesures de

4. Lorsque ε_x^- est supérieur au pas de mesure, les changements de l'environnement qui lui sont associés ont déjà été mesurés et pris en compte à l'instant t_{x-1}. De la même manière, avoir $\varepsilon_x^+ > \alpha$ risque de provoquer le chevauchement des opérations d'adaptation si les différences contextuelles se succèdent au rythme des mesures.

l'environnement effectuées aux autres instants de l'exécution (t_1, t_3, etc.) ne permettent pas de distinguer de changements contextuels et n'initient pas de nouveaux cycles d'adaptation.

III.2.2 Décomposition physique

Le modèle conceptuel proposé s'applique à la description d'un système distribué construit sur des briques matérielles et logicielles appelées **composants du système**. Un premier niveau de distribution d'un système d'information consiste à séparer la plate-forme en plusieurs composants matériels. Cette distribution est représentable physiquement et les composants se distinguent les uns des autres. Pour chacun d'entre eux, il est possible de le transporter, de le mettre sous tension, de l'éteindre, d'effectuer des opérations logicielles, etc. L'association de l'ensemble des composants constitue la **plate-forme du système** à un instant donné de son exécution. Formellement, la fonction $Platform(t_x)$ retourne à un instant t_x l'ensemble des composants du système, désignés par un identifiant de la forme $cmp_{<indice>}$:

$$Platform(t_x) = \{cmp_1, cmp_2, \ldots, cmp_y, \ldots, cmp_n\}$$

Notion de rôle d'un composant

Un système d'information interactif, qui stocke, traite et transforme de l'information à l'initiative et à destination d'utilisateurs, distingue en général les niveaux des données, des traitements et du client. Du point de vue de l'architecture adoptée, un système d'information en tiers répartis implémente ces niveaux sur l'ensemble de ses composants physiques. D'après la décomposition fonctionnelle proposée par , **les composants d'un système distribué adoptent généralement l'une des fonctions suivantes** :

- les **composants clients**, constituent la couche d'interaction avec l'utilisateur. Ces composants ont la charge de la présentation des données du système en réponse à une requête de l'utilisateur ainsi que de l'interprétation des entrées interactives de l'utilisateur ;
- les composants de gestion des données, aussi appelés **composants serveurs de données** importent et exportent les sous-ensembles d'information du système nécessaires à l'exécution de tâches par les utilisateurs. Ces composants garantissent le stockage, la sauvegarde et l'accès aux informations, organisées en structures de données informatiques ;
- les **composants serveurs de traitement** hébergent les procédures d'analyse et de rendu des informations du système. Ces composants modifient les informations du système en réponse aux requêtes des utilisateurs et illustrent les résultats de ces requêtes en retournant des vues sur les données concernées.

Ces "familles" de fonctions associées aux composants sont souvent spécialisées à l'exécution. Par exemple, pour un système donné, il peut exister différents types de traitements, implémentés sur deux ensembles distincts de composants. De la même manière, il peut être souhaitable de distinguer différents ensembles de clients en fonction de catégories d'utilisateurs (p. ex. utilisateurs piétons contre utilisateurs marins au large). Dans le modèle proposé, **les fonctions spécialisées du système constituent un ensemble de rôles**. En généralisant cette notion de rôle, le modèle de description à l'exécution considère qu'**un**

système distribué associe un certain nombre de rôles à ses composants et que **plusieurs composants peuvent tenir un même rôle**. Chaque rôle est en général assimilable à l'une des familles fonctionnelles, mais leur catégorisation précise dépend de la nature du système implémenté et est établie lors de sa conception.

Dans la suite, "$ROLES$" désigne l'ensemble $\{\text{roleA}, \text{roleB}, \ldots, \text{roleN}\}$ des rôles associés aux composants du système et la fonction $Role(cmp_y)$ retourne le rôle du composant cmp_y. L'ensemble des composants de la plate-forme qui tiennent le rôle "role $\in ROLES$" à un instant d'exécution t_x est retourné par la fonction $Cmps(\text{role}, t_x)$ [5] :

$$\forall\ \text{role} \in ROLES,\ t_x \in T :\ Cmps(\text{role}, t_x) = \{cmp_y \in Platform(t_x) \mid Role(cmp_y) = \text{role}\}$$

L'union des ensembles de composants par rôles fournit une autre façon de formaliser la notion de plate-forme à un moment donné de l'exécution :

$$\forall\ t_x \in T :\ Platform(t_x) = \bigcup_{\substack{\text{role} \\ \in ROLES}} Cmps(\text{role}, t_x)$$

Notation : par convention, le $i^{\text{ième}}$ composant de rôle "role $\in ROLES$" est noté $C\text{role}_i$, avec "C" pour "Composant". De cette manière, l'ensemble des composants d'un rôle donné peut s'écrire :

$$\forall\ \text{role} \in ROLES,\ t_x \in T :\ Cmps(\text{role}, t_x) = \{C\text{role}_1, C\text{role}_2, \ldots, C\text{role}_i, \ldots, C\text{role}_n\}$$

Le système de suivi côtier distingue deux rôles de traitements et considère que les fonctionnalités offertes aux utilisateurs en mer restent les mêmes : les clients sont tous associés à un même rôle. Dans ce cas, les rôles des plates-formes retenus sont $ROLES = \{\text{u}, \text{p1}, \text{p2}, \text{d}\}$, avec "d" pour "data", "u" pour "user", "p1" pour "processing 1" et "p2" pour "processing 2". La plate-forme du système, distribuée sur cinq composants (deux clients, deux serveurs de traitement, un serveur de données) à l'instant d'exécution t_1, et sa décomposition par rôles sont retournées par :

$$Platform(t_1) = \{C\text{u}_1, C\text{u}_2, C\text{p1}_1, C\text{p2}_1, C\text{d}_1\} \qquad Cmps(\text{d}, t_1) = \{C\text{d}_1\}$$
$$Cmps(\text{p1}, t_1) = \{C\text{p1}_1\} \qquad Cmps(\text{p2}, t_1) = \{C\text{p2}_1\}$$
$$Cmps(\text{u}, t_1) = \{C\text{u}_1, C\text{u}_2\}$$

Dans un système multi-composant, chaque tiers subit l'influence de son environnement. Les biais d'exécution des systèmes distribués énoncés par démontrent que le fonctionnement d'un tiers est rarement sûr, et que les occasions de pannes, de coupures réseau, etc., sont nombreuses. Ces aléas de fonctionnement sont représentés dans le modèle de description proposé par des variations de l'ensemble des composants au cours du temps. La référence temporelle t_x passée en paramètre de la fonction $Platform(t_x)$ et des décompositions par rôles permettent de connaître quels sont les composants actifs du système à l'instant d'exécution

5. Dans la suite, la notation "$\{e_x \in E \mid p_y\}$" est adoptée pour *définir un ensemble en compréhension*. Cette définition se lit "Ensemble des éléments e_x de l'ensemble E vérifiant la propriété p_y".

t_x. Si un évènement à l'exécution fait sortir un composant de son fonctionnement normal, alors celui-ci n'est plus considéré parmi l'ensemble des constituants actifs.

Avec un temps d'exécution discrétisé, il n'est possible de connaître l'évolution de la plate-forme qu'aux instants précis de T. Par exemple, l'initialisation du système à l'instant t_0 est déclenchée au moment du démarrage logiciel du premier composant. La plate-forme à t_0 contiendra uniquement ce premier élément (c-à-d. quels que soient le système ou le nombre et le type de composants, $|Platform(t_0)| = 1$). Potentiellement, d'autres composants peuvent s'initialiser et devenir actifs entre t_0 et la mesure suivante à t_1. Dans ce cas, chaque initialisation de composant supplémentaire est considérée comme une modification de l'environnement du système, mesurée et prise en compte lors du cycle d'adaptation suivant. Le système est en fonctionnement entre t_0 et t_{end}. Du point de vue de la plate-forme, cela signifie qu'à chaque instant de T il existe au moins un composant du système actif durant l'exécution :

$$\forall\ t_x \in T\ :\ Platform(t_x) \neq \emptyset$$

III.2.3 Décomposition géographique

La plate-forme du système distribué a été définie comme un ensemble de composants variant au cours du temps. Ces évolutions temporelles sont matérialisées à des moments précis de l'exécution auxquels des mesures de l'environnement du système sont effectuées. Cette première décomposition permet de décrire des systèmes distribués supportant l'initialisation ou l'arrêt de composants lors de l'exécution.

Pour introduire la mobilité du système dans le modèle proposé, ces composants sont localisés dans l'espace physique. Un ensemble de **régions d'exécution** marquent leur position et l'étendue de leur couverture opérationnelle. Étant associées aux composants, ces régions d'exécution évoluent dans le temps et l'espace en accompagnement des variations de la plate-forme. Dès lors, l'**environnement d'exécution** du système, rassemblant les régions d'exécution à un moment donné, constitue le miroir géographique de la plate-forme et des composants.

En complément des régions d'exécution, les **régions d'intérêt** sont définies par rapport à des informations sur des lieux ou des évènements de l'environnement du système. Elles permettent d'affiner les cas de variations contextuelles en exprimant différentes situations d'usage, tandis que les régions d'exécution servent à la prévision des variations du matériel et du transport d'information. Ces régions, rassemblées dans l'**environnement d'intérêt** du système, s'ajoutent aux régions d'exécution pour constituer son **espace géographique**.

A un instant donné de l'exécution, les relations entre les composants de la plate-forme et les régions de l'espace géographique définissent le contexte d'exécution et permettent la mesure de trois sous-ensembles (Chap II.2.2 : Identité/Système/Infrastructure)

- l'**identité** des composants est connue par leur appartenance à l'un des rôles du système. Généralement, ces rôles sont assimilables à un comportement particulier entre tiers clients, serveur de traitement ou serveur de données ;
- la topologie du **système** est déduite des relations entre les régions de l'espace géographique et entre les constituants de plate-forme. Elle peut être déterminée par l'association entre les régions d'exécution et les composants ;

– l'état de **l'infrastructure** est exprimé dans les relations entre régions et entre composants et régions. Par exemple, les composants qui génèrent deux régions distinctes n'échangent pas directement d'information.

Régions d'exécution

Le fonctionnement du système distribué se fonde sur l'échange d'informations entre les différents tiers, depuis la source des données, jusqu'à l'interface utilisateur. En considérant l'espace géographique dans lequel un système s'exécute, il est possible de doter les composants du système de régions représentant les espaces dans lesquels chaque composant agit et est capable de diffuser et de recevoir des informations. Dans le modèle proposé, tout composant de la plate-forme génère une et une seule région d'exécution et toute région d'exécution est générée par un composant unique. Il est toutefois admis que la couverture spatiale d'une région puisse être nulle ou composée d'aires disjointes à un moment donné.

Étant donné un composant cmp_y du système, la **région d'exécution** "reg_y" retournée par la fonction $Region(cmp_y)$ représente l'espace dans lequel ce composant agit au sein du système. La région reg_y est dite "générée" par le composant cmp_y et le composant cmp_y est appellé "générateur" de la région reg_y. Une région d'exécution reg_y fait partie de l'ensemble $EnvExec(t_x)$ des régions générées par les composants actifs de la plate-forme. Cet ensemble définit l'**environnement d'exécution** du système à l'instant t_x. Formellement, à tout composant de la plate-forme à t_x s'associe une région d'exécution de $EnvExec(t_x)$:

$$\begin{aligned}\forall\ t_x \in T\ :\ EnvExec(t_x) &= \bigcup_{\substack{\text{cmp}\\ \in Platform(t_x)}} Region(cmp)\\ &= \left\{\begin{matrix} Region(cmp_1), Region(cmp_2), \ldots,\\ Region(cmp_k), \ldots, Region(cmp_n)\end{matrix}\right\}\\ &= \{reg_1, reg_2, \ldots, reg_k, \ldots, reg_n\}\end{aligned}$$

Quelle que soit une région reg_y prise parmi $EnvExec(t_x)$, il est possible de connaître le composant qui la génère par un appel à la fonction $Component(reg_y)$. Cette fonction constitue l'application réciproque de la fonction $Region(cmp_y)$ [6].

Notation : par convention, le $i^{\text{ième}}$ composant de rôle "role $\in ROLES$", noté $C\text{role}_i$, génère une région d'exécution notée $R\text{role}_i$, avec R pour "Région". L'ensemble des régions générées par des composants d'un rôle donné sont retournés par la fonction $EnvRole(\text{role}, t_x)$:

$$\begin{aligned}\forall\ \text{role} \in ROLES,\ t_x \in T\ :\ EnvRole(\text{role}, t_x) &= \bigcup_{\substack{cmp\\ \in Cmps(\text{role}, t_x)}} Region(cmp)\\ &= \left\{R\text{role}_1, R\text{role}_2, \ldots, R\text{role}_k, \ldots, R\text{role}_n\right\}\end{aligned}$$

6. Autrement dit : $Region\big(Component(reg_y)\big) = reg_y$ et $Component\big(Region(cmp_y)\big) = cmp_y$.

L'union des ensembles de familles de régions, repérées par le rôle de leur composant générateur, fournit une autre façon de formaliser la notion d'environnement d'exécution :

$$\forall\ t_x \in T\ :\ EnvExec(t_x) = \bigcup_{\substack{\text{role} \\ \in ROLES}} EnvRole(\text{role}, t_x)$$

Les trois familles de rôles d'exécution (clients, serveurs de traitement et serveurs de données) décrivent un environnement dans lequel les composants génèrent des régions d'exécution qui sont assimilables à l'une des familles suivantes :

- les **régions de traitement** désignent les espaces de l'environnement où les procédures de transformations et d'analyses sur des données sont disponibles ;
- les **régions de diffusion des données** désignent les espaces de l'environnement où les données du système sont stockées et disponibles ;
- les **régions des utilisateurs** désignent les espaces où le résultat de tout ou partie des traitements effectués par le système est utilisé.

Comme les régions d'exécution sont définies par l'ensemble des composants de la plate-forme, il existe une dépendance de l'environnement d'exécution par rapport à la plate-forme. En particulier, si la plate-forme est vide, l'environnement d'exécution est inexistant. Lorsqu'un composant est soustrait à la plate-forme, alors la région qu'il génère disparaît de l'environnement d'exécution. A l'inverse, si une région d'exécution disparaît de l'ensemble de l'environnement d'exécution, alors le composant qui la supporte ne fait plus partie de la plate-forme. Cette double dépendance s'écrit plus formellement :

$$\forall\ t_x, t_y \in T, cmp_k \in Platform(t_y)\ :\ cmp_k \notin Platform(t_x) \Rightarrow Region(cmp_k) \notin EnvExec(t_x)$$
$$\forall\ t_x, t_y \in T, reg_l \in EnvExec(t_y)\ :\ reg_l \notin EnvExec(t_x) \Rightarrow Component(reg_l) \notin Platform(t_x)$$

Ces relations impliquent un nombre de régions d'exécution identique au nombre de composants quel que soit l'instant de l'exécution [7] :

$$\forall\ t_x \in T\ :\ |Platform(t_x)| = |EnvExec(t_x)|$$

À un instant donné $t_x \in T$, la **couverture spatiale** d'une région $reg_y \in EnvExec(t_x)$, retournée par $Cover(reg_y, t_x)$, représente l'ensemble des localisations à partir desquelles il est possible de recevoir des informations en provenance du composant $Component(reg_y)$, générateur de la région reg_y. La dépendance existentielle entre régions et composants n'implique pas pour autant que la couverture spatiale d'une région de l'environnement d'exécution soit représentable dans l'espace géographique. Plus spécifiquement, la couverture d'une région est vide si un composant est actif mais ne propose aucun service, s'il ne dispose que d'un support réseau limité ou encore s'il n'échange pas d'information avec un autre composant.

7. Les sous-ensembles de l'environnement et de la plate-forme héritent de cette équivalence cardinale. $\forall\ t_x \in T,\ \text{role} \in ROLES\ :\ |Cmps(\text{role}, t_x)| = |EnvRole(\text{role}, t_x)|$.

Dans le cas du système de suivi côtier à t_1, les cinq composants génèrent l'ensemble de régions d'exécution et les décompositions par rôles suivants :

$$EnvExec(t_1) = \{R\mathrm{u}_1, R\mathrm{d}_1, R\mathrm{p1}_1, R\mathrm{u}_2, R\mathrm{p2}_1\} \qquad EnvRole(\mathrm{d}, t_1) = \{R\mathrm{d}_1\}$$
$$EnvRole(\mathrm{p1}, t_1) = \{R\mathrm{p1}_1\} \qquad EnvRole(\mathrm{u}, t_1) = \{R\mathrm{u}_1, R\mathrm{u}_2\}$$
$$EnvRole(\mathrm{p2}, t_1) = \{R\mathrm{p2}_1\}$$

La figure III.2a représente les couvertures spatiales des régions d'exécution à t_1. Les régions de l'espace dans lesquelles données et traitements sont disponibles à un instant donné découlent des capacités de diffusion de données des composants de la plate-forme. Dans l'exemple du système de suivi côtier, les couvertures spatiales des régions d'exécution correspondent aux aires d'émission des composants générateurs.

Régions d'intérêt

Une large proportion des systèmes contextuels et adaptatifs présentés au chapitre II proposent de prendre en compte la localisation de l'utilisateur dans la construction du contexte. Ces systèmes réagissent en fait par des adaptations en fonction de la position ou des déplacements des utilisateurs, représentés du point de vue du système par les composants clients. L'ensemble de ces localisations où se produisent des adaptations du système, mais qui ne sont pas associées à la perte ou au gain de capacités de transmission d'information par l'entrée ou la sortie d'une région d'exécution, définissent des régions d'intérêt. Les relations entre les composants, les utilisateurs et cet ensemble de régions d'intérêt sont jugées importantes pour l'adaptation.

Contrairement aux régions d'exécution, les régions d'intérêt sont désignées à partir de données du système et non pas à partir d'aires de diffusion. La représentation des régions d'exécution offre une perspective de description centrée sur les aspects "système et communication" qui s'avère appropriée pour la modélisation du contexte système. Les régions d'intérêt constituent un outil d'analyse de l'exécution d'un système supplémentaire. Elles permettent de distinguer différents contextes d'usages à l'exécution et d'identifier les processus d'adaptation qu'ils déclenchent.

Dans l'exemple du système de suivi côtier (Fig. III.2), la zone de navigation interdite est une région d'intérêt du système. Les données qui définissent sa localisation géographique sont disponibles au niveau du composant de gestion des données (c-à-d. $C\mathrm{d}_1$). Cette région est utilisée pour caractériser différents contextes d'utilisation du système. En particulier, le système propose un comportement différent selon que les utilisateurs sont à l'intérieur de la région ou à l'extérieur. Lorsqu'un composant client (donc un utilisateur) franchit la frontière de cette région d'intérêt "Navigation Interdite", le système apporte une réponse fonctionnelle appropriée. Dans cette situation, le comportement le plus immédiat serait de guider l'utilisateur vers l'extérieur de la zone interdite. L'idée d'une région d'intérêt pour discriminer différents comportements du système est présente dans de nombreux travaux d'informatique ambiante (Chap. II.5.1). Dans cette optique, la salle café du Xerox Park constitue un exemple de région d'intérêt définie à partir des données enregistrées dans le système ParkTab et qui permet d'adapter le système selon que des utilisateurs se situent à l'extérieur ou à l'intérieur de la salle café ().

Plus généralement, une région d'intérêt notée reg_y fait partie de l'ensemble $EnvInt(t_x)$ des n régions d'intérêt du système à l'instant t_x :

$$\forall\ t_x \in T\ :\ EnvInt(t_x) = \{reg_1, reg_2, \ldots, reg_k, \ldots, reg_n\}$$

L'élément de la plate-forme qui héberge les données de définition d'une région d'intérêt est appelé "composant hôte" (host). Cet hôte $cmp_y \in Platform(t_x)$ d'une région d'intérêt $reg_z \in EnvInt$ est retourné par la fonction $Host(reg_z)$. Potentiellement un composant de la plate-forme peut servir d'hôte à plusieurs régions d'intérêt. L'ensemble des régions d'intérêt dont un composant cmp_y est l'hôte est retourné par la fonction $IntSet(cmp_y)$. Une définition plus formelle de la fonction $EnvInt(t_x)$ retourne l'ensemble des régions d'intérêt dont les hôtes font partie de la plate-forme à t_x :

$$\forall\ t_x \in T\ :\ EnvInt(t_x) = \bigcup_{\substack{cmp_y \in \\ Platform(t_x)}} IntSet(cmp_y)$$

Tout comme les régions d'exécution, une région d'intérêt évolue dans le temps et dans l'espace et est associée à une couverture spatiale à tout instant de l'exécution. En considérant une région d'intérêt $reg_y \in EnvInt(t_x)$, la couverture spatiale $Cover(reg_y, t_x)$ correspond à l'ensemble des localisations de l'environnement désigné par les données de définition de reg_y à t_x. Cette couverture est susceptible de varier dans le temps lorsque les données qui définissent reg_y sont modifiées. De la même manière, si entre t_{x-1} et t_x, les informations de définition d'une région d'intérêt sont supprimées du composant qui les hébergeait ou si ce composant disparaît de la plate-forme, alors cette région d'intérêt n'est plus identifiable et ne fera plus partie de $EnvInt(t_x)$. En conséquence, une région d'intérêt existe lorsque ses données de définition sont disponibles sur un composant hôte de la plate-forme. Plus formellement :

$$\forall\ t_x \in T,\ reg_y \in EnvInt(t_x)\ :\ Host(reg_y) \in Platform(t_x)$$

Comme dans le cas des régions d'exécution pour lesquelles une série de rôles sont définis, il est souhaitable de pouvoir distinguer plusieurs familles ou *types* de régions d'intérêt dans un même système. Par exemple, plusieurs zones de navigation interdites sont regroupées dans une même famille de régions d'intérêt, alors que des zones repérant les frontières de villages pourraient constituer une autre famille. Les types des régions d'intérêt sont issus de l'ensemble $TYPES = \{\text{typeA}, \text{typeB}, \ldots, \text{typeN}\}$.

Notation : par convention, la $i^{\text{ième}}$ région d'intérêt de type "type $\in TYPES$" est noté $R\text{type}_i$, avec R pour "Région". L'ensemble des régions d'intérêt d'un type donné est retourné par la fonction $EnvType(\text{type}, t_x)$:

$$\forall\ t_x \in T, \text{type} \in TYPES\ :\ EnvType(\text{type}, t_x) = \left\{R\text{type}_1, R\text{type}_2, \ldots, R\text{type}_k, \ldots, R\text{type}_n\right\}$$

L'union de ces sous-ensembles typologiques de régions fournit une autre façon de formaliser la notion d'environnement des régions d'intérêt à un moment donné de l'exécution :

$$\forall\ t_x \in T\ :\ EnvInt(t_x) = \bigcup_{\substack{\text{type} \\ \in TYPES}} EnvType(\text{type}, t_x)$$

Dans le cas du système de suivi côtier, à t_1, seule l'aire de navigation interdite constitue une région d'intérêt. Elle est associée aux régions d'intérêt de type "s" (pour "re<u>s</u>tricted"). Notée $R\text{s}_1$, cette région appartient à l'ensemble des régions d'intérêt à t_1 : $EnvInt(t_1) = \{R\text{s}_1\}$. Unique région d'intérêt du système de suivi côtier, $R\text{s}_1$ a pour hôte le composant serveur de données $C\text{d}_1$. Si ce composant vient à disparaître de la plate-forme, alors la région $R\text{s}_1$ ne peut plus être exploitée par les autres composants du système. Dans ce cas, le système de suivi perd la capacité à signaler aux clients leur présence dans la zone interdite.

Environnement géographique du système

Les régions d'intérêt et les régions d'exécution constituent l'environnement géographique du système distribué. Noté $EnvGeo(t_x)$, il rassemble $EnvInt(t_x)$ et $EnvExec(t_x)$ à t_x :

$$\forall\ t_x \in T\ :\ EnvGeo(t_x) = EnvInt(t_x) \cup EnvExec(t_x)$$

L'environnement géographique représente les frontières du système. La combinaison des couvertures spatiales des régions qui le constituent détermine la couverture du système à un moment donné. Dans l'exemple du système de suivi côtier, à t_1, l'environnement géographique $EnvGeo(t_1)$ est constitué des régions d'exécution liées aux capacités d'émission des cinq composants du système et de la région d'intérêt $R\text{s}_1$ qui traduit pour le système les frontières de la zone de navigation interdite :

$$EnvGeo(t_1) = \{R\text{s}_1, R\text{u}_1, R\text{d}_1, R\text{p}1_1, R\text{u}_2, R\text{p}2_1\}$$

Espaces accédés

L'environnement géographique, les régions et les composants expriment la situation d'exécution à un moment donné. Les espaces accédés forment l'image atemporelle de l'ensemble des situations rencontrées à l'exécution. L'espace accédé d'une région d'intérêt ou d'exécution est une couverture spatiale qui correspond à l'union des couvertures spatiales de la région considérée au cours de l'exécution. Autrement dit, l'espace accédé d'une région d'exécution reg_y référence l'ensemble des couvertures de régions dans lesquelles le composant générateur de reg_y a été, est ou sera actif entre t_0 et t_{end}. De la même manière, l'espace accédé d'une région d'intérêt reg_z représente l'ensemble des couvertures spatiales désignées à partir des données de définition de int_y entre t_0 et t_{end}.

En considérant que $Cover(reg_a, t_x) \cup Cover(reg_a, t_y)$ désigne l'union des couvertures spatiales d'une région reg_a à t_x et t_y, l'espace accédé par cette région reg_a est une couverture

spatiale définie par la fonction $Acc(reg_a)$:

$$\forall\ reg_a \in \bigcup_{t_x \in T} EnvGeo(tx)\ \ : Acc(reg_a) = \bigcup_{t_y \in T} f(reg_a, t_y)$$

$$\text{avec } f(reg_a, t_y) = \begin{cases} Cover(reg_a, t_y) & \text{ssi } reg_a \in EnvGeo(t_y) \\ \emptyset & \text{autrement} \end{cases}$$

Cette définition de l'espace accédé d'une région se décline sur les sous-ensembles de régions. Quel que soit le rôle role $\in ROLES$ et le type type $\in TYPES$, $Acc(R\text{role}_a)$ et $Acc(R\text{type}_b)$ désignent respectivement les espaces accédés par une région d'exécution $R\text{role}_a$ et par une région d'intérêt $R\text{type}_b$.

La Figure III.4b représente l'ensemble des espaces accédés dans le cas du système de suivi côtier, en considérant que :

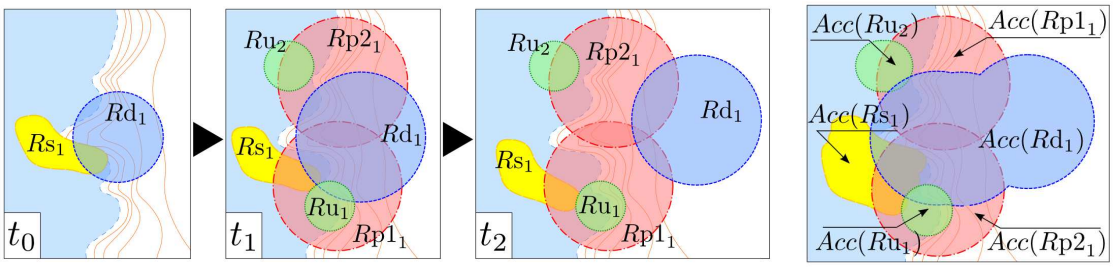
- le composant $C\text{d}_1$ est le premier à s'initialiser. Son démarrage marque l'instant t_0 du système. Il est rejoint à t_1 par l'ensemble des autres composants ;
- l'instant d'exécution t_2 est le dernier instant de mesure du système (t_2=t_{end}) ;
- comme $C\text{d}_1$ est l'hôte de la région d'intérêt $R\text{s}_1$, celle-ci apparaît pour le système dès l'instant de l'initialisation.

Les espaces accédés du système de suivi sont retournés par $Acc(R\text{s}_1)$, $Acc(R\text{u}_1)$, $Acc(R\text{u}_2)$, $Acc(R\text{p1}_1)$, $Acc(R\text{p2}_1)$, et $Acc(R\text{d}_1)$. Quel que soit $t_x \in T$, ces couvertures spatiales sont définies comme :

$$Acc(R\text{d}_1) = \begin{pmatrix} Cover(R\text{d}_1, t_0) \cup Cover(R\text{d}_1, t_1) \\ \cup Cover(R\text{d}_1, t_2) \end{pmatrix} \qquad Acc(R\text{u}_2) = \begin{pmatrix} \emptyset \cup Cover(R\text{u}_2, t_1) \cup \\ Cover(R\text{u}_2, t_2) \end{pmatrix}$$

$$Acc(R\text{p1}_1) = \begin{pmatrix} \emptyset \cup Cover(R\text{p1}_1, t_1) \cup \\ Cover(R\text{p1}_1, t_2) \end{pmatrix} \qquad Acc(R\text{p2}_1) = \begin{pmatrix} \emptyset \cup Cover(R\text{p2}_1, t_1) \cup \\ Cover(R\text{p2}_1, t_2) \end{pmatrix}$$

$$Acc(R\text{s}_1) = \begin{pmatrix} Cover(R\text{s}_1, t_0) \cup Cover(R\text{s}_1, t_1) \\ \cup Cover(R\text{s}_1, t_2) \end{pmatrix} \qquad Acc(R\text{u}_1) = \begin{pmatrix} \emptyset \cup Cover(R\text{u}_1, t_1) \cup \\ Cover(R\text{u}_1, t_2) \end{pmatrix}$$



(a) Évolution des couvertures spatiales des régions durant l'exécution

(b) Espaces accédés par les régions

Figure III.4 – Evolution des régions du système de suivi cotier entre t_0 et t_2

Les composants immobiles durant l'exécution, et qui n'ont pas vu leur région de diffusion varier, définissent un espace accédé strictement identique à leur région de diffusion. C'est

notamment le cas des composants clients et des composants serveurs de traitement. Lors de l'exécution décrite du système de suivi, seule la région de diffusion de données et la région d'intérêt désignant la zone de navigation interdite ont vu leur couverture spatiale varier. Les espaces accédés de ces régions diffèrent sensiblement de leurs couvertures spatiales aux différents instants de l'exécution.

Les relations existantes entre les espaces accessibles "conceptuels" et les espaces accédés "à l'exécution" forment le nœud qui permet de passer du modèle conceptuel à la description de l'exécution. Ces liens seront précisées lors de la description des étapes du processus de conception (Chap. III.3). Avant d'explorer plus avant cette approche de modélisation, la suite de cette section formalise les notions de relations entre régions et de contexte d'exécution. Ces deux définitions seront adaptées pour être utilisées comme des outils de spécification des cas d'adaptation lors de l'analyse conceptuelle.

Relations entre régions

Un contexte d'exécution du système a un instant donné repose sur les relations que les différentes régions entretiennent. Ces relations fournissent un vocabulaire pour résumer les propriétés des composants entre eux, des régions d'intérêt entres elles et des composants par rapport aux régions d'intérêt. Plus spécifiquement, la notion de relation entre régions est déclinée selon trois cas de figure :

- deux régions d'exécution sont en relation lorsque *leurs composants sont connectés et peuvent échanger des informations*,
- deux régions d'intérêt sont en relation dès lors que *leurs couvertures spatiales s'intersectent*,
- une région d'exécution et une région d'intérêt sont en relation lorsque le composant générateur de la région d'exécution *appartient à la couverture de la région d'intérêt.*

Pour pouvoir établir la validité d'une relation entre régions, ces trois cas de figure doivent être formalisés. Pour cela, les notions de *connection entre composants* et de d'*intersection de couvertures spatiales* sont définies. La formalisation proposée de la relation entre régions repose sur ces fonctions et compose l'union des différents cas de figures.

Dans la suite, la fonction $Loc(cmp_y, t_x)$ retourne la localisation du composant cmp_y à l'instant t_x. Cette localisation peut être située à l'intérieur ou à l'extérieur de la couverture spatiale d'une région d'intérêt ou d'exécution. En considérant la définition de la couverture spatiale comme l'ensemble des localisations recouvertes par une région à un moment donné, et quelle que soit une région $reg_y \in EnvGeo(t_x)$ et un composant $cmp_z \in Platform(t_x)$, la relation $Loc(cmp_z, t_x) \in Cover(reg_y, t_x)$ signifie la présence du composant cmp_z dans la couverture spatiale de la région reg_y à l'instant t_x.

Intuitivement, un composant se situant dans une région d'exécution où sont diffusées des informations à la capacité de récupérer ces informations et de les traiter. L'appartenance équivoque d'un composant à la couverture spatiale de la région d'exécution d'un autre traduit en termes géographiques la capacité à échanger des informations entre ces deux composants. Plus formellement quels que soient deux composants $cmp_a, cmp_b \in Platform(t_x)$, lorsque la localisation du composant cmp_a fait partie de la couverture spatiale de la région $Region(cmp_b)$ et réciproquement, alors l'échange d'informations entre ces deux composants est possible à

t_x. Dans la suite la fonction $Connect(cmp_a, cmp_b, t_x)$ retourne 1 si cmp_a et cmp_b peuvent échanger des informations à t_x, et 0 sinon :

$$\forall\ t_x \in T,\ cmp_a, cmp_b \in Platform(t_x)\ :$$
$$Connect(cmp_a, cmp_b, t_x) = \begin{cases} 1 & \text{ssi } \begin{pmatrix} Loc(cmp_a, t_x) \in Cover(Region(cmp_b), t_x) \wedge \\ Loc(cmp_b, t_x) \in Cover(Region(cmp_a), t_x) \end{pmatrix} \\ 0 & \text{autrement} \end{cases}$$

L'intersection entre deux régions est définie par rapport à leurs couvertures spatiales. Cette intersection est vérifiée s'il existe des localisations qui, à un instant donné de l'exécution, appartiennent à la fois à la couverture de chacune des régions. Formellement, pour deux régions $reg_a, reg_b \in EnvGeo(t_x)$, la fonction $Inter(reg_a, reg_b, t_x)$ retourne 1 si les couvertures spatiales de reg_a et reg_b s'intersectent à t_x et 0 autrement [8] :

$$\forall\ t_x \in T,\ reg_a, reg_b \in EnvGeo(t_x)\ :$$
$$Inter(reg_a, reg_b, t_x) = \begin{cases} 1 & \text{ssi } \big(\exists\ loc \in Cover(reg_a, t_x) \mid loc \in Cover(reg_b, t_x)\big) \\ 0 & \text{autrement} \end{cases}$$

La relation entre deux régions de l'environnement géographique à un moment donné est identifiée par la fonction booléenne $Relate(reg_a, reg_b, t_x)$ qui retourne 1 lorsque reg_a et reg_b sont liées à t_x et 0 autrement. Cette fonction repose sur $Inter(\ldots)$ et $Connect(\ldots)$ et regroupe les différents cas de figures qui valident la relation entre deux régions :

$$\forall\ t_x \in T,\ reg_a, reg_b \in EnvGeo(t_x)\ :\ Relate(reg_a, reg_b, t_x) =$$
$$\begin{cases} 1 & \text{ssi } \begin{pmatrix} \Big(reg_a, reg_b \in EnvExec(t_x)\ \wedge\ Connect\begin{pmatrix} Component(reg_a), \\ Component(reg_b), t_x \end{pmatrix} = 1\Big)\ \vee \\ \big(reg_a, reg_b \in EnvInt(t_x)\ \wedge\ Inter(reg_a, reg_b, t_x) = 1\big)\ \vee \\ \Big(\begin{pmatrix} reg_a \in EnvInt(t_x) \\ \wedge\ reg_b \in EnvExec(t_x) \end{pmatrix}\ \wedge\ Loc\big(Component(reg_b), t_x\big) \in Cover(reg_a, t_x)\Big)\ \vee \\ \Big(\begin{pmatrix} reg_b \in EnvInt(t_x) \\ \wedge\ reg_a \in EnvExec(t_x) \end{pmatrix}\ \wedge\ Loc\big(Component(reg_a), t_x\big) \in Cover(reg_b, t_x)\Big) \end{pmatrix} \\ 0 & \text{autrement} \end{cases}$$

Le formalisme proposé traduit en termes ensemblistes les trois cas de figure précédemment décrits textuellement et pour lesquels deux régions sont en relation. Par exemple, dans le cas du système de suivi côtier à t_1, les composants $C\mathrm{p}1_1$ et $C\mathrm{u}_2$ échangent des informations et leurs régions d'exécution respectives $R\mathrm{p}1_1$ et $R\mathrm{u}_2$ sont liées à cet instant (c-à-d. $relate(R\mathrm{p}1_1, R\mathrm{u}_2, t_1) = 1$). A l'inverse, même si leurs couvertures spatiales à t_1 s'intersectent, les régions $R\mathrm{p}2_1$ et $R\mathrm{u}_1$ ne sont pas liées : le composant de traitement $C\mathrm{p}2_1$ n'est pas inclus dans la couverture de la région $R\mathrm{u}_1$ et il ne peut pas recevoir d'information du client $C\mathrm{u}_1$ (c-à-d. $relate(R\mathrm{p}2_1, R\mathrm{u}_1, t_1) = 0$).

8. Dans la suite, la notation "$\exists\ e_x \in E \mid p_y$" est adoptée pour exprimer des assertions de quantification existentielle. Cette proposition logique se lit "Il existe au moins un élément e_x dans l'ensemble E tel que e_x vérifie la propriété p_y". Les assertions de quantification universelle peuvent être *vraies* ou *fausses*

III.2.4 Décomposition contextuelle

Le **contexte d'exécution** est construit par rapport à l'ensemble des relations entre régions à un moment donné. Il synthétise l'état de la plate-forme par 1) une vue des transferts d'informations qu'il est possible d'établir entre les composants, 2) la situation des composants par rapport aux régions d'intérêt et 3) la situation des régions d'intérêt entres elles. Le contexte d'exécution s'établit à deux niveaux. Le **contexte d'exécution global** pose un regard extérieur au système et caractérise l'ensemble des relations entre régions. En ce sens, ce niveau du contexte d'exécution est une reprise des paires de régions en relation à un moment donné. Le **contexte d'exécution local** est la vue des relations entre régions depuis un composant en particulier. Cette vue locale, mesurée au sein de chaque composant du système, est fournie comme clé de hachage aux processus d'adaptation de ce composant. En connaissant le contexte d'exécution local, ces processus recherchent l'ensemble des modifications à apporter au fonctionnement du composant pour passer du contexte d'exécution précédent au contexte courant.

Contexte d'exécution global

A un instant d'exécution t_x, le contexte global $State(t_x)$ retourne l'ensemble des paires de régions prises parmi $EnvGeo(t_x)$ et qui sont en relation. Plus formellement :

$$\forall\ t_x \in T\ :\ State(t_x) = \\ \big\{(reg_a, reg_b) \in EnvGeo(t_x)^2 \mid Relate(reg_a, reg_b, t_x) = 1\ \wedge\ reg_a \neq reg_b\big\}$$

Par exemple, dans le cas du système de suivi côtier, à chaque instant de l'exécution, un nouveau contexte global est mesuré :

$$State(t_0) = \emptyset \qquad State(t_2) = \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\big\}$$
$$State(t_1) = \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1), (R\mathrm{d}_1, R\mathrm{p}1_1)\big\}$$

La figure III.5 illustre différentes notations du contexte global. Dans son écriture sous forme de graphe (Fig. III.5(a, b & c)), les noeuds sont composés des régions de l'environnement géographique à l'instant de la mesure. Les arcs du graphe représentent les relations entre régions.

La notation sous forme de matrice représente en lignes et colonnes l'ensemble maximal des régions de l'environnement (Fig. III.5(d)) [9]. Par convention, une cellule noire représente le cas de deux régions en relation, une cellule blanche représente deux régions non liées et une cellule orange indique qu'au moins l'une des régions ne fait pas partie de l'environnement au moment de la mesure et que, de ce fait, sa relation aux autres régions ne peut pas être déterminée.

Cette notation matricielle donne une idée de la difficulté de prise en compte des différentes situations contextuelles. Il existe potentiellement une réponse fonctionnelle adaptée

9. Ces représentations sont optimisées pour qu'une relation entre régions distinctes soit dénotée une et une seule fois dans chaque matrice.

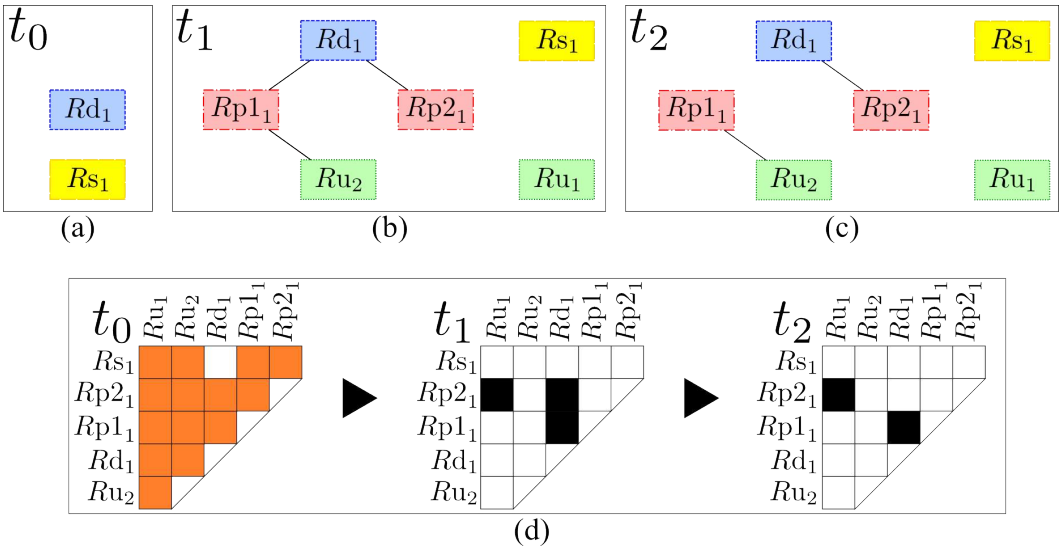


Figure III.5 – Graphe des relations entre régions et notation matricielle du contexte d'exécution dans le cas du système de suivi côtier

à chaque contexte d'exécution et le nombre de contextes qu'il est possible de décrire à l'exécution croît exponentiellement par rapport au nombre de régions de l'environnement. Considérant un système qui construit au maximum n régions au cours de son exécution, l'ensemble de ces régions définit C_2^n relations [10], soit autant de possibilités d'appel à la fonction $Relate(\ldots)$. Le nombre de contextes d'exécution exprimables pour un système donné est dans ce cas $2^{C_2^n}$. Dans le cas du système de suivi côtier, le nombre maximum de régions n vaut 6. Le nombre de contextes exprimables est $2^{C_2^6} = 2^{15} = 32768$.

Le nombre important de contextes d'exécution pose un défi conceptuel majeur. Dans la démarche de modélisation proposée dans la suite de ce chapitre, des simplifications dans le calcul de l'ensemble des contextes d'exécution seront effectuées pour qu'un même ensemble d'adaptations puisse être appliqué dans plusieurs situations d'exécution. Avec la même volonté de limitation de la combinatoire des relations entre régions, le système distribué sera décomposé en différents sous-ensembles de services construisant chacun un nombre de régions limité à l'exécution.

Contexte d'exécution local

Le contexte d'exécution local se distingue du contexte global en effectuant une mesure des relations entre régions selon les connaissances de l'environnement propres à chaque composant. Par exemple, dans le cas du système de suivi côtier, le contexte global reste une vue post-exécution, établie par un superviseur (p.ex. la personne qui analyse l'exécution du système). Ce contexte global ne peut pas être établi au niveau de l'ensemble des composants au moment de l'exécution. Par exemple, à t_1, le second composant client Cu_2, dont la région d'exécution est isolée des autres et qui n'échange d'information avec aucun composant, n'a

10. $C_k^n = \frac{n!}{k!(n-k)!}$ désigne en probabilités le nombre de choix possibles de k éléments parmi n éléments discernables.

aucun moyen de connaître quelles sont les relations entre les régions du reste du système, ni même s'il existe d'autres composants ou régions actifs ailleurs dans l'environnement. Cet isolement se reflète au niveau de son contexte d'exécution local, qui consiste en un ensemble vide de relations.

Le contexte local représente l'ensemble des paires de régions en relation tel qu'il peut être déterminé par un composant donné. Dans la suite, $ExecRel(reg_y, t_x)$ retourne l'ensemble des régions d'exécution en relation avec une région $reg_y \in EnvGeo(t_x)$:

$$\forall\ t_x \in T,\ reg_y \in EnvGeo(t_x)\ :\ ExecRel(reg_y, t_x) = \\ \big\{reg_z \in EnvExec(t_x) \mid Relate(reg_y, reg_z, t_x) = 1\ \wedge\ reg_y \neq reg_z\big\}$$

Le contexte local d'un composant cmp_y retourne les paires de régions en relation qui sont issues de l'union de l'ensemble des régions d'intérêt et des régions d'exécution accessibles depuis le composant, c'est-à-dire, qui sont liées directement ou par transitivité de voisinage à la région dérivée par cmp_y. Formellement, ce contexte d'exécution local $LocState(cmp_y, t_x)$ s'écrit :

$$\forall\ t_x \in T,\ cmp_y \in Platform(t_x)\ :\ LocState(cmp_y, t_x) = \\ \left\{\begin{matrix}(reg_a, reg_b) \in EnvGeo(t_x)^2 \mid Relate(reg_a, reg_b, t_x) = 1\ \wedge\ reg_a \neq reg_b\ \wedge \\ reg_a, reg_b \in ExecNeigh\big(Region(cmp_y), \emptyset, t_x\big) \cup IntNeigh\big(Region(cmp_y), t_x\big)\end{matrix}\right\}$$

avec $ExecNeigh(reg_y, A, t_x) =$

$$\Big\{reg_z \in ExecRel(reg_y, t_x)\ \cup\ ExecNeigh\big(reg_z, A \cup \{reg_z\}, t_x\big) \mid reg_z \notin A\Big\}$$

et $IntNeigh(reg_y, t_x) =$

$$\Big\{reg_z \in EnvInt(t_x) \mid Region\big(Host(reg_z, t_x)\big) \in ExecNeigh(reg_y, \emptyset, t_x)\Big\}$$

La fonction $ExecNeigh(reg_y, \emptyset, t_x)$ retourne l'ensemble des régions d'exécution dont le composant cmp_y peut avoir connaissance à t_x. Dans l'exemple du système de suivi côtier à t_1, $ExecNeigh(R\mathrm{u}_2, \emptyset, t_1)$ retourne $\{R\mathrm{u}_2, R\mathrm{p}1_1, R\mathrm{d}_1, R\mathrm{p}2_1\}$, soit l'ensemble des régions qui forment les nœuds du sous-graphe représenté en figure III.5(b).

La fonction $IntNeigh(reg_y, t_x)$ recherche les régions d'intérêt définies par les composants générant les régions d'exécution de $ExecNeigh(\ldots)$. Dans l'exemple du suivi côtier, $IntNeigh(R\mathrm{u}_2, t_1)$ retourne un ensemble contenant la région d'intérêt $R\mathrm{s}_1$: la région $R\mathrm{d}_1$, générée par le composant $C\mathrm{d}_1$ hôte de $R\mathrm{s}_1$, est contenue dans l'ensemble $ExecNeigh(R\mathrm{u}_2, \emptyset, t_1)$.

La fonction $LocState(cmp_y, t_x)$ effectue l'union des ensembles $ExecNeigh(\ldots)$ et $IntNeigh(\ldots)$ et retourne les paires de régions qui y sont en relation. Dans le cas du système

de suivi côtier, les contextes d'exécution locaux au second client sont retournés par :

$$\begin{aligned} LocState(C\mathrm{u}_2, t_1) &= \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\big\} \\ LocState(C\mathrm{u}_2, t_2) &= \big\{(R\mathrm{u}_2, R\mathrm{p}1_1)\big\} \end{aligned}$$

À t_2, il n'est pas possible du point de vue du composant client $C\mathrm{u}_2$ de s'informer sur l'existence d'une relation entre les régions $R\mathrm{d}_1$ et $R\mathrm{p}2_1$. Cette paire de régions n'apparaît pas dans le contexte local $LocState(C\mathrm{u}_2, t_2)$, même si elle fait partie du contexte global $State(t_2)$. Plus généralement, le contexte global recense l'ensemble des contextes locaux. L'union des paires de régions des contextes locaux fournit une autre façon d'exprimer $State(t_x)$:

$$\forall\ t_x \in T\ :\ State(t_x) = \bigcup_{\substack{cmp \\ \in Platform(t_x)}} LocState(cmp, t_x)$$

Dans le cas du système de suivi côtier à t_2, l'union des contextes locaux vérifie par cette équivalence d'écriture son égalité à l'ensemble du contexte global tel que décrit à la section précédente :

$$\begin{aligned} State(t_2) &= \begin{pmatrix} LocState(C\mathrm{u}_1, t_2) \cup LocState(C\mathrm{u}_2, t_2) \cup LocState(C\mathrm{p}2_1, t_2) \\ \cup\, LocState(C\mathrm{d}_1, t_2) \cup LocState(C\mathrm{p}1_1, t_2) \end{pmatrix} \\ &= \emptyset \cup \big\{(R\mathrm{u}_2, R\mathrm{p}1_1)\big\} \cup \big\{(R\mathrm{u}_2, R\mathrm{p}1_1)\big\} \cup \big\{(R\mathrm{p}2_1, R\mathrm{d}_1)\big\} \cup \big\{(R\mathrm{p}2_1, R\mathrm{d}_1)\big\} \\ &= \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\big\} \end{aligned}$$

Au niveau de chaque composant, les contextes locaux détectés déclenchent des changements fonctionnels qui ont éventuellement été prévus par les concepteurs. Par exemple, le contexte local vide mesuré à t_0 par le composant serveur de données signifie qu'il agit pour lui même uniquement. Une adaptation possible à son niveau peut être de baisser son activité de service d'information dans l'attente de l'établissement de nouvelles relations avec d'autres régions d'exécutions ou d'autres régions d'intérêt. De la même manière, le composant serveur de traitement $C\mathrm{p}1_1$ perdant la source de données à t_2, mesure à son niveau, le contexte $\big\{(R\mathrm{u}_2, R\mathrm{p}1_1)\big\}$ et adapte sa réponse aux requêtes du client en lui indiquant, par exemple, que les données sur la zone de navigation interdite peuvent ne plus être à jour.

En généralisant ces exemples, il est possible d'imaginer au maximum pour chaque composant autant de cas d'adaptations qu'il y a de contextes locaux mesurables à partir de ces composants. La suite de ce chapitre prend le point de vue des concepteurs du système et se concentre sur la définition des contextes d'exécutions locaux qu'il est utile de prendre en compte pour adapter le comportement d'un composant.

III.3 Description conceptuelle de l'environnement d'un système mobile et distribué

La description du système à l'exécution montre qu'un système mobile et distribué réagit par des adaptations à des changements de son environnement. Ces variations sont matérialisées au sein de chaque composant par l'état d'un contexte d'exécution local. Sans qu'il soit nécessaire (ni même humainement possible dans la plupart des cas) de spécifier et de distinguer l'ensemble des contextes et des cas d'adaptation, l'une des tâches à la conception est d'identifier quels sont les contextes d'exécution locaux déclencheurs d'adaptations qui vont dans le sens des objectifs fixés en termes d'amélioration des valeurs interactives et utilitaires (cf. II.3.3). Dans la suite, une méthode d'identification de ces contextes "clés" pour l'adaptation est proposée. Cette méthode spécifie l'ensemble des espaces accessibles des régions de l'environnement, étudie leurs relations, et construit des règles d'inclusion ou d'exclusion automatiques de relations entre régions pour réduire l'ensemble des contextes à étudier et focaliser la suite de la modélisation sur ces situations géographiquement réalistes.

L'environnement géographique et les composants de la plate-forme forment les primitives pour l'analyse et la construction du contexte d'exécution à un moment donné. Dans une perspective conceptuelle, ces éléments doivent être déterminés en premier lieu, préalablement à la recherche des contextes d'exécution et des adaptations qu'ils engendrent. Plus spécifiquement, il s'agit pour l'équipe de développement du système de s'interroger sur le modèle géographique du système et sur sa distribution en composants. Cette première étude analytique se déroule en deux temps et questionnements :

1. *Quels sont les composants du système ? Quels sont leurs rôles et leurs caractéristiques techniques ?* Lors de cette étape de **description physique**, les concepteurs dressent la liste des matériels sur lesquels se répartit le système. Pour chaque composant matériel, les caractéristiques techniques sont relevées. Elles caractérisent entre autres les aires d'influence et la mobilité géographique des composants. A l'issue de cette première étape, la plate-forme d'exécution est prévue, et ses composants sont dotés d'une couverture spatiale qui représente les limites de son espace d'utilisation à un moment donné de l'exécution ;
2. *Comment se décompose l'espace géographique du système ? Quelles sont les régions d'exécution ? Quelles sont les régions d'intérêt ?* Les composants, les régions associée et leurs couvertures spatiales déterminent un ensemble que les concepteurs peuvent localiser dans l'espace du système pour en effectuer une **description géographique**. Les couvertures spatiales des régions constituent l'image géographique des conditions d'exécution que le système rencontrera une fois implémenté. Ces couvertures spatiales variables constituent une primitive influençant dans une large mesure la détermination des différentes situations d'exécution.

A l'issue de ces premières étapes d'analyse conceptuelle, les composants et les régions du système sont déterminés et l'étude de leur mobilité amène les concepteurs à spécifier l'ensemble des espaces accessibles qui bornent les déplacements des couvertures spatiales des régions de l'environnement. Comme les régions lors de l'exécution, ces espaces accessibles entretiennent des relations qui permettent de filtrer l'ensemble des contextes d'exécution pour ne conserver comme candidats au déclenchement d'adaptations que les contextes qui pourront être observés à l'exécution. Cette étape forme un processus d'**analyse et de raisonnement**

spatial qui est complété par l'énonciation de l'ensemble des contextes potentiellement rencontrés à l'exécution lors d'une étape automatisée de **description contextuelle**.

III.3.1 Description physique

L'architecture du système mobile et distribué est décrite à la conception en termes de plate-forme multi-composants, de rôles des composants et de liens dénotant des échanges d'informations entre ces composants. Pour déterminer l'ensemble de la plate-forme et les rôles de chaque composant, il est essentiel pour les concepteurs de pouvoir connaître au préalable l'objectif du système et au moins partiellement quelles seront les tâches des utilisateurs, et quelles informations ils seront amenés à manipuler. Cette description de la tâche par une approche centrée utilisateur menée en parallèle de la description physique sera présentée au prochain chapitre de cette étude. Les liens entre l'analyse des tâches et la description physique des composants de la plate-forme y seront établis plus rigoureusement.

La suite de cette section admet que l'analyse des tâches fournit les renseignements nécessaires au dimensionnement de la plate-forme et à la description des rôles des composants. Par exemple, dans le cas du système de suivi côtier, les clients désignent des marins, plaisanciers, etc. Cette cible du système a été spécifiée en amont de la description de la plate-forme. De la même manière, la nature des traitements d'informations (p. ex. présentation de données sur une carte numérique) et des données gérées (p. ex. localisation de la zone de navigation interdite) est connue au moment de la détermination de la plate-forme distribuée et des rôles de ses composants.

Composants de la plate-forme

En parallèle d'une description des tâches et de la construction du modèle des données du système, les concepteurs spécifient l'ensemble des constituants de la plate-forme. Cette description physique définit une version "atemporelle" de la plate-forme à l'exécution. En l'occurrence, il s'agit de décrire tous les composants qui pourront appartenir à $Platform(t_x)$ lors de l'exécution et quel que soit le moment t_x d'exécution choisi. Si $PLATFORM$ référence cet ensemble de composants décrits à la conception, alors, lors de l'exécution, tout composant de $Platform(t_x)$ est issu de $PLATFORM$:

$$\begin{pmatrix} \forall\ t_x \in T,\ cmp_x \in Platform(t_x)\ : \\ cmp_x \in PLATFORM \end{pmatrix} \Leftrightarrow \begin{pmatrix} \forall\ t_x \in T\ : \\ Platform(t_x) \subseteq PLATFORM \end{pmatrix}$$

L'étape de description des composants de la plate-forme permet de distinguer deux cas de figure :

- lorsque le système est déployé sur une nouvelle architecture, les concepteurs décrivent un ensemble de composants et une plate-forme non encore existants. Cette option permet une grande latitude dans la définition de la distribution du système ;
- lorsque le système est déployé sur une architecture déjà existante, les concepteurs décrivent une plate-forme qui n'est pas forcément idéale pour l'exécution de leur

système. Dans ce cas, l'étude des variations d'exécution et la proposition d'adaptation est une manière de pallier ces contraintes.

Par exemple, dans le cas du système de suivi côtier, les concepteurs se sont appliqués à utiliser au mieux les composants à leur disposition, et plus particulièrement l'ensemble des serveurs de traitement et des données. Un cas de conception plus favorable leur aurait permis de déployer une nouvelle architecture matérielle et de positionner des composants de traitement diffusant plus loin au large, et un serveur de données qui assurerait une certaine permanence dans l'accès aux informations par les serveurs de traitement.

Description des rôles des composants

Après la spécification des composants de la plate-forme, les concepteurs décrivent les différents rôles d'exécution que pourront prendre les composants. Ces rôles dépendent essentiellement de la part du logiciel système que les composants devront assumer à l'exécution. Par exemple, un système distribué qui reparti ses données entre plusieurs composants se définit à la conception par un ensemble de rôles de gestion de données. Les rôles d'exécution déterminées par les concepteurs forment l'ensemble $ROLES$ utilisé lors de la description du système à l'exécution.

Ces rôles et leur association à l'un ou l'autre des composants de la plate-forme ne sont pas absolus. Par exemple, il est possible pour un composant de traitement d'héberger un ensemble de données, ou pour un serveur de données, d'effectuer certains traitements. De la même manière, un ordinateur personnel exécutant un système non distribué comme un traitement de texte pourrait être vu dans le modèle proposé comme un composant client qui effectuerait des traitements locaux à partir de données servies sur ce même client. Plus généralement, l'assimilation d'un composant à l'une ou l'autre des catégories doit être effectuée en connaissance du rôle et des fonctions qu'il assume principalement au sein du système. Dans le cas du traitement de texte décrit précédemment, le rôle d'interface avec l'utilisateur est retenu pour désigner ce composant comme un client de rôle "user". Ce composant $C\text{user}_1$ fait partie du sous-ensemble $Cmps(user, t_x)$ bien que des traitements soient effectués et que des données soient utilisées localement.

Les composants du système sont estampillés par leur rôle lors de la conception, et l'ensemble $CMPS(\text{role})$ retourne les composants qui tiendront le rôle "role" à l'exécution du système. Cet ensemble surclasse la description à l'exécution des composants qui tiennent le rôle "role"$\in ROLES$:

$$\forall\ t_x \in T,\ \text{role} \in ROLES\ :\ Cmps(\text{role}, t_x) \subseteq CMPS(\text{role})$$

L'association d'un rôle à un composant décrit dans la plate-forme n'est pas systématique. Lors de la conception, il est parfois utile de pouvoir décrire le comportement de composants qui ne sont pas connus à la conception, mais dont le rôle a été précisé. Notamment, dans le cas de la conception des systèmes multi-utilisateurs, le nombre de clients et les caractéristiques de leurs plates-formes n'est pas établi à la modélisation. En généralisant, il existe pour chaque rôle établi à la conception un certain nombre de composants qui l'implémenteront lors de l'exécution. Deux cas de figure se présentent pour les concepteurs :

- les composants qui tiendront un rôle "role $\in ROLES$" lors de l'exécution sont connus au moment de la conception. Dans ce cas, l'ensemble des composants $CMPS$(role) qui tiennent ce rôle peut être précisé exactement (p. ex. $CMPS(\text{role}) = \{C\text{role}_1, C\text{role}_2\}$).
- les composants qui tiendront un rôle "role $\in ROLES$" lors de l'exécution ne sont pas connus au moment de la conception. Dans ce cas, l'ensemble $CMPS$(role) contient un composant "générique", noté $C\text{role}_*$, et qui désigne n'importe quel(s) composant(s) dont le(s) rôle(s) sera/seront "role" lors de l'exécution.

Dans le cas du système de suivi côtier, en corrélant la description des tâches à la plate-forme dont ils disposent, les concepteurs ont prévu quatre rôles pour les composants :

- le rôle d'utilisateur "u" est dévolu au(x) marin(s) qui accède(nt) au système via un client mobile depuis leur(s) bateau(x),
- le rôle de serveur de données "d" est attribué au(x) composant(s) qui stocke(nt) l'ensemble des informations du système,
- le rôle de traitement "p1" désigne le(s) composant(s) qui effectue(nt) une première partie des tâches du système (p.ex. rendu cartographique de la zone interdite à la navigation),
- le rôle de traitement "p2" est associé au(x) composant(s) qui effectue(nt) une seconde partie des tâches du système (p.ex. analyse de dangerosité d'une situation maritime).

Dans cet exemple, les concepteurs ont choisi de répartir les traitements entre deux composants dont les rôles d'exécution "p1" et "p2" diffèrent. Selon la convention de notation introduite lors de la description à l'exécution, ces composant sont dénotés respectivement $C\text{p1}_1$ et $C\text{p2}_1$. En terme d'accès aux fonctionnalités du système, cette division des traitements entre plusieurs composants signifie que l'utilisateur ne pourra effectuer l'intégralité des tâches prévues que si son composant client a accès à l'ensemble des composants de traitement.

Par rapport à l'infrastructure en place, les concepteurs établissent avec certitude les composants associés au rôle serveur :

$$CMPS(\text{d}) = \{C\text{d}_1\} \qquad CMPS(\text{p1}) = \{C\text{p1}_1\} \qquad CMPS(\text{p2}) = \{C\text{p2}_1\}$$

L'ensemble des composants qui prendront le rôle "Ru" d'utilisateur du système n'est pas précisé à la conception. Le système de suivi côtier s'adresse à un nombre a priori inconnu et variable de clients. Les clients sont représentés par un composant générique :

$$CMPS(\text{u}) = C\text{u}_*$$

La plate-forme du système de suivi décrite à la conception rassemble les trois composants serveurs connus et associe le composant générique représentant les utilisateurs :

$$PLATFORM = \{C\text{d}_1, C\text{p1}_1\, C\text{p2}_1, C\text{u}_*\}$$

III.3.2 Description géographique

L'étape de description géographique du système établit l'ensemble des régions d'intérêt, puis étudie les variations des localisations de ces régions et de leurs composants générateurs ou hôtes pour déterminer les espaces accessibles qu'occuperont leurs couvertures spatiales à chaque moment de l'exécution.

Détermination des régions d'intérêt

Les régions d'intérêt sont déterminées par rapport à un ensemble d'informations localisées géographiquement. Différentes solutions sont possibles pour définir les régions d'intérêt. Par exemple, les frontières d'une ville sous la forme d'une série de coordonnées géodésiques, déterminent une région d'intérêt dont la couverture recouvre tout l'espace de la ville. Un périmètre de sécurité peut être représenté par une région d'intérêt déterminée par la coordonnée d'un point central et un rayon. Un autre exemple de région d'intérêt mobile est proposé dans la mise en œuvre d'un outil de suivi de régate (Chap. VI.1). Dans ce système, ce sont les bateaux en course qui définissent une région d'intérêt mobile dont les frontières sont modifiées au cours du temps.

Dans tous ces exemples, une région d'intérêt est construite à partir des données manipulées par le système. Cette notion de région désignée à partir d'informations localisées du système se retrouve souvent en informatique mobile. Dans ces systèmes, la position d'un client dans l'environnement par rapport à des objets de référence localisés est régulièrement employée pour distinguer différents cas d'utilisation et proposer un contenu adapté à une situation géographique particulière. Ces objets référents "géo-localisés" correspondent aux régions d'intérêt du modèle proposé. Leur détermination par les concepteurs découle soit d'une lecture d'un ensemble de données existantes, soit de l'analyse de la tâche des utilisateurs :

- dans le cas d'une détermination à la lecture des données du système, les concepteurs n'ont pas de préjugé sur l'utilité de la construction d'une région d'intérêt. Le propos de sa détermination est de provoquer l'émergence d'un ensemble de situations contextuelles qu'il aurait été délicat de caractériser par une approche analytique de la tâche de l'utilisateur. Par exemple dans le cas d'un logiciel de cartographie d'une ville, l'analyse des données manipulées permet de mettre en œuvre une région d'intérêt désignant cette ville. Sans que cela ne soit apparu de façon impérative à l'analyse des tâches, les fonctionnalités du système peuvent potentiellement être adaptées selon qu'un utilisateur se situe à l'intérieur de la ville ou non ;
- à l'inverse, dans le cas d'une détermination après analyse des tâches des utilisateurs, les concepteurs savent que le système devra effectuer des différenciations entre des situations d'utilisation en se basant sur les positions relatives des utilisateurs et des composants par rapport à des régions de l'environnement. Dans ce cas, ces différenciations apparaissent dès l'analyse de la tâche et les comportements subséquents ont déjà été établis. Les régions d'intérêt sont prévues mais les données qui les déterminent ne sont pas forcément établies.

Dans le premier cas, la post-détermination des régions d'intérêt et l'analyse géographique qui suivra agissent comme un générateur d'idées qui permet d'affiner les tâches des utilisateurs en fonction de situations contextuelles faciles à détecter et à mettre en œuvre. Dans le cas de la pré-détermination des région d'intérêt, l'analyse des déplacements des autres composants autour de ces régions et les uns par rapport aux autres permet de préciser quels composants sont les plus à même de devenir les hôtes des régions d'intérêt et d'héberger leurs données de définition. Les régions d'intérêt décrites lors de la conception sont rassemblées au sein de l'ensemble noté "$ENVINT$". Comme pour les composants de la plateforme, cet ensemble

surclasse la description à l'exécution des régions d'intérêt :

$$\forall\ t_x \in T\ :\ EnvInt(t_x) \subseteq ENVINT$$

Au niveau du système de suivi côtier, la région d'intérêt $R\mathrm{s}_1$ est post-déterminée. Autrement dit, les données de définition d'une région interdite à la navigation sont présentes dans le système et sont utiles à la réalisation d'une tâche d'utilisateur (p. ex. illustration de cette zone sur une carte numérique), mais au niveau de l'analyse des tâches menée préalablement, le comportement du système n'est pas fondamentalement affecté par le positionnement des composants vis-à-vis de $R\mathrm{s}_1$. A ce stade de la modélisation du système, les concepteurs font un "pari" sur l'intérêt de représenter des régions interdites à la navigation, désignées par le type "s″ (re<u>s</u>tricted), pour offrir une meilleure utilité ou "utilisabilité" à l'exécution. Ils qualifient la région $R\mathrm{s}_1$ pour l'étude géographique de ses relations aux autres régions et composants du système et l'incluent dans l'ensemble des régions d'intérêt décrites à la conception : $ENVINT = \{R\mathrm{s}_1\}$.

Détermination des régions d'exécution

Les régions d'exécution décrites lors de la conception font partie de l'ensemble noté $ENVEXEC$ qui surclasse la description des régions d'exécution d'un système en fonctionnement :

$$\forall\ t_x \in T\ :\ EnvExec(t_x) \subseteq ENVEXEC$$

Dans la plupart des cas, les dimensions des régions de l'environnement et leurs couvertures spatiales sont estimées à la conception plutôt que mesurées. Formellement, chaque composant est associé à une région d'exécution qui représente ses capacités "théoriques" d'envoi de données (le tableau I.1 au chapitre I.2.2 répertorie quelques estimations de la couverture réseau dans le cas d'envoi de données par des technologies sans fil). Lorsqu'une région d'exécution est représentée par un composant générique, l'estimation des caractéristiques techniques peut se faire d'après un état de l'art des solutions du marché. Par exemple, dans le cas de clients représentés par un composant générique, et s'agissant de donner une estimation de capacités de transmission, il semble raisonnable à l'heure actuelle de s'appuyer sur des envois/réception en Wi-Fi de faible puissance qui couvrent environ une cinquantaine de mètres autour du client.

Les composants génériques et régions qu'ils génèrent constituent une première simplification conceptuelle par rapport à l'observation du système à l'exécution. Plus particulièrement, s'il existe plusieurs composants de la plate-forme se rapportant à la description conceptuelle d'un composant générique, alors les relations entre les régions qu'ils génèrent ne sont pas considérées comme discriminantes pour l'adaptation du système. Dans le système de suivi côtier, les composants clients $C\mathrm{u}_1$ et $C\mathrm{u}_2$ à l'exécution "implémentent" le composant générique $C\mathrm{u}_*$. La relation entre les régions d'exécution qu'ils génèrent (c-à-d. la valeur retournée par $Relate(R\mathrm{u}_1, R\mathrm{u}_2, t_x)$, quel que soit $t_x \in T$) n'a pas d'impact sur l'adaptation. Ce choix des concepteurs signifie que si deux marins se rencontrent et ont la possibilité d'échanger

des informations via leur client, cet échange n'est toutefois pas jugé utile pour l'amélioration de l'utilité ou de l'utilisabilité du système[11].

Pour le système de suivi côtier, l'équipe de conception précise que les composants de la plate-forme (c-à-d. $C\mathrm{d}_1$, $C\mathrm{p1}_1$, $C\mathrm{p2}_1$) génèrent des régions d'exécution $R\mathrm{d}_1$, $R\mathrm{p1}_1$ et $R\mathrm{p1}_2$ décrites à l'exécution. Le composant générique $C\mathrm{u}_*$ représentant les clients de rôle "u", est associé à une région d'exécution "générique" nommée $R\mathrm{u}_*$. L'ensemble des régions d'exécution décrites lors de la conception du système de suivi côtier est alors : $ENVEXEC = \{R\mathrm{u}_*, R\mathrm{d}_1, R\mathrm{p1}_1, R\mathrm{p2}_1\}$.

Environnement géographique à la conception

Comme lors de la description à l'exécution, les régions d'intérêt rejoignent l'ensemble des régions d'exécution pour former l'environnement géographique du système. A la différence d'une analyse de l'exécution, l'ensemble des régions déterminés à la conception est maximal. Dans la suite, $ENVGEO$ est l'ensemble des région du système qu'il est possible de retrouver à l'exécution. Cet ensemble est constitué par l'union des régions d'exécution et d'intérêt :

$$ENVGEO = ENVEXEC \cup ENVINT$$

$ENVGEO$ surclasse la description des régions de l'environnement d'un système à l'exécution :

$$\forall\ t_x \in T\ :\ EnvGeo(t_x) \subseteq ENVGEO$$

Détermination des espaces accessibles

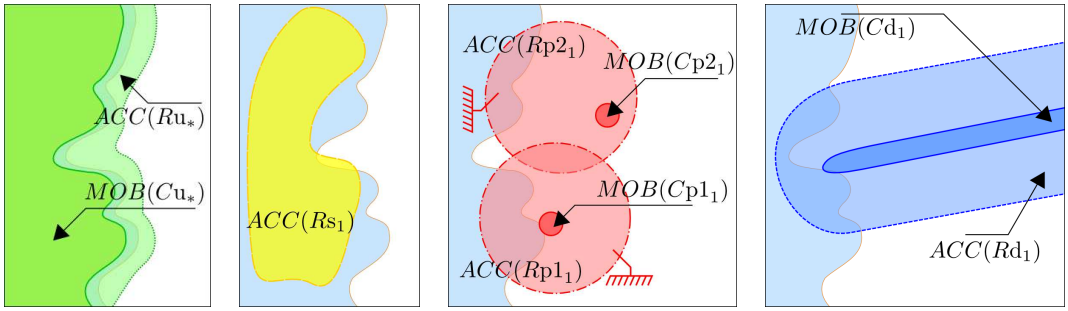
Dans la plupart des systèmes mobiles décrits dans l'état de l'art, il est possible de borner la mobilité des composants dans des portions de l'espace qui représentent les limites fonctionnelles du système. Ces régions constituent des *espaces de mobilité* à l'intérieur desquels les composants du système évoluent à l'exécution. De la même manière, un ensemble d'*espaces accessibles* délimite les recouvrements des couvertures spatiales de chaque région de l'environnement. À la conception, les espaces accessibles et les espaces de mobilité représentent les limites fixées du comportement "normal" du système. Lors de l'exécution, le système garde un fonctionnement cohérent avec sa description conceptuelle lorsque :

- un composant est restreint à l'intérieur du périmètre de son espace de mobilité ;
- la couverture spatiale une région de l'environnement est inclue dans l'espace accessible cette région.

Pour décrire les espaces accessibles et de mobilité, les concepteurs déterminent à l'avance les évolutions possibles des régions de l'environnement et des composants générateurs des régions d'exécution. À chaque région de l'environnement est associé un espace accessible. Quelle que soit une région d'intérêt ou d'exécution $reg_x \in ENVGEO$, l'espace accessible retourné par la fonction $ACC(reg_x)$ recouvre l'ensemble des localisations de l'environnement qui pourront faire partie de la couverture de reg_x à un moment donné de l'exécution.

11. Le choix de minorer l'impact d'un échange entre utilisateurs en considérant qu'ils appartiennent tous à un même "groupe" représenté par une plate-forme générique sera équilibré par une approche de collaboration multi-utilisateurs pour la définition de la dimension "utilisateurs" du contexte d'usage (cf. Chap V).

En plus des espaces accessibles aux régions du systèmes, les espaces de mobilité désignent les couvertures spatiales recouvrant les déplacements possibles des composants. Quel que soit un composant de la plate-forme $cmp_x \in PLATFORM$, l'espace de mobilité retourné par la fonction $MOB(cmp_x)$ désigne l'ensemble des localisations où le composant cmp_x peut être situé à l'exécution.



(a) Régions des utilisateurs (b) Zone de navigation interdite (c) Régions des traitements (d) Région des données

Figure III.6 – Espaces accessibles pouvant être occupés par les régions et espaces de mobilité pouvant être occupés par les composants

Dans le cas du système de suivi côtier, les espaces accessibles et les espaces de mobilité représentés en figure III.6 ont été établis par les concepteurs en coordination avec des futurs utilisateurs et les commanditaires du système.

Les utilisateurs du système, des marins naviguant le long de la côte, sont limités par celle-ci. Les frontières de l'espace de mobilité $MOB(C\mathrm{u}_*)$ des composants clients, représentés par le composant générique $C\mathrm{u}_* \in PLATFORM$, suivent la ligne de côte. L'espace accessible $ACC(Region(C\mathrm{u}_*)) = ACC(R\mathrm{u}_*)$ de la région d'exécution $R\mathrm{u}_* \in ENVEXEC$ générée par ce composant est à peine plus large que l'espace de mobilité. Les concepteurs estiment que les capacités de transmission des clients leur permettent d'émettre depuis la côte vers l'intérieur des terres, dans les limites raisonnables des capacités d'un réseau Wi-Fi (Fig. III.6a).

De la même manière, le composant serveur de données $C\mathrm{d}_1$, mobile à l'exécution, se déplace le long d'une route est/ouest depuis la côte. Son espace de mobilité $MOB(C\mathrm{d}_1)$ et l'espace accessible $ACC(R\mathrm{d}_1)$ de la région qu'il génère et dans lequel le composant transmettra des informations sont représentés par une couverture spatiale de forme oblongue d'est en ouest (Fig. III.6d).

Les composants de traitement et sont fixes à l'exécution. Leurs espaces de mobilité $MOB(C\mathrm{p}1_1)$ et $MOB(C\mathrm{p}2_1)$ se limitent à l'emplacement des composants eux-mêmes (Fig. III.6c). Les concepteurs estiment que les espaces accessibles $ACC(R\mathrm{p}1_1)$ et $ACC(R\mathrm{p}2_1)$ des régions de traitements générées par ces composants sont aussi figés à l'exécution. Les composants $C\mathrm{p}1_1$ et $C\mathrm{p}2_1$ conservent leur capacité à émettre dans $R\mathrm{p}1_1$ et $R\mathrm{p}2_1$ tant que l'un et l'autre font partie de la plate-forme à l'exécution.

L'espace accessible $ACC(R\mathrm{s}_1)$ de la région d'intérêt est établi à partir de données de positionnement de la zone interdite à la navigation (Fig. III.6b). Par exemple, les concepteurs peuvent étudier un historique des positions de cette région et estimer une enveloppe maximale de positionnement. Souvent, il arrive que l'espace accessible d'une région d'intérêt soit fixe à

l'échelle de temps d'une exécution du système. Par exemple, les frontières d'une ville désignent une région d'intérêt dont l'espace accessible est identique la région d'intérêt elle-même puisque les frontières de la ville ne varient pas à l'échelle d'une exécution.

La description des espaces accessibles et des déplacements des composants peut être relativement souple. Notamment, les couvertures spatiales des espaces accessibles et des espaces de mobilité peuvent être légèrement amplifiées à la conception sans que cela ne nuise à l'analyse géographique par raisonnement spatial. Lors de cette étape, l'essentiel est de représenter des relations réalistes entre les espaces accessibles des régions et les espaces de mobilité des composants. Par exemple, une région d'intérêt représentant la zone de course interdite, dont la couverture aurait été prévue un peu plus large garderait malgré tout sa relation d'intersection avec l'espace de mobilité des composants clients des utilisateurs.

III.3.3 Analyse spatiale

La description de l'exécution du système a montré que la multiplication du nombre de régions de l'environnement géographique, liée à l'augmentation du nombre de composants sur lesquels le système est distribué, fait croître très rapidement le nombre de cas de contextes d'exécution différentiables. Si la mobilité est généralisée à l'ensemble du système (composants, régions d'exécution et régions d'intérêt), et que cette mobilité n'est pas restreinte à certaines portions de l'espace, alors si n représente le nombre de régions de l'environnement, il existe $2^{C_2^n}$ cas de contextes d'exécution globaux différentiables (cf. III.2.4). À l'exclusion des systèmes distribués les plus simples, il devient vite délicat d'étudier les unes après les autres toutes ces situations à la conception. Plus le système contient de régions, et plus il devient difficile de définir un comportement propre à chacun de ces contextes. Afin de limiter cette croissance, des contraintes au niveau de la mobilité des régions ont été introduites par la définition des espaces accessibles et de mobilité. L'étape de modélisation spatiale construit à partir de ces espaces accessibles l'ensemble des relations potentielles entre des régions lors de l'exécution. Cet ensemble détermine les cas de contextes qui pourront être rencontrés à l'exécution.

Relations entre espaces accessibles

Les relations entre espaces accessibles sont introduites comme une première méthode de simplification de l'ensemble des situations contextuelles, en conservant uniquement pour la suite de l'analyse conceptuelle les contextes qui sont physiquement réalisables à l'exécution. Par exemple, en considérant des composants mobiles de la plate-forme, s'ils évoluent dans des espaces de mobilité distants, alors les régions d'exécutions qu'ils génèrent ne seront jamais en relation. Les cas de contextes d'exécution qui incluent ces relations peuvent être soustraits de l'ensemble des contextes : ils ne seront jamais mesurés à l'exécution. Plus généralement, un jeu de règles peut être appliqué pour filtrer les contextes d'exécution. Dans la suite, la notion de fixité d'une région est définie et les règles de filtrage des contextes sont énoncées et illustrées.

Propriété de fixité des régions Du point de vue de l'analyse conceptuelle, une région de l'environnement est fixée à l'exécution dans deux situations :

- une région d'intérêt $reg_y \in ENVINT$ est fixe si les données qui la définissent ne varient pas lors de l'exécution. Cette fixité se traduit au niveau spatial par l'équivalence entre l'espace accessible de cette région et sa couverture spatiale à n'importe quel moment de l'exécution (c-à-d. $\forall\ t_x \in T\ :\ Cover(reg_y, t_x) = ACC(reg_y)$) ;
- une région d'exécution $reg_y \in ENVEXEC$ est fixe si tout à la fois son composant générateur $cmp_y = Component(reg_y)$ est immobile, et si sa couveture spatiale ne varie pas à l'exécution. Cette condition de fixité est vérifiée quand l'espace accessible est égal à la couverture de la région d'exécution quel que soit $t_x \in T$ et quand la localisation du composant est identique à tout instant (c-à-d. $\forall\ t_x \in T\ :\ Cover(reg_y, t_x) = ACC(reg_y) \wedge \{Loc(cmp_y, t_x)\} = MOB(cmp_y)$).

Les propriétés de fixité des régions sont établies à la conception, lors de l'étude des mouvements des régions et des composants pour déterminer les espaces accessibles. Dans la suite, la fonction $Fixed(reg_y)$ retourne 1 si la région $reg_y \in ENVGEO$ est fixe et 0 sinon. Dans l'exemple du système de suivi côtier, les concepteurs qui s'appliquent à déployer le système sur une infrastructure en place déterminent la fixité des régions de traitement. Les composants serveurs $C\mathrm{p}1_1$ et $C\mathrm{p}2_1$ ne sont pas mobiles et les couvertures spatiales de leurs régions d'exécution $R\mathrm{p}1_1$ et $R\mathrm{p}2_1$ restent constantes. Les autres régions d'exécution ou régions d'intérêt sont mobiles. Formellement :

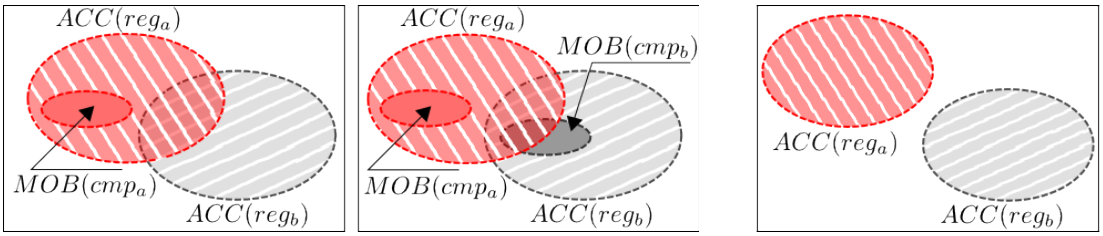
$$Fixed(R\mathrm{s}_1) = Fixed(R\mathrm{u}_*) = Fixed(R\mathrm{d}_1) = 0 \qquad Fixed(R\mathrm{p}1_1) = Fixed(R\mathrm{p}2_1) = 1$$

Ensemble de règles sur les espaces accessibles Les règles proposées sont appliquées par les concepteurs après avoir déterminé les composants de la plate-forme, les régions de l'environnement, les espaces accessibles de ces régions et les espaces de mobilité des composants. Elles s'appliquent à la description de relations binaires entre espaces qui forcent soit l'exclusion soit l'inclusion de relations entre régions dans les contextes mesurés à l'exécution. Plus formellement, **l'inclusion systématique d'une relation** entre deux régions $reg_a, reg_b \in ENVGEO$ signifie que lors de l'exécution $\forall\, t_x \in T,\ Relate(reg_a, reg_b, t_x) = 1$ et $(reg_a, reg_b) \in State(t_x)$. À l'inverse, **l'exclusion systématique d'une relation** entre deux régions reg_a et reg_b signifie que $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 0$ est systématiquement mesuré à l'exécution et $(reg_a, reg_b) \notin State(t_x)$. Dans la suite, les ensembles $Included$ et excluded répertorient les paires de régions systématiquement en relation ou systématiquement hors relation :

$$Excluded = \left\{(reg_a, reg_b) \in ENVGEO^2 \mid \forall\ t_x \in T :\ relate(reg_a, reg_b, t_x) = 1\right\}$$
$$Included = \left\{(reg_a, reg_b) \in ENVGEO^2 \mid \forall\ t_x \in T :\ relate(reg_a, reg_b, t_x) = 0\right\}$$

Dans le modèle proposé, deux relations d'exclusion systématique sont retenues. Les paires de régions auxquelles s'appliquent ces règles d'exclusion sont répertoriés par l'ensemble $Excluded$ et ne font jamais partie des contextes mesurées à l'exécution. Les règles d'exclusion s'appliquent quelles que soient les propriétés de fixité des régions :

1. un composant doit évoluer dans la couverture d'une région d'exécution ou d'une région d'intérêt de l'environnement pour qu'une relation puisse être établie entre sa région d'exécution et la région considérée de l'environnement (Fig. III.7a). Plus formellement, soient $reg_a \in ENVEXEC$ une région d'exécution et $reg_b \in ENVGEO$, une région d'intérêt ou d'exécution. Lorsque l'espace de mobilité $MOB(Component(reg_a))$

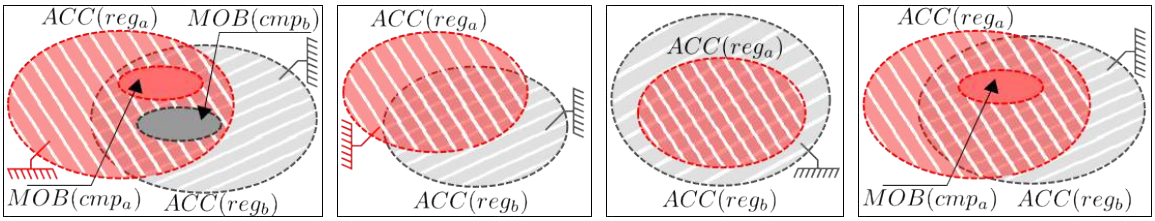


(a) (b)

Figure III.7 – Illustration des règles d'exclusion systématique des relations entre régions de l'environnement

du composant générateur de reg_a est en dehors de l'espace accessible $ACC(reg_b)$, alors les régions reg_a et reg_b ne seront pas en relation à l'exécution et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 0$;

2. les couvertures de deux régions d'intérêt doivent pouvoir s'intersecter pour que ces régions soient en relation (Fig. III.7b). Plus précisément, soient deux régions d'intérêt $reg_a, reg_b \in ENVINT$ et leurs espaces accessibles $ACC(reg_a)$ et $ACC(reg_b)$. Si ces espaces accessibles ne s'intersectent pas, alors les régions reg_a et reg_b ne se croiseront pas à l'exécution et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 0$.



(a) (b) (c) (d)

Figure III.8 – Illustration des règles d'inclusion systématique des relations entre régions de l'environnement

De la même manière, l'approche conceptuelle proposée retient quatre règles d'inclusion automatique de relations entre régions, particulièrement lorsque les régions considérées sont fixes et que leurs espaces accessibles correspondent à ces régions elles-mêmes [12]. Les paires de régions auxquelles s'appliquent ces règles d'inclusion sont répertoriées par l'ensemble $Included$ et sont systématiquement présentes dans les contexte mesurés à l'exécution :

1. deux régions d'exécution sont en relation systématique si à tout instant le composant générateur de l'une est inclus dans la couverture spatiale de l'autre (Fig. III.8a). Plus formellement, soient $reg_a, reg_b \in ENVEXEC$, deux régions d'éxecution et leurs composants générateurs $cmp_a = Component(reg_a)$ et $cmp_b = Component(reg_b)$. Avec $Fixed(reg_a) = Fixed(reg_b) = 1$, si l'espace de mobilité $MOB(cmp_b)$ est inclus dans l'espace accessible $ACC(reg_a)$ et qu'à l'inverse, l'espace $ACC(reg_b)$ recouvre $MOB(reg_a)$, alors les régions reg_a et reg_b sont en relation systématique et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 1$;
2. deux régions d'intérêt fixes dont les espaces accessibles s'intersectent auront leurs couvertures spatiales en intersection systématique à l'exécution (Fig. III.8b). Plus formellement, soient $reg_a, reg_b \in ENVINT$, deux régions d'intérêt telles que $Fixed(reg_a) =$

12. Dans la figure III.8, les symboles "Masse" signifient la fixité de la région qu'ils désignent.

$Fixed(reg_b) = 1$. lorsque les espaces accessibles $ACC(reg_a)$ et $ACC(reg_b)$ s'intersectent, la relation de ces régions fixes est systématique et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 1$;

3. une région d'intérêt dont l'espace accessible est entièrement contenu dans l'espace accessible d'une seconde région d'intérêt fixe est toujours en relation avec celle-ci (Fig. III.8c). Plus formellement, soient $reg_a, reg_b \in ENVINT$ deux régions d'intérêt telles que $Fixed(reg_b) = 1$. Si l'espace accessible $ACC(reg_b)$ recouvre intégralement l'espace accessible $ACC(reg_a)$, alors les régions reg_a et reg_b sont en relation systématique et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 1$;
4. une région d'exécution est en relation systématique avec une région d'intérêt fixe dès lors que le composant générateur de la région d'exécution est contraint à l'intérieur de la couverture de la région d'intérêt (Fig. III.8d). Plus formellement, soit $reg_a \in ENVEXEC$, une région d'exécution générée par le composant $cmp_a = Component(reg_a)$, et $reg_b \in ENVINT$, une région d'intérêt fixe (c-à-d. $Fixed(reg_b) = 1$). Si l'espace de mobilité $MOB(cmp_a)$ du composant est contenu dans la l'espace accessible $ACC(reg_b)$, alors les régions reg_a et reg_b sont en relation systématique et $\forall\ t_x \in T,\ Relate(reg_a, reg_b, t_x) = 1$.

Les concepteurs du système de suivi côtier effectuent une superposition des régions accessibles et recherchent dans les relations topologiques obtenues des cas d'application des règles d'inclusion et d'exclusion (Fig. III.9(a)). Dans ce cas, 1) les espaces de mobilité $MOB(C\mathrm{p1}_1)$ et $MOB(C\mathrm{p2}_1)$ sont systématiquement en dehors de l'espace accessible $ACC(R\mathrm{s}_1)$ de la zone d'interdiction de navigation (Fig. III.9(b & c)), 2) l'espace de mobilité $MOB(C\mathrm{p2}_1)$ est en dehors de l'espace accessible $ACC(R\mathrm{u}_*)$ de la région des utilisateurs (Fig. III.9(e)), 3) les espaces accessibles $ACC(R\mathrm{p1}_1)$ et $ACC(\mathrm{p2}_1)$ ne recouvrent pas mutuellement les espaces de mobilité $MOB(C\mathrm{p1}_1)$ et $MOB(C\mathrm{p2}_1)$ (Fig. III.9(e)). D'après la première règle d'exclusion systématique, ces trois cas de figure éliminent la possibilité d'un ensemble de relations entre régions lors de l'exécution. Plus spécifiquement :

$$\forall\ t_x \in T,\ Relate(R\mathrm{p1}_1, R\mathrm{s}_1, t_x) = 0 \qquad Relate(R\mathrm{p2}_1, R\mathrm{s}_1, t_x) = 0$$
$$\forall\ t_x \in T,\ Relate(R\mathrm{p2}_1, R\mathrm{u}_*, t_x) = 0 \qquad Relate(R\mathrm{p1}_1, R\mathrm{p2}_1, t_x) = 0$$

L'ensemble des relations systématiquement incluses reste vide (c-à-d. $Included = \emptyset$). L'ensemble des relations automatiquement exclues reflète les quatre impossibilités de relations avec les régions des traitements :

$$Excluded = \big\{(R\mathrm{p1}_1, R\mathrm{s}_1), (R\mathrm{p2}_1, R\mathrm{s}_1), (R\mathrm{p2}_1, R\mathrm{u}_*), (R\mathrm{p1}_1, R\mathrm{p2}_1)\big\}$$

III.3.4 Description conceptuelle des contextes d'exécution

A la conception, l'écriture des contextes d'exécution défini les ensembles de relations entre régions susceptibles de se produire lors de l'exécution. Dans la suite de l'analyse conceptuelle du système, une réponse adaptative particulière devrait être implémentée pour chacun des contextes décrits. En pratique, les concepteurs étudient la validité des relations entre les régions de $ENVGEO$ et énoncent une série de configurations topologiques. Un contexte d'exécution s'écrit comme un ensemble de paires de régions en relation à l'exécution. La différence essentielle par rapport à la description contextuelle donnée à l'exécution provient

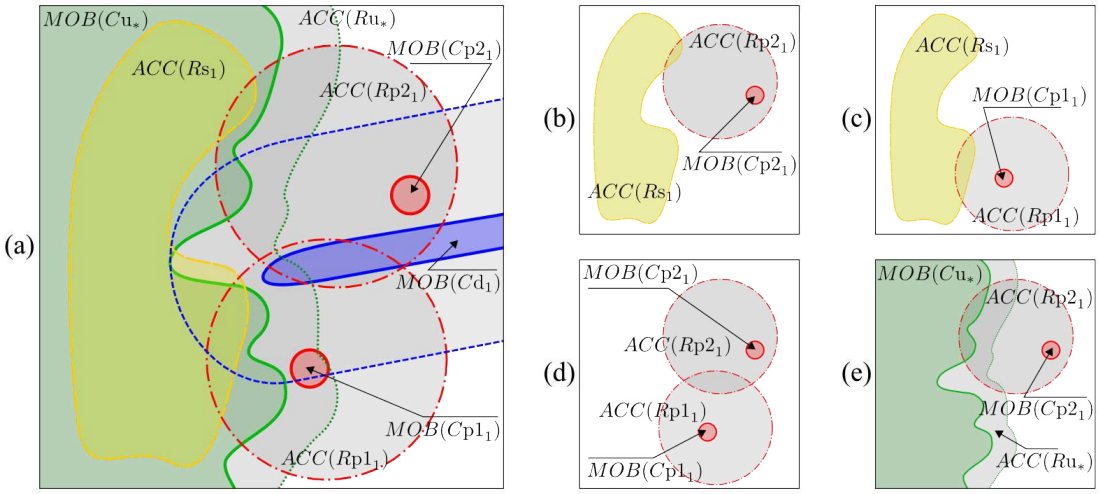


Figure III.9 – Superposition des espaces accessibles et des espaces de mobilité pour la recherche des règles d'inclusion et d'exclusion systématiques

du fait que les régions de rôles génériques ne sont pas divisées en fonction du nombre de composants qui implémentent ce rôle. De plus, un contexte d'exécution décrit à la conception ne sera pas forcément reproduit à l'exécution. La seule certitude motivant la description conceptuelle des contextes est qu'il existe une probabilité que le système rencontre telle ou telle configuration à l'exécution et doive s'y adapter.

Les règles d'inclusion ou d'exclusion définissent quelles paires de régions se retrouveront automatiquement dans chaque contexte d'exécution et quelles sont celles qui en seront systématiquement exclues. L'ensemble des paires de régions dont les relations peuvent varier à l'exécution est noté "$Pairs$" et se définit comme :

$$Pairs = \{(reg_a, reg_b) \in ENVGEO^2 \mid (reg_a, reg_b) \notin Included \cup Excluded\}$$

Les contextes d'exécution que le système pourrait rencontrer sont rassemblés dans l'ensemble $STATES$. Il est constitué par l'ensemble des parties de $Pairs$ auxquelles s'ajoutent les paires de régions en relations systématiques de $Included$. En considérant que $\mathcal{P}(SET)$ désigne l'ensemble des parties de l'ensemble SET [13], l'ensemble des contextes d'exécution $STATES$ s'écrit :

$$STATES = \{\mathcal{P}(Pairs) \cup Included\}$$

Le nombre d'états d'exécution observables est donné par le cardinal de cet ensemble. Avec $|ENVGEO|$ désignant le nombre de régions de l'environnement décrites lors de la conception :

$$|STATES| = 2^{|Pairs|} = 2^{\left(C_2^{|ENVGEO|} - (|Included| + |Excluded|)\right)}$$

13. L'ensemble $\mathcal{P}(SET)$ des parties d'un ensemble SET est défini par : $\mathcal{P}(SET) = \{A : A \subseteq SET\}$. Par exemple, si $SET = \{a, b, c\}$, alors $\mathcal{P}(SET)$ est l'ensemble des combinaisons de lettres $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, SET\}$.

Par rapport au nombre $2^{C_2^n}$ de configurations contextuelles qu'il est possible d'imaginer avec un environnement décomposé à la conception en n régions, les relations d'inclusion et d'exclusion éliminent $2^{C_2^n} - |STATES|$ cas qui, dans les faits, ne pourront pas être observés.

Dans l'exemple du système de suivi côtier, pour lequel les ensembles *Included* et *Excluded* sont respectivement vides et égaux à $\{(R\mathrm{p}1_1, R\mathrm{s}_1), (R\mathrm{p}2_1, R\mathrm{s}_1), (R\mathrm{p}2_1, R\mathrm{u}_*), (R\mathrm{p}1_1, R\mathrm{p}2_1)\}$ et pour lequel l'environnement décrit à la conception contient cinq régions (quatre d'exécution : $ENVEXEC = \{R\mathrm{u}_*, R\mathrm{d}_1, R\mathrm{p}1_1, R\mathrm{p}2_1\}$; une d'intérêt : $ENVINT = \{R\mathrm{s}_1\}$), le nombre de cas de contextes d'exécution $|STATES|$ vaut :

$$|STATES| = 2^{\left(C_2^{|ENVGEO|} - (|Included| + |Excluded|)\right)} = 2^{\left(C_2^5 - 4\right)} = 2^{10-4} = 64$$

Appliquée au cas du système de suivi côtier, la modélisation spatiale des mobilités des régions fait passer l'ensemble des contextes d'exécution d'un potentiel de 1024 situations [14] à 64 cas réellement observables et potentiellement utiles pour distinguer différents niveaux d'adaptation. Sans pour l'instant s'intéresser à la sémantique des contextes du système, l'analyse spatiale des comportements des composants de la plate-forme permet de réduire la complexité conceptuelle. Par cette étude préalable à l'analyse fonctionnelle, les concepteurs répertorient les contextes dans lesquels le système évoluera à l'exécution et excluent les contextes qui ne se produiront pas.

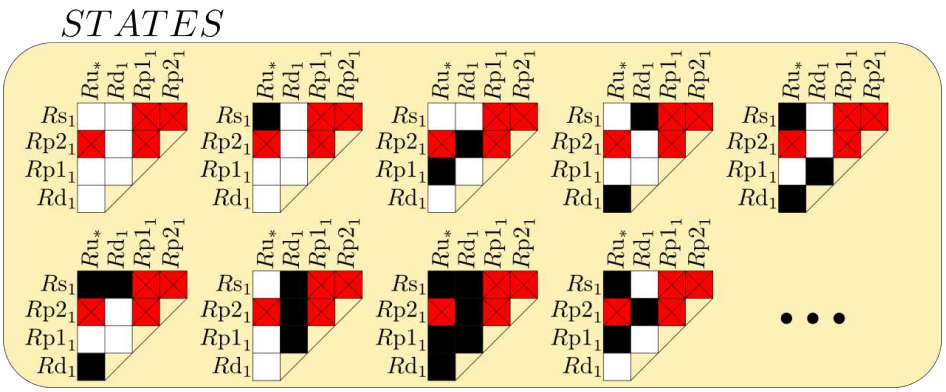


Figure III.10 – Représentation graphique de quelques contextes d'exécution de l'ensemble $STATES$

La figure III.10 propose une représentation graphique de l'ensemble des contextes d'exécution pour le système de suivi côtier. Chaque matrice y représente un contexte ; les cas d'exclusion automatique entre deux régions se retrouvent dans chacune et ont été marqués en rouge barré d'une croix. De la même manière, les cas d'inclusion automatique de relation seraient marqués en vert. Comme lors de la description à l'exécution, une cellule blanche signale la non relation de deux régions. A l'inverse, une cellule noire signale leur relation.

14. avec cinq régions décrites à la conception, $2^{C_2^5} = 1024$.

III.4 Conclusion

Ce chapitre a proposé une approche géographique pour la description de l'exécution d'un système mobile et distribué et établi la première étape d'une démarche de conception pour la mise en œuvre de ces systèmes.

Dans un premier temps, la plate-forme d'un système distribué a été décomposée en composants répartis en familles fonctionnelles. Ces composants mobiles génèrent un ensemble de régions d'exécution qui représentent les limites des capacités de diffusion des composants. Des régions d'intérêt sont définies par rapport aux données localisées traitées par le système. Elle permettent de caractériser des situations contextuelles où un composant se situe à l'intérieur ou à l'extérieur de zones sémantiquement significatives de l'environnement du système. Les régions d'exécution et les régions d'intérêt forment l'environnement du système. Leurs relations à un instant donné sont appelées "contexte d'exécution". A chaque contexte d'exécution, correspond possiblement un comportement du système, et des opérations d'adaptation sont effectuées lors de la transition entre différents contextes.

Le défi conceptuel posé par la désignation des contextes avant l'exécution est adressé par l'étude des espaces accessibles dans lesquels évoluent les régions de l'environnement. Les relations qui s'établissent entre ces espaces accessibles, alliées à la fixité de certaines régions, permettent d'énoncer une série de règles d'inclusion ou d'exclusion de relations entre régions. Ces relations forcées restreignent l'ensemble des contextes d'exécution aux situations "vraisemblables" d'un point de vue géographique.

Le cadre illustratif proposé par le système de suivi côtier, décrit à la fois lors de l'exécution puis à la conception, illustre cette approche géographique. Les concepteurs du système, qui s'adaptent à une plate-forme en place, déterminent les types de composants et de régions puis les espaces de mobilité et de diffusion. La définition d'une région générique pour désigner l'espace où évoluent les composants clients et l'application des règles d'inclusion ou d'exclusion de relation simplifient drastiquement l'ensemble des contextes à modéliser pour distinguer différents niveaux d'adaptation.

Dans la suite de la modélisation, détaillée au chapitre IV, cette approche conceptuelle par la description géographique de l'environnement d'exécution intègre le niveau plus général d'une démarche de conception centrée utilisateur. La connaissance des tâches des utilisateurs réduit encore l'ensemble des contextes d'exécution en éliminant les contextes qui peuvent se produire, mais qui ne sont pas d'un intérêt immédiat pour l'adaptation du système. Dans une seconde étape de simplification, les contextes restants sont groupés en fonction des processus d'adaptation qu'ils déclenchent et des objets adaptés. Des comportements adaptatifs sont associés à chacun de ces sous-ensembles contextuels à l'issue de la combinaison de l'approche centrée utilisateur et de la modélisation géographique.

CHAPITRE IV

Intégration de l'environnement spatial dans la modélisation d'un système interactif

Sommaire

IV.1 Modèle de conception pour la mise en œuvre des contextes d'exécution 110
IV.1.1 Cadre de conception unifié : une approche pour le développement de systèmes interactifs 110
IV.1.2 Cadre de conception étendu : proposition pour l'intégration de la modélisation géographique 112
IV.2 Étapes de corrélation avec les contextes d'exécution 114
IV.2.1 Primitives du cadre de conception unifié : scénario nominal, modèle des tâches et des informations 115
IV.2.2 Recherche d'équivalences entre contextes d'exécution 118
IV.2.3 Arbre des tâches situé 128
IV.3 Conclusion 135

La caractérisation des contextes d'exécution par la prise en compte de l'espace géographique d'un système, présentée au chapitre précédent, permet de décrire les conditions que ce système rencontrera lors de son exécution. Les relations qui s'établissent entre des régions d'exécution et d'intérêt déterminent l'ensemble des contextes qu'il est nécessaire d'intégrer lors de la conception d'un système. La démarche d'intégration proposée consiste à enrichir un modèle de conception reconnu et utilisé, adapté à la modélisation de systèmes interactifs, pour lui adjoindre une analyse de différents niveaux d'exécution repérés lors de la modélisation spatiale des relations entre régions du système.

Ces étapes de corrélation entre la démarche de conception d'un système interactif et les différents niveaux d'exécution disponibles dans l'environnement du système constituent le cœur de ce chapitre. Dans un premier temps, les contextes d'exécution provoquant des

adaptations équivalentes sont rassemblés en groupes appelés "chaînes d'équivalence". Dans un second temps, ces chaînes d'équivalence sont corrélées à l'analyse des tâches menée par les concepteurs du système en concertation avec les futurs utilisateurs. Plus spécifiquement, pour structurer les actions réalisables par les utilisateurs, une organisation des tâches sous la forme d'un arbre articulé logiquement est retenue. Les tâches formant les embranchements et les feuilles de cet arbre sont estampillées par les contextes qui garantissent les ressources nécessaires à leur bon déroulement. Potentiellement, certaines chaînes n'occupent aucune place dans l'arbre et il existe dans ce cas des situations d'exécution pour lesquelles aucune réponse fonctionnelle n'a encore été prévue. Pour intégrer ces situations, de nouvelles tâches forment des branches et sont ajoutées à l'arbre initial. Ces tâches représentent des alternatives au cas d'exécution nominal et garantissent que les utilisateurs disposent d'un minimum de service quelle que soit leur situation d'exécution.

Le résultat de l'association entre l'analyse des tâches et les contextes d'exécution est schématisée par un **arbre des tâches situé**. Cet arbre résume à la fois les tâches des utilisateurs et du système, les informations manipulées par ces tâches et les contextes d'exécution dans lesquels peuvent être effectuées ces tâches. L'arbre des tâches situé alimente l'ingénierie du système, d'une part du côté du client en fournissant un guide pour la structuration du dialogue et de l'interface et, d'autre part, du côté des tiers métiers et données en proposant une répartition des procédures et des données sur les composants du système. Ce chapitre décrit les différentes étapes de la construction d'un arbre des tâches situé dans le cas d'un système mobile et distribué.

IV.1 Modèle de conception pour la mise en œuvre des contextes d'exécution

Cette section présente un cadre de conception "unifié" et dédié à la modélisation de systèmes interactifs. Ce cadre de conception est utilisé comme support pour l'intégration des contextes d'exécution décrits par l'étude géographique de l'environnement et des relations entre régions. Le cadre de conception résultant, dit "étendu", offre un modèle cohérent pour décrire des systèmes contextuels distinguant les dimensions d'usage et d'exécution.

IV.1.1 Cadre de conception unifié : une approche pour le développement de systèmes interactifs

Modéliser un système interactif implique d'identifier et de prendre en compte les besoins et les attentes des futurs utilisateurs pour initier l'ingénierie d'un système qui réponde à leurs besoins. Les démarches de *conception centrées utilisateur* () et leurs dérivées comme la *conception participative* () ou la *conception par scénarios* () intègrent le point de vue des utilisateurs dans le processus de développement logiciel. En général, un échantillon représentatif de la population des futurs utilisateurs est associé à l'équipe de concepteurs et développeurs. Ensemble, ils spécifient les étapes de la conception (p. ex. par l'analyse des tâches et des intentions des utilisateurs) et décrivent le profil et les besoins d'un stéréotype d'utilisateur (). Ces descriptions sont prises en compte par les concepteurs et les développeurs pour proposer

des solutions acceptables du point de vue de l'interaction Humain-machine et du contenu informationnel présenté.

Plusieurs démarches de conception de systèmes d'information interactifs et contextuels qui impliquent l'utilisateur ont été proposées (; ;). Elles se conforment au cadre de référence unifié pour la modélisation d'un système interactif proposé par . En premier lieu, elles définissent les objectifs des utilisateurs du système, puis énoncent les étapes nécessaires à l'accomplissement de ces objectifs par un ensemble de *tâches* s'appuyant sur des *informations* caractérisant les données d'entrée-sortie utilisées par le système. Cette description est détaillée au sein d'un document textuel : *le scénario nominal*. Il s'agit d'une description informelle du fonctionnement du système construite à partir des besoins des utilisateurs. Dans ce sens, les experts du domaine, les concepteurs du système et éventuellement les futurs utilisateurs se réunissent en groupes de travail pour écrire ce scénario. Ce document servira de fil conducteur à l'équipe de développement pour la suite du processus de conception du système. Le scénario nominal doit être aussi détaillé que possible. Son analyse doit permettre de spécifier les fonctionalités implémentées par un premier prototype du système.

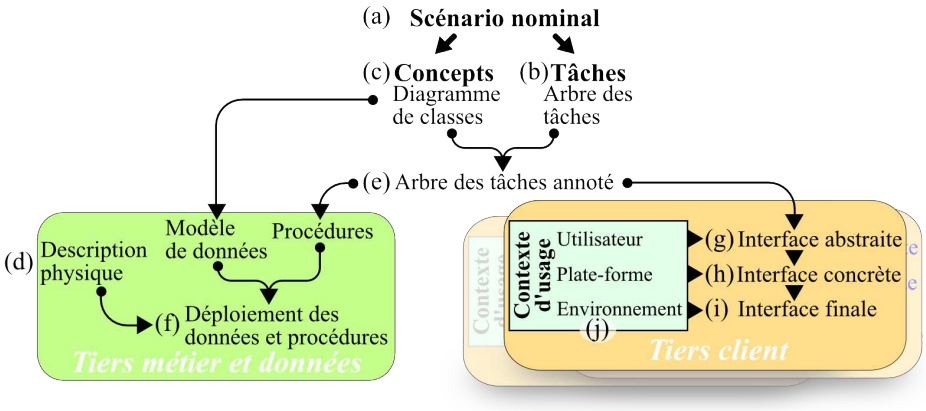


Figure IV.1 – Cadre de conception et de mise en œuvre d'un système d'information sensible au contexte

L'écriture du scénario nominal pour la description des tâches s'établit comme le point d'entrée d'une large majorité des méthodes de conception centrées utilisateur () (Fig. IV.1(a)). En partant de cette spécification, les tâches et les informations du domaine sont décomposées et modélisées selon une conception descendante :

1. organisation des informations du système en un modèle conceptuel de données (Fig. IV.1(c)). Par exemple, les informations sont structurées au sein d'un schéma de classes ;
2. décomposition et organisation des tâches de l'utilisateur par un modèle fonctionnel en général sous la forme d'un arbre articulé par des opérateurs d'ordonnancement (Fig. IV.1(b)). Décoration de cet arbre par les concepts manipulés en entrée et sortie (Fig. IV.1(e)) ;
3. conception des tiers métier et données : cette étape aborde le choix de la distribution physique des composants du système (Fig. IV.1(d)), l'implémentation du modèle de données et du noyau fonctionnel, et le déploiement sur l'architecture (Fig. IV.1(f)) ;
4. conception du tiers client () : l'interface abstraite décompose l'interaction graphique avec le système en dédiant un espace de dialogue à chaque tâche (Fig. IV.1(g)). L'interface concrète organise les éléments d'interaction (p. ex. listes, menus, boutons, etc.)

au sein de chaque espace de dialogue (Fig. IV.1(h)). Enfin, l'interface finale propose une implémentation logicielle de l'interface concrète, en se reposant sur une bibliothèque de composants graphiques (Fig. IV.1(i)).

Les étapes de conception des tiers "métier", "données" et "client" peuvent être semi-automatiques. Pour le tiers métier, l'organisation des informations est assortie de méthodes de manipulation pour former un diagramme de classes, et un squelette de code est généré. Pour le tiers client, les différents niveaux d'abstraction de l'interface utilisateur sont générés à partir d'une description formalisée des tâches (). L'ensemble du processus de conception est éventuellement répété. Chaque itération ajuste ou enrichit le cycle précédent en complétant les scénarios d'utilisation et en répercutant ces modifications dans les différentes étapes de l'implémentation, jusqu'à obtenir une version du système qui satisfasse au mieux les exigences et les besoins des utilisateurs ().

Le contexte d'usage, qui adapte dynamiquement les capacités interactives du système, est pris en compte au niveau de la conception du tiers client. Le cadre de référence unifié distingue trois dimensions contextuelles dans l'environnement immédiat de l'utilisateur : le contexte utilisateur, le contexte de la plate-forme client, et le contexte d'environnement d'utilisation (Fig. IV.1(j)). Chaque triplet du contexte d'usage engendre la conception d'une partie cliente qui est *adaptée* pour l'utilisateur considéré.

IV.1.2 Cadre de conception étendu : proposition pour l'intégration de la modélisation géographique

Selon le cadre de conception unifié, la modélisation d'un système interactif est principalement centrée sur l'ingénierie de l'interaction humain-machine du tiers client et l'impact d'un contexte dynamique reste cantonné aux utilisateurs et à leur environnement proche, c'est-à-dire à des *contextes d'usage* particuliers. Selon cette approche du cadre de conception unifié, percevoir l'environnement présente un intérêt si l'exécution doit être modifiée pour s'adapter à une situation nouvelle à la suite d'un changement dans l'une des dimensions du contexte d'usage (c-à-d. dimensions de l'utilisateur, de la plate-forme ou de l'environnement proche (Chap. II.2.2)). Ces trois dimensions du contexte d'usage sont répercutées lors des dernières étapes de la modélisation, lorsque les niveaux d'interaction avec les utilisateurs sont spécifiés (Fig. IV.1(j)). Deux postulats du cadre de conception unifié justifient l'intégration du contexte d'utilisation lors des dernières étapes de la démarche de conception :

- le cadre de conception unifié considère que les composants métier et données ne sont pas contraints par l'environnement : à tout moment, ils devraient tenir leur rôle dans le système.
- de la même manière, le cadre de conception unifié considère que les tâches sont décrites à un haut niveau et ne sont pas sensibles aux variations du contexte d'usage.

Ces deux hypothèses, valides dans le cas du développement de logiciels interactifs non distribués ou statiques, sont mises à mal dans le cas des systèmes mobiles et distribués, sensibles aux aléas de l'environnement au niveau de l'ensemble des composants. Pour gérer au mieux ces aléas, et **pour déployer un système d'information distribué dans un environnement géographique dynamique, il est nécessaire de considérer, en plus du contexte d'usage, les variations du contexte d'exécution** au niveau de chaque composant. Le système déployé est fortement contraint par sa situation dans l'environnement, dans

lequel l'évolution de la localisation spatiale de ses composants joue un rôle prépondérant. Plus spécifiquement, un système distribué subit les variations d'un ensemble de contextes d'exécution qui n'apparaissent pas au sein du modèle de conception unifié [1]. Pour prendre en compte ces situations, **le cadre de conception d'un système d'information interactif est étendu et ajoute la description géographique de l'environnement comme entrée du processus de modélisation** (Fig. IV.2).

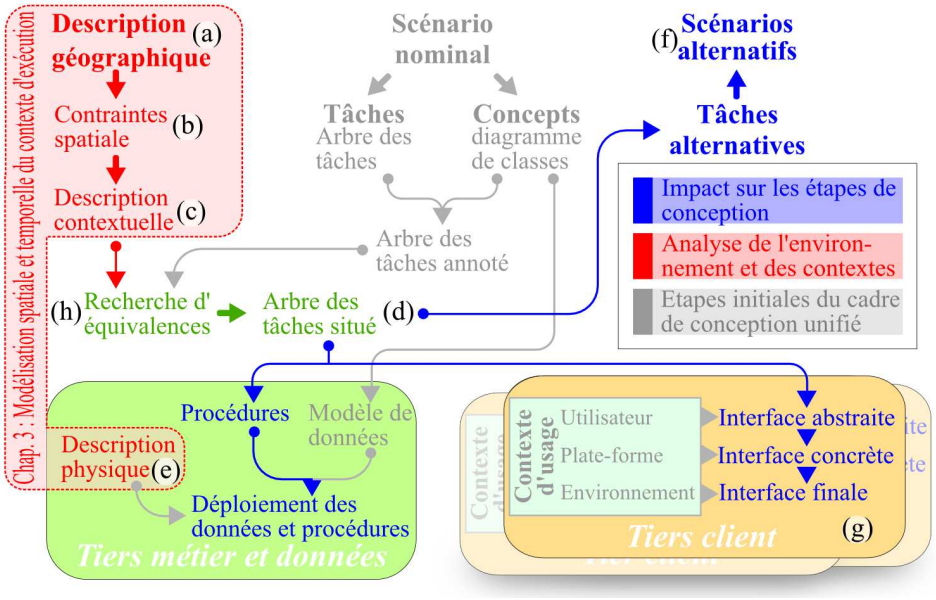


Figure IV.2 – Cadre de conception étendu

La proposition de cadre de conception étendu prend en compte les variations de l'environnement d'exécution du système sous la forme de contraintes qui s'ajoutent à la description des tâches issues du scénario nominal. Ces contraintes enrichissent l'arbre des tâches par de nouvelles branches correspondantes aux tâches effectuées dans les contextes d'exécution qui ne sont pas nominaux. Cette approche étendue du cadre de conception unifié permet de caractériser *l'influence de l'environnement d'exécution du système sur le déroulement du scénario nominal* sans perdre la capacité à exprimer différents contextes d'usage.

La description conceptuelle de l'environnement d'un système mobile et distribué par une approche géographique, présentée au chapitre III, modélise la dynamique des composants d'un système mobile par un ensemble de régions (Fig. IV.2(a)), où l'étude des contraintes et des équivalences entre configurations topologiques (Fig. IV.2(b)) détermine un ensemble de contextes d'exécution. Ces étapes caractérisent l'ensemble des situations d'exécution que le système peut rencontrer (Fig. IV.2(c)). Le cadre de conception étendu adjoint au cadre de conception unifié deux étapes supplémentaires de modélisation pour permettre l'intégration des différents contextes d'exécution :

1. En simplifiant à l'extrême son cadre d'utilisation, le modèle unifié offre un cadre d'analyse du contexte d'usage pertinent pour un contexte d'exécution donné, mais reste limité quand il s'agit de prendre en compte de multiples situations d'exécution.

- une première étape considère l'ensemble des contextes produits à l'issue de l'analyse des mobilités des régions, et recherche des équivalences en termes de modifications fonctionnelles entre deux ou plusieurs contextes d'exécution (Fig. IV.2(h)) ;
- les contextes ainsi groupés sont ensuite mis en relation avec les tâches de l'utilisateur et les informations manipulées au sein d'un arbre des tâches "situé". Ce nouvel outil de conception exprime les possibilités fonctionnelles offertes dans une situation d'exécution donnée par rapport aux tâches de l'utilisateur (Fig. IV.2(d)).

Lorsque des contextes d'éxecution ne sont liés à aucune tâche de l'arbre annoté, il existe des situations d'exécution potentielle qui n'ont pas été décrites par le groupe de réflexion à l'origine du scénario nominal. Afin que le système réponde de façon appropriée il est nécessaire de prévoir un comportement adapté à ces situations. De nouvelles tâches sont imaginées au sein de scénarios alternatifs (Fig.IV.2(f)). Elles seront effectuées à l'initiative du système ou de l'utilisateur lorsque les conditions ne permettront le déroulement du scénario nominal.

La description physique de l'architecture des composants sert de support au système de régions mis en place. En particulier, celle-ci doit permettre la détection des variations dans les intersections des régions d'exécution et d'intérêt pour que les changements de contextes d'exécution soient mesurés et répercutés au niveau du système (figure IV.2(e)). Par exemple, pour pouvoir détecter la relation d'une région d'exécution avec une région d'intérêt, le composant générateur de la région d'intérêt doit connaître et rendre disponible sa localisation au reste du système. En parallèle de cette description de la plate-forme, les tâches situées sont à l'origine de la partie cliente. La modélisation du tiers client décompose en actions d'interaction et de dialogue les tâches du scénario nominal, mais aussi les tâches alternatives. Comme l'ensemble des situations d'exécution est pris en compte et représenté dans un arbre des tâches commun, l'ensemble des cas d'exécution est établi au sein d'un outil commun. L'implémentation du client bénéficie de cette approche et la mise en œuvre d'une interaction humain-machine *cohérente* et *robuste*[2] face aux changements de contexte du système est facilitée (figure IV.2(g)).

IV.2 Étapes de corrélation avec les contextes d'exécution

Dans la suite de ce chapitre, le cadre de conception étendu est présenté plus en détail. Issues du cadre de conception unifié, les notions de scénario nominal, de modèle d'arbre des tâches et de structuration des informations sont illustrés dans le cas du système de suivi côtier introduit au chapitre précédent. PLus précisément, la corrélation entre la modélisation des contextes d'exécution et l'analyse des tâches s'effectue par le regroupement de contextes équivalents. Avec les chaînes de contextes comme point d'entrée, la construction de l'arbre des tâches situé est effectuée en trois temps : 1) distribution des procédures et des données sur la plate-forme, 2) analyse des conditions nécessaires pour le déroulement de l'arbre des tâches annoté, 3) augmentation de l'arbre des tâches annoté par de nouvelles branches qui expriment des comportements alternatifs au cas nominal.

Les déploiements des tiers métiers et client sur les composants de la plate-forme, et l'implémentation du dialogue et de l'interface Humain-machine à partir de l'arbre des tâches

2. La cohérence et la robustesse sont considérées au sens des critères d'évaluation de l'interaction Humain-machine ().

situé sont généralement liés au cas d'application choisi. Ces dernières étapes, centrées sur l'ingénierie du système, sont illustrées au chapitre VI.2.

IV.2.1 Primitives du cadre de conception unifié : scénario nominal, modèle des tâches et des informations

Le cadre de conception unifié met en place une méthodologie pour la construction d'un système interactif. Dans ce sens, la démarche est guidée dans son ensemble par la caractérisation des besoins des futurs utilisateurs du système. En appliquant une démarche "centrée utilisateur" à chaque étape de la modélisation prévue par le cadre de conception unifié, l'équipe de conception évalue les buts et les intentions des utilisateurs avant de proposer des services qui y répondent. Le premier rôle des concepteurs à l'initiation du processus est d'organiser une réponse fonctionnelle pour permettre à l'utilisateur d'atteindre ses objectifs. L'expression de ces objectifs reste un pré-requis à la démarche de conception et les **scénarios d'utilisation** constituent un outil pour y parvenir [3]. Le scénario nominal en particulier décrit textuellement l'ensemble des actions menées par l'utilisateur pour atteindre son objectif. Ces actions peuvent être déterminées à partir d'entretiens avec les utilisateurs d'un système pré-existant. Par exemple, le tableau IV.1 présente un entretien entre des utilisateurs marins et des concepteurs pour la modélisation d'un poste de navigation numérique.

Tableau IV.1 – Exemple d'entretien entre concepteurs et utilisateurs

[...]

MARIN 2 : Moi je commence par dire à l'opérateur machine de faire tourner le moteur à vide.

MARIN 1 : C'est vrai, je fais cela aussi. Et après je règle le cap sur le poste de navigation...

CONCEPTEUR 1 : Dans l'ordre : 1) Mise en route des moteurs, 2) Choix du cap, 3) Réglage du cap

MARIN 2 : Après on va choisir et régler la vitesse sur le poste de navigation. Sinon le bateau n'avance pas.

CONCEPTEUR 2 : Bien sûr. Après cela, le bateau va partir vers le bon cap et à la bonne vitesse ?

MARIN 1 : Oui, comme prévu. C'est assez simple.

CONCEPTEUR 2 : Et si dans le chemin vers le port, un bateau vous coupe la route ?

MARIN 1 : Dans ce cas nous avons le choix, personnellement, je réduis ma vitesse...

MARIN 2 : ... Moi par contre, je modifie le cap pour effectuer une manœuvre d'évitement.

CONCEPTEUR 1 : Si je comprends bien, vous dites qu'à tout moment vous pouvez jouer soit sur le cap, soit sur la vitesse ?

MARIN 1 : Oui, bien sûr. C'est normal, non ?

CONCEPTEUR 2 : C'est qu'avec votre description du démarrage du bateau, on pouvait croire qu'il fallait impérativement régler le cap avant la vitesse.

MARIN 1 : C'est-à-dire que l'on pourrait aussi régler d'abord la vitesse. En fait, notre poste de commande est divisé en deux parties, l'une pour le cap et l'autre pour la vitesse, avec à chaque fois 2 boutons qui nous permettent soit d'incrémenter, soit de décrémenter le cap ou la vitesse.

CONCEPTEUR 2 : Vous avez 4 boutons devant vous ?

MARIN 2 : Voilà, et il n'y a aucun ordre particulier pour les utiliser, on peut même agir en même temps sur le cap et la vitesse.

[...]

Dans le cas de la modélisation sans système pré-existant, le panel des futurs utilisateurs choisi pour participer à la conception peut définir des besoins, mais n'a pas d'a priori sur la manière de décomposer ces objectifs en une succession d'opérations. Dans ce cas, les

3. D'autres méthodes pour l'expression des besoins des utilisateurs existent. Par exemple, la **GOMS** (Goals, Operators, Methods and Selection rules ()) étudie les actions et les performances des utilisateurs *in-situ* pour déterminer ses objectifs et la meilleure façon de les atteindre. Dans d'autres approches, les **Personas** remplacent ou enrichissent les scénarios d'utilisation et décrivent un ensemble de profils d'utilisateurs types que les concepteurs confrontent à leurs analyses à chaque étape de la modélisation (). Plus généralement, la norme ISO sur la conception centrée utilisateur recense un ensemble de méthodes et d'outils pour l'appréhension des objectifs des utilisateurs et leur participation au sein du processus de conception (cf. http ://www.iso.org/iso/catalogue_detail.htm ?csnumber=21197).

concepteurs jouent un rôle moteur lors de l'écriture du scénario nominal. En concertation avec les futurs utilisateurs, ils imaginent une manière acceptable de décomposer les objectifs des utilisateurs en séries d'opérations vis-à-vis du système.

Que les utilisateurs décrivent un système pré-existant ou qu'avec les concepteurs ils imaginent une nouvelle décomposition des tâches pour atteindre leur objectif, le scénario nominal est systématiquement défini à partir de ces entretiens comme un compromis acceptable entre les services que le système peut offrir et la succession de tâches idéale pour les utilisateurs. Le tableau IV.2 présente un cas de scénario nominal dérivé des entretiens entre les concepteurs et les marins de l'exemple précédent. Il décrit une situation d'utilisation du système où l'utilisateur atteint son objectif (c-à-d. "se rendre à un endroit", marqué sur fond vert dans le tableau IV.2) sans rencontrer d'erreur ou de comportement inapproprié de la part du système.

Tableau IV.2 – Exemple de scénario nominal

Scénario "Poste de navigation"
Pour se rendre à un endroit donné, le marin va devoir déplacer son **bateau**. Cette tâche consiste en deux opérations : d'une part régler **le cap** et d'autre part régler **la vitesse**. ~~Le marin règle le cap en l'augmentant ou en le diminuant et, de la même façon, il augmente ou diminue la vitesse~~. Sur ces embarcations, la **vitesse maximale** atteinte est de 25 nœuds.

À partir du scénario nominal, les informations du domaine repérées par les concepteurs (figurées en jaune dans le tableau IV.2) sont détaillées et organisées dans un diagramme de classes qui fournit un premier patron pour la conception des tiers métier et données. De la même manière, les tâches décrites dans le scénario (signifiées en bleu dans le tableau IV.2) sont décomposées en une série d'actions de l'utilisateur et peuvent être organisées sous forme d'arbre dont les feuilles représentent les séquences de sous-tâches à effectuer pour accomplir la tâche parente. Nous utiliserons la notation CTT (*Concur Task Tree* ()) qui propose de différentier les *tâches utilisateur*, les *tâches d'interaction* et les *tâches système* qui concourent à la réalisation d'une *tâche abstraite* représentant l'intention de l'utilisateur au plus haut niveau. Les tâches d'interaction et les tâches système sont implémentées par des procédures et des méthodes et impliquent l'intervention du système. Les tâches utilisateur sont du ressort de l'utilisateur seul et ne nécessitent pas d'utilisation ou d'intervention du système. CTT articule ces trois catégories de tâches par un ensemble d'opérateurs. Pour limiter la complexité de l'association entre les contextes d'exécution et le modèle des tâches, seuls les plus courants des opérateurs de CTT sont retenus dans cette étude. Plus spécifiquement, les relations binaires de succession (notés ">>") et de mise en parallèle (notés "|||") et les opérateurs unaires d'itération (notés "$*$") et d'optionalisation (notés "[]") sont conservées. En considérant deux tâches T1 et T2 :

- "T1 >> T2" signifie que T2 est validée par T1 et lui succède,
- "T1 ||| T2" signifie que T1 et T2 peuvent être effectuées dans n'importe quel ordre,
- "$*$T1" signifie que la tâche T1 est itérative et peut s'effectuer plusieurs fois de suite,
- "[T1]" signifie que T1 est optionnelle.

L'arbre des tâches annoté, proposé en figure IV.3, correspond au second niveau de description de l'organisation des tâches, lorsque les opérations effectuées par les utilisateurs sont confrontées au modèle des informations du domaine (Fig. IV.1(e)). Les tâches décrites dans cet arbre sont articulées par les opérateurs CTT. Ces articulations lèvent les ambiguïtés de la lecture de l'arbre des tâches et différencient les manières pour un utilisateur d'atteindre un objectif.

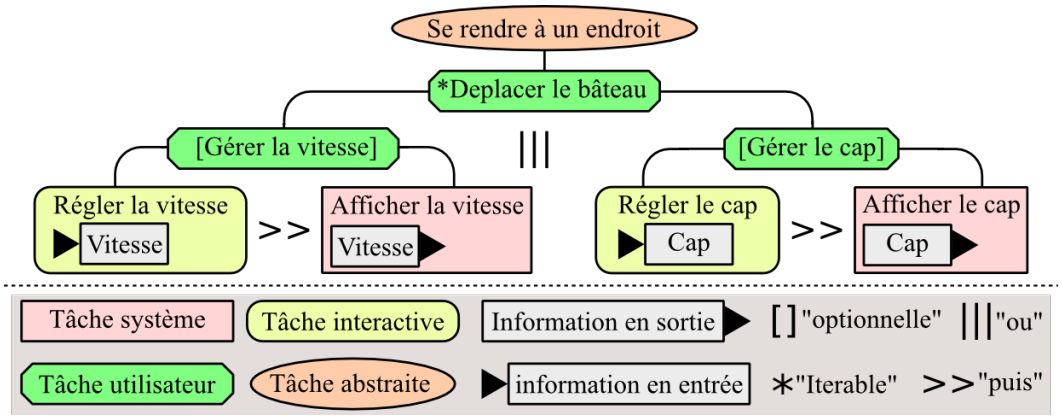


Figure IV.3 – Exemple d'arbre des tâches annoté

Dans l'exemple proposé, l'arbre des tâches illustre la décomposition issue de l'analyse du scénario de la tâche abstraite "Se rendre à un endroit" en un ensemble de sous-tâches. Cette tâche abstraite consiste en une répétition d'une tâche utilisateur de déplacement d'un bateau, elle-même effectuée soit en réglant le cap, soit en réglant la vitesse. Comme ces deux dernières sous-tâches restent optionnelles, il est possible d'effectuer un déplacement du bateau sans régler ni le cap ni la vitesse. Dans ce cas, le mouvement continue avec les valeurs de cap et de vitesse initiales. À plus bas niveau, les tâches interactives et les tâches système sont annotées par les informations manipulées en entrée et en sortie. Par exemple, la tâche interactive "Régler le cap" consiste pour l'utilisateur à fournir en entrée une valeur de cap a priori différente de la valeur courante. De la même manière, pour "Afficher le cap", le système va présenter en sortie la nouvelle valeur courante du cap. Les tâches système qui apparaissent dans l'arbre représentent le retour fait à l'utilisateur à l'issue d'une chaîne de traitement plus ou moins complexe. Par exemple, avant d'afficher le cap, le système veille à ce que ce cap soit effectivement mis en œuvre par les commandes du gouvernail, etc.

Les informations qui annotent les tâches et les sous-tâches forment des pré-requis à l'exécution de celles-ci. Plus spécifiquement, une tâche s'effectue de façon nominale dans la mesure où 1) l'ensemble de ses sous-tâches articulées par des opérateurs de succession accède aux informations d'entrée/sortie qui les annotent et 2) au moins l'une de ses sous-tâches articulées par l'opérateur de mise en parallèle accède aux informations d'entrée/sortie qui l'annotent. Dans la suite, les tâches et les sous-tâches décrites dans l'arbre des tâches font partie d'un ensemble noté $TASK$:

$$TASK = \{task_1, task_2, \ldots, task_x, \ldots, task_n\}$$

De la même manière, l'ensemble des informations qui annotent l'arbre des tâche est appelé $INFO$:

$$INFO = \{info_1, info_2, \ldots, info_x, \ldots, info_n\}$$

Quels que soient une tâche $task_x \in TASK$ et une information $info_y \in INFO$, les fonctions $AnnotateIn(task_x, info_y)$ et $AnnotateOut(task_x, info_y)$ retournent 1 lorsque $info_y$ *annote* la tâche $task_x$ en entrée (respectivement en sortie) ou 0 sinon. En considérant une tâche $task_x \in TASK$, les fonctions $InfoIn(task_x)$ et $InfoOut(task_x)$ retournent respectivement les sous-ensembles $\{info_1, info_2, \ldots, info_k, \ldots, info_n\} \subseteq INFO$ des informations utilisées en entrée ou en sortie par la tâche $task_x$:

$$\forall\ task_x \in TASK\ :\ InfoIn(task_x) = \{info_y \in INFO \mid AnnotateIn(task_x, info_y) = 1\}$$
$$\forall\ task_x \in TASK\ :\ InfoOut(task_x) = \{info_y \in INFO \mid AnnotateOut(task_x, info_y) = 1\}$$

Dans l'exemple du poste de guidage de bateau, ces fonctions retournent :

$$InfoOut(\text{“Afficher le cap”}) = InfoIn(\text{“Régler le cap”}) = \{\text{“Cap”}\}$$
$$InfoOut(\text{“Afficher la vitesse”}) = InfoIn(\text{“Régler la vitesse”}) = \{\text{“Vitesse”}\}$$

Pour pouvoir régler le cap ou la vitesse, le système doit avoir accès à l'ensemble des données informatiques qui permettent d'inférer le cap ou la vitesse. Par exemple, si les informations sont installées sur un composant de modèle de données, et que les traitements qui implémentent une tâche annotée par ces informations sont déployés sur un composant client distinct, il est nécessaire que le composant de gestion de données et le composant client puissent échanger des données. Dans ce cas, en termes de description géographique, leurs régions d'exécution sont en relation. Ces contraintes sur l'exécution d'une tâche ou d'une sous-tâche seront formalisées par la corrélation entre l'analyse de l'environnement et la décomposition des tâches, puis représentées dans un arbre des tâches situé.

IV.2.2 Recherche d'équivalences entre contextes d'exécution

Avant la production d'un arbre des tâches situé, une simplification sur l'ensemble $STATES$ des contextes d'exécution observables est effectuée. Cette simplification consiste en une recherche d'équivalences fonctionnelles entre différents contextes d'exécution afin de grouper les contextes qui déclenchent les mêmes mesures d'adaptation.

Dans un système interactif, l'objectif est d'offrir une flexibilité fonctionnelle pour l'utilisateur. Ce dernier, représenté au sein du système par un composant client associé à un contexte particulier, est désigné comme la cible de l'adaptation du système. Le contexte d'exécution de son point de vue local est déterminant. Si deux configurations différentes de régions sont perçues de la même manière au niveau de l'utilisateur, alors elles sont *équivalentes* et l'une comme l'autre donneront lieu à la même adaptation fonctionnelle du système du point de vue de l'utilisateur.

À titre d'exemple, quelles que soient les relations entre les régions d'exécution et les régions d'intérêt d'un système, tant qu'une région u_x d'un utilisateur n'est en relation avec aucune autre, alors le système reste inaccessible et cet utilisateur est isolé. Tant qu'il reste

isolé, il ne percoit pas l'impact des relations entre les autres régions d'intérêt ou d'exécution du système. Dans ces situations d'exécution, il doit disposer localement des informations et des capacités de traitement pour pouvoir effectuer sa tâche. Passer d'un contexte d'exécution où la région de l'utilisateur est en relation avec d'autres régions à un cas où l'utilisateur est isolé déclenchera des traitements locaux sur la plate-forme de l'utilisateur. Il est notamment possible de signaler à l'utilisateur qu'il a perdu les ressources du système et que le nombre des fonctionnalités dont il dispose s'est considérablement réduit.

Propriétés et fonctions d'équivalences

Les équivalences entre régions sont déterminées par un ensemble de propriétés formalisées par des fonctions de la forme $f\text{prop}_i(ec_x, ec_y)$. Ces fonctions prennent en entrée deux contextes d'exécution ec_x et ec_y de l'ensemble des contextes $STATES$. La fonction d'équivalence $f\text{prop}_i(\ldots)$ retourne 1 si selon l'énoncé de la propriété prop_i les contextes passés en paramètres provoquent les mêmes adaptations du système au niveau de l'utilisateur. La valeur 0 est retournée dans tous les autres cas de figure. Dans la suite de ce chapitre, chaque propriété d'équivalence est décrite textuellement puis formellement et est illustrée par un exemple dérivé du cas d'étude du système de suivi côtier développé au chapitre III. D'après cet exemple, les utilisateurs cibles de l'adaptation sont de rôle $\text{u} \in ROLES$. Ils sont représentés par un composant générique $CMPS(\text{u}) = C\text{u}_*$ et une région générique $Region(C\text{u}_*) = R\text{u}_*$.

Tableau IV.3 – Motivation des propriétés d'équivalence fonctionnelles

Nom de la règle	**Motivation**	**Traduction par une équivalence entre contextes d'exécution**
Utilisateur isolé	Un utilisateur qui n'atteint pas le système dispose de ressources identiques quel que soit le contexte du reste du système	Des contextes d'exécution sont équivalents tant qu'ils caractérisent la situation d'un l'utilisateur isolé du système
Région d'intérêt indéfinie	Les relations avec une région d'intérêt ne peuvent être déterminées que lorsque les données de définition de cette région sont disponibles	Des contextes d'exécution identiques à l'exception de relations non-déterminées impliquant une/des région(s) d'intérêt sont équivalents
Région d'intérêt distante	Dans l'optique de l'utilisateur, les relations des régions d'intérêt avec les autres régions du système ne sont pas forcément significatives	Des contextes d'exécution égaux à l'exception de relations entre des régions d'intérêts et des régions différentes de celle de l'utilisateur sont identiques
Contextes locaux identiques	Un composant est capable de s'adapter selon le contexte qu'il perçoit a son niveau. Ce contexte local peut être différent du contexte global du système	Des contextes d'exécution sont équivalents lorsque leurs mesures locales depuis le composant de l'utilisateur sont identiques
Relation non significative	Certaines relations entre composants n'ont pas de valeur sémantique et ne modifient pas le comportement du système du point de vue de l'utilisateur	Les contextes d'exécution égaux à l'exception d'une/de relation(s) jugée(s) non significative(s) par les concepteurs sont équivalents

Les propriétés d'équivalence, proposées ci-après, n'ont pas un caractère impératif pour l'implémentation du système. Leur usage permet de réduire la complexité de la tâche de conception du système en limitant le nombre de comportements adaptatifs à implémenter. Malgré cela, le choix est laissé aux concepteurs d'appliquer uniquement certaines propriétés d'équivalence, voire d'en ajouter de plus appropriées le cas échéant. Les règles retenues dans cette étude sont motivées par l'orientation interactive de la conception du système. Elles cherchent principalement à regrouper les contextes d'exécution selon qu'ils impliquent ou

non des différences d'exécution du point de vue de l'utilisateur. Le tableau IV.3 reprend chacune des propriétés détaillées dans la suite et motive de façon informelle les équivalences de contextes qu'elles caractérisent.

Les propriétés présentées dans cette étude se veulent les plus générales possibles et peuvent être réutilisées comme telles dans la plupart des cas d'étude. Des règles plus spécifiques et dépendantes du cas d'application pourraient par exemple considérer des équivalences de contextes se fondant sur des configurations particulières de relations entre régions plutôt que sur la simple présence ou absence d'une relation dans un contexte donné. Notamment, le premier cas d'étude développé au chapitre VI.1 illustrera quelques règles d'équivalence appropriées à un cas particulier, mais difficilement généralisables.

Dans la suite, les fonctions de vérification des n propriétés choisies ou définies par les concepteurs font partie du groupe $PROP = \{f\text{prop}_1, f\text{prop}_2, \ldots, f\text{prop}_x, \ldots, f\text{prop}_n\}$.

Propriété 1 (Utilisateur isolé)
Lorsqu'une région $R\text{u}_*$ d'un utilisateur cible de l'adaptation, portée par son composant $C\text{u}_*$, n'est en relation avec aucune région d'intérêt ou d'exécution, le système est hors de portée de cet utilisateur. Les contextes présentant une telle configuration sont équivalents.

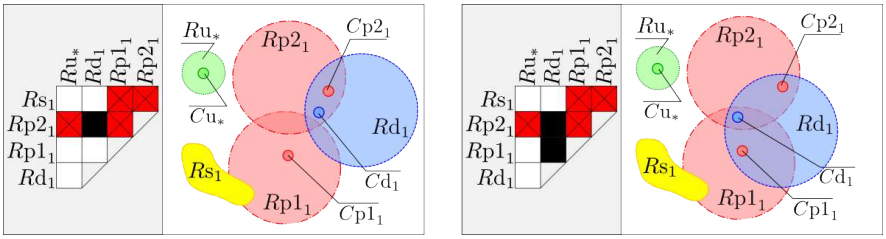


Figure IV.4 – Exemple d'équivalence entre contextes d'exécution par la propriété "utilisateur isolé"

Ce cas de figure est observé entre deux contextes d'exécution ec_x et ec_y lorsque la région des utilisateurs $R\text{u}_*$ n'est jamais impliquée dans une relation avec une autre région, quelle que soit la paire de régions en relation (reg_m, reg_n) appartenant à l'union $ec_x \cup ec_y$ des deux contextes (Prop. IV.1). Plus formellement et quels que soient les deux contextes d'exécution ec_x et ec_y de $STATES$, observés à deux instants de l'exécution, la fonction $f\text{prop1}(ec_x, ec_y)$ retourne 1 si ces contextes sont équivalents par la propriété "utilisateur isolé" et 0 autrement :

$$\forall\ ec_x, ec_y \in STATES\ :$$
$$f\text{prop1}(ec_x, ec_y) = \begin{cases} 1 & \text{ssi } \big(\forall\ (reg_m, reg_n) \in ec_x \cup ec_y\ :\ reg_m, reg_n \neq R\text{u}_*\big) \\ 0 & \text{autrement} \end{cases} \tag{IV.1}$$

Dans l'exemple du système de suivi côtier, les situations représentées en figure IV.4 montrent toutes deux une région de l'utilisateur isolée du reste du système. Le système reste toujours hors de portée de $C\text{u}_*$ et selon l'utilisateur, le contexte est interprété comme vide. Ces contextes d'exécution sont équivalents selon la propriété "Utilisateur isolé" et il est vérifié que $f\text{prop1}\big(\{(R\text{p2}_1, R\text{d}_1)\}, \{(R\text{p2}_1, R\text{d}_1), (R\text{p1}_1, R\text{d}_1)\}\big) = 1$.

Propriété 2 (Région d'intérêt indéfinie)
Pour détecter une relation entre une région d'exécution $reg_z \in ENVEXEC$ et une région d'intérêt $reg_m \in ENVINT$, le composant générateur $cmp_z = Component(reg_z)$ de la région d'exécution doit échanger des informations avec le composant $cmp_y = Host(reg_m)$, hôte de la région d'intérêt. Cet échange d'information est possible lorsque les régions d'exécution reg_z et $reg_y = Region(cmp_y)$ sont en relation. Même si la relation entre reg_z et reg_m existe, dès lors que $Relate(reg_z, reg_y) = 0$ cette relation n'est pas détectable par le système. Dans ce cas, les contextes d'exécution égaux à l'exception de la paire (reg_z, reg_m) et ne contenant pas la paire $(reg_z, Region(Host(reg_m)))$, sont équivalents.

Ce cas de figure est observé entre deux contextes d'exécution ec_x et ec_y lorsqu'il existe dans l'environnement géographique du système une région d'intérêt $reg_m \in ENVINT$ et une région d'exécution $reg_n \in ENVEXEC$ telles que (Prop. IV.2) :

- la paire (reg_m, reg_n) appartienne à l'union des contextes $ec_x \cup ec_y$ [4], et que ces contextes soient identiques à l'exception de cette relation ;
- le composant $Host(reg_m)$, qui héberge les informations définissant la région d'intérêt reg_m, génère une région $Region(Host(reg_m))$ qui n'est en relation avec la région d'exécution reg_n ni dans ec_x, ni dans ec_y.

Plus formellement et quels que soient deux contextes d'exécution ec_x et ec_y de $STATES$ observés lors de l'exécution, la fonction $f\text{prop2}(ec_x, ec_y)$ retourne 1 si ces contextes sont équivalents par la propriété "Région d'intérêt indéfinie" et 0 autrement :

$$\forall\ ec_x, ec_y \in STATES\ :$$

$$f\text{prop2}(ec_x, ec_y) = \begin{cases} 1 & \text{ssi} \begin{pmatrix} \exists\ reg_m \in ENVINT, reg_n \in ENVEXEC\ | \\ (reg_m, reg_n) \in ec_x \cup ec_y\ \wedge \\ ec_x - (reg_m, reg_n) = ec_y - (reg_m, reg_n) \wedge \\ \big(Region(Host(reg_m)), reg_n\big) \notin ec_x \cup ec_y \end{pmatrix} \\ 0 & \text{autrement} \end{cases} \quad \text{(IV.2)}$$

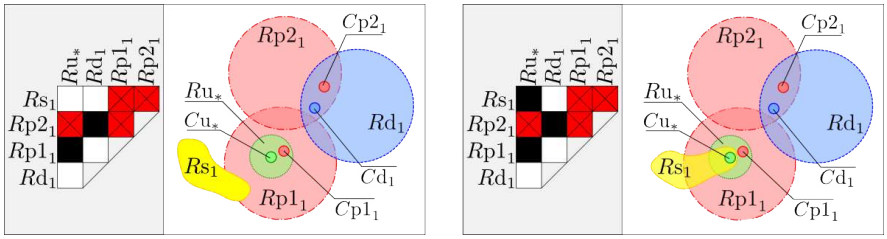


Figure IV.5 – Exemple d'équivalence entre contextes d'exécution par la propriété "Région d'intérêt indéfinie"

Dans l'exemple du système de suivi côtier, les situations représentées en figure IV.5 montrent deux contextes équivalents par la propriété "Région d'intérêt indéfinie". À tout instant de l'exécution, la région sémantique Rs_1 est définie par les données du composant Cd_1 qui génère la région d'exécution Rd_1. Comme dans aucun des deux contextes, cette région Rd_1 n'est en lien avec la région d'exécution Ru_*, la situation de la région

4. $(reg_m, reg_n) \in ec_x \cup ec_y$ vérifie l'appartenance de la paire (reg_m, reg_n) à l'un, à l'autre ou aux deux contextes ec_x et ec_y

Rs_1 est indéfinie pour l'utilisateur au moment de la mesure. Du point de vue des processus d'adaptation, ces deux situations sont indifférentiables et équivalentes. Il est vérifié que $f\text{prop2}\big(\{(Rp2_1, Rd_1), (Rp1_1, Ru_*)\}, \{(Rp2_1, Rd_1), (Rp1_1, Ru_*), (Rs_1, Ru_*)\}\big) = 1$.

Propriété 3 (Région d'intérêt distante)
Le positionnement de l'utilisateur par rapport à une région d'intérêt permet de distinguer deux niveaux fonctionnels, entre un utilisateur "acteur" à l'intérieur d'un espace géographique à la sémantique particulière (p. ex. ville, pièce, etc.) et un utilisateur "spectateur", observateur passif de la région d'intérêt. Du point de vue de l'utilisateur, les relations supplémentaires entre cette région d'intérêt et les autres régions d'exécution n'influencent pas l'adaptation du système. Dans ce cas, quelles que soient une région d'intérêt $reg_y \in ENVINT$ et une région d'utilisateur Ru_*, un contexte ne contenant pas (Ru_*, reg_y) et qui inclue une relation entre reg_y et une région $reg_z \neq Ru_*$ est équivalent à un contexte identique mais n'incluant pas (reg_y, reg_z).

Ce cas de figure est observé entre deux contextes d'exécution ec_x et ec_y s'il existe une région d'intérêt $reg_m \in ENVINT$ et une région quelconque $reg_n \in ENVGEO$ telles que (Prop. IV.3) :

- reg_n n'est pas la région des utilisateurs ($reg_n \neq Ru_*$) ;
- la paire (reg_m, reg_n) appartient à l'union des contextes $ec_x \cup ec_y$;
- ec_x et ec_y sont identiques à l'exception de la paire (reg_m, reg_n).

Plus formellement et quels que soient deux contextes d'exécution ec_x et ec_y de $STATES$ observés lors de l'exécution, la fonction $f\text{prop3}(ec_x, ec_y)$ retourne 1 si ces contextes sont équivalents par la propriété "Région d'intérêt distante" et 0 autrement :

$$\forall\ ec_x, ec_y \in STATES\ :$$
$$f\text{prop3}(ec_x, ec_y) = \begin{cases} 1 & \text{ssi} \left(\begin{array}{l} \exists\ \text{reg}_m \in ENVINT, reg_n \in ENVGEO - Ru_*\ | \\ ec_x - (reg_m, reg_n) = ec_y - (reg_m, reg_n)\ \wedge \\ (reg_m, reg_n) \in ec_x \cup ec_y \end{array} \right) \\ 0 & \text{autrement} \end{cases} \quad \text{(IV.3)}$$

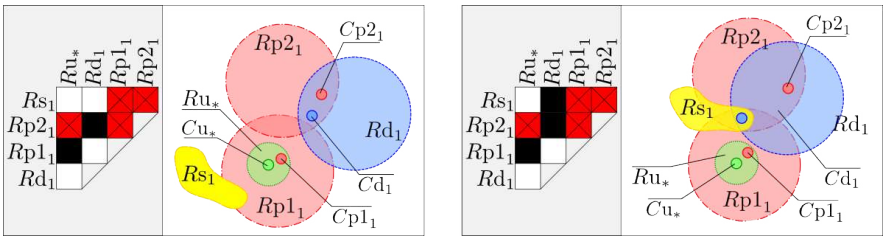


Figure IV.6 – Exemple d'équivalence entre contextes d'exécution par la propriété "Région d'intérêt distante"

Dans l'exemple du système de suivi côtier, les situations représentées en figure IV.6 montrent deux contextes équivalents par la propriété "Région d'intérêt distante". Dans le premier contexte, la région d'intérêt Rs_1 est distante du reste du système. Dans le second cas, la relation entre cette région et la région d'exécution Rd_1 est établie et fait partie du contexte d'exécution. Du point de vue de l'utilisateur, cette différence n'est pas fondamentale. Le fait que le composant Cd_1 soit inclus dans l'aire de navigation restreinte n'implique pas de réajustement fonctionnel à

son niveau. Ces deux situations sont indifférenciées et équivalentes. Il est vérifié que $f\text{prop3}\big(\{(R\text{p}2_1, R\text{d}_1), (R\text{p}1_1, R\text{u}_*)\}, \{(R\text{p}2_1, R\text{d}_1), (R\text{p}1_1, R\text{u}_*), (R\text{s}_1, R\text{d}_1)\}\big) = 1$.

Propriété 4 (Contextes locaux identiques)
A un instant donné, le contexte d'exécution mesuré au niveau de l'utilisateur inclut les relations entre régions dont il a connaissance uniquement, et qui font partie de son contexte local. Les réactions d'adaptation sont prises sur la base de ce contexte local. Les relations entre régions extérieures au contexte local de l'utilisateur n'impactent pas l'adaptation du système. Deux contextes d'exécution partageant une même interprétation locale au niveau de l'utilisateur sont équivalents.

Ce cas de figure est observé entre deux contextes d'exécution ec_x au temps t_a et ec_y au temps t_b s'il existe un type de région d'utilisateur $R\text{u}_*$ tel que les contextes locaux $LocState(R\text{u}_*, t_a)$ et $LocState(R\text{u}_*, t_b)$ mesurés à partir de cette région d'utilisateur soient identiques (Prop. IV.4). Plus formellement, quels que soient deux contextes d'exécution $ec_x, ec_y \in STATES$, observés lors de l'exécution aux instants t_a et t_b, la fonction $f\text{prop4}(ec_x, ec_y)$ retourne 1 si ces contextes sont équivalents du point de vue d'un utilisateur générique de région $R\text{u}_*$ par la propriété "Contextes locaux identiques" et 0 autrement :

$$\forall\ ec_x, ec_y \in STATES\ :$$
$$f\text{prop4}(ec_x, ec_y) = \begin{cases} 1 & \text{ssi} \begin{pmatrix} \exists\ R\text{u}_* \in ENVEXEC\ \mid \\ LocState(R\text{u}_*, t_a) = LocState(R\text{u}_*, t_b) \wedge \\ LocState(R\text{u}_*, t_a) \subseteq ec_x\ \wedge\ LocState(R\text{u}_*, t_b) \subseteq ec_y \end{pmatrix} \\ 0 & \text{autrement} \end{cases} \tag{IV.4}$$

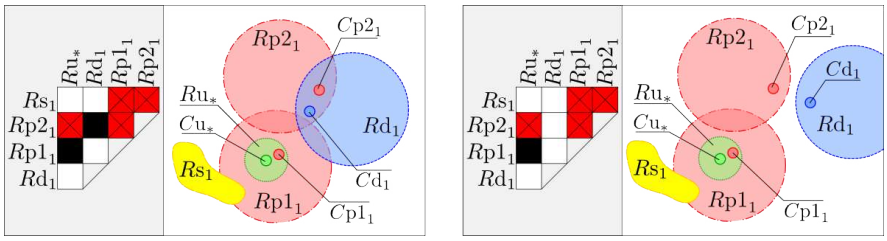


Figure IV.7 – Exemple d'équivalence entre contextes d'exécution par la propriété "Contextes locaux identiques"

Dans les contextes d'exécution illustrés par la figure IV.7, la relation entre la seconde région de traitement $R\text{p}2_1$ et la région des données $R\text{d}_1$ n'est pas détectable du point de vue de l'utilisateur. Son contexte local est identique pour chaque situation contextuelle : $LocState(R\text{u}_*, t_a) = LocState(R\text{u}_*, t_b) = \{(R\text{u}_*, R\text{p}1_1)\}$. Les mesures d'adaptation éventuellement mises en œuvre à son niveau sont identiques et les contextes d'exécution sont équivalents par la propriété "Contextes locaux identiques". Il est vérifié que $f\text{prop4}\big(\{(R\text{p}2_1, R\text{d}_1), (R\text{p}1_1, R\text{u}_*)\}, \{(R\text{p}1_1, R\text{u}_*)\}\big) = 1$.

Propriété 5 (Relation non significative)
Certaines relations entre régions d'exécution n'engendrent pas forcement d'adaptation particulière pour un utilisateur du système. En considérant qu'un ensemble de relations non

significatives entre régions d'exécution est déterminé à la conception en toute connaissance des besoins des utilisateurs et de leurs tâches, alors contexte qui inclue une relation de cet ensemble est équivalent à un contexte identique à cette relation prêt[5].

Soit $EXCLUDED$, l'ensemble des paires de régions représentant les relations entre régions jugées non pertinentes pour l'adaptation. La propriété "Relation non significative" est observée entre deux contextes d'exécution ec_x et ec_y s'il existe une paire de région (reg_m, reg_n) dans $EXCLUDED$ telle que cette relation appartienne à $ec_x \cup ec_y$, et que ces contextes soient identiques à l'exception de cette relation (Prop. IV.5). Plus formellement, quels que soient deux contextes ec_x et ec_y de $STATES$ observés lors de l'exécution, la fonction $f\text{prop5}(ec_x, ec_y)$ retourne 1 si ces contextes sont équivalents du point de vue d'un utilisateur générique u_* par la propriété "Relation non significative" et 0 autrement :

$$\forall\ ec_x, ec_y \in STATES\ :$$
$$f\text{prop5}(ec_x, ec_y) = \begin{cases} 1 & \text{ssi } \begin{pmatrix} \exists\ (reg_m, reg_n) \in EXCLUDED \mid (reg_m, reg_n) \in ec_x \cup ec_y \\ \wedge\ \ ec_x - (reg_m, reg_n) = ec_y - (reg_m, reg_n) \end{pmatrix} \\ & \text{avec } EXCLUDED \text{ l'ensemble des relations non significatives} \\ 0 & \text{autrement} \end{cases} \tag{IV.5}$$

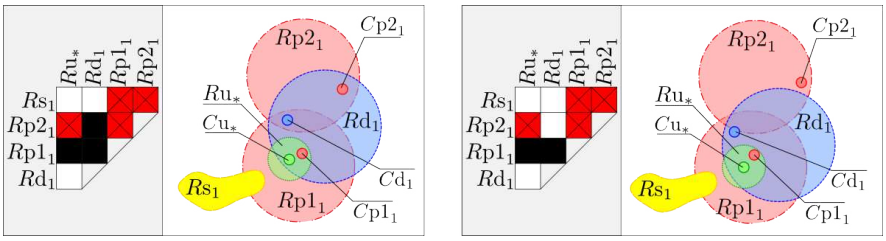


Figure IV.8 – Exemple d'équivalence entre contextes d'exécution par la propriété "Relation non significative"

Dans l'exemple du système de suivi côtier, compte tenu des limitations physiques dans la mobilité des composants, les régions d'exécution $R\text{p2}_1$ et $R\text{u}_*$ ne sont jamais en relation. Les utilisateurs du système, désignés par le rôle u, n'accèdent jamais à cette région de service $R\text{p2}_1$ et les relations qu'elle établit avec d'autres régions d'exécution du système n'impactent pas les fonctionnalités offertes aux utilisateurs. Par exemple, la relation entre $R\text{p2}_1$ et la région des données $R\text{d}_1$ ne permet pas d'offrir un meilleur service aux utilisateurs et les concepteurs excluent cette relation : $EXCLUDED = \{(R\text{p2}_1, R\text{d}_1)\}$. Dans ce cas, les deux contextes d'exécution présentés en figure IV.8 sont équivalents par la propriété "Relation non significative" et il est vérifié que $f\text{prop5}\big(\{(R\text{p2}_1, R\text{d}_1), (R\text{p1}_1, R\text{u}_*), (R\text{p1}_1, R\text{d}_1)\}, \{(R\text{p1}_1, R\text{u}_*), (R\text{p1}_1, R\text{d}_1)\}\big) = 1$.

5. Rendre une relation non significative entre deux régions d'exécution implique la même portée conceptuelle que si cette relation était exclue pour des causes physiques, comme décrit au chapitre III.3.3.

Chaînes de contextes équivalents

Des paires de contextes équivalents, issues de l'application des propriétés, peuvent être combinées pour former des chaînes de contextes générant la même adaptation pour l'utilisateur. Pour garantir l'adaptabilité du système, un comportement fonctionnel et des adaptations doivent être prévus et implémentés pour chaque chaîne identifiée. La mesure de l'environnement d'exécution effectuée au niveau de l'utilisateur permet de lui proposer un comportement du système approprié à sa situation d'exécution.

La formation des chaînes d'équivalence est automatique et compose des groupes de contextes à partir de l'ensemble $STATES$ et des propriétés d'équivalence choisies par les concepteurs. Plus spécifiquement, les contextes équivalents sont groupés en binômes. Lorsqu'un contexte donné est présent dans deux binômes, alors ceux-ci sont groupés et forment une chaîne d'équivalence. Par exemple, en considérant trois contextes ec_a, ec_b, ec_c de $STATES$, si les contextes ec_a et ec_b sont équivalents par la propriété "Région d'intérêt distante" et que ec_b et ec_c sont équivalents par la propriété "Contextes locaux identiques", alors ces trois contextes font partie d'une même chaîne d'équivalence, dont ec_b est le liant.

Dans la suite, $chain_x$ est un élément de l'ensemble $CHAINS = \{chain_1, chain_2, \ldots, chain_k, \ldots, chain_n\}$ des n chaînes de contextes équivalents. L'ensemble $CHAINS$ est construit à partir des fonctions de $PROP$ en deux étapes :

- construction de l'ensemble des binômes de contextes d'exécution équivalents, en application des propriétés retenues par les concepteurs et en ajoutant une propriété implicite d'équivalence d'un contexte à lui-même ;
- recherche itérative dans cet ensemble des éléments qui peuvent être associés par l'identité de certains de leurs contextes (c-à-d. les liants entre groupes équivalents).

L'ensemble des paires de contextes équivalents (ec_x, ec_y) est déduit en explorant deux à deux les équivalences entre les contextes d'exécution de $STATES$. Deux contextes ec_x et ec_y forment un bînome équivalent (ec_x, ec_y) si :

- il existe une propriété $f\text{prop}_z \in PROP$, telle que selon cette propriété, ec_x et ec_y soient équivalents ;
- ec_x et ec_y sont égaux au sens de la théorie des ensembles et contiennent strictement les mêmes paires de régions. Dans ce cas, les relations entre régions sont identiques selon les deux contextes.

Formellement, $Pairs(PROP)$ désigne la fonction retournant l'ensemble des paires de contextes équivalents par les propriétés vérifiées par les fonctions de $PROP$. Quels que soient deux contextes d'exécution ec_x et ec_y de $STATES$:

$$Pairs(PROP) = \left\{ (ec_x, ec_y) \mid \left(\begin{array}{l} \left(\exists\ f\text{prop}_z \in PROP \mid f\text{prop}_z(ec_x, ec_y) = 1\right) \\ \vee\ ec_x = ec_y \end{array} \right) \right\}$$

L'ensemble des chaînes de contextes est constitué par une recherche des groupements possibles parmi les paires équivalentes. Dans la suite, la fonction $Chain(ec_x, A)$, paramétrée par un contexte $ec_x \in STATES$ et un ensemble de contextes A, retourne l'ensemble des contextes qui sont équivalents, de proche en proche, à ec_x. Cette fonction s'écrit récursivement, en augmentant l'ensemble A des contextes équivalents à ec_x détectées lors d'une itération et qui ne font pas déjà partie de cet ensemble (Prop. IV.6). L'ensemble des

chaînes d'équivalence $CHAINS$ s'écrit formellement :

$$CHAINS = \bigcup_{ec_i \in STATES} \big(Chain(ec_i, \{ec_i\}) - \emptyset\big)$$
$$\text{avec } Chain(ec_i, A) = \left\{ \begin{matrix} ec_j \in STATES \ \cup \ Chain\big(ec_j, A \cup \{ec_j\}\big) \ | \\ (ec_i, ec_j) \in Pairs(PROP) \ \wedge \ ec_j \notin A \end{matrix} \right\} \tag{IV.6}$$

Cette définition de l'ensemble des chaînes de contextes $CHAINS$ en fait une partition de l'ensemble des contextes d'exécution $STATES$. Autrement dit, chaque élément de $STATES$ est présent au moins une fois dans un élément de $CHAINS$ et les éléments de $CHAINS$ sont disjoints deux à deux. Cette nouvelle partition des contextes d'exécution garantit qu'au moins chaque situation d'exécution se trouve représentée au sein d'une chaîne d'équivalence, quitte dans le cas le plus défavorable à générer autant de chaînes d'équivalence que de contextes d'exécution [6].

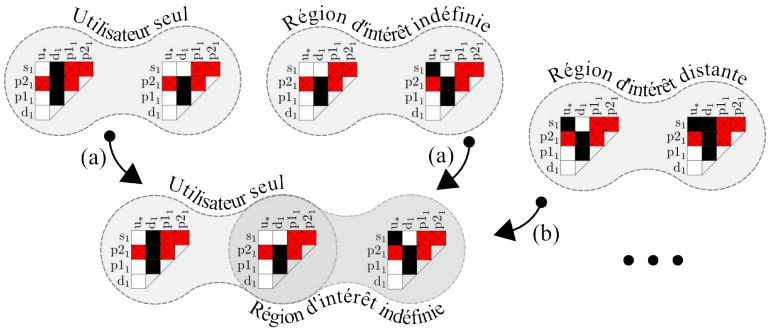


Figure IV.9 – Exemple de construction d'une chaîne d'équivalence par association de binômes de contextes d'exécution

Dans l'exemple du suivi côtier, les paires de contextes équivalents sont formées en appliquant les cinq propriétés d'équivalence (c-à-d. pour les concepteurs, $PROP = \{f\text{prop}_1, f\text{prop}_2, f\text{prop}_3, f\text{prop}_4, f\text{prop}_5\}$). Les 64 contextes d'exécution éventuellement mesurables qui ont été déterminés à l'issue de la modélisation des relations entre régions forment la source de la recherche des équivalences. Par exemple, les contextes $\{(R\text{d}_1, R\text{s}_1), (R\text{d}_1, R\text{p}2_1), (R\text{d}_1, R\text{p}1_1)\}$ et $\{(R\text{d}_1, R\text{p}2_1), (R\text{d}_1, R\text{p}1_1)\}$ forment une paire de contextes équivalents par la propriété "Utilisateur isolé" (Fig. IV.9) [7]. Une fois l'ensemble $Pairs(PROP)$ généré, les chaînes d'équivalence sont construites en cherchant des contextes d'exécution communs à différentes paires de $Pairs(PROP)$. Par exemple, la figure IV.9(a) illustre l'association au sein d'une chaîne de trois contextes de deux paires équivalentes et partageant un élément commun. Par extension, d'autres paires s'agrègent à ce noyau et prolongent la chaîne d'équivalence (Fig. IV.9(b)). À l'issue de la recherche d'équivalences, cette chaîne contiendra finalement 16 éléments (Fig. IV.10(a)).

L'étape de génération des chaînes de contextes équivalents retourne dans le cas du système de suivi côtier un ensemble $CHAINS$ de douze chaînes d'équivalence (Fig. IV.10). Le principal groupe rassemble les contextes équivalents à la situation où l'utilisateur est

6. Ce cas de figure se présente si les concepteurs ne définissent aucune propriété d'équivalence.

7. Pour la figure IV.9 et les suivantes, la notation matricielle des contextes d'exécution est simplifiée en omettant le "R" initial lorsqu'il n'y a pas d'ambiguïté sur la dénomination d'une région.

extérieur au reste du système et où les tâches qu'il peut effectuer reposent uniquement sur les capacités de traitement et les données disponibles localement à son composant client (Fig. IV.10(a)). La situation contextuelle la plus "riche", qui caractérise le maximum des relations entre régions, est associée dans un groupe d'équivalence à trois autres contextes moins riches mais qui néanmoins permettent la mise en oeuvre des mêmes services au niveau du client (Fig. IV.10(b)). A priori, cet ensemble de contextes particuliers représente des situations d'exécution idéales, durant lesquelles un maximum de ressources sont disponibles pour l'utilisateur.

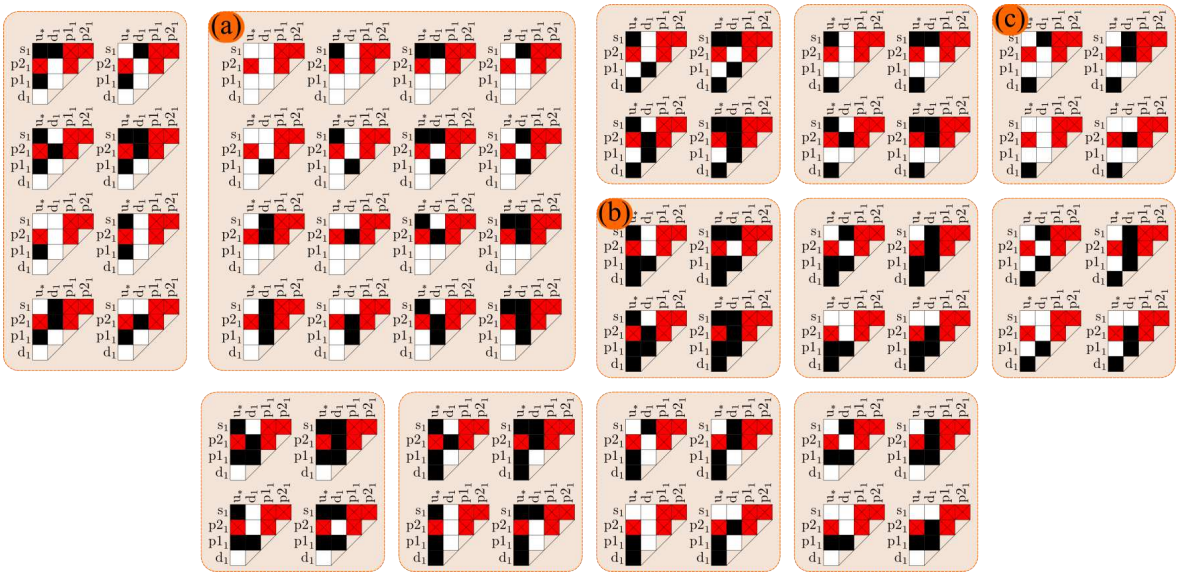


Figure IV.10 – Groupes de contextes équivalents constituant l'ensemble $CHAINS$ dans l'exemple du système de suivi côtier

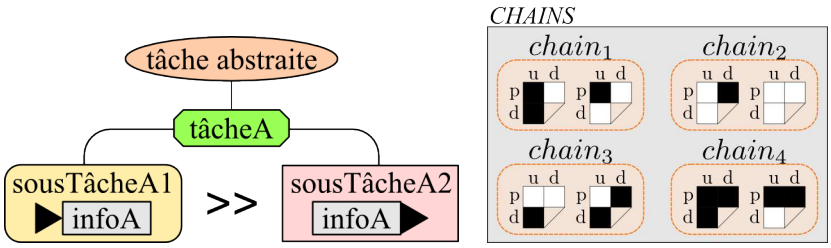
Les autres groupes de contextes équivalents demandent une étude plus fine des situations dans lesquelles se trouve l'utilisateur pour savoir ce que devient sa tâche dans ces situations particulières. Pour cela, les concepteurs détaillent plus spécialement les contextes de chaque groupe incluant le moins de relations entre régions et s'interrogent sur ce qu'il est possible d'offrir en termes de fonctionnalités dans ces situations. Par exemple, si le contexte à l'exécution figure parmi le groupe représenté en figure IV.10(c), les concepteurs se posent la question de quelles fonctionnalités offrir à un utilisateur capable d'échanger des informations avec le serveur de données $C\mathrm{d}_1$. Plus généralement, les descriptions des contextes et des chaînes d'équivalences reposent sur des critères géographiques ou logiques qui, hormis différents état de la plate-forme, n'engendrent pas de signification conceptuelle particulière. Pour l'instant, les rôles des contextes n'ont pas été définis. Dans l'hypothèse d'une conception logicielle, un sens fonctionnel est donné à ces configurations de ressources. Plus spécifiquement, les tâches des utilisateurs sont associées à chaque groupe de contextes qui permet de les exécuter. Des lors, les groupes de contextes s'enrichissent d'une sémantique fonctionnelle et il est possible, connaissant un contexte, d'énoncer les fonctionnalités qu'il active.

IV.2.3 Arbre des tâches situé

Les concepteurs, en concertation avec des utilisateurs et les commanditaires du système, ont énoncé un scénario nominal et l'ont organisé au sein d'un arbre des tâches annoté (Chap. IV.2.1). Dans un système distribué et mobile, les contraintes de relations entre régions d'exécution et d'intérêt et les capacités des composants à échanger des informations limitent le déroulement du cas nominal à un sous-ensemble de situations d'exécution rencontrées par l'utilisateur. Ce sous-ensemble est un élément parmi l'ensemble des chaînes d'équivalence décrites précédemment et qui appellent chacune un comportement du système approprié. Idéalement, un comportement du système devrait être décrit pour chaque élément de $CHAINS$. Ces comportements alternatifs au scénario et à la tâche nominale viennent enrichir l'arbre des tâches annoté pour former un arbre des tâches situé. La construction de l'arbre des tâches situé à partir de l'arbre des tâches annoté s'effectue en trois temps :

- le choix d'une **répartition des procédures et des données** sur les composants du système permet de définir des propositions localisées comme "la tâche $Task_a$ est réalisable dans la région d'exécution reg_x" ou "l'information $Info_b$ est disponible dans la région d'exécution reg_y et désigne une région d'intérêt reg_z" ;
- la **contextualisation de l'arbre des tâches annoté** assigne aux feuilles et branches de l'arbre annoté les chaînes de contextes qui valident les conditions d'exécution requises par les tâches désignées ;
- l'**ajout de tâches alternatives** permet d'associer des nouveaux comportements aux groupes de contextes équivalents qui ne sont pas encore assignés à des nœuds de l'arbre des tâches.

L'arbre des tâches situé désigne l'arbre annoté augmenté des branches représentant l'ensemble des alternatives au cas nominal et dans lequel les conditions nécessaires à l'exécution d'une tâche ou sous-tâche sont signifiées par l'association entre les groupes de contextes équivalents et les noeuds de l'arbre.



(a) Arbre des tâches annoté (b) Chaînes d'équivalence

Figure IV.11 – Exemple simplifié de décomposition des tâches et des contextes d'exécution

La suite de cette section illustre la construction d'un arbre des tâches situé à partir d'un exemple simplifié de tâche annotée telle qu'illustrée en figure IV.11a. Les concepteurs ont déterminé un fonctionnement nominal pour lequel une tâche "tâcheA" est effectuée par une tâche interactive "sousTâcheA1" qui demande l'information "infoA" en entrée, suivie par une tâche système "sousTâcheA2" qui produit "infoA" en sortie. Par ailleurs, la décomposition de la plate-forme est elle aussi simplifiée en trois composants, $C\mathrm{d}_1$, $C\mathrm{p}_1$ et $C\mathrm{u}_1$, qui forment les régions d'exécution $R\mathrm{d}_1$, $R\mathrm{p}_1$ et $R\mathrm{u}_1$. La modélisation spatiale des relations entre ces régions produit un ensemble de huit contextes d'exécution. En admettant un utilisateur désigné par le

rôle "u" et sa région d'exécution $R\mathrm{u}_1$, l'étape de recherche des chaînes d'équivalence regroupe ces contextes d'exécution en quatre chaînes, illustrées en figure IV.11b, et ci-après nommées $chain_1$ à $chain_4$.

Répartition des procédures et des données

Afin d'identifier les régions qui permettent l'exécution d'une tâche donnée, et vérifier le moment et le lieu où cette tâche pourra être effectuée par l'utilisateur, les informations et les tâches interactives ou du système sont étiquetées par les régions d'exécution où ces services sont diffusés. En considérant qu'une tâche $Task_a$ est implémentée par les procédures d'un composant de traitement cmp_x, alors la région d'exécution $reg_x = Region(cmp_x)$, générée par ce composant, constitue une étiquette de la tâche $Task_a$. De la même manière, une information $Info_b$ est étiquetée par une région d'exécution $reg_y = Region(cmp_y)$ si le composant cmp_y générant cette région est chargé de la gestion et du stockage des données informatiques qui définissent $Info_b$. Les régions constituent un ensemble d'étiquettes sur les tâches et les informations. Ces ensembles sont retournés par les fonctions $TaskTag(Task_a)$ et $InfoTag(Info_b)$:

- $TaskTag(Task_a)$ retourne l'ensemble des régions d'exécution $\{reg_1, reg_2, \ldots, reg_k, \ldots, reg_n\} \subseteq ENVEXEC$ dont les composants générateurs hébergent chacun les procédures pour le traitement de la tâche $Task_a$. Cet ensemble $TaskTag(Task_a)$ *étiquette* la tâche $Task_a$;
- $InfoTag(Info_b)$ retourne l'ensemble des régions d'exécution $\{reg_1, reg_2, \ldots, reg_k, \ldots, reg_n\} \subseteq ENVEXEC$ dont les composants générateurs hébergent chacun les données de définition de l'information $Info_b$. L'ensemble des régions $InfoTag(Info_b)$ *étiquette* l'information $Info_b$.

Cette répartition des traitements et des données est effectuée par les concepteurs, en connaissance des rôles de chaque composant. Dans l'exemple simplifié de système à trois composants, le composant $C\mathrm{p}_1$ est plus particulièrement dédié aux traitements alors que le composant $C\mathrm{d}_1$ héberge un système de gestion de données propre à assurer l'accès et la sauvegarde des informations du système. En toute logique, les concepteurs décident d'implémenter les procédures pour effectuer les tâches interactives au sein du composant $C\mathrm{p}_1$ et dédient la gestion de l'information $Info_A$ au composant $C\mathrm{d}_1$. Dans ce cas, les étiquetages des tâches et informations sont les suivants :

$$\begin{aligned} TaskTag(\text{sousTâcheA1}) &= \{R\mathrm{p}_1\} \\ TaskTag(\text{sousTâcheA2}) &= \{R\mathrm{p}_1\} \\ InfoTag(\text{infoA}) &= \{R\mathrm{d}_1\} \end{aligned}$$

Du point de vue de la décomposition de l'espace du système en un ensemble de régions, la relation entre deux régions d'exécution $reg_a, reg_b \in ENVEXEC$ au niveau spatial dénote, de la part de leurs composants générateurs, une capacité d'utilisation mutuelle de ressources au niveau du système. Lorsque à un instant t_x de l'exécution la paire (reg_a, reg_b) fait partie d'un contexte d'exécution $State(t_x)$ alors :

- les tâches et les informations implémentées au niveau du composant $cmp_a = Component(reg_a)$, générateur de la région reg_a, sont accessibles au composant $cmp_b = Component(reg_b)$, générateur de la région reg_b ;
- de manière symétrique, les tâches et les informations implémentées au niveau du composant cmp_b, sont accessibles au composant cmp_a.

Les traitements qui implémentent une tâche doivent accéder aux informations associées à cette tâche en entrée et en sortie pour pouvoir s'exécuter de façon optimale. Ces procédures doivent pouvoir recevoir les informations nécessaires depuis un ou plusieurs composant(s) serveur de données. Au niveau des régions d'exécution, cela se traduit par les relations entre une région $reg_x \in ENVEXEC$, qui étiquette une tâche $task_a$ donnée, et les régions d'exécution $\{reg_1, \ldots, reg_n\} \subseteq ENVEXEC$ qui étiquettent les informations nécessaires en entrée et en sortie pour réaliser $task_a$. D'autre part, les tâches représentées dans l'arbre annoté s'adressent explicitement à l'utilisateur du système. Dans le sens de l'utilisateur vers le système, les tâches interactives demandent sa participation dans l'objectif de modifier des informations ou de commander un traitement. À l'inverse, du système vers l'utilisateur, les tâches du système représentent les réponses fournies aux requêtes de l'utilisateur par le système. Chaque tâche interactive ou système de l'arbre annoté est mobilisante pour l'utilisateur et nécessite l'utilisation de son composant client, que ce soit pour saisir ou pour recevoir des informations. Du point de vue des relations entre régions d'exécution, une tâche interactive ou système $task_a$ peut s'effectuer si la région représentant le type d'utilisateur dont la tâche est décomposée est en relation avec l'une des régions d'exécution qui étiquettent $task_a$.

Ces deux pré-requis au déroulement d'une tâche constituent une condition d'exécution relative à un contexte donné. Plus formellement, une tâche $task_a$ est exécutable lorsque le contexte d'exécution mesuré figure parmi une chaîne d'équivalence $chain_x$ tel qu'il existe un contexte $ec_r \in chain_x$ vérifiant :

- l'existence d'une région d'exécution reg_m parmi l'ensemble des régions $TaskTag(task_a)$, telle que cette région soit accessible à l'utilisateur et que leur relation, signifiée par la paire $(reg_m, R\mathrm{u}_*)$, figure dans le contexte ec_r ;
- l'existence dans le contexte ec_r d'une région d'exécution reg_n appartenant à l'ensemble des régions $InfoTag(info_z)$ et qui soit liée à la région des traitement reg_m, quelle que soit l'information $info_z$ nécessaire en entrée ou en sortie à la tâche $task_a$.

Dans la suite, la fonction $Exec(task_a, chain_x)$ retourne 1 s'il existe un contexte d'exécution parmi $chain_x \in CHAINS$ qui réunisse les conditions nécessaires au déroulement de la tâche $task_a$ et 0 autrement :

$$\forall\ task_a \in TASK, chain_x \in CHAINS\ :$$

$$Exec\binom{task_a,}{chain_x} = \begin{cases} 1 & \text{ssi} \begin{pmatrix} \exists\ ec_r \in chain_x \mid \begin{pmatrix} \exists\ reg_m \in TaskTag(task_a) \mid \\ (reg_m, R\mathrm{u}_*) \in ec_r \end{pmatrix} \\ \wedge \begin{pmatrix} \forall\ info_z \in InfoOut(task_a) \cup InfoIn(task_a)\ : \\ \exists\ reg_n \in InfoTag(info_z) \mid (reg_m, reg_n) \in ec_r \end{pmatrix} \end{pmatrix} \\ & \text{avec } R\mathrm{u}_*\text{, une région d'un utilisateur du système} \\ 0 & \text{autrement} \end{cases}$$

Dans l'exemple de système simplifié à trois composants proposé dans cette section, seul le groupe de contextes équivalents désigné par $chain_4$ présente les conditions d'exécution

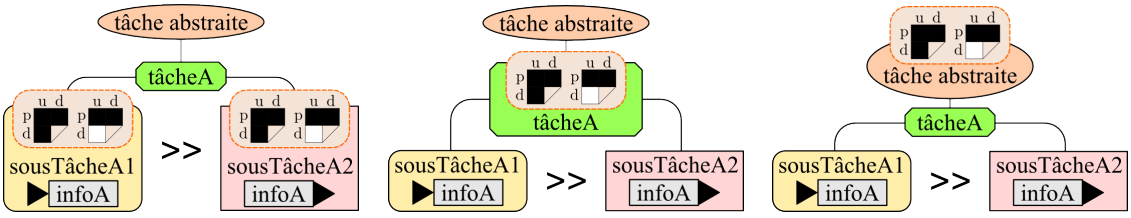
requises pour effectuer les tâches "sousTâcheA1" et "sousTâcheA2". Dans les autres chaînes formées, soit l'utilisateur n'accède pas aux traitements qui implémentent les tâches (cas de la chaîne $chain_2$), soit l'utilisateur accède aux traitements mais ceux-ci ne disposent pas des informations en entrée/sortie (cas de la chaîne $chain_1$), ou encore l'utilisateur accède aux informations, mais ne dispose pas des traitements pour effectuer les tâches (cas de la chaîne $chain_3$). Par exemple, dans le cas de la tâche "sousTâcheA1", il est vérifié que :

$$Exec(\text{sousTâcheA1}, chain_1) = 0 \qquad Exec(\text{sousTâcheA1}, chain_2) = 0$$
$$Exec(\text{sousTâcheA1}, chain_3) = 0 \qquad Exec(\text{sousTâcheA1}, chain_4) = 1$$

Contextualisation de l'arbre des tâches annoté

L'arbre des tâches annoté, qui comporte les informations d'entrée/sortie liées à chaque tâche (Fig. IV.11a(a)), peut être mis en contexte en associant à chaque tâche interactive ou système l'ensemble des chaînes de contextes qui les valident les conditions d'exécution minimales nécessaires au déroulement d'une tâche. Cette association d'une chaîne d'équivalence à une tâche signifie que la tâche considérée est conditionnée par la chaîne d'équivalence. Lorqu'une chaîne d'équivalence conditionne une tâche, alors, cette tâche et l'ensemble de ses sous-tâches ne peuvent être effectués que si le contexte d'exécution mesuré fait partie de la chaîne d'équivalence considérée. Dans le cadre de cette étude, une tâche est effectuée dans un seul contexte. Autrement dit, le contexte rencontré à la racine de l'arbre, est considéré permanent jusqu'aux feuilles. Les cas qui impliquent un changement de contexte en cours de réalisation d'une tâche, ou lors de la succession de sous-tâches, ne sont pas considérés. Pour l'utilisateur, cette restriction signifie qu'il doit conserver les mêmes conditions d'exécution lorsqu'il effectue une tâche de l'arbre.

Les groupes de contextes représentant les chaînes d'équivalence sont ajoutés à l'arbre de tâches annoté et permettent une lecture graphique des conditions d'environnement nécessaires au déroulement d'une tâche. Par exemple, la figure IV.12a représente une forme développée d'arbre des tâches annoté et mis en contexte où les tâches "sousTâcheA1" et "sousTâcheA2" sont conditionnées par la chaîne d'équivalence $chain_4$.



(a) Forme développée (b) Factorisation intermédiaire (c) Forme contextualisée

Figure IV.12 – Construction d'une représentation d'un arbre de tâches annoté mis en contexte

Cette forme développée est un intermédiaire entre l'arbre annoté et sa version contextualisée. L'objectif est de faire remonter les contraintes d'exécution au maximum jusqu'à ce qu'il existe un groupe d'équivalence qui conditionne la tâche racine (Fig. IV.12b). Cette forme factorisée au maximum constitue la version contextuelle de l'arbre annoté (Fig. IV.12c). Elle est obtenue par l'application de règles de transformation entre différentes versions d'arbres

des tâches pour faire remonter les contraintes contextuelles vers les nœuds parents. La figure IV.13 présente ces factorisations en considérant trois tâches $task_a$, $task_b$ et $task_c$, qui peuvent être indifféremment des tâches abstraites, interactives, système ou utilisateur, et $chains_x$ et $chains_y$, deux ensembles de chaînes d'équivalence telles que $\{chains_x, chains_y\} \subseteq CHAINS$. Différents types de factorisations sont distingués :

- lorsque deux tâches sont articulées par un opérateur "ou", alors leur tâche parente peut être effectuée indifféremment par l'une comme par l'autre. Cette tâche parente se trouve conditionnée par l'union des chaînes d'équivalence qui valident les tâches filles. Le résultat de la factorisation proposée en figure IV.13b se lit "*Si le contexte mesuré fait partie des chaînes d'équivalence $chains_x$ ou $chains_y$, alors la tâche $task_a$ peut être effectuée. Cette tâche sera effectuée par la sous-tâche $task_b$ si le contexte mesuré fait partie d'une chaîne de $chain_x$, ou par la sous-tâche $task_c$ sinon.*" ;
- si deux tâches sont articulées par un opérateur "ou" et que les contextes qui les conditionnent sont les mêmes, alors leur tâche parente est conditionnée par ce même ensemble de chaînes d'équivalence et il n'est plus nécessaire de distinguer les chaînes qui conditionnent les tâches filles. Le résultat de la factorisation proposée en figure IV.13c se lit "*Si le contexte mesuré fait partie d'un élément de $chains_x$, alors la tâche $task_a$ peut être effectuée indifféremment soit par la sous-tâche $task_b$ soit par la sous-tâche $task_c$.*" ;
- si deux tâches doivent s'opérer l'une à la suite de l'autre, alors il existe au moins une chaîne d'équivalence commune à ces sous-tâches pour qu'elles puissent être effectuées dans un même contexte. Leur tâche parente est conditionnée par ces groupes d'équivalence communs. Le résultat de la factorisation proposée en figure IV.13a se lit "*Si le contexte d'exécution mesuré appartient à un élément de l'union de $chain_x$ et $chain_y$, alors la tâche $task_a$ peut être effectuée par l'enchaînement des sous-tâches $task_b$ et $task_c$*" ;
- une tâche parente qui n'est opérée que par une unique sous-tâche se trouve conditionnée par les mêmes chaînes de contextes que celle-ci. Le résultat de la factorisation proposée en figure IV.13d se lit "*Si le contexte mesuré figure dans un élément de $chain_x$, alors la tâche $task_a$ peut être effectuée par sa sous-tâche $task_b$*" ;
- lorsqu'une sous-tâche ne peut être effectuée dans aucune des conditions d'exécution proposées dans l'environnement du système, alors sa tâche parente ne peut elle même être effectuée et il n'est pas nécessaire d'implémenter cette branche de l'arbre. Le résultat de la factorisation proposée en figure IV.13e se lit "*Une tâche qui n'est pas conditionnée par au moins une chaîne d'équivalence n'est pas réalisable, et ses sous-tâches peuvent etre supprimées de l'arbre*".

Dans le cas le plus favorable, en ayant au départ associé les chaînes de contextes aux niveaux les plus bas de l'arbre annoté, et après plusieurs tours de factorisation, la tâche abstraite à la racine de l'arbre annoté se trouve elle aussi conditionnée par au moins une chaîne de contextes. Dans ce cas, il existe au moins un contexte d'exécution dans lequel l'utilisateur peut atteindre son objectif tel qu'il a été déterminé à l'analyse du scénario nominal. L'exemple de décomposition de tâche qui illustre cette section est contextualisé après deux étapes de factorisation (Fig. IV.12(a, b et c)). À l'issue, il est possible d'effectuer la tâche abstraite et donc le scénario nominal si l'utilisateur mesure un contexte parmi $chain_4$.

Dans le cas défavorable, la tâche racine n'est toujours pas conditionnée après l'étape de factorisation maximale. Dans cette situation, l'arbre des tâches ne peut être effectué dans

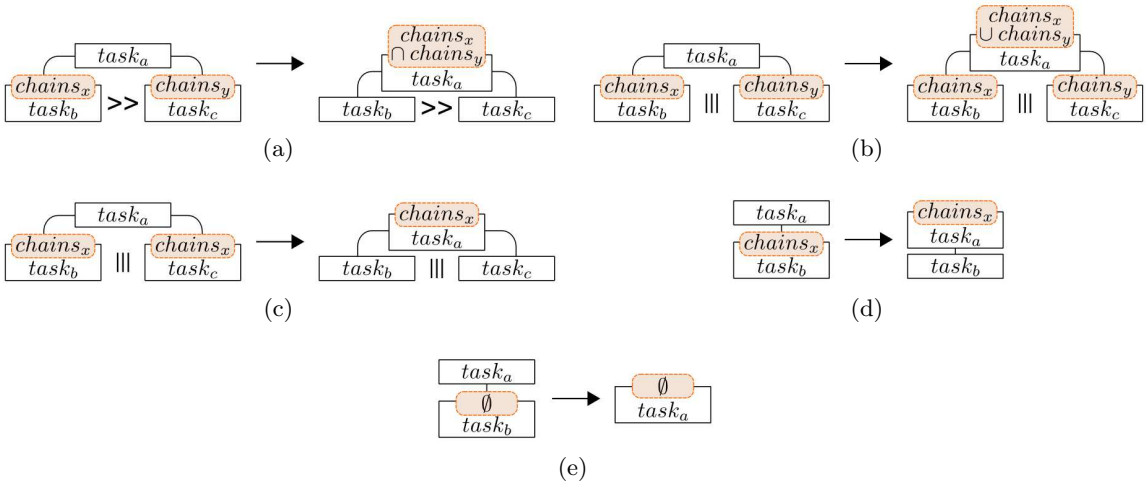


Figure IV.13 – Transformations par factorisation de contraintes pour la construction d'un arbre contextualisé

aucune des conditions d'exécution rencontrées[8]. Avec l'objectif de satisfaire à la fois les exigences des commanditaires du système et les besoins des utilisateurs, il est essentiel qu'au moins un contexte d'exécution garantisse le déroulement du scénario nominal. Pour corriger ce problème éventuel, les concepteurs peuvent agir sur plusieurs leviers complémentaires :

- l'arbre des tâches peut être réorganisé pour contourner les problèmes liés à l'ordonnancement des tâches. Par exemple l'opérateur "ou" est moins contraignant que l'opérateur "et" du point de vue de la factorisation des conditions d'exécution vers les tâches parentes ;
- si la réorganisation de l'arbre dans son ensemble est trop contraignante, la distribution des procédures parmi les composants peut être repensée de façon à trouver des conditions suffisantes pour effectuer les niveaux bas de l'arbre. Dans le cas des tâches qui ne nécessitent pas de traitement complexe, un moyen souvent efficace pour ajouter des conditions valides pour effectuer ces tâches consiste à consacrer une partie des ressources du composant client pour effectuer des micro-traitements et impliquer un composant de traitement en moins dans le contexte d'exécution ;
- similairement, les concepteurs peuvent distribuer les données plus équitablement entre les composants, de façon à ce que l'indisponibilité d'une information ne bloque plus l'exécution d'une tâche pour une chaîne de contextes donnée. Par exemple, il est envisageable de répliquer certaines données aux niveaux de composants de traitement, voir sur la plate-forme de l'utilisateur, pour que ces informations soient disponibles quel que soit le contexte[9] ;
- enfin, il est éventuellement possible de revenir sur le niveau de la distribution des composants et sur leurs capacités mobiles pour définir des espaces accessibles qui per-

8. En général ces problèmes apparaissent quand une tâche parente à un ensemble de tâches articulées par des opérateurs de succession ne trouve pas de contexte commun qui permette de valider la succession des tâches filles. Dans ce cas, cette tâche parente elle-même n'est pas effectuable. De proche en proche, certaines branches qui incluent éventuellement la racine deviennent inopérables.

9. Dans ce cas, l'utilisateur voit son rôle changer, son client peut être assimilé à un composant de traitement ou à un serveur de données et les régions utilisateur et données/traitement peuvent être confondues. Une telle configuration est illustrée au chapitre VI.2, dans l'exemple d'un système de visite de campus pour lequel les régions de traitement et de données sont systématiquement appairées.

mettent de nouvelles relations entre régions et un nombre plus important de contextes et de groupes d'équivalence. Ces groupes supplémentaires viennent éventuellement conditionner des tâches qui ne pouvaient pas initialement être effectuées.

Ajout de tâches alternatives

La dernière étape de l'intégration de la description géographique des contextes d'exécution consiste à imaginer des comportements alternatifs pour les cas d'exécution où les contextes mesurés ne figurent pas parmi l'arbre des tâches mis en contexte. Pour garantir un service maximum quelle que soit la situation, les concepteurs ajoutent un ensemble de tâches alternatives, conditionnées par des chaînes d'équivalence non encore utilisées dans l'arbre annoté, de façon à ce que chaque chaîne se trouve représentée dans l'arbre des tâches et conditionne la racine de l'arbre. Éventuellement, les utilisateurs sont impliqués dans la détermination des tâches alternatives. Avec les concepteurs, ils réfléchissent au sens que prennent les chaînes de contextes restantes du point de vue de l'utilisateur, puis ils proposent de nouvelles tâches appropriées dans ces situations.

Dans l'exemple proposé, les chaînes de contextes $chain_1$, $chain_2$ et $chain_3$ ne conditionnent aucune partie de l'arbre de tâches mis en contexte (Fig. IV.12c). Les concepteurs portent leur attention sur ces trois chaînes de contextes pour définir de nouvelles tâches. Notamment, la chaîne $chain_2$ rassemble les situations où l'utilisateur se retrouve isolé. Dans ce cas, un comportement alternatif du système pourrait être de prévenir l'utilisateur qu'il se situe en dehors des aires de service du système. Cette tâche système alternative "alternA" effectue ses traitements sur la seule plate-forme disponible dans ce contexte, le composant $C\mathrm{u}_1$ de l'utilisateur, et ne nécessite pas d'information particulière en entrée/sortie. Dans ce cas, avec $TaskTag(\mathrm{alternA}) = \{R\mathrm{u}_1\}$ et $InfoIn(\mathrm{alternA}) = InfoOut(\mathrm{alternA}) = \emptyset$, il est possible d'exécuter "alternA" dans l'un des contextes de $chain_2$ et $Exec(\mathrm{alternA}, chain_2) = 1$. Cette tâche est ajoutée à l'arbre sous la forme d'une alternative à la tâche "tâcheA". De la même manière, des comportements alternatifs sont définis pour les chaînes d'équivalence $chain_1$ et $chain_3$.

Entre chaque ajout de nouvelles branches, l'arbre est re-factorisé. Une fois que chaque alternative a été intégrée, l'ensemble des chaînes d'équivalence conditionne la racine de l'arbre, et chaque contexte de $STATES$ constitue un point d'entrée dans la décomposition des tâches (Fig. IV.14). De cette manière, il existe une partie de l'arbre, incluant des tâches d'interaction ou système, qui peut être effectué dans n'importe quelle contexte d'exécution mesuré.

L'arbre des tâches situé résulte de l'ajout de tâche alternatives et de la prise en compte de l'ensemble des chaînes de contextes équivalents. Cet élément, nœud du cadre de conception étendu, peut être utilisé pour quantifier la notion de *valeur utilitaire* du système à un moment donné. Cette mesure, associée à la valeur interactive d'une interface client permet d'estimer la valeur que prendra le système pour un utilisateur (Chap. II.3.3). Une première approche possible pour mesurer l'utilité du système consiste à faire le rapport entre le nombre de nœuds de l'arbre des tâches situé qu'il est possible d'effectuer dans un contexte donné et le nombre de nœuds total de l'arbre. Dans le cas d'un système implémentant l'arbre situé présenté en figure IV.14, un utilisateur isolé pourra effectuer deux tâches parmi six (c-à-d. la tâche abstraite et la tâche système "alternA"). La valeur d'utilité est dans cas estimée à 1/3.

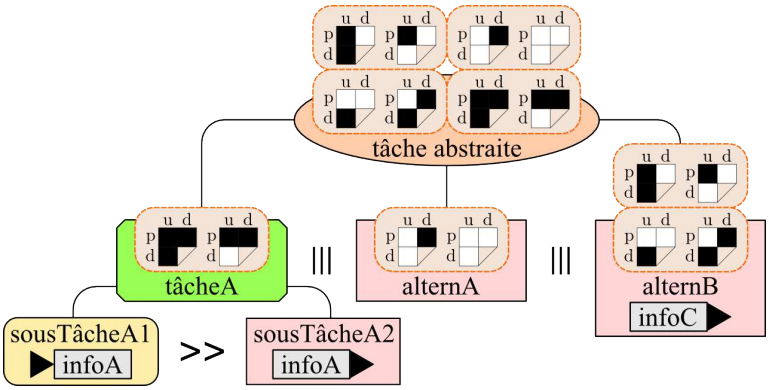


Figure IV.14 – Exemple d'arbre des tâches situé

Lorsque le contexte mesuré implique plus de relations entre régions, et que la tâche "tâcheA" est effectuable, ce rapport s'inverse et l'utilité du système passe à 2/3, son maximum.

IV.3 Conclusion

Ce chapitre présente une méthode de corrélation entre une description géographique par un ensemble de régions d'un système mobile et distribué et une méthodologie de conception de système interactif centrée sur l'analyse des besoins et des tâches des utilisateurs.

La démarche proposée pour établir cette corrélation utilise comme base un modèle de référence dans le cadre de la conception de systèmes interactifs. Ce cadre de conception unifié part d'une analyse scénarisée des besoins des utilisateurs pour établir une décomposition de ses tâches sous la forme d'un arbre. Cet arbre représente à la fois les fonctionnalités qui devront être implémentées au sein de procédures du système et les informations dont ces procédures se serviront en entrée et en sortie. Le cadre unifié propose à partir de cet arbre de tâches annoté de concevoir et d'implémenter de façon semi-automatique les niveaux du dialogue et de l'interface avec les utilisateurs. Cette implémentation du tiers client propose notamment de tenir compte du contexte d'usage de l'utilisateur pour adapter le style d'interface et de dialogue.

La mise en œuvre au sein d'un système interactif d'une variété de situations d'exécution d'un système distribué et mobile demande des adaptations plus profondes que les modifications d'usage supportées par le cadre de conception unifié. Notamment, les contextes d'exécution caractérisés par les relations entre régions d'exécution ou d'intérêt sont susceptibles de venir altérer l'enchaînement des tâches, voire d'invalider complètement le scénario nominal prévu par les concepteurs. Dans l'objectif d'intégrer cette versatilité des situations d'exécution, le cadre de conception étendu est proposé comme une version augmentée du cadre de conception unifié. Ce guide de conception reprend l'ensemble des contextes d'exécution déterminés à l'issue des étapes de description géographique et les confronte au modèle des tâches des utilisateurs.

Deux étapes permettent cette corrélation entre les contextes d'exécution modélisés et l'arbre des tâches nominal. Dans un premier temps, une recherche d'équivalence est effectuée

parmi les contextes d'exécution pour grouper sous forme de chaînes d'équivalence l'ensemble des contextes qui produisent les mêmes conditions d'exécution du point de vue de l'utilisateur et pour lesquels le système offrira un même niveau de fonctionnalité. Ensuite, ces chaînes de contextes viennent conditionner les branches de l'arbre des tâches annoté pour constituer un arbre des tâches situé. Celui-ci rassemble, dans une même organisation des tâches, les branches décomposant la tâche "nominale", et des branches complémentaires, imaginant des nouvelles réponses fonctionnelles adaptées à des contextes d'exécution ne permettant pas le déroulement du cas nominal. L'arbre des tâches situé constitue le noeud à partir duquel peut être implémenté une interface client robuste et cohérente. Les détails de répartition des informations et des traitements entre les différents composants distribués de la plate-forme fournissent un second point d'entrée en vue du déploiement des données et des procédures. Ces derniers aspects constituent les étapes d'ingénierie du système et dépendent trop des cas d'application choisis pour être décrits en termes conceptuels. Un exemple d'ingénierie d'un client mobile à partir de la spécification conceptuelle de l'arbre des tâches situé sera exposé au chapitre VI.1.

Le cadre de conception étendu offre une solution pour mettre en œuvre la sous-dimension d'exécution/système d'un contexte plus global qui inclut le contexte d'usage. Le chapitre suivant s'intéresse à cette dernière partie de la notion de contexte et plus particulièrement à l'apport d'un système fondé sur la notion de relation entre régions pour caractériser la dimension "utilisateur" du contexte d'usage. Cette approche s'applique au cas des systèmes multi-utilisateurs et passe par la notion de groupes dont la catégorisation ne s'effectue plus par un niveau d'expertise, ou par des critères sociaux, mais se fonde plutôt sur le contexte d'exécution que les utilisateurs du groupe rencontrent. En sachant qu'à un contexte d'exécution donné correspondent un ensemble fonctionnel déterminé et une situation géographique particulière, des utilisateurs mesurant simultanément ce contexte peuvent former un groupe cohérent pour la détermination d'un profil commun et l'application d'adaptation du contenu, de l'interface et du style d'interaction proposés.

CHAPITRE V

Utilisation des contextes d'exécution pour la personnalisation

Sommaire

V.1 Mesure implicite des préférences des utilisateurs 141
V.1.1 Création d'un profil par l'apport d'expérience 141
V.1.2 Méthode hybride pour la construction de profils dans un système mobile et distribué 144
V.2 Propositions pour la construction de profils et la mesure de scores 147
V.2.1 Création d'un profil utilisateur 149
V.2.2 Perspective multi-utilisateurs et regroupement de profils 156
V.3 Conclusion 166

Le cadre de développement d'un système mobile et distribué étudié au chapitre IV associe à chaque situation d'exécution, ou groupe de situations, un comportement approprié de la part du système. Ces comportements visent à offrir un maximum de fonctionnalités à l'utilisateur compte tenu des ressources disponibles parmi les composants de la plate-forme, et compte tenu de la localisation de l'utilisateur par rapport à un ensemble de régions d'intérêt. Les relations existantes entre les régions établissent un contexte d'exécution à un instant donné et ce contexte d'exécution permet d'effectuer tout ou partie de la tâche d'un utilisateur.

L'approche des contextes d'exécution fournit des outils d'aide à la conception pour la caractérisation de différents niveaux d'exécution d'un même système. Les adaptations effectuées par le système à l'exécution consistent à passer d'un niveau fonctionnel à un autre. Par exemple, la perte d'une relation entre deux régions d'exécution peut signifier du point de vue du système la perte d'une source de données. Dans ce cas, le système doit signifier cette perte à l'utilisateur et adapter les fonctionnalités dont il dispose pour invalider celles qui requièrent les ressources perdues pour fonctionner. Ce processus d'adaptation se fait essentiellement aux dépens de l'utilisateur et de sa capacité à atteindre l'ensemble de ses objectifs. Les réponses du système lors de dégradations des conditions d'exécution peuvent être vues comme les "moins pires" des adaptations. Elles établissent un consensus entre les capacités

de la plate-forme et les besoins des utilisateurs ciblés. Intégrer ces contextes d'exécution très tôt lors de la conception, au moment de la description de la tâche, permet de constituer un arbre des possibilités offertes à l'utilisateur en connaissance des conditions d'exécution. Cette description dans un arbre des tâches situé est à la base de l'implémentation à la fois des tiers client et métier/données. Au niveau du client en particulier, l'arbre situé permet d'identifier dans un même outil les différents niveaux fonctionnels et les adaptations que le système est susceptible d'opérer lors de l'exécution. Cette connaissance de l'ensemble des niveaux fonctionnels permet une implémentation plus immédiate du dialogue et de l'interface.

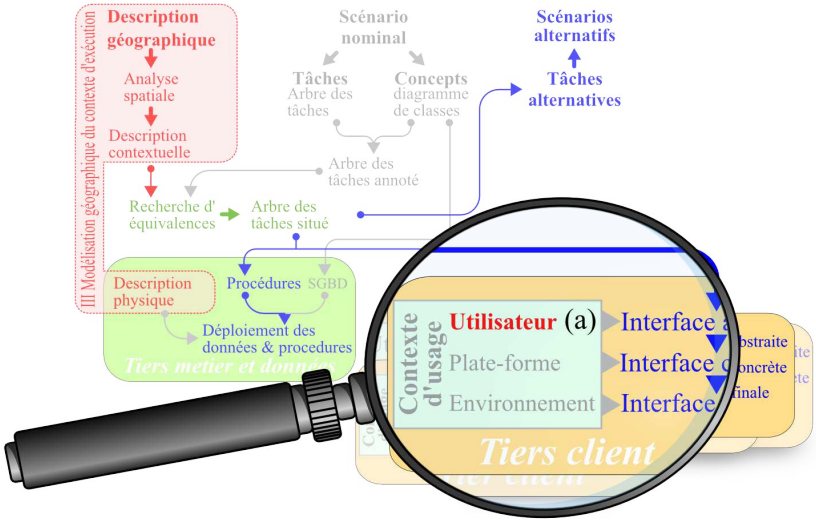


Figure V.1 – Intégration des mesures du contexte d'usage pour l'individualisation du client

La démarche proposée par le cadre de conception étendu permet une intégration naturelle des variations de la plate-forme distribuée. Dans ces étapes de conception, le modèle ne propose pas la même latitude d'adaptation pour les utilisateurs cibles. Plus particulièrement, les intentions, les buts et les préférences des utilisateurs ne sont plus remis en cause une fois le scénario nominal et l'arbre des tâches décrits. Dans les systèmes distribués où l'utilisateur est lui-même mobile, cette permanence des attentes des utilisateurs est très discutable et l'étude bibliographique du chapitre I a montré que l'environnement proche de l'utilisateur influence ses besoins et son comportement vis-à-vis du système. Pour garantir que le système offre une palette d'exécution suffisante pour satisfaire les utilisateurs, la dimension complémentaire du contexte d'usage est intégrée à la modélisation du client (Fig. V.1). Ses trois composantes, "Utilisateur", "Plate-forme cliente" et "Environnement", permettent d'affiner dynamiquement les fonctionnalités et l'interaction offertes dans un contexte d'exécution donné, et éventuellement d'adapter le système à des comportements d'utilisation qui n'auraient pas été précisés avant l'exécution.

Ce chapitre propose de se servir de la dimension "Utilisateur" du contexte d'usage pour individualiser le système et offrir une certaine plasticité à l'exécution malgré une définition stricte des besoins des utilisateurs lors de la conception. L'approche d'individualisation proposée s'appuie sur une double conjecture :

- dans un premier temps, il est possible de repérer implicitement les comportements interactifs, ou les préférences fonctionnelles et informationnelles des utilisateurs au moment de l'exécution ;
- ensuite, ces informations récoltées peuvent servir à la composition de profils communs entre plusieurs utilisateurs, pour faire émerger des préférences relatives à un contexte d'exécution donné.

Ces deux étapes spécialisent le comportement du système à l'exécution sans que les concepteurs n'aient à prévoir à l'avance le résultat de cette spécialisation. Plus spécifiquement, le système compose à partir de profils d'utilisateurs un ensemble de préférences propres à chaque contexte d'exécution et adapte (dans ce cas il individualise) à la fois l'interface et le contenu proposés dans ces contextes d'exécution. Avant de détailler dans le reste du chapitre l'individualisation d'un système modélisé selon l'approche du cadre de conception étendu, la suite de cette section motive le besoin de l'individualisation et décrit les différents leviers qui permettent sa mise en oeuvre.

Intérêt de l'individualisation dans les systèmes mobiles

L'individualisation des systèmes mobiles est motivée par l'usage croissant de nouvelles plates-formes d'interaction en mobilité. Les innovations portées par ces plates-formes, et notamment leur capacité à la localisation géographique, élargissent les domaines d'application des systèmes mobiles. En particulier, le succès des dernières générations de téléphones portables, d'organisateurs ou de systèmes de navigation va de pair avec le développement des systèmes mobiles et distribués qui augmentent de façon significative la quantité d'informations accessibles à un utilisateur mobile.

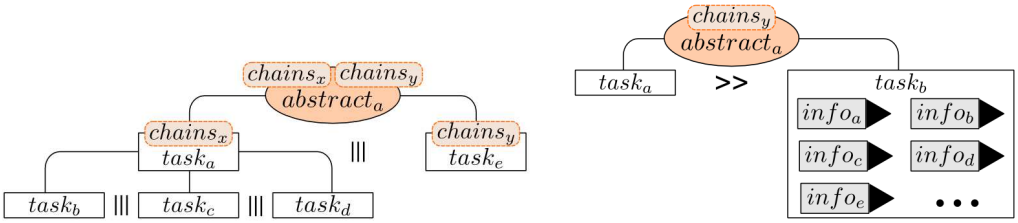
Contrastant avec cette richesse d'information disponible, les plate-formes mobiles deviennent plus petites et portables et il est souvent souhaitable d'effectuer un filtrage des données présentées à l'utilisateur pour individualiser son expérience du système en fonction de ses besoins du moment. Cette démarche d'individualisation a pour but d'éviter à l'utilisateur d'être inondé par des informations ou des fonctionnalités qui ne seraient pas pertinentes pour lui. Par exemple, dans le cas des systèmes localisés, il peut être approprié de conditionner l'affichage d'informations relativement à l'endroit où se trouve l'utilisateur.

Pour arriver à un bon niveau de personnalisation, il est utile d'établir un profil des préférences d'un utilisateur. Dans la suite de ce chapitre, différentes solutions pour la construction de profils sont présentées. Au final, il est proposé de se servir d'un profil pour identifier les informations les plus appropriées pour un contexte et un utilisateur donnés. Les profils d'utilisateurs sont les éléments atomiques des algorithmes d'individualisation qui permettent l'implémentation d'une grande variété de processus d'adaptation. Il est possible en travaillant sur le contenu des profils, sur leur association, ou sur leur diffusion dans l'espace du système, de dépasser largement le cadre de la personnalisation dédiée à un seul utilisateur. Notamment, ce chapitre s'attache à la notion de groupes d'utilisateurs pour la construction de profils et à la diffusion et au partage de préférences au cours de l'exécution.

Leviers logiciels pour opérer l'individualisation à partir de profils

L'application de profils pour individualiser un système est une méthode d'adaptation dont les mesures contextuelles appartiennent au contexte d'usage (Chap. II.2.2). D'après le cadre de conception étendu, la mesure du contexte d'usage est complémentaire à la mesure du contexte d'exécution. Le contexte d'exécution détermine à haut niveau les capacités fonctionnelles du système et impacte le déroulement des tâches des utilisateurs. Le contexte d'usage s'attache plus particulièrement à la spécialisation du client et constitue un point d'entrée pour distinguer différentes manières d'organiser les niveaux de l'interface avec l'utilisateur (Fig. V.1(a)). L'application d'adaptations en conséquence de mesures du contexte d'usage s'effectue dans des circonstances d'exécution particulières qui limitent la portée des modifications applicables. En particulier, il n'est pas possible d'augmenter l'ensemble des fonctionnalités à disposition de l'utilisateur puisque ce maximum fonctionnel est déjà limité par le contexte d'exécution courant. En revanche, à l'inverse, il est possible et parfois souhaitable d'appliquer des filtres sur les fonctionnalités et les informations offertes au niveau des clients mobiles pour limiter l'utilisation et l'interaction des utilisateurs à ce qui leur est réellement utile.

Cette étude présente deux cas généraux, repérés au sein de l'arbre situé d'un système donné et qui forment des candidats potentiels pour l'application de règles de filtrage à partir de profils d'utilisation (Fig. V.2). Une première application de l'adaptation à un contexte d'usage considère que s'il existe plusieurs décompositions de tâches permettant à un utilisateur d'atteindre son objectif, il est possible de proposer un ordonnancement de ces décompositions basé sur les préférences des utilisateurs. Pour un système implémentant l'arbre des tâches illustré en figure V.2a, un utilisateur mesurant un contexte d'exécution parmi $chains_x$ peut réaliser une tâche abstraite $abstract_a$ en effectuant indifféremment l'une des trois sous-tâches $task_b$, $task_c$ ou $task_d$. Si le profil d'un utilisateur mesure les fréquences auxquelles il effectue chacune des trois sous-tâches, une adaptation envisageable au niveau du client porterait à l'avant plan les outils pour réaliser les sous-tâches les plus souvent effectuées.



(a) Choix entre plusieurs façons d'effectuer une (sous-)tâche

(b) (Sous-)Tâche mobilisant un ensemble important d'informations

Figure V.2 – Exemples d'arbres situés pour lesquels l'individualisation peut être appliquée

La seconde application de filtrages contextuels porte sur les informations mobilisées lors de l'exécution d'une tâche ou sous-tâche. Dans les situations où le système est amené à offrir un ensemble important d'informations pour se conformer aux besoins des utilisateurs, il est possible d'ordonner la présentation de ces informations, voire de cacher certaines d'entre elles pour limiter la charge cognitive de l'utilisateur et simplifier sa lecture des données présentées. Dans le cas d'un système implémentant l'arbre situé détaillé en figure V.2b, un utilisateur mesurant un contexte d'exécution parmi $chains_y$ effectuera la tâche abstraite $abstract_a$ en

deux étapes $task_a$ puis $task_b$. La sous-tâche $task_b$ mobilise un ensemble de données en sortie, qui sont produites vers l'utilisateur. Si un profil de préférences de l'utilisateur contient un ensemble ordonné de scores sur ces informations, il est possible de présenter le résultat informationnel de $task_b$ en cohérence avec ces scores. Par exemple, les informations les plus importantes d'après les utilisateurs, et qui obtiennent un score élevé, peuvent être présentées en premier lieu au niveau de l'interface.

Chacun des cas d'application présentés repose sur des profils d'utilisateurs pour appliquer des adaptations au niveau de la présentation du contenu et de la présentation des outils de manipulation de ce contenu. La section suivante analyse plus avant cette notion de profils. Plus particulièrement, un ensemble de méthodes pour la construction de profils personnels est analysé et une proposition de construction adaptée aux systèmes distribués conçus selon l'approche du cadre de conception étendu est détaillée.

V.1 Mesure implicite des préférences des utilisateurs

Les mesures de la dimension utilisateur du contexte revient à générer un profil utilisateur qui résume l'ensemble des propriétés contextuelles en relation avec l'utilisateur (p.ex. son curriculum, ses préférences d'usage, etc.) pouvant être utiles pour l'adaptation du comportement du système au niveau de la plate-forme cliente. Les profils d'utilisateurs, abordés au chapitre II.3.1, sont construits soit par le **renseignement explicite** de la part des utilisateurs, soit par l'application de **profils pré-déterminés** décrits par les concepteurs, ou soit par un **renseignement implicite** des mesures du profil inférées à partir des actions des utilisateurs. Cette dernière approche est plus souvent développée dans le cas des systèmes distribués et mobiles. Bien que le renseignement implicite de profils constitue une approximation de préférences, les bénéfices de cette méthode pour un utilisateur, éventuellement en déplacement, sont doubles :

- contrairement à l'application de profils pré-déterminés, la personnalisation du contenu et des fonctionnalités reste dynamique à l'exécution ;
- par rapport au renseignement explicite du profil, la mesure implicite permet à l'utilisateur de rester concentré sur son action en cours sans avoir à fournir d'indications de préférences pour bénéficier d'adaptations contextuelles.

Dans une première partie, cette section détaille et critique les principales méthodes de mesure implicite de profil personnel ou de profil de groupe. Dans la suite, une méthode hybride de constitution de profil est proposée pour adresser la difficulté de mise en oeuvre des approches classiques de construction de profils. Cette méthode hybride s'appuie particulièrement sur l'infrastructure d'un système mobile et distribué développé selon l'approche du cadre de conception étendu (Chap. IV.1.2), et est illustrée par un ensemble de métriques appropriées dans le cas de la consultation d'informations localisées sur une carte numérique.

V.1.1 Création d'un profil par l'apport d'expérience

Selon , deux techniques de mesures sont couramment adoptées pour l'initialisation implicite d'un profil utilisateur. La **construction par analyse de contenu** (content-based recommendation) considère les comportements passés d'un individu particulier comme des

indicateurs de son comportement à venir. Par exemple, un utilisateur ayant depuis un mois consulté chaque matin en arrivant au bureau la même série de sites Web d'informations fournit des données de référence permettant de déterminer de façon fiable les chances de reproduction de ce motif les matins suivants. Un navigateur Internet, analysant la navigation Web de cet employé de bureau, déduirait de la répétition des consultations une mesure de la dimension utilisateur précisant par exemple les catégories de sites consultés par l'utilisateur en début de journée.

La **construction par analyse conjointe** (collaborative recommendation) considère les comportements passés d'un ensemble d'individus groupés selon une valeur de proximité contextuelle comme un indicateur du comportement à venir d'un utilisateur particulier, associé au groupe considéré. Par exemple, si l'ensemble des employés d'un bureau consulte les mêmes sites d'information en arrivant au travail, alors l'association par proximité contextuelle d'environnement (dans ce cas, le partage d'un même espace de bureau) d'un utilisateur au groupe des employés du bureau fournit une information statistique fiable sur les chances d'observer une reproduction des comportements de navigation des autres occupants du bureau par ce nouvel employé. Comme dans le cas de la construction par analyse de contenu, un navigateur Internet peut tirer bénéfice de cette analyse statistique des comportements de groupes pour compléter les mesures contextuelles de la dimension utilisateur et présenter les pages Web d'informations attendues dès l'arrivée de l'utilisateur sur son lieu de travail.

Ces deux approches pour la construction implicite de profils utilisateurs disposent d'avantages et d'inconvénients. L'approche par analyse de contenu permet une personnalisation au grain des individus, mais nécessite la collection implicite d'une quantité importante de données sur les habitudes passées des utilisateurs avant de pouvoir établir des vérités statistiques complétant les mesures du profil utilisateur. Dans le domaine de la personnalisation implicite, ce passage d'un profil vide à un profil utilisable est connu comme le problème du **démarrage à froid** (Cold start initialization ()). En corollaire de ce problème, les profils générés souffrent d'une certaine **inertie du contenu** liée à la difficulté de mesurer des changements de comportement par une analyse statistique ().

A l'inverse, l'approche par analyse conjointe est moins sensible à l'inertie de la génération des profils par analyse de contenu et élimine le problème du démarrage à froid en associant le profil du groupe à tout nouvel utilisateur s'y rattachant. Toutefois, selon , le regroupement d'utilisateurs pour la définition de profils personnels biaise la notion de personnalisation en ne créant que de nouveaux stéréotypes d'utilisateurs, qui satisfont les utilisateurs en général, mais aucun d'entre eux en particulier. corroborent cette analyse : pour les auteurs, les profils générés en groupe peuvent s'avérer contre-productifs s'ils sont appliqués à des individus ne se reconnaissant pas dans la population de leur groupe. L'un des principaux défis pour répondre à ces critiques consiste à déterminer la bonne **métrique d'association** des utilisateurs et la constitution des groupes. Cette problématique de dimensionnement des groupes se rapporte aux notions d'intelligence artificielle de *sur-apprentissage* et *sous-apprentissage* statistique () : avec trop peu de contraintes dans les critères d'association d'utilisateurs, les groupes deviennent trop importants et les mesures de la dimension utilisateur n'ont plus rien de personnel ; à l'inverse, avec des critères d'association trop stricts, les groupes n'ont pas les dimensions nécessaires pour faire ressortir des mesures généralisables à un profil partagé.

Les deux techniques de mesure s'opposent dans leurs avantages et inconvénients. De nombreuses expérimentations pour la personnalisation implicite reposent sur des méthodes hybrides qui tentent "d'associer les avantages de ces techniques *[par analyse conjointe et analyse de contenu]* sans hériter de leurs inconvénients" (). classe différentes propositions pour l'hybridation des techniques de mesure (Tab. V.1). En général, le but recherché en combinant les deux techniques est la résolution des problèmes de démarrage à froid et d'inertie des profils constatés à l'utilisation de la technique d'analyse de contenu ().

Tableau V.1 – Méthodes pour l'hybridation des techniques de mesure de la dimension utilisateur (d'après)

Méthode	**Description**	**Exemple**
Pondération	Les mesures par analyse conjointe et par analyse de contenu sont pondérées puis combinées pour former un profil utilisateur commun.	(;)
Alternance	Le système alterne entre les mesures par analyse conjointe et par analyse de contenu selon la situation à l'exécution (p.ex. présence nécessaire d'autres d'utilisateurs pour effectuer une analyse conjointe).	(;)
Mitigeur	Les mesures par analyse conjointe et par analyse de contenu sont associées (mais pas combinées) dans un profil commun. Le système produit une adaptation en choisissant l'une ou l'autre des valeurs du profil.	(;)
Cascade	Une technique de mesure est utilisée pour affiner ou modérer les résultats de l'autre (p.ex, seules les mesures par analyse conjointe qui corroborent l'analyse de contenu sont enregistrée en tant que profil).	(;)
Augmentation	Le profil utilisateur initialisé en application d'une technique est fourni en entrée de l'autre (p.ex. les mesures conjointes servent de base d'application aux mesures par analyse de contenu).	(;)

Méthodes de mesure implicite d'un profil

En pratique, chaque technique de mesure implicite de la dimension utilisateur, que ce soit par analyse de contenu, par analyse conjointe, ou alors par une méthode hybride, repose sur l'enregistrement et l'étude statistique d'objets contextuels. Pour la dimension utilisateur, ces objets sont appelés *indicateurs implicites d'intérêt* (Implicit Interest Indicator, ()). Leur mesure repose sur l'étude non-intrusive des comportements des utilisateurs, des requêtes qu'ils formulent, et des informations qui leur sont présentées (). Ces indicateurs sont relevés au moment de l'exécution et au niveau de la plate-forme cliente par le système d'information lors des interactions avec l'utilisateur. Dans le précédent exemple de la consultation des sites d'information par les employés de bureau, l'objet du contexte constitué par la liste chronologique des sites consultés est exploité comme primitive pour l'analyse de contenu comme pour l'analyse conjointe. Issue de l'observation discrète (c-à-d. non-intrusive) des actions de navigation des utilisateurs, cette liste de consultation est un exemple d'indicateur implicite

d'intérêt utilisé pour la génération de modèles statistiques de préférences des utilisateurs et leur enregistrement sous forme de profil.

L'utilisation d'indicateurs d'intérêt implicites fait l'objet de nombreuses expérimentations et est rencontrée dans la plupart des domaines de recherche pour lesquels la personnalisation représente un enjeu. Notamment, ils ont été particulièrement développés pour l'analyse de l'activité de navigation dans les pages de sites Internet. présentent l'utilisation d'indicateurs implicites dans le cadre de la personnalisation Web. L'extraction de mots clés, l'analyse des signets, les actions de copier/coller, les sélections de texte à la souris, les liens pointés, ou encore les demandes d'impression ou de sauvegarde de documents figurent parmi les mesures discrètement répertoriées (;). Les auteurs montrent, à partir des mesures effectuées sur un corpus de plus de 2500 cas de navigation, que chaque mesure proposée apporte des éléments pour la qualification des intérêts et préférences des utilisateurs vis-à-vis des pages consultées ou des éléments présentés au sein d'une même page. poussent ces expérimentations dans le domaine spatial et mesurent la position du curseur lors de la navigation. Les auteurs montrent que le curseur est souvent utilisé comme guide de lecture dans les pages de texte, et qu'il existe un lien fort entre les endroits de pause du curseur et les éléments prégnants pour l'utilisateur. suggèrent que ces "informations récoltées par l'enregistrement et l'analyse de l'activité du curseur peuvent être très utiles pour inférer ce que l'utilisateur pense". ; appliquent ces analyses dans le cas de l'interaction à la souris sur une carte numérique. Les auteurs enregistrent des sessions d'utilisation et post-analysent les traces de souris et les variations de niveau de zoom, d'orientation et de placement de la carte pour produire un profil d'utilisateur contenant la liste ordonnée par niveau d'intérêt des éléments géo-référencés affichés sur la carte numérique. Dans le domaine de l'information spatiale, utilisent des indicateurs de localisation pour affiner les requêtes soumises à un portail Web de réservation hôtelière. Les auteurs proposent un système prédictif qui, à partir des chemins de liens des utilisateurs du portail web, recommande le prochain contenu. Pour produire les recommandations, ce système considère en entrée une mesure de proximité géographique entre les éléments désignés par les hyperliens (p.ex. les hôtels ou les restaurants d'une ville). Les retours des utilisateurs (en l'espèce, ont-ils suivi ou non le lien proposé par le système ?) permettent d'ajuster les scores attribués aux recommandations. En extension, les auteurs envisagent la catégorisation implicite des utilisateurs selon les proximités de leurs parcours et proposent d'appliquer cette classification à la personnalisation du contenu et de la présentation des pages du portail.

V.1.2 Méthode hybride pour la construction de profils dans un système mobile et distribué

L'infrastructure d'un système d'information mobile et distribué développé selon l'approche du cadre de conception étendu fournit un support pour la construction à l'exécution de profils qui adressent les principaux inconvénients de la mise en œuvre de la mesure implicite des préférences des utilisateurs. A l'exécution, les utilisateurs d'un tel système rencontrent selon leurs déplacements ou selon les mobilités des autres composants du système une série de contextes d'exécution. Éventuellement, les contextes mesurés au niveau de certains clients peuvent être identiques. Dans ce cas, ces utilisateurs partagent un même contexte d'exécution et les propriétés suivantes sont vérifiées :

- les utilisateurs dans un même contexte d'exécution peuvent effectuer un même sous-ensemble de l'arbre des tâches. Par cette **proximité des tâches**, ils accèdent au niveau de leurs plates-formes clientes à un sous-ensemble fonctionnel identique et manipulent les mêmes informations en entrée/sortie ;
- une certaine **proximité d'environnement** est garantie entre les utilisateurs qui partagent un même contexte d'exécution. Par exemple, deux utilisateurs ayant leurs régions d'exécution en relation avec une région d'exécution tierce partagent une même proximité spatiale à cette région et à son composant générateur [1].

Proximité des tâches et proximité d'environnement constituent une **métrique d'association** des utilisateurs jugée pertinente pour la construction de profils par analyse conjointe. Dans ce cas, les problèmes de **démarrage à froid** sont limités par l'application de profils de groupe propres à chaque contexte d'exécution. D'autre part, comme un même système fait cohabiter plusieurs contextes d'exécution, il existe à un instant donné autant de profils de groupe que de contextes d'exécution et les utilisateurs, en passant de l'un à l'autre, limitent l'**inertie du contenu** de leur profil en appliquant à chaque changement de contexte d'exécution le profil de groupe associé au contexte qu'ils rejoignent.

Les critères de mesures implicites pour constituer un tel profil hybride reposent nécessairement sur le cas d'application choisi. Dans les exemples précédents, des mesures implicites portant sur la navigation au sein d'une page Internet ont illustré les différentes méthodes de construction de profils. La suite de cette étude détaille la constitution de profils de groupe à partir de critères appropriés à la mesure d'intérêts qui portent sur des éléments localisés dans l'environnement (p. ex. bâtiments, montagne, bosquet, etc.) et qui sont reportés au niveau du client sur une carte numérique.

Ce choix d'un système de cartographie mobile comme cas d'application pour la constitution de profils est motivé par le nombre important des systèmes nomades qui traitent une information localisée et proposent à l'utilisateur de le documenter sur certains éléments de son environnement géographique (cf. les exemples du chapitre I). D'autre part, les nouvelles capacités des clients mobiles à exploiter un ensemble de services d'information géographique comme Google Maps (http ://maps.google.com/), MapQuest (http ://www.mapquest.com/), Bing Maps (http ://www.bing.com/maps/) ou des produits open-source comme OpenStreetMap (http ://www.openstreetmap.org/) facilitent l'implémentation de service reposant sur des informations spatiales. Plus particulièrement, dans le cas de systèmes mobiles, Les données de localisation de l'utilisateur associées à ces outils de cartographie numérique facilitent la présentation l'espace géographique proche de l'utilisateur et des éléments qui l'entourent.

D'autre opart, le cas d'application de cartographie mobile retenu permet des mesures implicites discrètes portant sur les éléments représentés sur la carte et disponibles aux alentours de l'utilisateur. Ces mesures permettent de déterminer des scores, élément par élément, répartis dans deux catégories :

- d'une part les **scores d'interaction** classent les éléments localisés selon les préférences des utilisateurs en interprétant leurs actions vis-à-vis de l'interface et des informations présentées sur la carte numérique. L'action de sélection d'un élément sur

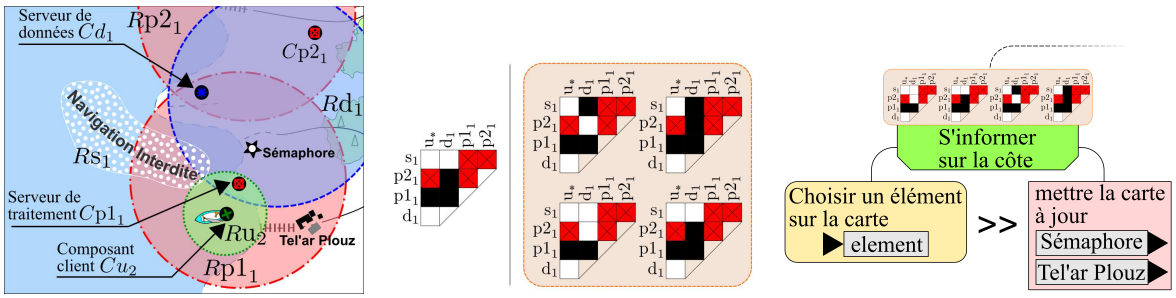
1. Les similarités entre utilisateurs d'un même contexte d'exécution dépassent le cadre de la proximité spatiale et le terme "proximité d'environnement" est choisi préférablement à "proximité spatiale". Par exemple, des utilisateurs dans un même espace physique partagent potentiellement les mêmes conditions météo, le même encombrement sonore, etc.

la carte numérique est un exemple d'indicateur d'intérêt implicite pour déterminer un score d'interaction. Les mesures de scores d'interaction appliquées à la cartographie numérique ont été notamment développées par . Cette contribution est reprise dans cette étude pour la constitution des profils de groupe.

– d'autre part, les **scores d'environnement** classent les éléments localisés selon les préférences des utilisateurs inférées à partir de leur mobilité et de la situation physique qu'ils occupent par rapport à ces éléments. En supposant que la prégnance d'un élément dans l'environnement d'un utilisateur est proportionnelle à la distance séparant l'utilisateur et l'élément, le système assume qu'un utilisateur s'approchant d'un ou stationnant auprès d'un élément dénote son intérêt envers cet élément.

La suite de ce chapitre présente une méthode de construction implicite de profils à l'exécution. Dans un premier temps, les profils sont initialisés pour chaque utilisateur, en tenant compte à la fois de ses interactions avec le client et de son évolution dans l'espace du système. Dans un deuxième temps, la construction de profils est placée dans une perspective multi-utilisateurs, où un ensemble de clients contribue à la définition de préférences communes qui peuvent être partagées à travers l'espace du système. La construction de profils personnels, puis de profils communs est illustrée par un exemple d'utilisation du système de suivi côtier. Un cas d'application des profils communs pour la personnalisation du contenu d'un service de cartographie mobile sera développé par le second cas d'étude proposé au chapitre VI.2.

Exemple d'utilisation du système de suivi côtier Le système de suivi côtier fournit un support d'illustration à la constitution de profils pour la personnalisation. Cet exemple suppose que le système se trouve, à un instant t_3 de l'exécution, dans la situation où un utilisateur accède via son client $C\mathrm{u}_2$ aux services fournis par le composant de traitement $C\mathrm{p1}_1$. Cet utilisateur bénéficie d'un outil de cartographie qui lui permet de localiser des éléments sur la côte alentour et de détailler des informations sur ces éléments (Fig. V.4). Le composant $C\mathrm{p1}_1$ accède de son côté à l'ensemble des informations disponibles sur le composant de gestion de données $C\mathrm{d}_1$ (Fig. V.3a).



(a) Situation d'exécution rencontrée à un instant t_3

(b) Contexte d'exécution à t_3 et chaîne d'équivalence à laquelle il appartient

(c) Partie de l'arbre des tâches réalisable dans les conditions d'exécution à t_3

Figure V.3 – Exemple de tâche réalisable dans le cas du système de suivi côtier à t_3 par l'utilisateur sur son composant $C\mathrm{u}_2$

Dans la situation présentée à t_3, un utilisateur et son client $C\mathrm{u}_2$ sont présents dans la région d'accès aux traitements du composant $C\mathrm{p1}_1$ et le contexte d'exécution ec_1 mesuré à

son niveau répertorie les relations entre régions suivantes (Fig. V.3b) :

$$ec_1 = LocState(C\mathrm{u}_2, t_3) = \{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\}$$

Ce contexte est répertorié dans un groupe d'équivalence qui conditionne l'exécution d'une partie de l'arbre des tâches. Dans ces circonstances d'exécution, l'utilisateur représenté par le composant $C\mathrm{u}_2$ peut notamment "S'informer sur la côte". Cette tâche consiste dans un premier temps à sélectionner un élément de la côte sur une carte numérique affichée par le client (Fig. V.4 - écran gauche). Au niveau du dialogue avec l'utilisateur, cette sélection s'effectue par un pointage sur l'un des labels référençant les éléments de la carte (Fig. V.4(c)). Cette sous-tâche interactive provoque la modification par le système des données affichées sur la carte (Fig. V.4 - écran droit). En particulier, différents détails sur l'élément sélectionné sont présentés à l'utilisateur (Fig. V.4(e)). Dans cet exemple, un sémaphore et le village de "Tel'ar Plouz" sont deux éléments de la côte représentés sur la carte numérique de l'utilisateur. Pour affiner sa sélection, l'utilisateur dispose encore d'outils de cartographie classiques, accessibles sous la forme de boutons de commande sur la gauche de l'interface (Fig. V.4(b)).

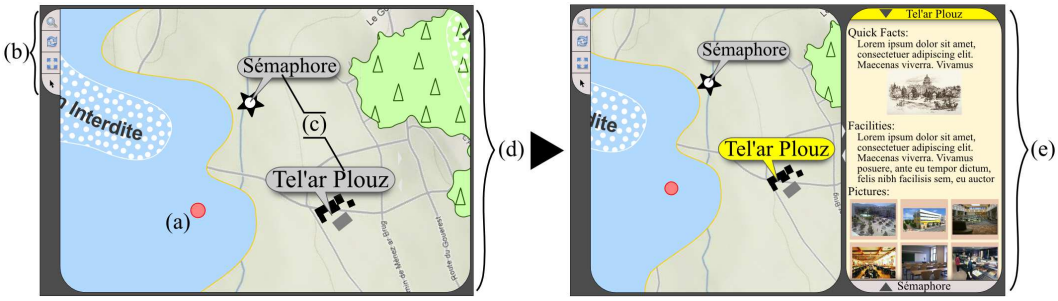


Figure V.4 – Exemple d'affichages cartographiques au niveau de l'interface du client $C\mathrm{u}_2$ pour la tâche "S'informer sur la côte" réalisable à t_3

Cet exemple permet d'illustrer, dans la suite de ce chapitre, les deux aspects de la construction de profils par mesure implicite d'intérêt. Sans changer de contexte d'exécution entre différents instants de l'exécution, les utilisateurs peuvent se situer physiquement à des distances variables des éléments "Sémaphore" et "Tel'ar Plouz" représentés sur leur carte numérique. Ces variations de localité permettent de distinguer différents scores d'environnement. D'autre part, les éléments affichés sur leur clients mobiles permettent l'initialisation de scores d'interaction par la mesure implicite des actions de l'utilisateur sur l'interface.

V.2 Propositions pour la construction de profils et la mesure de scores

L'approche proposée pour la caractérisation de la dimension "Utilisateur" du contexte d'exécution passe par la construction de profils de groupe. Ces profils regroupent les préférences d'un ensemble d'utilisateurs par rapport à un contenu informationnel présenté dans un contexte d'exécution donné. Plus spécifiquement, un profil de groupe est constitué par les scores de groupe de chaque éléments pour lesquels des préférences d'utilisateurs peuvent

être déterminées (Fig. V.5(a)). Les scores de groupe des éléments sont constitués à chaque instant de l'exécution à partir de deux mesures :

- **la partie instantanée** représente les préférences groupées d'un ensemble d'utilisateurs partageant un même contexte d'exécution, des mêmes tâches, et un même contenu informationnel (Fig. V.5(f)). Dans ce contexte d'exécution, un profil personnel est défini pour chaque individu (Fig. V.5(d)). En mêlant des mesures implicites d'interaction et de déplacement, ce profil ordonne les préférences vis-à-vis du contenu proposé. L'agrégation des profils personnels à un moment de l'exécution détermine un ensemble de scores moyens instantanés. Scores moyens et profils personnels sont construits à partir des mesures de la dimension utilisateur effectuées lors du dernier intervalle d'exécution (c-à-d. entre t_x et t_{x-1}) ;
- **la partie historique** rapporte les valeurs des scores de groupe construits aux instants d'exécution précédents (c-à-d. avant t_x : t_{x-1}, t_{x-2}, etc., jusqu'à l'instant d'initialisation t_0) (Fig. V.5(e)). Ces scores de groupe précédents mesurent les préférences passées des utilisateurs ayant déjà interagi dans le contexte d'exécution considéré.

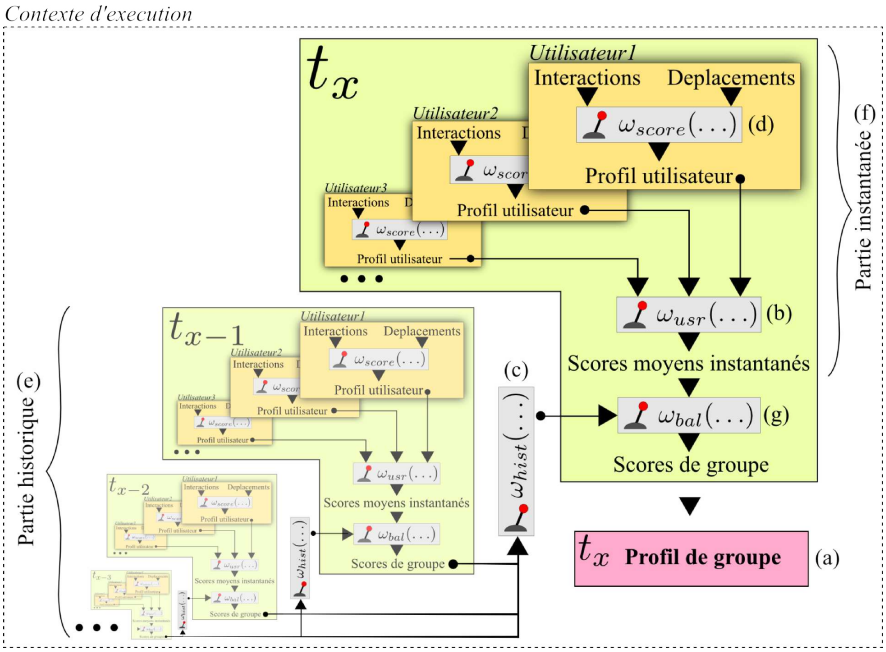


Figure V.5 – Étapes de la construction d'un profil de groupe à un instant de l'exécution

L'ensemble de ces contributions à la construction des scores de groupe est pondéré par des fonctions définies au moment de la conception. Ces fonctions permettent d'adapter le comportement et le contenu des profils de groupe au cas d'application développé et d'optimiser les recommandations basées sur ces profils. Les concepteurs spécifient ces fonctions lors de l'implémentation du système pour coder la manière dont les profils se génèreront lors de l'exécution :

- la fonction $\omega_{score}(\ldots)$ détermine pour chaque score formant un profil d'utilisateur quelle est la part des préférences revenant aux mesures d'interaction ou revenant aux mesures d'environnement (Fig. V.5(d)). Selon la définition donnée de cette fonction lors de la conception, la mesure de préférences peut s'appliquer que l'utilisateur soit mobile ou non ;

- la fonction $\omega_{usr}(\ldots)$ paramètre l'apport de chaque utilisateur dans la construction des scores moyens (Fig. V.5(b)). Cette fonction permet de regrouper les utilisateurs en différentes classes définies à la conception, selon des critères de mobilité, d'expérience, d'usage du système, etc ;
- la fonction $\omega_{hist}(\ldots)$ définit quelle est l'importance de chacun des scores de groupe passés lors du calcul d'un score de groupe à l'instant courant (Fig. V.5(c)). Cette fonction permet de renforcer lors de l'exécution certains profils passés, par exemple lorsque les recommandations dérivées de ce profil avaient été favorablement accueillies par les utilisateurs ;
- la fonction $\omega_{bal}(\ldots)$ partage la part d'historique et la part d'instantané dans le calcul des scores de groupe (Fig. V.5(g)). Cette fonction équilibre une balance entre des profils d'inertie importante, reposant sur des scores passés, et des profils à forte variabilité, reposant pour l'essentiel sur des scores instantanés.

Le paramétrage du calcul des scores par des fonctions définies au moment de la conception offre à la démarche de construction de profil la plasticité suffisante pour produire des recommandations par *analyse de contenu*, par *analyse conjointe* ou par un mélange de ces deux approches (Tab. V.1). Par exemple, dans le cas d'une fonction de pondération $\omega_{usr}(\ldots)$ répercutant le profil d'un utilisateur unique en entrée des scores moyens, la chaîne de construction de profil de groupe devient équivalente à une construction par analyse de contenu.

V.2.1 Création d'un profil utilisateur

L'objectif de la création d'un profil est de déterminer le niveau d'intérêt d'un utilisateur par rapport à l'ensemble des informations qui sont représentées sur sa carte numérique sous forme d'éléments et qui font partie de son environnement proche dans le monde réel. Les éléments dont le score peut etre évalué forment l'ensemble des informations qui, dans un contexte d'exécution donné, sont offertes à l'utilisateur et pour lesquelles il peut signifier implicitement ou explicitement son intérêt [2]. Dans le cas d'application choisi, les éléments mesurables consistent en une représentation numérique d'informations localisées dans l'environnement de l'utilisateur. Dans la suite, le rôle "u" est systématiquement utilisé pour désigner les utilisateurs du système et leurs composants clients. En considérant un contexte d'exécution local $ec_k = LocState(C\mathrm{u}_y, t_x)$, mesuré à partir d'une plate-forme cliente $C\mathrm{u}_y$ à un instant t_x, les éléments mesurables disponibles dans ce contexte d'exécution à l'instant sont retournés par $Elems(ec_k, t_x) = \{elm_1, elm_2, \ldots, elm_j, \ldots, elm_n\}$. Cet ensemble est une sous-partie de $ELEMS$, qui désigne l'union des éléments mesurables dans chaque contexte et quel que soit l'instant d'exécution.

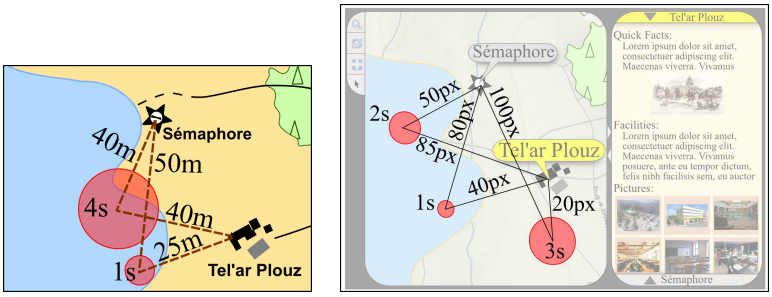
Dans le cas du système de suivi côtier (Fig. V.4), les éléments disponibles dans le contexte $ec_1 = \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\big\}$ à l'instant t_3 dépendent de la part de l'arbre des tâches activées et des informations d'entrée/sortie disponibles. Dans cette configuration particulière, la sous-tâche "Mettre la carte à jour" est active et mobilise en sortie des informations localisant le sémaphore et le village côtier. Ces informations sont à la fois représentées sur la carte numérique au niveau du composant client et font partie de l'environnement physique de l'utilisateur. Dans ce cas, ces deux éléments peuvent obtenir un

2. Les éléments pour lesquels un score d'intérêt peut être calculé sont appelés "éléments mesurables" par la suite.

score de préférence et constituent les dimensions du profil de l'utilisateur :

$$Elems(ec_1, t_3) = \{ \text{ “Sémaphore”, “Tel'ar Plouz”} \}$$

Le modèle de construction de profil propose une remise à jour régulière des scores et des profils. Pour un système construit selon le cadre de conception étendu, s'exécutant entre t_0 et t_{end}, les mesures des changements de contexte d'exécution sont effectuées avec un pas α, identique au pas de détection des changement de contextes (Chap. III.2.1). Une mise à jour des profils effectuée avec cette même régularité est proposée : les actions implicites des utilisateurs sont monitorées dans l'intervalle $[t_{x-1}, t_x[$ et les scores et profils sont mis à jour à partir de ces mesures à t_x.



(a) Niveau de l'environnement

(b) Niveau de l'interaction

Figure V.6 – Actions implicites de l'utilisateur mesurées entre t_2 et t_3

Les stratégies identifiées pour compléter les scores de chaque élément sont de deux natures dont l'évaluation quantitative est proposée dans la suite. Localement à un contexte d'exécution ec_x, tel que l'élément soit mesurable dans ec_x :

- le **score d'environnement** est construit à partir des déplacements et des pauses des utilisateurs de ce contexte (Fig. V.6a) ;
- le **score d'interaction** est construit à partir de différents indicateurs implicites d'intérêt lors de la manipulation par l'utilisateur des fonctionnalités disponibles dans ec_x sur son client.

Ces contributions au score d'un élément doivent être balancées. Il est possible que l'intérêt de certains utilisateurs très peu mobiles soit plutôt révélé par leurs interactions. Dans ce cas, le score d'environnement devrait être minimisé. A l'inverse, les utilisateurs très mobiles sont enclins à moins de manipulations sur leur client et fournissent un score d'interaction de moindre importance.

Score d'interaction

L'enregistrement de l'activité du pointeur donne une indication sur la réflexion des utilisateurs à propos des éléments qui lui sont présentés. En effet, de nombreux travaux ont montré que les positions du curseur et ses temps de pause révèlent les intérêts des utilisateurs (; ;). Dans le domaine de la cartographie numérique, montrent que ces indicateurs gardent toute leur pertinence et que l'utilisateur se sert de son curseur comme point de focalisation

de son attention, un peu comme un enfant apprendrait à lire en soulignant les mots avec son doigt. Dans ses travaux, les interactions cartographiques sont systématiquement associées à une vue particulière sur les données. Par exemple, un utilisateur d'un outil de cartographie peut agrandir une région particulière, orienter la carte, ou encore la positionner en latitude et longitude. L'ensemble de ces manipulations définit une vue qui est un indicateur d'intérêt à part entière.

L'approche proposée du score d'interaction est construite sur la base des indicateurs d'intérêt du curseur, adaptée à la manipulation cartographique par la prise en compte de la vue affichée par le client comme élément de pondération. Par exemple, une pause de curseur dans une vue présentant un niveau de zoom important est considérée comme une indication plus pertinente de l'intérêt d'un utilisateur que le même cas de figure, mais avec une vue plus générale. Dans la suite, une fonction de pondération $\omega_{view}(C\mathrm{u}_y, t_x)$ rapporte l'importance relative d'une vue affichée à t_x sur le client $C\mathrm{u}_y$ d'un utilisateur du système lors du calcul du score d'interaction. Si entre t_a et t_b, l'utilisateur agrandit la portée de la zone géographique affichée sur son composant $C\mathrm{u}_y$ en "dé-zoomant", alors le poids de la vue à t_b est inférieur au poids de la vue à t_a : $\omega_{view}(C\mathrm{u}_y, t_b) < \omega_{view}(C\mathrm{u}_y, t_a)$.

Le score d'interaction d'un élément combine le poids de la vue aux indicateurs révélés par les actions du pointeur. En particulier, les endroits de l'écran où le pointeur hésite entre deux mouvements sont enregistrés lors de chaque intervalle de mesure. Pondérées par la durée d'hésitation, les distances entre un élément affiché et les différents endroits d'hésitation du curseur forment le score de cet élément. Étant donné un contexte utilisateur mesurant le contexte d'exécution ec_k, la fonction $MouseHes(C\mathrm{u}_y, ec_k, t_x)$ retourne l'ensemble des positions de l'écran $\{mh_1, mh_2, ..., mh_n\}$ où des hésitations du pointeur du client $C\mathrm{u}_y$ ont été enregistrées à l'analyse des mesures implicites du curseur dans l'intervalle $[t_{x-1}, t_x[$. La durée d'hésitation du curseur à une position $mh_j \in MouseHes(C\mathrm{u}_y, ec_k, t_x)$ de l'écran est retournée par la fonction $HesTmp(mh_j)$. Enfin, le score d'interaction d'un élément $elm_i \in Elems(ec_k, t_x)$ pour un utilisateur assimilé à son composant $C\mathrm{u}_y$ est retourné par la fonction $UIScore(elm_i, C\mathrm{u}_y, t_x)$:

$$\forall\ t_x \in T, elm_i \in ELEMS, C\mathrm{u}_y \in Cmps(\mathrm{u}, t_x)\ :\ UIScore(elm_i, C\mathrm{u}_y, t_x) =$$
$$\begin{cases} \omega_{view}(C\mathrm{u}_y, t_x) \times \displaystyle\sum_{\substack{mh_l \in MouseHes(C\mathrm{u}_y,\\ LocState(C\mathrm{u}_y, t_x), t_x)}} \frac{HesTmp(mh_l)}{d(elm_i, mh_l)} & \text{ssi } \begin{array}{c} elm_i \in Elems(\\ LocState(C\mathrm{u}_y, t_x), t_x) \end{array} \\ 0 & \text{autrement} \end{cases}$$

avec $d(elm_i, mh_l) \in \mathbb{R}^*_+$, la distance strictement positive à l'écran entre elm_i et mh_l

La définition du score d'interaction considère la proximité des éléments à un historique des hésitations de la souris, dans le contexte de la vue affichée à l'instant du calcul. Plus une pause de souris est longue, plus saillante devient sa position à l'écran. En conséquence, les meilleurs scores sont attribués aux éléments proches des endroits d'hésitation les plus longs. Ces scores sont pondérés par la valeur de la vue courante et le score d'un élément augmente proportionnellement avec le poids de la vue.

Dans l'exemple du suivi côtier, l'utilisateur a déjà passé quelques temps à interagir sur sa plate-forme $C\mathrm{u}_2$ avec la carte numérique et les informations disponibles. Ses actions pour affiner la présentation de la carte numérique produisent différentes vues. Lors de la

conception, les poids de ces vues ont été échelonnées entre 0 pour la vue la plus générale et 10 pour la vue la plus fine. Les scores des éléments "Sémaphore" et "Tel'ar Plouz' peuvent être calculés à t_3 à partir des interactions de l'utilisateur et du poids de la vue courante. En considérant l'ensemble des pauses du curseur enregistrées entre t_2 et t_3 (Fig. V.6b) et un poids de la vue affichée à t_3 égal à $\omega_{View}(C\mathrm{u}_2, t_3) = 4$, dans ce cas :

$$UIScore(\text{“Tel'ar Plouz”}, C\mathrm{u}_2, t_3) = 4 \times \left(\frac{3}{20} + \frac{1}{40} + \frac{2}{85}\right) = 0.8$$

Le score de l'élément "Sémaphore", retourné par la fonction $UIScore(\text{“Sémaphore”}, C\mathrm{u}_2, t_3)$ vaut 0.32. A cet instant, et d'après la mesure implicite des interactions de l'utilisateur, l'élément "Sémaphore" a moins d'importance que la ville côtière. Le système infère une distinction de préférence entre ces éléments qui, une fois associée à des scores dépendant de la localisation de l'utilisateur, peut constituer un profil instantané de cet utilisateur.

Score d'environnement

Dans le cas d'utilisateurs mobiles dans leur environnement, le score d'interaction peut être enrichi par l'enregistrement des déplacements et des pauses de l'utilisateur. La relation implicite entre la position du curseur et l'intérêt des utilisateurs envers le contenu désigné est adaptée dans l'environnement physique. Dans ce cas, la proximité réelle entre un individu et un élément de l'environnement peut être significative d'un intérêt de l'utilisateur envers l'élément proche. Par exemple, les utilisateurs d'un système mobile peuvent s'arrêter ou effectuer des pauses dans leurs déplacements à proximité d'un élément de l'environnement avant d'interroger le système sur cet élément. Ces endroits de pauses, où l'utilisateur interagit avec le système, sont considérés, dans le modèle de construction de profil proposé, comme des indicateurs d'intérêt pour les éléments alentours. Ces indicateurs sont quantifiés de manière similaire aux scores d'interaction, en tenant à la fois compte des durées des pauses des utilisateurs et des distances physiques entre eux et les éléments de l'environnement, mesurées au cours du dernier intervalle d'exécution.

Le calcul du score d'environnement d'un élément repose sur la fonction $BreakPos(C\mathrm{u}_y, ec_k, t_x)$ qui retourne l'ensemble $\{bp_1, bp_2, ..., bp_n\}$ des localisations spatiales où un utilisateur mesurant un contexte d'exécution ec_k et représenté au sein du système par son composant client $C\mathrm{u}_y$, a effectué des pauses entre t_{x-1} et t_x. La durée d'une pause $bp_l \in BreakPos(C\mathrm{u}_y, ec_k, t_x)$ de cet ensemble est retourné par la fonction $BreakTmp(bp_l)$. Le score d'environnement d'un élément $elm_i \in Elems(ec_k, t_x)$ pour un utilisateur assimilé à

son composant $C\mathrm{u}_y$ est retourné par la fonction $LocScore(elm_i, C\mathrm{u}_y, t_x)$:

$$\forall\ t_x \in T, elm_i \in ELEMS, C\mathrm{u}_y \in Cmps(\mathrm{u}, t_x)\ :\ LocScore(elm_i, C\mathrm{u}_y, t_x) = \begin{cases} \sum\limits_{\substack{bp_l \in BreakPos(C\mathrm{u}_y,\\ LocState(C\mathrm{u}_y,t_x),t_x)}} \dfrac{BreakTmp(bp_l)}{d(elm_i, bp_l)} & \text{ssi } \begin{array}{c} elm_i \in Elems(\\ LocState(C\mathrm{u}_y, t_x), t_x) \end{array} \\ 0 & \text{autrement} \end{cases}$$

avec $d(elm_i, bp_j) \in \mathbb{R}_+^*$, la distance métrique strictement positive entre elm_i et bp_l

Lorsque l'utilisateur mesure à t_x un contexte local égal à $LocState(C\mathrm{u}_y, t_x)$, et sachant que les lieux et les durées de pause de l'utilisateur ont été mesurés implicitement dans l'intervalle $[t_{x-1}, t_x[$, la définition proposée du score d'environnement d'un élément correspond à l'addition des ratios entre la durée d'une pause et la distance du lieu de cette pause à l'élément considéré. Le score d'environnement d'un élément est proportionnel à la durée des pauses de l'utilisateur et inversement proportionnel à la distance entre l'utilisateur et l'élément. Par exemple, si les éléments mesurables désignent des bâtiments de l'environnement, plus un utilisateur passe de temps à observer ou à se situer à proximité d'un bâtiment, plus le score d'environnement de ce bâtiment augmente.

Dans l'exemple du système de suivi côtier, l'utilisateur s'est déplacé et a effectué deux pauses de quatre et une seconde dans l'intervalle $]t_2, t_3]$, pendant lesquelles il a utilisé le service de cartographie (Fig. V.6a). Ces deux pauses ont été enregistrées dans le contexte d'exécution de l'utilisateur $LocState(C\mathrm{u}_2, t_3) = \{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\}$ et les scores des éléments de l'environnement sont calculés à partir des durées de ces pauses et de leurs localisations. Par exemple, le score d'environnement $LocScore(\text{“Sémaphore”}, C\mathrm{u}_2, t_3)$ est attribué à l'élément “Sémaphore” à l'instant t_3 :

$$LocScore(\text{“Sémaphore”}, C\mathrm{u}_2, t_3) = \frac{1}{50} + \frac{4}{40} = 0.12$$

Le score d'environnement de l'élément “Tel'ar Plouz” est retourné par la fonction $LocScore(\text{“Tel'ar Plouz”}, C\mathrm{u}_2, t_3) = 0.14$. Les scores d'environnement corroborent les scores d'interaction et confirment la préférence pour le village côtier par rapport au sémaphore.

Combinaison des scores et profil d'un utilisateur

Les indicateurs implicites d'intérêt quantifient pour chaque élément présenté à un utilisateur des scores d'interaction et d'environnement. Ces scores forment les primitives du profil d'un individu. Dans l'approche proposée, une combinaison d'indicateurs implicites est effectuée pour construire un profil global contenant une liste d'éléments et leurs scores.

Réduire l'intérêt des utilisateurs à un score sur une dimension au lieu de deux n'est pas anodin et ne saurait être réduit à la simple moyenne arithmétique de deux scores. Notamment, la moyenne de deux scores d'interaction et d'environnement en opposition donne un score global neutre alors que la sémantique de l'opposition est souvent plus riche. Par exemple, lorsqu'un un utilisateur se tient à proximité d'un bâtiment (score d'environnement élevé),

mais que dans le même temps, il n'interagit pas du tout avec les données cartographiques à sa disposition (score d'interaction faible), cela ne signifie pas forcément un intérêt neutre pour cet élément. Comme illustré par cet exemple, il existe un ensemble de situations pour lesquelles il est souhaitable de donner plus d'importance au critère d'interaction par rapport au critère d'environnement (et vice-versa) dans le calcul d'un score global. Cette importance relative d'un score par rapport à l'autre est introduite dans le calcul du score global par une fonction de pondération adaptée au système développé et définie par les concepteurs. Cette fonction $\omega_{score}(elm_i, C\mathrm{u}_y, t_x)$ est définie dans l'intervalle réel $[0, 1]$, relativement à un élément mesurable elm_i, pour une plate-forme utilisateur $C\mathrm{u}_y$ donnée, et à un instant de l'exécution du système t_x. En considérant la définition pseudo-formalisée du score global d'un élément (Prop. V.1), les concepteurs peuvent limiter l'impact d'une partie du score jugée non pertinente :

$$Score(\ldots) = \big(\omega_{score}(\ldots) \times UIScore(\ldots) + (1 - \omega_{score}(\ldots)) \times LocScore(\ldots)\big) \qquad \text{(V.1)}$$

Plus spécifiquement, lorsque $\omega_{score}(\ldots) = 0$, le score d'interaction est annulé et le score global est égal au score d'environnement. À l'inverse, si $\omega_{score}(\ldots) = 1$, seul le score d'interaction est conservé comme score global. Cette approche de pondération du score global permet aux concepteurs d'envisager par un même outil la définition de profils d'utilisateurs mobiles ou non. Par exemple, le score de mobilité n'est pas essentiel pour déterminer l'intérêt d'un utilisateur accédant au système depuis son bureau. Dans ce cas, le poids attribué à cet utilisateur et à sa plate-forme devrait tendre vers 1. De la même manière, il est possible de rendre la pondération dynamique à l'exécution et de faire augmenter la part du score d'environnement avec la quantité de déplacements des utilisateurs. La fonction $Score(elm_i, C\mathrm{u}_y, t_x)$ effectue une moyenne pondérée des scores d'interaction et d'environnement et retourne le score global d'un élément elm_i à l'instant t_x :

$$\forall\, t_x \in T, elm_i \in ELEMS, C\mathrm{u}_y \in Cmps(\mathrm{u}, t_x) \;:$$

$$Score(elm_i, C\mathrm{u}_y, t_x) = \begin{cases} \left(\begin{array}{c} \omega_{score}(elm_i, C\mathrm{u}_y, t_x) \times \dfrac{UIScore(elm_i, C\mathrm{u}_y, t_x)}{Sum_{UIScore}(C\mathrm{u}_y, t_x)} + \\ \big(1 - \omega_{score}(elm_i, C\mathrm{u}_y, t_x)\big) \times \dfrac{LocScore(elm_i, C\mathrm{u}_y, t_x)}{Sum_{LocScore}(C\mathrm{u}_y, t_x)} \end{array}\right) & \text{ssi } elm_i \in Elems\big(LocState(C\mathrm{u}_y, t_x), t_x\big) \\ 0 & \text{autrement} \end{cases}$$

$$\text{avec } Sum_{UIScore}(C\mathrm{u}_y, t_x) = \begin{cases} \sum\limits_{elm_k \in Elems(LocState(C\mathrm{u}_y, t_x), t_x)} UIScore(elm_k, C\mathrm{u}_y, t_x) & \text{ssi } \left(\begin{array}{c} \exists\, elm_l \in Elems\big(LocState(C\mathrm{u}_y, t_x), t_x\big) \mid \\ UIScore(elm_l, C\mathrm{u}_y, t_x) \neq 0 \end{array}\right) \\ 1 & \text{autrement} \end{cases}$$

$$\text{et } Sum_{LocScore}(C\mathrm{u}_y, t_x) = \begin{cases} \sum\limits_{elm_k \in Elems(LocState(C\mathrm{u}_y, t_x), t_x)} LocScore(elm_k, C\mathrm{u}_y, t_x) & \text{ssi } \left(\begin{array}{c} \exists\, elm_m \in Elems\big(LocState(C\mathrm{u}_y, t_x), t_x\big) \mid \\ LocScore(elm_m, C\mathrm{u}_y, t_x) \neq 0 \end{array}\right) \\ 1 & \text{autrement} \end{cases}$$

Ce formalisme de calcul des scores reprend l'idée de balance entre interaction et environnement et garantit une réponse quels que soient les éléments passés comme variables de la fonction. Il est ainsi possible demander le score de n'importe quel élément, qu'il soit mesurable ou non et qu'il appartienne ou non à l'ensemble $Elems\big(LocState(C\mathrm{u}_y, t_x), t_x\big)$ des

éléments affichés sur le client à t_x. Une valeur de score nulle est attribuée aux éléments qui ne sont pas dans $Elems(\ldots)$. Les parts des scores d'interaction et d'environnement sont exprimées par la proportion du score d'un élément particulier par rapport à la somme des scores de chaque élément $\left(\text{c-à-d. } \frac{UIScore(\ldots)}{Sum_{UIScore}(\ldots)} \text{ et } \frac{LocScore(\ldots)}{Sum_{LocScore}(\ldots)}\right)$. Les sous-fonctions $Sum_{UIScore}(C\mathrm{u}_y, t_x)$ et $Sum_{LocScore}(C\mathrm{u}_y, t_x)$ retournent respectivement la somme des scores d'interaction et la somme des scores d'environnement obtenus par les éléments mesurables dans le contexte local de l'utilisateur à t_x. Si aucun élément n'est mesurable dans ce contexte, ces fonctions retournent arbitrairement la valeur 1 pour ne pas annuler les dénominateurs lors des calculs des proportions des scores.

La collection des scores de chaque élément constitue le profil $UsrProf(C\mathrm{u}_y, t_x)$ d'un utilisateur référencé par son client $C\mathrm{u}_y$ pour un instant et un contexte d'exécution donnés :

$$\forall\, t_x \in T, C\mathrm{u}_y \in Cmps(\mathrm{u}, t_x) \;:\; UsrProf(C\mathrm{u}_y, t_x) = \bigcup_{\substack{elm_i \in Elems(\\ LocState(C\mathrm{u}_y, t_x), t_x)}} Score(elm_i, C\mathrm{u}_y, t_x)$$

Ces profils sont appliqués à la personnalisation de l'interface et des données pour un utilisateur particulier. Selon les choix des concepteurs, les éléments de contenu présentant un plus faible intérêt pour l'utilisateur peuvent être par exemple cachés ou placés en arrière plan cartographique. A l'inverse, les éléments pour lesquels l'utilisateur montre un intérêt important peuvent être accentués sur son interface cliente.

Dans le cas du système de suivi côtier, les concepteurs font, par exemple, le choix d'accorder une égale importance à la part d'interaction et à la part d'environnement lors du calcul d'un score global d'un élément. Dans ce cas, la fonction de pondération est définie constante :

$$\forall \begin{array}{c} t_x \in T,\ C\mathrm{u}_y \in Cmps(\mathrm{u}, t_x), \\ elm_i \in Elems\big(LocState(C\mathrm{u}_y, t_x), t_x\big) \end{array} \;:\; \omega_{score}(elm_i, C\mathrm{u}_y, t_x) = 0.5$$

Le score de l'élément représentant le village côtier "Tel'ar Plouz" est retourné à t_3 et pour l'utilisateur du client $C\mathrm{u}_2$ par :

$$Score(\text{“Tel'ar Plouz”}, C\mathrm{u}_2, t_3) = 0.5 \times \frac{0.8}{0.8 + 0.32} + (1 - 0.5) \times \frac{0.14}{0.14 + 0.12} = 0.62$$

De la même manière, le score de l'élément "Sémaphore" est retourné par $Score(\text{“Sémaphore”}, C\mathrm{u}_2, t_3) = 0.37$. Le profil de cet utilisateur à t_3 est constitué des scores globaux des deux éléments avec lesquels il interagit :

$$\begin{aligned} UsrProf(C\mathrm{u}_2, t_3) &= \big\{Score(\text{“Tel'ar Plouz”}, C\mathrm{u}_2, t_3), Score(\text{“Sémaphore”}, C\mathrm{u}_2, t_3)\big\} \\ &= \{0.62, 0.37\} \end{aligned}$$

Dans le cas d'une application multi-utilisateurs, les profils construits par les utilisateurs d'un même contexte peuvent être comparés et des préférences communes peuvent éventuellement être extraites parmi les intérêts des utilisateurs. Cette approche de construction groupée d'un profil commun permet de proposer aux nouveaux utilisateurs dans un contexte d'exécution donné de recevoir un ensemble de recommandations issues d'un consen-

sus entre les utilisateurs présents et de démarrer son utilisation du système avec un profil. Les avantages attendus des profils de groupe, ainsi qu'une méthode de construction appropriée aux systèmes à contextes d'exécution multiples, sont détaillés dans la suite de ce chapitre.

V.2.2 Perspective multi-utilisateurs et regroupement de profils

La personnalisation d'un système cartographique a été envisagée sous l'angle de la construction d'un profil propre à un utilisateur. Dans un contexte donné, celui-ci accède à un sous-ensemble de fonctionnalités et d'informations selon la part de l'arbre des tâches situé pouvant être effectuée dans ces conditions d'exécution. Les informations qui sont représentées sur son client et qui sont physiquement situées dans l'environnement de cet utilisateur constituent un ensemble d'éléments qu'il est possible d'ordonner selon l'intérêt implicite de l'utilisateur à un moment donné. Cet intérêt implicite est déduit des interactions et des déplacements de l'utilisateur et est quantifié sous la forme de scores attribués aux éléments à un moment donné de l'exécution. La collection des scores de chaque élément détermine le profil d'un d'utilisateur. Ce profil peut être pris en compte pour ajuster le contenu présenté.

Cette approche de construction de profils par *analyse de contenu* (), tout en combinant deux sources complémentaires de mesures implicites d'intérêt, reste sensible aux problèmes d'initialisation à froid () et de variabilité des scores (). Dans le cas des systèmes mobiles à plusieurs niveaux d'exécution et qui personnalisent le contenu présenté par l'application de profils personnels, ces deux problèmes se révèlent à l'exécution :

- un nouvel utilisateur du système qui arrive dans un contexte d'exécution donné ne peut pas bénéficier de prime abord de recommandation de contenu. Son profil est toujours vide à l'initialisation ;
- avec un système construisant des profils pour chaque contexte d'exécution rencontré, ce problème de profil vide se retrouve dès lors que l'utilisateur mobile change de conditions d'exécution et de contenu informationnel présenté ;
- dans la méthode de construction de profils présentée, les éventuelles mesures d'intérêt révélées avant l'intervalle d'exécution courant n'ont pas d'impact dans le calcul des scores des éléments. A chaque instant $t_x \in T$ de l'exécution, le profil d'un utilisateur est entièrement reconstruit.

L'approche complémentaire de construction de profils par *analyse conjointe* a été présentée comme permettant de réduire l'impact des problèmes d'inertie et de démarrage à froid rencontrés avec les profils individuels. Selon cette approche, les profils peuvent être partagés et construits communément à un ensemble d'utilisateurs, dès lors qu'il existe une métrique d'association pertinente entre ces utilisateurs. Dans le cas des systèmes distribués décrits jusqu'à présent, les contextes d'exécution jouent le rôle d'une métrique d'association. Plus spécifiquement, des utilisateurs d'un système mesurant localement un même contexte d'exécution sont associés et construisent en groupe un profil local à ce contexte. Comme dans le cas des profils individuels, les profils de groupe contiennent des scores associés aux différents éléments de l'environnement. Ceux-ci ne constituent plus uniquement des intérêts individuels mais composent un consensus entre les utilisateurs du groupe. Dans le cas de systèmes mobiles distinguant différents contextes d'exécution, cette approche tend à corriger les problèmes de la personnalisation par profils individuels reportés au chapitre II.5.2 :

- pour limiter l'effet de "démarrage à froid", un profil de préférences résultant d'un consensus entre des membres mesurant un même contexte d'exécution peut être proposé à un nouvel arrivant comme profil personnel initial ;
- les variations rapides de profils peuvent être modérées en tenant compte des profils courants aux n derniers instants d'exécution ;
- les intérêts déduits des préférences communes peuvent être rapportés au niveau de chaque utilisateur sous la forme de recommandations dont la richesse dépasse la simple mise en valeur des intérêts individuels. Par exemple, un élément présentant beaucoup d'intérêt pour une majorité d'utilisateurs peut être suggéré comme digne d'intérêt à un utilisateur qui ne se focalise pas sur cet élément.

Regroupement d'utilisateurs

Un système mobile et distribué autorise en général un ensemble d'utilisateurs à accéder au service de manière simultanée. Cette possibilité a été introduite dans le modèle conceptuel sous la forme de composants et de régions particulières, dits "génériques" et qui traduisent les comportements d'une classe de composants indénombrables à la conception. Si plusieurs utilisateurs accèdent à un même système à un moment donné, et que ce système établit un ensemble de contextes d'exécution différentiables au niveau des utilisateurs, il est alors possible que des utilisateurs partagent un même contexte d'exécution. Par exemple en considérant deux clients $C\mathrm{u}_a$ et $C\mathrm{u}_b$, désignés lors de la conception par le composant générique $C\mathrm{u}_*$, dès lors qu'à un instant t_x de l'exécution, les contextes d'exécution $LocState(C\mathrm{u}_a, t_x)$ et $LocState(C\mathrm{u}_b, t_x)$ mesurés localement sur chacun des composants sont égaux, alors les utilisateurs désignés par ces composants accèdent aux mêmes tâches et informations, partagent des mêmes conditions d'environnement et peuvent être groupés pour définir des intérêts communs et construire un profil de groupe. Les utilisateurs du système de rôle "u", représentés par leurs composants clients $C\mathrm{u}_1$, $C\mathrm{u}_2$, etc., et qui figurent dans un contexte d'exécution ec_k à un moment donné t_x de l'exécution font partie de l'ensemble $Users(\mathrm{u}, ec_k, t_x)$:

$$\begin{aligned} \forall\ ec_k \in STATES, t_x \in T\ :\ & \\ Users(\mathrm{u}, ec_k, t_x) &= \{cmp_k \in Cmps(\mathrm{u}, t_x) \mid LocState(cmp_k, t_x) = ec_k\} \\ &= \{C\mathrm{u}_1, C\mathrm{u}_2, \ldots, C\mathrm{u}_y, \ldots, C\mathrm{u}_n\} \end{aligned}$$

Les utilisateurs dont les plates-formes figurent dans cet ensemble contribuent à l'instant t_x à la définition du profil de groupe relatif au contexte d'exécution ec_k.

Par exemple, dans le cas du système de suivi côtier, à un instant t_5 de l'exécution, deux clients $C\mathrm{u}_1$ et $C\mathrm{u}_2$ se rejoignent et partagent un même contexte d'exécution (Fig. V.7). Dans ce cas :

$$LocState(C\mathrm{u}_1, t_5) = LocState(C\mathrm{u}_2, t_5) = \big\{(R\mathrm{u}_*, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\ \mathrm{d}_1, R\mathrm{p}2_1)\big\}$$

et

$$Users\big(\mathrm{u}, \big\{(R\mathrm{u}_*, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, \mathrm{p}2_1)\big\}, t_5\big) = \{C\mathrm{u}_1, C\mathrm{u}_2\}$$

Le contexte d'exécution $ec_1 = \big\{(R\mathrm{u}_2, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}1_1), (R\mathrm{d}_1, R\mathrm{p}2_1)\big\}$ valide les conditions d'exécution suffisantes pour effectuer la tâche "S'informer sur la côte" (Fig. V.3c) et ces deux

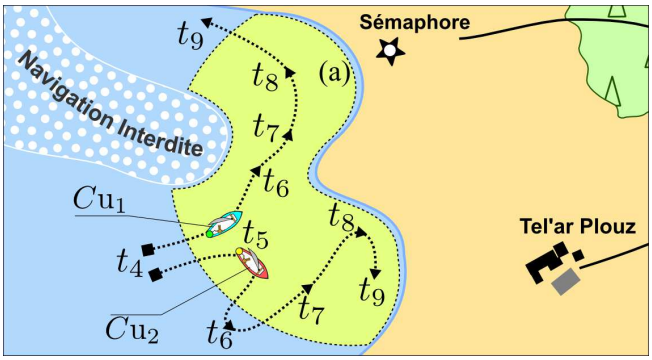


Figure V.7 – Parcours de deux utilisateurs du système entre t_4 et t_9

utilisateurs accèdent au même moment et dans un espace relativement proche à un outil cartographique qui leur permet d'obtenir des informations sur les éléments "Tel'ar Plouz" et "Sémaphore"(Fig. V.4). Leurs profils personnels attribuent des scores d'intérêt à ces éléments et deviennent arithmétiquement comparables.

La suite de cette section montre comment passer d'un ensemble de profils personnels à un moment donné à un score moyen instantané puis à un profil de groupe tenant compte de l'ensemble des scores instantanés aux différents moments de l'exécution. L'exemple du système de suivi côtier illustre cette construction de profil commun et considère le parcours éventuel de deux utilisateurs entre t_4 et t_9 (Fig. V.7). Ces utilisateurs, ci-après désignés par la dénomination de leur plate-forme cliente, construisent des scores individuels à chaque étape de l'exécution pour les éléments disponibles dans leur contexte d'exécution courant. Le tableau V.2 présente notamment les scores obtenus par les éléments "Tel'ar Plouz" (TaP) et "Sémaphore" (Sem) à chaque instant d'exécution et pour chaque utilisateur.

Tableau V.2 – Scores individuels des éléments Sem et TaP entre t_4 et t_9

x	4	5	6	7	8	9
$Score(TaP, Cu_2, t_x)$	0*(a)*	0*(b)*	0*(c)*	0.9	0.66	0.83
$Score(TaP, Cu_1, t_x)$	0*(a)*	0*(b)*	0.24	0.45	0	0*(d)*
$Score(Sem, Cu_2, t_x)$	0*(a)*	1	0*(c)*	0.1	0.33	0.17
$Score(Sem, Cu_1, t_x)$	0*(a)*	1	0.74	0.55	1	0*(d)*

Les scores individuels de ces éléments peuvent être construits par un utilisateur qui accède et interagit avec ces informations cartographiques dans le contexte d'exécution ec_1. En faisant l'hypothèse que le composant de gestion de données Cd_1, sa région d'exécution associée Rd_1, et la région d'intérêt Rs_1 désignant l'aire de navigation interdite restent immobiles entre t_4 et t_9, la zone désignée en figure V.7(a) représente l'espace dans lequel un utilisateur du système mesure localement un contexte d'exécution égal à ec_1 et où il accède au service cartographique et aux éléments "Sem" et "TaP" [3]. À l'instant t_4, les deux clients sont en dehors de la région de service. A cet instant, leur contexte local est vide

3. Cet espace correspond à l'union des localisations des clients de rôle générique $Cu*$ pour lesquelles le contexte local mesuré vaut $ec_1 = \{(Ru_2, Rp1_1), (Rd_1, Rp1_1), (Rd_1, Rp2_1)\}$. En dehors de cet espace, soit la relation entre la région de traitement $Rp1_1$ et la région d'un utilisateur Ru_* n'est plus garantie, soit l'utilisateur sort de son espace accessible, soit la relation entre Ru_* et la région d'intérêt Rs_1 enrichit le contexte ec_1.

($LocState(C\mathrm{u}_1, t_4) = LocState(C\mathrm{u}_2, t_4) = \emptyset$), “Sem” et “TaP” ne sont pas mesurables et les préférences des utilisateurs par rapport à ces éléments restent indéterminées (Tab. V.2(a)). Entre t_4 et t_5, une avarie réseau temporaire empêche la diffusion des informations concernant le village côtier, et le score de l'élément “TaP” correspondant au village est indéterminé à t_5 (Tab. V.2(b)). Enfin, lorsque les clients $C\mathrm{u}_1$ ou $C\mathrm{u}_2$ quittent l'aire de service à t_6 et t_9, ils changent de contexte d'exécution et perdent leur capacité à attribuer des scores aux éléments disponibles dans le contexte ec_1 (Tab. V.2(c) et (d)).

Score moyen instantané d'un élément

Les scores de groupe sont construits à partir des scores individuels de chaque membre du groupe. Dans une première étape, un score moyen instantané est construit pour chaque élément mesurable. Il représente une synthèse des scores entre les utilisateurs du groupe, pour le dernier intervalle de mesures implicites, entre t_{x-1} et t_x.

Importance relative des utilisateurs du groupe Les scores personnels d'un ensemble d'utilisateurs, attribués à des mêmes éléments, peuvent être comparés et éventuellement mis en commun pour constituer un profil de groupe. La moyenne arithmétique des scores par éléments constitue l'approche la plus immédiate pour effectuer la synthèse d'un ensemble de scores personnels. Cette approche considère de façon égale tous les utilisateurs et donne à leurs scores personnels une même importance. Toutefois, l'expérience et les connaissances des utilisateurs envers les éléments qu'ils notent peuvent être nuancées. Notamment, en considérant deux utilisateurs $C\mathrm{u}_a$ et $C\mathrm{u}_b$ de rôle “u” mesurant le contexte ec_k à t_x et associés au sein du groupe $Users(\mathrm{u}, ec_k, t_x)$:

- lorsque l'utilisateur $C\mathrm{u}_a$ a passé plusieurs intervalles d'exécution dans le contexte ec_k, alors que l'utilisateur $C\mathrm{u}_b$ découvre seulement les éléments disponibles dans ce contexte, le premier utilisateur a potentiellement acquis une plus grande expérience des éléments affichés sur sa carte. Il a passé plus de temps que le second utilisateur à interagir avec ces éléments, à la fois via la médiation du service de cartographie, mais aussi dans son environnement réel. L'apport d'expérience de $C\mathrm{u}_a$ est plus important que la contribution initiale de $C\mathrm{u}_b$;
- de la même manière, il peut être souhaitable de distinguer différents niveaux de privilège entre un utilisateur $C\mathrm{u}_a$ qualifié d'expert et un utilisateur $C\mathrm{u}_b$ novice. Dans ce cas, les scores personnels de l'utilisateur expert prennent le pas sur les recommandations issues des préférences de l'utilisateur novice.

Dans l'approche proposée de mise en commun des scores individuels, le choix des différentiations entre utilisateurs est laissé à l'appréciation des concepteurs du système. Un crédit d'expérience, de privilège ou de tout autre critère de différentiation entre un ensemble d'utilisateurs peut être répercuté lors du calcul des valeurs moyennes des éléments en augmentant l'importance des scores personnels de certains utilisateurs par rapport à d'autres. Cette importance des utilisateurs les uns par rapport aux autres est représentée par une fonction de pondération $\omega_{usr}(elm_i, C\mathrm{u}_k, t_x)$, spécifiée à la conception, et qui retourne le poids associé au score personnel accordé par un utilisateur $C\mathrm{u}_k$ à un élément elm_i à l'instant d'exécution t_x (Fig. V.5(b)). Quelle que soit la définition choisie, cette fonction $\omega_{usr}(\ldots)$ renvoie des poids

réels positifs :

$$\forall\ t_x \in T,\ C\mathrm{u}_k \in Cmps(\mathrm{u}, t_x),\ elm_i \in ELEMS\ :\ \omega_{usr}(elm_i, C\mathrm{u}_k, t_x) \in \mathbb{R}^+$$

Par exemple, lors de la construction d'un score commun d'un élément elm_1 à un instant t_x, lorsque $\omega_{usr}(elm_1, C\mathrm{u}_a, t_x) = 2 \times \omega_{usr}(elm_1, C\mathrm{u}_b, t_x)$, l'intérêt de l'utilisateur $C\mathrm{u}_a$ pour l'élément elm_1 est considéré avec deux fois plus d'importance que le score donné par l'utilisateur $C\mathrm{u}_b$. La définition de cette fonction de pondération à la conception permet d'envisager des comportements adaptés au cas d'application, depuis une approche équitable pour laquelle chaque score d'utilisateur obtient un même poids, jusqu'aux comportements adaptés selon l'exécution, où les poids des scores dépendent de l'instant, de l'utilisateur ou de l'élément considéré.

Dans l'exemple du système de suivi côtier, les concepteurs accordent plus d'importance aux scores d'utilisateurs ayant passé du temps à interagir avec les éléments qu'ils notent. Ils définissent la fonction de poids dynamique suivante :

$$\omega_{usr}(elm_i, C\mathrm{u}_j, t_x) = \sum_{y=0}^{x} f(y, C\mathrm{u}_j),\ \text{avec}\ f(y, C\mathrm{u}_j) = \begin{cases} 1 & \text{ssi}\ \left(C\mathrm{u}_j \in Users(\mathrm{u}, LocState(C\mathrm{u}_j, t_x), t_y)\right) \\ 0 & \text{autrement} \end{cases}$$

Dans ce cas, le poids du score d'un élément elm_i correspond au nombre d'intervalles d'exécution que l'utilisateur a passé dans le contexte où cet élément est disponible depuis l'instant t_0 d'initialisation du système. Par exemple, entre $t_5 + \epsilon$ et $t_7 - \epsilon$, l'utilisateur $C\mathrm{u}_2$ est resté en dehors de l'aire de service et l'utilisateur $C\mathrm{u}_1$ a passé plus de temps à profiter du système et des éléments à sa disposition. Avec la définition adoptée de la fonction de pondération, cette différence d'expérience est reflétée par les poids des utilisateurs à t_7 :

$$\forall\ elm_i \in \{\mathrm{TaP}, \mathrm{Sem}\}\ :\ \left(\omega_{usr}(elm_i, C\mathrm{u}_1, t_7) = 3\right) > \left(\omega_{usr}(elm_i, C\mathrm{u}_2, t_7) = 2\right)$$

Définition du score moyen d'un élément Les scores personnels sont construits en mesurant implicitement les interactions et les déplacements des utilisateurs lors d'une session qui couvre la durée du dernier intervalle d'exécution $[t_{x-1}, t_x[$. En se reposant sur les scores personnels, le score moyen d'un élément est valable sur ce même intervalle, et ces scores moyens sont construits avec la même fréquence que les scores personnels, à chaque instant de l'exécution. Étant donné un contexte d'exécution ec_k, la fonction $MeanVal(elm_i, ec_k, t_x)$ retourne le score moyen attribué à un élément elm_i, déterminé à partir des scores pondérés de l'ensemble des utilisateurs mesurant à t_x un contexte local égal à ec_k :

$$\forall\ t_x \in T, elm_i \in ELEMS, ec_k \in STATES\ :\ MeanVal(elm_i, ec_k, t_x) = \begin{cases} \displaystyle\sum_{\substack{elm_j \in \\ Users(\mathrm{u}, ec_k, t_x)}} \left(\frac{\omega_{usr}(elm_i, C\mathrm{u}_j, t_x)}{Sum_{\omega_{usr}}(elm_i, ec_k, t_x)} \times \frac{Score(elm_i, C\mathrm{u}_j, t_x)}{Sum_{Score}(C\mathrm{u}_j, ec_k, t_x)} \right) & \text{ssi}\ \underset{(\mathrm{u}, ec_k, t_x)}{Users} \neq \emptyset \\ \frac{1}{|Elems(ec_k, t_x)|} & \text{ssi}\ \begin{smallmatrix} Users(\mathrm{u}, ec_k, t_x) = \emptyset \\ \wedge Elems(ec_k, t_x) \neq \emptyset \end{smallmatrix} \\ 0 & \text{autrement} \end{cases}$$

$$\text{avec } \underset{(elm_i, ec_k, t_x)}{Sum_{\omega_{usr}}} = \begin{cases} \sum\limits_{\substack{C\mathrm{u}_l \in \\ Users(\mathrm{u}, ec_k, t_x)}} \omega_{usr}(elm_i, C\mathrm{u}_l, t_x) & \text{ssi } \left(\begin{smallmatrix} \exists\ C\mathrm{u}_m \in Users(\mathrm{u}, ec_k, t_x) \mid \\ \omega_{usr}(elm_i, C\mathrm{u}_m, t_x) \neq 0 \end{smallmatrix} \right) \\ 1 & \text{autrement} \end{cases}$$

$$\text{et } \underset{(C\mathrm{u}_j, ec_k, t_x)}{Sum_{Score}} = \begin{cases} \sum\limits_{\substack{elm_l \in \\ Elems(ec_k, t_x)}} Score(elm_l, C\mathrm{u}_j, t_x) & \text{ssi } \left(\begin{smallmatrix} \exists\ elm_m \in Elems(ec_k, t_x) \mid \\ Score(elm_m, C\mathrm{u}_j, t_x) \neq 0 \end{smallmatrix} \right) \\ 1 & \text{autrement} \end{cases}$$

Cette définition du score moyen d'un élément distingue deux cas de figure en fonction de la présence d'utilisateurs au sein d'un contexte d'exécution :

- lorsqu'aucun utilisateur n'interagit avec les éléments disponibles dans un contexte donné ec_k à l'instant t_x du calcul des scores moyens (c-à-d. $Users(\mathrm{u}, ec_k, t_x) = \emptyset$), chaque élément reçoit le même score $1/|Elems(ec_k, t_x)|$. Par exemple, si trois éléments disponibles dans un contexte ec_k sont mesurables à un instant t_b, mais qu'aucun utilisateur ne mesure ce contexte, alors un score moyen de $\frac{1}{|Elems(ec_k, t_b)|} = \frac{1}{\text{(nombre d'éléments)}} = 1/3$ est attribué à chaque élément ;
- s'il existe des utilisateurs interagissant avec l'élément elm_i (c-à-d $Users(\mathrm{u}, ec_k, t_x) \neq \emptyset$), alors un score moyen pondéré est attribué à cet élément. Ce score moyen correspond à la somme des scores pondérés de chaque utilisateur du groupe. Plus précisément, dans la somme des valeurs individuelles, le quotient $\frac{\omega_{usr}(elm_i, C\mathrm{u}_j, t_x)}{Sum_{\omega_{usr}}(elm_i, ec_k, t_x)} = \frac{\text{[Poids d'un utilisateur]}}{\text{[Somme des poids des utilisateurs]}}$ retourne une valeur dans l'intervalle réel $[0, 1]$ et qui signifie la part du poids d'un utilisateur relativement aux poids des autres utilisateurs du groupe. De la même manière, le quotient $\frac{Score(elm_i, C\mathrm{u}_j, t_x)}{Sum_{Score}(C\mathrm{u}_j, ec_k, t_x)} = \frac{\text{Score d'un élément}}{\text{Somme des scores des éléments}}$ retourne une valeur dans $[0, 1]$ qui correspond à la part d'un score individuel relativement à l'ensemble des scores attribués par cet utilisateur.

Les fonctions $Sum_{\omega_{usr}}(\ldots)$ et $Sum_{Score}(\ldots)$ retournent respectivement la somme des poids des utilisateurs et la valeur totale des scores attribués par un utilisateur. Ces fonctions assurent aussi que les diviseurs des quotients de $MeanVal(\ldots)$ ne sont pas nuls [4].

Dans le cas du système de suivi côtier, à l'instant t_7, les contextes mesurés au niveau des plates-formes $C\mathrm{u}_2$ et $C\mathrm{u}_1$ sont identiques et valent $LocState(C\mathrm{u}_1, t_7) = LocState(C\mathrm{u}_2, t_7) = ec_1$. Les utilisateurs de ces plates-formes sont groupés : $Users(\mathrm{u}, ec_1, t_7) = \{C\mathrm{u}_1, C\mathrm{u}_2\}$. Ces utilisateurs interagissent avec les éléments "TaP" et "Sem" et construisent entre t_6 et t_7 les scores individuels de ces éléments. A l'instant t_7, La valeur moyenne de l'élément "TaP" est composée à partir des scores individuels des deux utilisateurs dans le contexte ec_1 : $MeanVal(\mathrm{TaP}, ec_1, t_7) = 0.63$ (Annexe A.1). A l'instant t_4, comme aucun utilisateur n'accède au système dans le contexte d'exécution ec_1, les scores moyens sont distribués équitablement entre les éléments mesurables : $MeanVal(\mathrm{TaP}, ec_1, t_4) = MeanVal(\mathrm{Sem}, ec_1, t_4) = \frac{1}{2}$ (Annexe A.2).

4. Ce cas de figure peut se présenter lorsque tous les poids des utilisateurs valent 0 ou si les scores d'un utilisateur donné valent chacun 0.

Score de groupe d'un élément

La valeur moyenne d'un élément est valide à un instant d'exécution t_x et représente un score collectif basé sur la détermination implicite des préférences des utilisateurs lors du dernier intervalle d'exécution $[t_{x-1}, t_x[$. Ces scores de préférences moyennes restent instantanés dans le sens où ils ne prennent pas en compte les scores attribués lors des intervalles d'exécution précédents, c'est-à-dire entre t_0 et t_{x-1}. Le score de groupe d'un élément combine ces scores immédiats aux mesures passées pour atténuer les variations entre les scores obtenus à chaque instant d'exécution (Fig. V.5(g)).

Poids des profils et des scores moyens précédents ont souligné la nécessité de maintenir des profils à jour et de mesurer régulièrement les préférences des utilisateurs pour éviter une trop grande inertie des profils et des recommandations du système en opposition avec les intentions du moment des utilisateurs. A l'extrême opposé, les mesures et scores construits jusqu'ici reposent tous sur des mesures du dernier intervalle d'exécution. Potentiellement, les profils personnels et les scores moyens changent du tout au tout entre deux intervalles d'exécution et plutôt qu'une trop grande inertie, les profils personnels sont assujettis à une forte variabilité.

Au niveau de la construction des profils de groupe, pour éviter les écueils de l'excès d'inertie ou de variabilité, l'impact que doit avoir l'historique des préférences lors du calcul du profil de groupe à l'instant t_x est laissé au jugement des concepteurs du système. Par exemple, lorsque l'intervalle d'exécution est suffisamment important, il peut être suffisant de générer un profil de groupe uniquement à partir des scores et profils personnels construits lors du dernier intervalle. A l'inverse, dans des systèmes qui mesurent les changements de contexte en quasi temps-réel, le pas d'exécution est très court et ces intervalles de mesure ne sont probablement pas suffisants pour inférer des préférences et compléter les scores et profils. Dans ce cas, il devient nécessaire de tenir compte de plusieurs intervalles d'exécution lors de la construction d'un profil de groupe. Les concepteurs peuvent aussi choisir une approche mitigée qui considère que l'historique récent des scores est plus important que les anciennes mesures pour le calcul d'un profil commun à un moment donné.

Ces différents cas de figure sont répercutés lors du calcul des profils de groupe en considérant qu'à un instant t_x, le score de groupe d'un élément est déduit à la fois des scores moyens construits à t_x et du score de groupe précédent, construit à t_{x-1}. Dans ce cas, le score moyen rapporte les préférences instantanées des utilisateurs et le score de groupe précédent représente par récursivité l'historique des scores de groupe depuis l'instant t_0 d'initialisation. Deux fonctions définies lors de la conception et appliquées au cas d'étude modélisé permettent d'adapter la part d'historique dans le calcul d'un score de groupe et l'importance d'un score de groupe précédent. En considérant un score de groupe construit à l'instant t_x pour un élément elm_i disponible dans un contexte ec_k :

- la fonction de pondération $\omega_{hist}(elm_i, ec_k, t_n, t_x)$ retourne un poids (réel positif) qui dénote l'importance attribuée au score de groupe obtenu par l'élément elm_i à l'instant t_n par rapport au score de groupe de cet élément à t_x ;
- la fonction de balance $\omega_{bal}(elm_i, ec_k, t_x)$ définit l'équilibre entre la part historique et la part instantanée des scores d'éléments lors de la construction des scores de

groupe. Cette fonction retourne un résultat dans $[0, 1]$ qui représente la proportion de l'historique des préférences par rapport aux scores instantanés.

Les fonctions $\omega_{hist}(\ldots)$ et $\omega_{bal}(\ldots)$ sont définies par les concepteurs au moment de l'implémentation du système et respectent les contraintes suivantes :

$$\forall\ t_x, t_n \in T,\ elm_i \in ELEMS,\ ec_k \in STATES\ :\ \omega_{hist}(elm_i, ec_k, t_n, t_x) \in \mathbb{R}^+$$

et

$$\forall\ t_x \in T,\ elm_i \in ELEMS,\ ec_k \in STATES\ :\ \omega_{bal}(elm_i, ec_k, t_x) \in \mathbb{R}^+$$

Par exemple, lorsque $\omega_{hist}(elm_1, ec_1, t_2, t_3) = 2 \times \omega_{hist}(elm_1, ec_1, t_1, t_3)$, alors le score de groupe qu'avait obtenu un élément elm_1 à t_2 est considéré deux fois plus important que son ancien score à t_1 lors du calcul du score de cet élément elm_1 à t_3. Si la balance passé/instantané vaut à t_3 $\omega_{bal}(elm_1, ec_1, t_3) = \frac{3}{4}$, alors l'historique des scores de groupe de l'élément elm_1 constituera trois quarts de son nouveau score à t_3.

Dans le cas du système de suivi côtier, les concepteurs définissent les fonctions $\omega_{bal}(\ldots)$ et $\omega_{hist}(\ldots)$ suivantes :

$$\omega_{hist}(elm_i, ec_k, t_n, t_x) = \begin{cases} \frac{1}{x-n} & \text{ssi } n - x \leq 2 \wedge n < x \\ 0 & \text{autrement} \end{cases}$$

et

$$\omega_{bal}(elm_i, ec_k, t_x) = \begin{cases} 1 - \frac{1}{|Users(\mathrm{u}, ec_k, t_x)|} & \text{ssi } |Users(\mathrm{u}, ec_k, t_x)| > 0 \\ 0 & \text{autrement} \end{cases}$$

Dans cette exemple de balance passé/instantané, plus il y a d'utilisateurs à un moment donné dans un contexte d'exécution, plus la fonction $\omega_{bal}(\ldots)$ tend vers 1 et plus la part d'historique prend d'importance par rapport aux scores instantanés. Ce choix conceptuel permet d'assurer que lorsqu'un nombre important d'utilisateurs rejoint un contexte, les préférences établies par les utilisateurs déjà présents dans ce contexte ne sont pas réduites à portion congrue au sein du profil de groupe. A l'inverse, cette définition garantit qu'un utilisateur seul dans un contexte applique uniquement ses préférences instantanées comme profil de groupe. Par exemple, lorsque l'utilisateur $C\mathrm{u}_1$ se retrouve à t_6 seul dans le contexte ec_1, quel que soit l'élément $elm_i \in \{\mathrm{TaP}, \mathrm{Sem}\}$, $\omega_{bal}(elm_i, ec_1, t_6) = 1 - \frac{1}{|\{C\mathrm{u}_1\}|} = 0$, et le profil de groupe est identique au profil personnel de cet utilisateur [5].

La définition proposée de la fonction $\omega_{hist}(\ldots)$, garantit un impact plus important aux mesures de scores récentes. Plus spécifiquement, les scores de groupe établis lors des deux instants d'exécution précédents conservent un impact sur les scores de groupe présents. Les scores établis avant t_{x-2} sont considérés comme négligeables et se voient attribuer un poids nul. Le score "historique" construit à partir des profils des deux étapes précédentes est jugé par les concepteurs comme un bon compromis entre l'inertie du contenu passé et la variabilité

5. Selon les choix d'adaptation réalisés en application des profils de groupe, il peut être souhaitable qu'un utilisateur seul à construire un profil de groupe voie ses préférences instantanées intégralement et immédiatement répercutées dans ce profil, sans tenir compte de la part d'historique de ses préférences. Le cas d'application du système d'information campus présenté au chapitre VI.2 illustre cette situation et montre comment ce comportement permet d'assurer une certaine cohérence dans les modifications apportées à la présentation des éléments.

des mesures instantanées. Par exemple, lors du calcul d'un score de groupe à t_4, et quel que soit l'élément $elm_i \in \{\text{TaP}, \text{Sem}\}$ choisi, le poids $\omega_{hist}(elm_i, ec_1, t_3, t_4) = \frac{1}{4-3} = 1$ du score de cet élément est deux fois plus important que son poids $\omega_{hist}(elm_i, ec_1, t_2, t_4) = \frac{1}{2}$ à l'étape antérieure.

Définition du score de groupe Le score de groupe d'un élément résulte de la balance entre les valeurs de scores moyens instantanés d'un élément et une valeur représentant l'historique des scores de groupe. Cette valeur d'historique, qui définit partiellement le score de groupe à t_x à partir du score de groupe à t_{x-1}, implique un calcul récursif des scores d'un élément à un instant donné. Soit $GrpScore(elm_i, ec_k, t_x)$, la fonction retournant le score de groupe à t_x de l'élément mesurable elm_i, disponible dans le contexte d'exécution ec_k, $GrpScore(\ldots)$ est définie par récursion et implique les fonctions de pondération $\omega_{hist}(\ldots)$ et $\omega_{bal}(\ldots)$:

$$\forall\ t_x \in T, ec_k \in STATES, elm_i \in ELEMS\ :$$
$$\begin{cases} GrpScore(elm_i, ec_k, t_0) = MeanVal(elm_i, ec_k, t_0) \\ GrpScore(elm_i, ec_k, t_x) = \big(1 - \omega_{bal}(elm_i, ec_k, t_x)\big) \times MeanVal(elm_i, ec_k, t_x) + \\ \qquad \omega_{bal}(elm_i, ec_k, t_x) \times \displaystyle\sum_{j=1}^{x} \left(\frac{\omega_{hist}(elm_i, ec_k, t_{x-j}, t_x)}{Sum_{\omega_{hist}}(elm_i, ec_k, t_x)} \times GrpScore(elm_i, ec_k, t_{x-j}) \right) \end{cases}$$
$$\text{avec } \underset{(elm_i, ec_k, t_x)}{Sum_{\omega_{hist}}} = \begin{cases} 1 & \text{ssi } \sum_{j=1}^{x} \omega_{hist}(elm_i, ec_k, t_{x-j}, t_x) = 0 \\ \displaystyle\sum_{j=1}^{x} \omega_{hist}(elm_i, ec_k, t_{x-j}, t_x) & \text{autrement} \end{cases}$$

À l'instant t_0, le score de groupe d'un élément est une copie du score moyen instantané de cet élément. Lorsque t_x est supérieur à t_0, le score de groupe est constitué par l'addition de deux sous-parties :

- d'une part la partie instantanée du score de l'élément est formée par la valeur moyenne instantanée de l'élément pondérée par la balance $\omega_{bal}(\ldots)$ $\big($c-à-d. $(1 - \omega_{bal}(\ldots)) \times MeanVal(\ldots)\big)$
- d'autre part, la partie historique du score de l'élément est constituée par la somme des anciennes valeurs de scores de groupe entre t_x et t_0, chacune d'elle étant pondérée par un poids $\omega_{hist}(\ldots)$ $\Big($c-à-d $\sum \Big(\frac{\omega_{hist}(\ldots)}{Sum_{\omega_{hist}}(\ldots)} \times GrpScore(\ldots) \Big)\Big)$. Cette somme est elle-même pondérée selon la proportion du score de groupe consacrée à la part d'historique $\big($c-à-d. $\omega_{bal}(\ldots) \times \sum(\ldots)\big)$.

Pour éviter les cas de diviseurs nuls, la fonction $Sum_{\omega_{hist}}(\ldots)$ retourne la somme des poids des scores précédents ou la valeur arbitraire 1 en cas d'égalité à 0 de l'ensemble des poids.

Dans l'exemple du système de suivi côtier, les définitions proposées des fonctions de pondération $\omega_{hist}(\ldots)$ et $\omega_{bal}(\ldots)$ permettent une distribution équitable entre les valeurs historiques et les valeurs instantanées. Le tableau V.3a présente pour chaque élément les scores moyens instantanés obtenus en combinant les scores personnels des utilisateurs Cu_1 et Cu_2.

Les scores de groupe, présentés dans le tableau V.3b, combinent ces moyennes instantanées avec les scores de groupe précédents et mettent en lumière le rôle de la fonction de

Tableau V.3 – Scores des éléments “TaP” et “Sem” entre t_4 et t_9

(a) Valeurs moyennes instantanées

x	4	5	6	7	8	9
$MeanVal$ (TaP, ec_1, t_x)	.5	0	.26	.63	.4	.83
$MeanVal$ (Sem, ec_1, t_x)	.5	1	.74	.37	.6	.17

(b) Scores de groupe

x	4	5	6	7	8	9
$GrpScore$ (TaP, ec_1, t_x)	.5	.25	.26	.44	.42	.83
$GrpScore$ (Sem, ec_1, t_x)	.5	.75	.74	.56	.58	.17

balance entre la part d'instantané et d'historique dans la composition des scores. Notamment, lorsque l'un ou l'autre des utilisateurs se retrouve seul dans le contexte ec_1 à t_6 et t_9, les scores de groupe sont égaux aux scores moyens à ces instants. À t_7, les scores moyens instantanés favorisent clairement l'élément “TaP” par rapport à l'élément “Sem”. Néanmoins, aux instants d'exécution précédents, les scores de groupe de l'élément “Sem” prédominaient. En tenant compte de cet historique des intérêts, les scores de groupe à t_7 inversent le rapport de préférences établi par les moyennes instantanées et infèrent un niveau d'intérêt plus élevé envers l'élément “Sem”.

Profil de groupe

Le profil de groupe à un instant donné est l'ensemble des scores de groupe des éléments disponibles à cet instant et dans un contexte d'exécution donné. Dans la suite, $GrpProf(ec_k, t_x)$ désigne la fonction retournant le contexte de groupe associé au contexte ec_k à l'instant t_x.

$$\forall\ t_x \in T, ec_k \in STATES \ : \ GrpProf(ec_k, t_x) = \bigcup_{\substack{elm_i \in \\ Elems(ec_k, t_x)}} GrpScore(elm_i, ec_k, t_x)$$

Dans le cas du système de suivi côtier, les scores de groupe entre t_4 et t_9 sont rassemblés et forment les profils de groupe à chaque instant d'exécution (Tab. V.4).

Tableau V.4 – Profils de groupe

x	4	5	6	7	8	9
$GrpProf$ (ec_1, t_x)	$\left\{\begin{matrix}.5,\\.5\end{matrix}\right\}$	$\{.75\}$	$\left\{\begin{matrix}.26,\\.74\end{matrix}\right\}$	$\left\{\begin{matrix}.44,\\.56\end{matrix}\right\}$	$\left\{\begin{matrix}.42,\\.58\end{matrix}\right\}$	$\left\{\begin{matrix}.83,\\.17\end{matrix}\right\}$

Ces profils de groupe peuvent être utilisés comme source de personnalisation du contenu présenté avec l'objectif de simplifier l'utilisation des outils mobiles. Une approche possible consiste à mettre en valeur les éléments qui s'adjugent un niveau d'intérêt élevé. Un système plus flexible pour l'utilisateur lui propose de jongler entre une personnalisation sur la base de son profil personnel et des recommandations multi-utilisateurs basées sur le profil de groupe. Une autre utilisation combinée des profils personnels et des profils de groupe permet

d'assister les utilisateurs dans la découverte du contenu qui présente le plus d'intérêt pour une majorité d'utilisateurs. En comparant les centres d'intérêt personnels et de groupe, un système peut produire des recommandations non intrusives en faveur des éléments favorisés par le groupe et qui sont restés indifférenciés au niveau du profil de l'utilisateur. Certaines des utilisations possibles des profils personnels et de groupe sont développées au chapitre VI.2. Notamment, le cas d'application du système d'information campus illustre la production de recommandations de contenu par la mise en valeur des éléments d'intérêt issus de ces profils.

V.3 Conclusion

La démarche de construction de profil pour effectuer des recommandations sur le contenu présenté aux utilisateurs a été introduite dans ce chapitre comme une manière supplémentaire d'assurer un maximum de flexibilité et de capacité à s'adapter de la part du système. Par rapport à la capacité à s'accommoder d'un ensemble de contextes d'exécution, les profils d'utilisateur et de groupe composent une partie du contexte d'usage et permettent de spécialiser les informations présentées dans chaque contexte d'exécution.

L'approche retenue pour la constitution de profils s'appuie sur la mesure implicite des préférences des utilisateurs vis-à-vis des éléments qui lui sont présentés. Plus spécifiquement, deux sources de mesures sont identifiées dans le cas de systèmes mobiles. D'une part, les interactions de l'utilisateur sur son client sont enregistrées et analysées, d'autre part, les propres déplacements de l'utilisateur dans l'environnement physique sont considérés comme des indicateurs d'intérêt pour les éléments alentours. A partir de ces mesures, une démarche de construction de profils, d'abord personnel, puis de groupe a été proposée. Elle s'appuie sur la décomposition d'un système mobile et distribué qui distingue un ensemble de contextes d'exécution pour proposer des profils adaptés aux recommandations de contenu disponible dans ces contextes d'exécution. Du point de vue du concepteur, cette méthode de construction de profils est paramétrée par un ensemble de fonctions de pondération. Lors de la construction d'un profil à un moment donné, ces fonctions permettent de jauger la contribution des anciens profils, des autres utilisateurs, des interactions sur le client, des déplacements de l'utilisateur, etc. Cette paramétrisation souple de la construction de profils a pour objectif d'assurer aux concepteurs la latitude nécessaire pour adapter une méthode générique de la construction de profils à une variété de cas d'application. L'exemple du système de suivi côtier a ainsi montré une construction de profil de groupe dynamique, dépendante du nombre d'utilisateurs mobilisés dans un contexte donné. Par rapport aux grandes familles de constructions implicites de profils par l'apport d'expérience et selon les choix de paramétrage retenus par les concepteurs, le modèle proposé est équivalent soit à un système à analyse de contenu, soit à un système à analyse conjointe, soit à une hybridation entre ces systèmes.

La construction implicite de profils a pour objectif de permettre la recommandation de contenu présenté à un utilisateur ou à un ensemble d'utilisateurs. En ayant construit à chaque instant de l'exécution à la fois un profil personnel et un profil de groupe, il est possible de choisir l'une ou l'autre de ces sources pour déduire des recommandations et adapter le contenu présenté. Un cas d'application des profils de groupe est proposé au chapitre VI.2.

CHAPITRE VI

Illustration de la démarche de modélisation

Sommaire

VI.1 Cas d'étude : système de suivi de régate 168
VI.1.1 Contexte du système de suivi de régate 169
VI.1.2 Descriptions géographique et contextuelle de l'exécution 170
VI.1.3 Analyse des tâches et construction d'un arbre situé 175
VI.1.4 Maquettage pour l'implémentation du tiers client 179
VI.2 Cas d'étude : système d'information de campus 181
VI.2.1 Contexte du système d'information campus 181
VI.2.2 Architecture distribuée au sein du campus 182
VI.2.3 Perspective multi-utilisateurs et adaptations 185
VI.3 Conclusion 195

Differents cadres d'étude ont accompagné et parfois motivé les recherches présentées dans ce mémoire. Dans ce chapitre, les scénarios de travail les plus impactants sont présentés. L'application des principes de conception et des algorithmes de personnalisation précédemment définis à ces cas d'étude constitue le cadre d'expérimentation des travaux de thèse.

Dans une première étude, la démarche de conception basée sur l'analyse de l'environnement d'exécution et sur la construction d'un arbre des tâches situé est appliquée à un cas scénarisé de développement logiciel. Ce cas d'étude imagine la conception d'un service localisé de suivi de compétition nautique qui viendrait s'ajouter à l'infrastructure et aux services déployés lors des journées du Grand Prix de voile de l'Ecole Navale. L'infrastructure existante alimente la décomposition de l'espace de fonctionnement du système en régions d'intérêt et d'exécution. La méthodologie de conception, détaillée dans les chapitres III et IV, tient compte de ces ressources distribuées et éventuellement mobiles pour mieux appréhender l'ensemble des cas de contexte d'exécution et les intégrer à une démarche de conception de système interactif. Ce premier exemple illustre l'initiation de la démarche de conception et s'achève par la construction de maquettes du client rendu disponible aux spectateurs. Plus spécifiquement, les contextes d'exécution liés à la plate-forme mise en œuvre lors du Grand Prix sont déterminés et combinés, puis des scénarios d'utilisation liés au suivi en temps réel

de compétition nautique sont associés à ces contextes. L'arbre situé et les maquettes du client sont proposés comme résultats de cette première étude. L'implémentation du client et la distribution des procédures et données sur la plate-forme dépasse le cadre de cet exemple de conception.

Le second cas d'étude concerne la mise en œuvre d'un système de recommandation de contenu pour un service de visite de campus universitaire. Cette étude, menée en partenariat avec l'University College Dublin, imagine une système de visite, conçu sur le mode de l'arbre des tâches situé, et pour lequel il est demandé de mettre en œuvre un processus de recommandation et de personnalisation du contenu et des éléments affichés sur le client. La méthode de calcul de scores et profils décrite au chapitre V.2 est particulièrement adaptée à ce cas de figure dans lequel un ensemble d'utilisateurs accède à différents contextes d'exécution. Le système de visite est présenté de façon succincte. La personnalisation de contenu est plus particulièrement illustrée en considérant un contexte particulier et en suivant les évolutions des utilisateurs qui mesurent ce contexte. Au cours du temps, des profils personnels, moyens et de groupe sont calculés et reflètent les scores obtenus par les éléments disponibles et présentés à l'utilisateur. L'implémentation d'une chaîne de simulation des adaptations permet de régler finement les paramètres du calcul des scores avant d'envisager le déploiement de l'application sur la plate-forme distribuée.

Les deux cas d'étude détaillés dans ce chapitre illustrent les facettes complémentaires du travail de thèse. L'approche conceptuelle reste plutôt illustrée par la modélisation du service de suivi. La personnalisation dynamique est appliquée dans le cadre du service de visite de campus. Cette distinction permet d'illustrer chaque aspect de la contribution sur des exemples appropriés et réalistes. Cependant, ces deux cas d'étude assez complets ont rendu nécessaire une répartition des efforts de développement et posé des limites à la démarche de validation expérimentale :

- dans le cas du système de suivi de régate, l'exemple d'application du cadre de conception étendu n'est pas complet. Il débute par l'analyse contextuelle et l'analyse des tâches des utilisateurs pour s'achever sur la construction d'un arbre situé et le maquettage du client. Les premières étapes de conception sont illustrées mais l'implémentation du client et le déploiement des données sont laissés de côté ;
- l'application du cadre de conception reprend à partir du prototypage client pour le système de visite de campus. Dans ce second cas d'étude, les premières étapes de la conception (c-à-d. l'analyse des tâches et des contextes) sont admises et très succinctement résumées. La chaîne de simulation implémentée en Java permet d'expérimenter la constitution de profils mais seuls des tests d'utilisation in-situ permettraient un réglage fin des paramètres du modèle.

VI.1 Cas d'étude : système de suivi de régate

Dans cette section, les étapes de la conception d'un système de suivi de régate en temps réel sont détaillées, en partant de la description d'une plate-forme distribuée existante, jusqu'au maquettage des différentes interfaces clientes, associées aux variations des contextes d'exécution. Ce premier cas d'étude se limite à la description des différentes fonctionnalités offertes aux utilisateurs selon leur contexte d'exécution et décrit des adaptations provoquées

par des changements au niveau du contexte d'exécution. Par souci de cohérence avec les besoins des utilisateurs, le contexte d'usage, et la personnalisation par la mesure implicite des préférences, ne sont pas abordés dans ce premier cas d'étude.

VI.1.1 Contexte du système de suivi de régate

La méthodologie décrite par le cadre de conception étendu, et qui implique une description de l'environnement d'exécution du système, est expérimentée dans le contexte d'un championnat annuel de voile [1]. Cet évènement rassemble un large public, entre compétiteurs, passionnés de voile et visiteurs découvrant le monde de la mer. Les régates du championnat ont lieu loin de la côte (c-à-d. environ 10km) et les spectateurs suivent avec difficulté l'évolution des bateaux au loin. Dans le cadre du projet NavTrack [2], labellisé par le pôle de compétitivité Images et Réseaux, un ensemble de service est mis en œuvre pour la durée de la compétition. Ces services ont pour objectif de faciliter l'information des utilisateurs à propos des évènements de la course au large. Par exemple, une bouée météo installée en mer répercute en temps réel les conditions de température et de vent vécues par les compétiteurs. Si leur équipement portable dispose de capacités adéquates, les spectateurs accèdent aussi à un service de vidéo qui diffuse depuis les boucles de course des images des équipages en manœuvre.

Pour proposer un complément d'information à ces services, un système de suivi à distance est mis en place. Un module de positionnement en temps réel a été développé et équipe les bateaux (Fig. VI.1(a)) (). Leurs positions GPS sont transmises par communication sans fil (bande ISM) à une station de collecte qui fait office de serveur de données (Fig. VI.1(c)). Les utilisateurs peuvent suivre l'évolution des bateaux sur un dispositif mobile (Fig. VI.1(b)) en accédant aux procédures de suivi diffusées depuis le serveur de traitement (Fig. VI.1(d)).

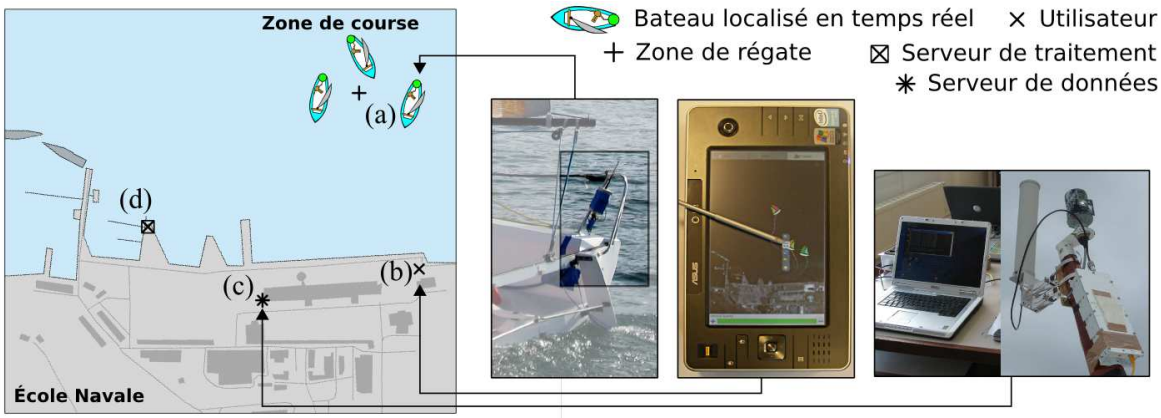


Figure VI.1 – Infrastructure du système de suivi en temps réel

Ce cas d'étude illustre les étapes de la conception d'un système d'information construit sur les trois composants matériels du système NavTrack, et qui intègre des données localisées. L'organisation des composants et les échanges d'informations entre ceux-ci ne sont pas

1. http ://www.gpen.ecole-navale.fr
2. http ://navtrack.ecole-navale.fr

fixés. Chaque composant, potentiellement mobile, rend l'architecture du système variable à l'exécution et l'adaptation nécessaire. Par exemple, le serveur de données étant différencié du serveur de traitement, il est possible que les procédures de traitements ne reçoivent plus d'informations à un moment de l'exécution. Ce contexte d'exécution doit être caractérisé à la conception en prévoyant un comportement approprié du système. L'utilisation de technologies de transmission sans fil dote chaque composant d'une aire d'influence dans laquelle il est capable d'envoyer et de recevoir des informations. Lors de la conception d'un tel système, ces régions et leurs intersections constituent la description géographique fournie en entrée du cadre de conception étendu.

VI.1.2 Descriptions géographique et contextuelle de l'exécution

La plate-forme du système de suivi est décomposée en trois familles de composants de rôles auxquels sont associées des régions d'exécution. Régions et composants peuvent être mobiles dans l'espace du système. Les relations entre régions, composants et espaces de mobilité permettent la définition des contextes d'exécution du système. Cette section présente ces mobilités et établit l'ensemble des contextes d'exécution et sa simplification dans le cas du système de suivi.

Plate-forme et régions du système

La description du fonctionnement prévu pour le système fait apparaître une répartition des rôles entre trois types de composants. Dans un premier temps, les concepteurs décrivent ces composants en leur attribuant des rôles :

- les composants de rôle "p", désignent des serveurs qui disposent de capacités de traitement nécessaires au déploiement de procédures de rendu cartographique d'informations ;
- les composants de rôle "d" hébergent et collectent en temps réel les données de positionnement des bateaux ;
- les composants utilisateurs de rôle "u" forment la cible du système et ces plates-formes sont capables d'afficher des informations localisées sur une carte numérique et supportent l'interaction avec celles-ci.

Dans le cas du système de suivi de régate, le service implémenté déploiera ces trois rôles sur la plate-forme pré-existante du projet NavTrack. Plus précisément, cette répartition s'effectue entre un composant $C\mathrm{p}_1$ de rôle "p', un composant $C\mathrm{d}_1$ de rôle "d", et un nombre arbitraire de clients de rôle "u", désignés lors de la conception par un composant générique $C\mathrm{u}_*$.

Cette définition de la distribution des composants du système d'information permet aux concepteurs de distinguer un ensemble de trois régions qui caractérisent l'environnement géographique d'exécution du système. La région $R\mathrm{p}_1$, générée par le composant $C\mathrm{p}_1$, représente l'espace où les procédures utiles pour la réalisation de traitements sur des données sont disponibles. La région $R\mathrm{d}_1$, générée par le composant $C\mathrm{d}_1$, représente l'aire de diffusion des informations où les données brutes du système sont disponibles. La région générique $R\mathrm{u}_*$ représente l'espace de l'utilisateur où le résultat de tout ou d'une partie des traitements effectués sont exploités.

En plus des régions d'exécution précédentes, et pour distinguer les possibilités de contextes d'exécution potentiellement valorisables, les concepteurs assimilent l'aire de course à une région d'intérêt possible des utilisateurs. Cette région est déterminée par l'empreinte spatiale de l'enveloppe des localisations des bateaux en course, et est associée au type $\mathrm{s} \in TYPES$ désignant la "région des bateaux suivis". L'environnement d'intérêt du système de suivi de régate comporte une région de type "s" nommée par les concepteurs "$R\mathrm{s}_1$". Le composant $C\mathrm{d}_1$ a pour rôle d'intégrer les positions des équipages et la région d'intérêt $R\mathrm{s}_1$ est formée à partir de ces renseignements. Ce serveur de données tient lieu d'hôte de la région d'intérêt à tout instant de l'exécution : $Host(\mathrm{s}_1) = C\mathrm{d}_1$.

À l'issue de cette première étape d'analyse de l'environnement d'exécution du système, les concepteurs ont décrit les composants de la plate-forme $PLATFORM$, les régions d'exécution $ENVEXEC$ générées par ces composants, et la région d'intérêt de $ENVINT$ déterminée par des positions de bateaux :

$$PLATFORM = \{C\mathrm{d}_1, C\mathrm{p}_1, C\mathrm{u}_*\} \quad ENVEXEC = \{R\mathrm{u}_*, R\mathrm{d}_1, R\mathrm{p}_1\} \quad ENVINT = \{R\mathrm{s}_1\}$$

L'ensemble de ces constituants détermine l'environnement d'exécution du système. La figure VI.2 propose une répartition possible de ces éléments à un instant t_x. Pour réaliser cette vue, les concepteurs s'appuient sur les dispositifs connus. En particulier, les composants de traitement et d'accès aux données font partie d'une infrastructure en place au moment de la conception. Les régions $R\mathrm{p}_1$ et $R\mathrm{d}_1$ sont déterminées à partir des capacités réseaux de ces composants. La région d'exécution générique d'un utilisateur $R\mathrm{u}_*$ est estimée en tenant compte d'une capacité de transmission d'une vingtaine de mètres, moyenne généralement admise pour les clients communicants par Wi-Fi (Tab. I.1).

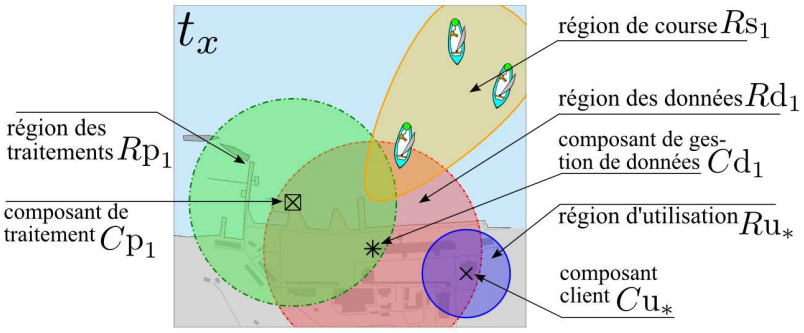


Figure VI.2 – Exemple de répartition des composants, des régions d'exécution et de la région d'intérêt à t_x

Cet exemple de situation rencontrée à t_x illustre un contexte d'exécution qui mesure une relation entre les régions $R\mathrm{p}_1$ et $R\mathrm{d}_1$: $State(t_x) = \{(R\mathrm{p}_1, R\mathrm{d}_1)\}$. Dans la suite, les concepteurs dimensionnent l'ensemble maximal des contextes à prendre en compte à la conception en appliquant des contraintes spatiales sur les régions et composants et en recherchant des équivalences entre contextes.

Contraintes spatiales

Dans ce système réparti sur trois composants et qui intègre une région d'intérêt, lorsqu'aucune limite n'est posée dans la mobilité des régions de l'environnement, il est possible de différencier jusqu'à soixante-quatre contextes d'exécution[3] : depuis le plus pauvre, pour lequel aucune région n'est liée à une autre, jusqu'au plus riche, pour lequel chaque région est en relation avec toutes les autres, et l'ensemble des contextes est maximal. Pour déterminer l'ensemble des contextes qui peuvent effectivement être mesurés à l'exécution et réduire le nombre de cas d'exécution qui rejoindront l'analyse des tâches, les concepteurs étudient la mobilité des constituants de la plate-forme et déterminent les espaces accessibles aux régions et aux composants.

Pour chaque région d'exécution, les concepteurs déterminent l'*espace accessible* à cette région et l'*espace de mobilité* de son composant générateur. Dans le cas du système de suivi de régate, l'implantation des serveurs $C\mathrm{p}_1$ et $C\mathrm{d}_1$ est contrainte par les possibilités techniques du site et les besoins des organisateurs. Plus spécifiquement, les ressources énergétiques nécessaires au fonctionnement de ces serveurs et la sensibilité des dispositifs de suivi des bateaux impliquent que les composants $C\mathrm{p}_1$ et $C\mathrm{d}_1$ soient immobiles. Les régions $R\mathrm{p}_1$ et $R\mathrm{d}_1$ générées par ces composants sont fixes à l'exécution. Les données diffusées pour le système de suivi doivent être disponibles dans l'espace dédié aux spectateurs, mais surtout rester hors de la zone de course. Les organisateurs décident de limiter la portée de diffusion des informations de course et fixent le composant de gestion de données $C\mathrm{d}_1$ suffisamment à l'écart de la zone de course. L'espace de mobilité $MOB(C\mathrm{d}_1)$ de ce composant est ponctuel et l'espace accessible $ACC(R\mathrm{d}_1)$ est égal à la région générée $R\mathrm{d}_1$ à tout instant de l'exécution (Fig. VI.3(a) : $MOB(C\mathrm{d}_1)$ & $ACC(R\mathrm{d}_1)$). Les procédures de traitement sont diffusées dans une région à forte concentration d'utilisateurs, près de l'arrivée des régates. Le composant serveur de traitement $C\mathrm{p}_1$ est fixe à tout moment de l'exécution. Comme dans le cas du serveur de données, son espace de mobilité $MOB(C\mathrm{p}_1)$ est ponctuel et l'espace accessible $ACC(R\mathrm{p}_1)$ est égal à la région d'exécution $R\mathrm{p}_1$ (Fig. VI.3(a) : $MOB(C\mathrm{p}_1)$ & $ACC(R\mathrm{p}_1)$). Les utilisateurs et leurs composants sont mobiles dans un espace $MOB(C\mathrm{u}_1)$ qui s'étend le long de la ligne de côte. La portée maximale de transmission de leurs clients limite l'espace accessible $ACC(R\mathrm{u}_*)$.

Comme pour les régions d'exécution, la couverture spatiale de la région d'intérêt peut varier à l'exécution en fonction des positions occupées par les bateaux suivis. Dans le cas d'une régate, des régions sont délimitées et établissent des frontières que les skippers ne doivent pas dépasser (c-à-d. les "ronds de course"). Cette zone maximale de la régate définit l'espace accessible $ACC(R\mathrm{s}_1)$ aux concurrents et aux bateaux suivis en temps-réel (Fig. VI.3(a) : $ACC(R\mathrm{s}_1)$).

Avec les espaces de mobilité de leurs composants générateurs et les espaces accessibles invariants à l'exécution, les régions $R\mathrm{d}_1$ et $R\mathrm{p}_1$ vérifient la propriété de fixité des régions d'exécution. À l'inverse, la région des utilisateurs et la zone de course sont mobiles au sein

3. Le nombre de contextes correspond au nombre de combinaisons de deux régions : soit $STATES$ l'ensemble des contextes d'exécution. Avec un nombre de régions $n = |ENVINT| + |ENVEXEC|$, $|STATES| = 2^{C_2^n} = 2^{C_2^4} = 64$ contextes sont dénombrés.

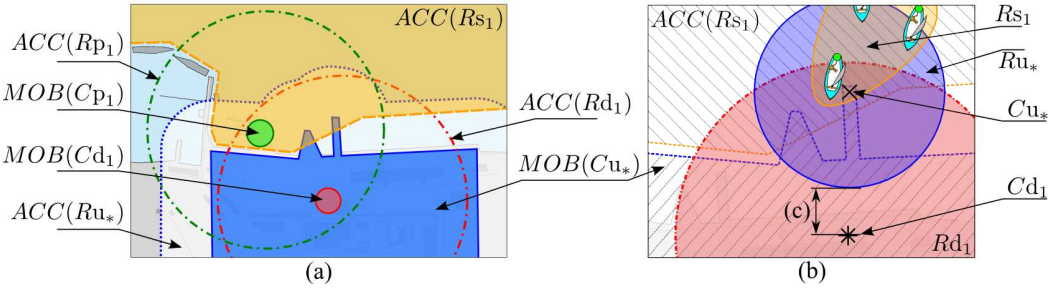


Figure VI.3 – Espaces accessibles et espaces de mobilité du système de suivi de régate

de leurs espaces accessibles et ne sont pas fixes à l'exécution :

$$Fixed(R\mathrm{d}_1) = Fixed(R\mathrm{p}_1) = 1 \qquad\qquad Fixed(R\mathrm{u}_*) = Fixed(R\mathrm{s}_1) = 0$$

En tenant compte des propriétés de fixité des régions et de la distribution géographique des espaces accessibles et espaces de mobilité, les concepteurs appliquent un ensemble de règles pour forcer l'inclusion systématique ou l'exclusion systématique d'une relation de région à un contexte d'exécution. Le système de suivi vérifie deux règles d'inclusion ou d'exclusion présentées au chapitre III.3.3 :

- les deux régions d'exécution fixes $R\mathrm{p}_1$ et $R\mathrm{d}_1$ sont en relation systématique. À tout instant de l'exécution le composant générateur de l'une est inclus dans la couverture de diffusion de l'autre. Leurs composants générateurs C_{d_1} et C_{p_1} sont toujours capables d'échanger des informations,
- la région d'exécution $R\mathrm{d}_1$ et la région d'intérêt $R\mathrm{s}_1$ ne sont jamais liées. L'aire de mobilité $MOB(C\mathrm{d}_1)$ est disjointe de l'espace accessible $ACC(R\mathrm{s}_1)$ de la course et le composant $C\mathrm{d}_1$ ne sera jamais inclus dans la zone des bateaux suivis.

D'autre part, dans le cas particulier présenté par ce système de suivi, les concepteurs énoncent une troisième règle d'exclusion conditionnelle. La région de l'utilisateur $R\mathrm{u}_*$, mobile dans $ACC(R\mathrm{u}_*)$, ne peut être à la fois en relation avec $R\mathrm{d}_1$ et $R\mathrm{s}_1$ (Fig. VI.3(b)). Quel que soit le moment de l'exécution t_x, si le composant client $C\mathrm{u}_*$ appartient à la région d'intérêt $R\mathrm{s}_1$, alors le composant serveur de données $C\mathrm{d}_1$ est en dehors de la région de diffusion du client. Dans ce cas, la relation entre la région de l'utilisateur $R\mathrm{u}_*$ et la région des données $R\mathrm{d}_1$ n'est pas vérifiée (Fig. VI.3(c)). À l'inverse, si la relation entre les régions des données et de l'utilisateur est établie à t_x (c-à-d. $Relate(R\mathrm{u}_*, R\mathrm{d}_1, t_x) = 1$), alors la relation $Relate(R\mathrm{u}_*, R\mathrm{s}_1, t_x)$ entre la région d'intérêt et la région de l'utilisateur vaut 0.

Ces règles permettent d'exclure de l'ensemble des contextes observables les configurations qui ne contiennent pas la paire $(R\mathrm{d}_1, R\mathrm{p}_1)$; qui contiennent la paire $(R\mathrm{d}_1, R\mathrm{s}_1)$; et qui contiennent à la fois $(R\mathrm{d}_1, R\mathrm{u}_*)$ et $(R\mathrm{u}_*, R\mathrm{s}_1)$. L'ensemble des contextes possibles est réduit à douze cas effectivement observables (Fig. VI.4). La notation matricielle de ces contextes illustre en vert les inclusions forcées, en rouge les exclusions forcées et en noir les relations entre régions [4]. Le contexte présenté en illustration de la répartition spatiale des composants est le plus pauvre en termes de relations entre régions. Il n'inclut que la relation forcée entre la région de traitement $R\mathrm{p}_1$ et la région des données $R\mathrm{d}_1$ (Fig. VI.4(a)).

4. Les étiquettes en lignes et colonnes désignent les régions de l'environnement. Le "R" initial est omis pour faciliter la lecture.

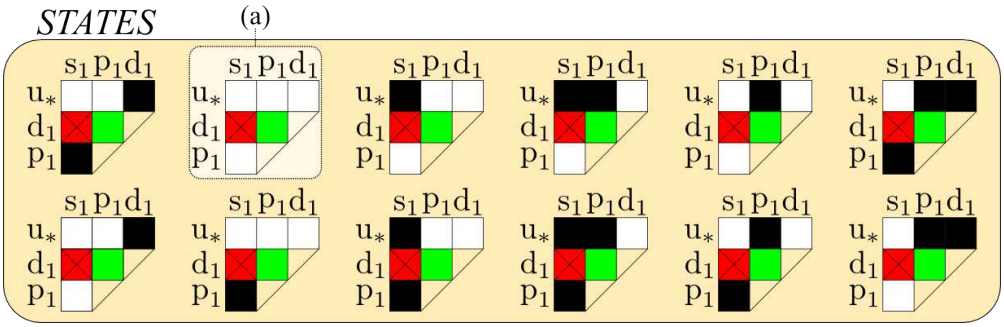


Figure VI.4 – Ensemble $STATES$ des contextes mesurables à l'exécution

Recherche d'équivalences

A partir de l'ensemble $STATES$ des contextes d'exécution, les concepteurs recherchent des équivalences fonctionnelles entre contextes. Au chapitre IV.2.2, un ensemble de propriétés d'équivalence entre contextes a été introduit. Ces propriétés permettent de regrouper deux contextes s'ils provoquent un même comportement fonctionnel du point de vue de l'utilisateur. Même si l'ensemble de règles proposées ne prétend pas à l'exhaustivité, les concepteurs du système de suivi de régate trouvent parmi cet ensemble les propriétés d'équivalence appropriées à leur cas d'étude. Plus particulièrement, ils décident d'appliquer trois propriétés d'équivalence :

- la propriété "Utilisateur seul" regroupe les situations contextuelles dans lesquelles un spectateur de la régate et sa plate-forme cliente n'ont pas de lien avec les autres composants du système ;
- la propriété "Région d'intérêt distante", appliquée au cas du système de régate, dénote l'équivalence entre deux contextes dans lesquels le serveur de traitement appartient ou non à la zone de course. Pour les concepteurs, la présence du composant de traitement à l'intérieur de la zone de course n'implique pas d'adapter le comportement du système pour l'utilisateur ;
- la propriété "Région d'intérêt indéfinie", représente l'équivalence entre une situation où un utilisateur est distant des régions d'exécution et une situation où il est uniquement inclus dans la zone de course. Cette inclusion est détectable à son niveau uniquement lorsque son client a connaissance des frontières de la région de course, communiquées par le composant $C\mathrm{d}_1$.

L'ensemble des propriétés d'équivalence est appliqué aux douze cas de contextes mesurables. Les paires de contextes équivalents sont regroupées en chaînes d'équivalence. lorsqu'un contexte donné est présent dans deux paires équivalentes, alors ces groupes forment une chaîne. Par exemple, la chaîne représentée en figure VI.5(a) est construite en combinant deux paires de contextes équivalents par la propriété "*Région d'intérêt indéfinie*" avec une paire de contextes équivalents par la propriété "*Utilisateur seul*". Les contextes de cette chaîne sont tous équivalents par transitivité. Dans ce cas, l'utilisateur ne peut jamais communiquer avec les composants actifs du système.

Les autres paires équivalentes ne comportent pas d'éléments communs et ne peuvent pas êtres liées en chaînes plus grandes. Après les combinaisons des groupes de contextes, cinq chaînes de contextes sont formées parmi les douze configurations contextuelles initiales (Fig.

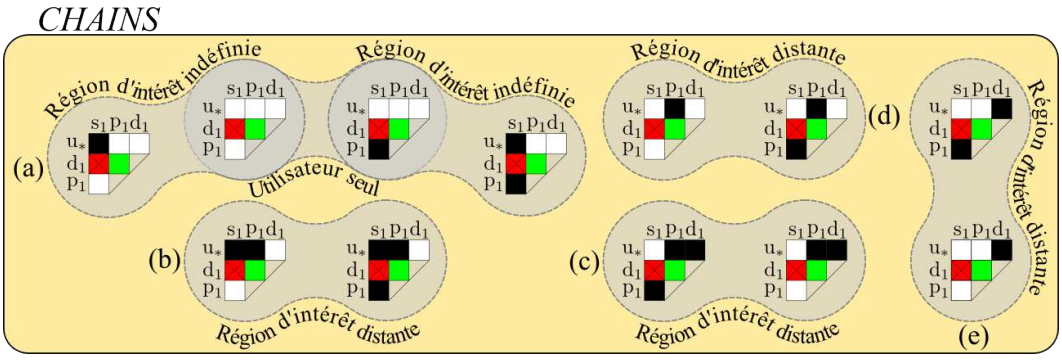


Figure VI.5 – Chaînes des contextes d'exécution équivalents du système de suivi régate

VI.5). Dans la suite, ces chaînes d'équivalence sont intégrées à un processus de conception d'un système interactif. Des comportements fonctionnels sont prévus par les concepteurs pour que le système de suivi de régate produise une réponse appropriée quelle que soit la situation contextuelle d'exécution de l'utilisateur.

VI.1.3 Analyse des tâches et construction d'un arbre situé

L'analyse des besoins des utilisateurs constitue avec la description géographique de l'environnement le second point d'entrée du modèle de conception étendu. Son application est présentée en partant du scénario nominal imaginé dans le cas du système de suivi de régate. Les étapes d'intégration de la description des contextes d'exécution et des chaînes d'équivalence sont plus précisément détaillées, avant de présenter des tâches alternatives issues de la combinaison du modèle de tâches au modèle d'environnement.

Scénario nominal et décomposition des tâches

Afin de concevoir le système de suivi, des utilisateurs potentiels et les organisateurs de la régate joignent l'équipe de développement. Ils expriment ensemble les besoins des utilisateurs, corrélés aux impératifs des organisateurs et rédigent un scénario nominal (tableau VI.1).

Les informations du domaine repérées par les concepteurs à la lecture du scénario (soulignées dans le tableau VI.1) sont organisées sous forme de classes et de relations. Par exemple, la notion de bateau est définie par les attributs de "Nom", de "Type" et d'"Année" au sein de la classe "Bateau". Les informations dynamiques comme la vitesse, le cap ou la position d'un bateau sont définies par les attributs de "Vitesse", de "Cap" et de "Localisation" au sein de la classe "DonnéesTempsRéel" associée aux objets de type "Bateau".

De la même manière, les tâches décrites dans le scénario (signifiées en gras dans le tableau VI.1) sont articulées au sein d'un arbre des tâches en notation CTT (Fig. VI.6). Dans le cas d'étude du suivi de régate, la tâche principale des utilisateurs consiste à "suivre la régate"

Tableau VI.1 – Extraits du scénario nominal établi pour le système de suivi de régate

Le système de suivi de régate s'exécute sur le terminal mobile de l'utilisateur et lui permet de ***suivre la régate*** *en temps réel. Le terminal fournit des outils de manipulation et propose une* ***carte de la zone de compétition*** *où évoluent les* ***bateaux*** *de course. L'utilisateur peut* ***définir sa zone cartographique d'intérêt*** *en fonction de* ***l'intérêt qu'il accorde à certains bateaux*** *en particulier. Il peut également choisir de suivre les* ***zones d'intérêt des autres utilisateurs****. Si un utilisateur manifeste son* ***intérêt pour un bateau spécifique****, les informations de positionnement (telles que la* ***localisation****, la* ***vitesse*** *et le* ***cap****) sont retournées. Elles sont accompagnées de méta-informations du voilier (****nom****,* ***équipage****,* ***année*** *et* ***images****). Lorsque l'utilisateur est proche de la zone de course, il peut* ***prendre des images du voilier et les partager*** *avec les autres utilisateurs.*

et comprend la sous-tâche "définir la zone cartographique d'intérêt", elle même effectuée soit par "s'intéresser à certains bateaux" soit par "suivre l'intérêt d'un autre utilisateur" [5].

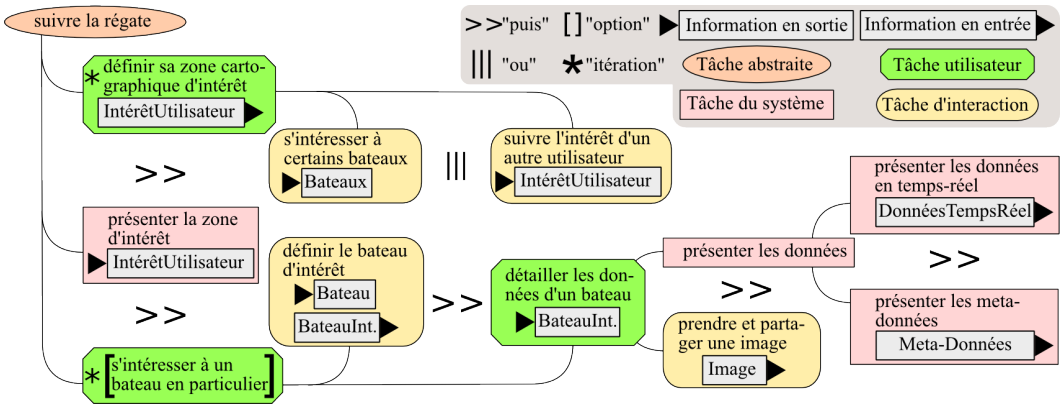


Figure VI.6 – Arbre des tâches annoté du système de suivi de régate

Les nœuds de l'arbre sont décorés par les informations issues du scénario nominal et manipulées en entrée et en sortie d'une tâche ou sous-tâche. La tâche d'interaction itérative "définir la zone cartographique d'intérêt" complète par exemple en sortie l'information "IntérêtUtilisateur" à partir d'une entrée constituée soit par un ensemble de bateaux si l'utilisateur effectue la sous-tâche "s'intéresser à certains bateaux", soit par la zone d'intérêt d'un autre utilisateur si la sous-tâche "suivre l'intérêt d'un autre utilisateur" est effectuée.

5. Dans le cas d'un accès simultané multi-utilisateurs, cette sous-tâche est motivée par le besoin éventuel de suivre les commentaires et recommandations d'autres utilisateurs. Par exemple, si un commentateur de la course porte son attention sur un équipage, il peut être intéressant pour l'audience que les informations présentées au même moment portent sur cet équipage particulier. Dans ce cas, l'audience suit l'intérêt du commentateur.

Construction de l'arbre des tâches situé

Les (sous-)tâches et les informations identifiées lors de l'analyse des tâches sont corrélées à la décomposition de la plate-forme distribuée. Plus précisément, les concepteurs déterminent quels composants hébergent les procédures qui implémentent une tâche donnée et quels sont les composants qui hébergent les données qui décrivent les informations du système. Dans le cas du système de suivi de régate, seules une région de traitement et une région de diffusion de données sont mises en place. Dès lors, l'ensemble des procédures implémentant les tâches de l'utilisateur sont disponibles dans la région $R\mathrm{p}_1$ du composant de traitement. De la même manière, toutes les informations du domaine sont enregistrées sur le serveur $C\mathrm{d}_1$ et disponibles dans la région des données $R\mathrm{d}_1$. La plate-forme de l'utilisateur est considérée comme un client léger, qui se contente de demander des traitements et d'accéder aux résultats.

Pour s'exécuter correctement, une tâche a besoin d'accéder aux informations qui lui sont associées en entrée et en sortie. En pratique, les procédures doivent pouvoir recevoir les informations nécessaires depuis un ou plusieurs serveurs de données. Dans le cas du système de suivi de régate, comme il existe une relation permanente entre les régions d'exécution $R\mathrm{p}_1$ et $R\mathrm{d}_1$, toutes les informations hébergées sur le serveur $C\mathrm{d}_1$ sont accessibles aux procédures de traitement implémentées sur le composant $C\mathrm{p}_1$. L'ensemble de la tâche est toujours réalisable dans la région d'exécution $R\mathrm{p}_1$. Du point de vue de l'utilisateur, cela signifie que dès lors qu'il accède aux services du serveur de traitement et que la relation $(R\mathrm{p}_1, R\mathrm{u}_*)$ fait partie de son contexte local, alors il peut effectuer l'ensemble de l'arbre des tâches nominales.

Parmi les chaînes d'équivalence issues de la description environnementale, seuls trois groupes de contextes parmi cinq caractérisent l'accès aux procédures du système par un utilisateur (Fig. VI.5(b, c & d)). Pour les contextes qui composent ces chaînes, la relation entre la région de l'utilisateur $R\mathrm{u}_*$ et la région des traitements $R\mathrm{p}_1$ est systématique. Si l'utilisateur mesure à t_x un contexte local parmi ceux-ci, alors il peut effectuer l'ensemble des branches qui décomposent la tâche "suivre la régate".

Afin de garantir que le système réponde aux attentes des utilisateurs dans l'ensemble de leur espace de mobilité, les concepteurs imaginent des comportements alternatifs du système pour les groupes de contextes restants (Fig. VI.5(a & e)), en enrichissant la tâche initiale par des cas alternatifs :

- lorsque la région de l'utilisateur est en relation avec la zone de diffusion des données sans être en relation avec la région de traitement, il accède à l'ensemble des données enregistrées sur $C\mathrm{d}_1$. Dans ce cas, les concepteurs considèrent qu'il est possible pour le client de supporter les traitements nécessaires à une lecture directe des méta-données sur les bateaux. La tâche "s'intéresser à un bateau en particulier" reste réalisable sans l'intervention du composant de traitement. Toutefois, la tâche de présentation des données temps-réel, qui nécessite une illustration cartographique et l'implication du serveur de traitement, est indisponible ;
- lorsque l'utilisateur n'est en relation avec aucune autre région d'exécution $R\mathrm{p}_1$ ou $R\mathrm{d}_1$, les concepteurs prévoient un traitement local au composant client $C\mathrm{u}_*$. Ce traitement présente une carte simplifiée à l'utilisateur et lui précise l'endroit où sont diffusés les services de suivi (c-à-d. les régions d'exécution $R\mathrm{d}_1$ et $R\mathrm{p}_1$).

En pratique, lors de l'implémentation de ces fonctionnalités, les développeurs auront à mettre en œuvre un système de détection de changement de contexte sous la forme d'un "écouteur

de service" au niveau du client. Ce détecteur devra tourner en tâche de fond et signalera toute nouvelle relation avec un composant actif du système et les éventuelles fonctionnalités que cette connexion permet d'activer.

D'un point de vue conceptuel, avec ces nouvelles procédures du système, chaque chaîne de contexte dispose d'un comportement associé et l'espace d'utilisation du système couvre tout l'espace de mobilité $MOB(R\mathrm{u}_*)$ de l'utilisateur. D'autre part, la position de l'utilisateur par rapport à la région d'intérêt $R\mathrm{s}_1$ permet de quantifier la notion de proximité à la zone de course, exprimée dans le scénario nominal ("Lorsque l'utilisateur est proche de la zone de course"). Cette proximité est établie lorsque la région de l'utilisateur est en relation avec la zone de course. Dans ces conditions d'exécution, un utilisateur pouvant effectuer chaque tâche du scénario nominal, se voit en plus autorisé à prendre et partager des photos de la course (c-à-d. tâche "Prendre et partager une image" du scénario nominal).

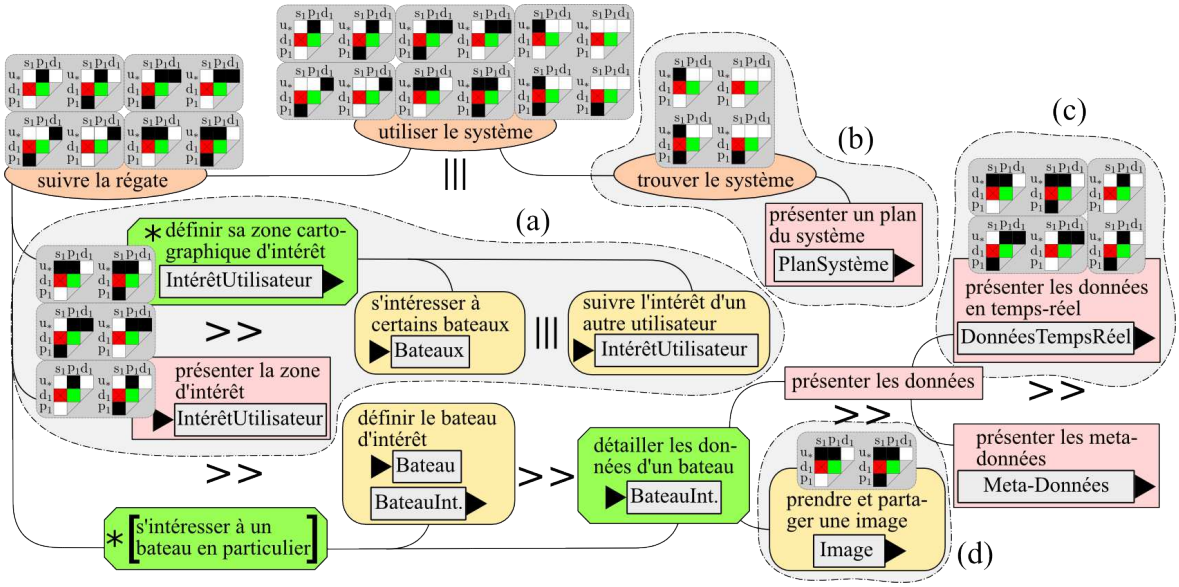


Figure VI.7 – Arbre des tâches situé du système de suivi de régate

Les chaînes de contextes sont intégrées à l'arbre des tâches sous la forme de pré-conditions nécessaires à l'exécution d'une tâche et à l'ensemble de ses sous-tâches. Les concepteurs appliquent la méthode de factorisation présentée au chapitre IV.2.3. Ils commencent par étiqueter les feuilles de l'arbre par les chaînes de contexte qui assurent les conditions d'exécution nécessaires à la sous-tâche désignée. Après plusieurs cycles de factorisations, les plus hautes branches et la racine sont à leur tour conditionnées et l'arbre devient situé (Fig. VI.7). Par exemple, les tâches d'enregistrement et de rendu du centre d'intérêt de l'utilisateur peuvent être effectuées quand le système se trouve dans l'un des états contextuels où la région des traitements est liée à la région de l'utilisateur (c-à-d quand $(R\mathrm{p}_1, R\mathrm{u}_*) \in LocState(C\mathrm{u}_*, t_x)$, Fig. VI.7(a)). Dans ces mêmes contextes, le serveur de traitement peut effectuer un rendu temps réel cartographique des informations de course (c-à-d. la tâche "présenter les données temps réel", Fig. VI.7(c)). Par contre la tâche "s'intéresser à un bateau en particulier" peut toujours être effectuée lorsque l'utilisateur est uniquement en relation avec la région de diffusion de données (c-à-d quand $(R\mathrm{d}_1, R\mathrm{u}_*) \in LocState(C\mathrm{u}_*, t_x)$

et $(R\mathrm{p}_1, R\mathrm{u}_*) \notin LocState(C\mathrm{u}_*, t_x)$). Dans ces contextes, seule la sous-tâche "présenter les méta-données" de la branche "détailler les données d'un bateau" est réalisable. Pour pouvoir partager une photo et effectuer la tâche "prendre et partager une image", l'espace utilisateur doit être lié à la zone de course. Autrement dit, le client $C\mathrm{u}_*$ doit se situer à l'intérieur de la région d'intérêt $R\mathrm{s}_1$. Une seule des chaînes de contextes équivalents représente ces conditions et réunit des contextes qui incluent la relation $(R\mathrm{s}_1, R\mathrm{u}_*)$ (Fig. VI.7(d)).

L'arbre des tâches situé peut encore être enrichi par les tâches alternatives déterminées par les concepteurs. Notamment, l'assistance du système pour la recherche d'une meilleure zone d'utilisation, comportement prévu du système lorsque l'utilisateur se retrouve isolé, est intégrée à l'arbre des tâches. Une nouvelle tâche abstraite "trouver le service", alternative à la tâche "suivre la régate", est effectuée à l'initiative du système quand l'utilisateur est hors de portée des régions actives (Fig. VI.7(b)). Cette nouvelle tâche permet de compléter les cas d'usages en imaginant un scénario complémentaire (Tab. VI.2). A l'implémentation, les procédures nécessaires à l'exécution de ce scénario alternatif devront étre pré-installées sur le client (seule plate-forme active), ou alors mises à disposition sous la forme de service Web accessible globalement. L'activation de ces procédures est conditionnée par la localisation, intérieure ou extérieure, de l'utilisateur par rapport à son espace de mobilité.

Tableau VI.2 – Scénario alternatif présentant le cas d'un utilisateur isolé

Un utilisateur marche le long du rivage et souhaite recevoir les informations de course sur son terminal mobile. Lorsqu'il arrive sur le site du grand prix, il est averti qu'il peut ***trouver le service*** *au niveau des régions des traitements ou des données et obtenir un meilleur support fonctionnel. Les instructions d'orientation vers ces régions lui sont* ***présentées sur un plan du système****.*

La lecture de l'arbre situé depuis la racine vers les feuilles, montre que quel que soit le contexte d'exécution pris parmi les douze mesurables, il existe toujours un chemin dans l'arbre menant à une tâche non abstraite. Dans ce cas, les concepteurs du système de suivi de régate sont sûrs de proposer une réponse fonctionnelle à l'utilisateur, indépendamment de la situation encourue.

VI.1.4 Maquettage pour l'implémentation du tiers client

L'arbre des tâches contraint par l'environnement, et augmenté des cas alternatifs, est le point d'entrée de l'implémentation de la partie cliente. Comme chaque état du système est caractérisé dans l'arbre, la présentation des fonctionnalités du cas nominal et des cas alternatifs est unifiée dans une interface homme-machine opérationnelle à tout moment lors de l'exécution du système.

Les concepteurs et développeurs se servent des contextes d'exécution qui annotent les nœuds de l'arbre des tâches comme indicateurs de la façon dont les procédures doivent être distribuées parmi les composants du système. Dans ce cas d'application, par exemple, la tâche "présenter un plan du système" s'exécute en intégralité sur la seule plate-forme disponible à ce moment : l'assistant personnel de l'utilisateur (composant $C\mathrm{u}_*$). La tâche "détailler les données d'un bateau" s'exécute indifféremment sur les composants $C\mathrm{p}_1$ et $C\mathrm{d}_1$ supports

des régions $R\mathrm{p}_1$ et $R\mathrm{d}_1$. Par contre, sa sous-tâche “présenter les données en temps réel” est implémentée uniquement sur le composant $C\mathrm{p}_1$.

Dans les maquettes de la figure VI.8, les adaptations des interfaces de la partie cliente sont illustrées pendant un exemple de parcours d'utilisateur au sein d'un environnement de régions en évolution. Dans cet exemple, à un instant t_x donné, le contexte d'exécution mesuré par le client (c-à-d. $LocState(C\mathrm{u}_*, t_x)$) est mis en correspondance avec la maquette l'interface utilisateur prévue pour leur permettre d'effectuer les tâches disponibles dans ce contexte.

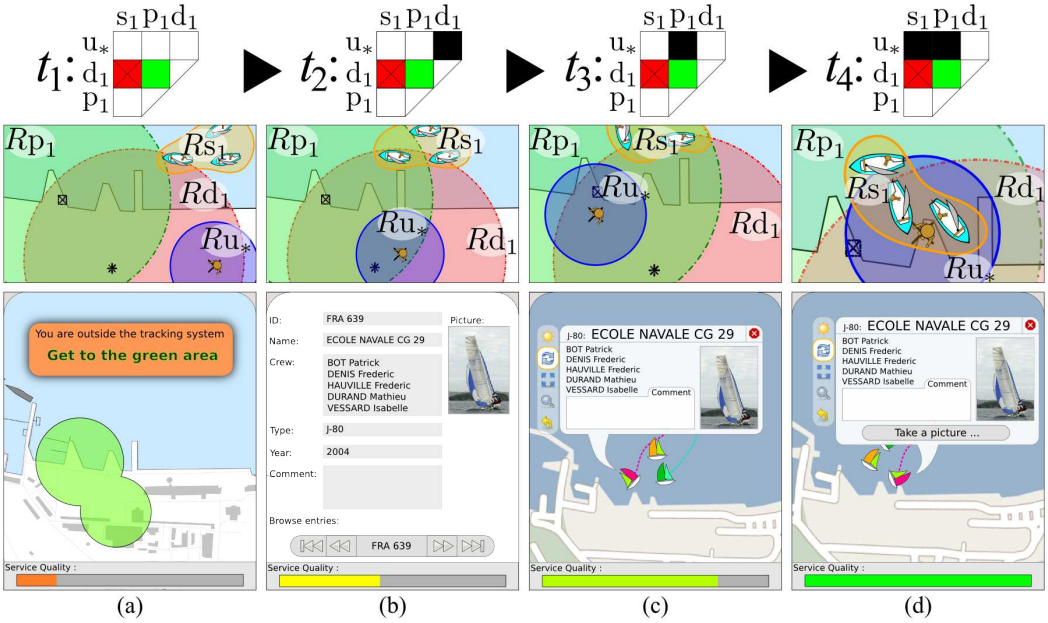


Figure VI.8 – Comportement du système modifié selon le contexte d'exécution rencontré

À l'instant t_1, l'utilisateur est seul et le système, dans ce cas réduit à la plate-forme cliente, présente une carte des zones où les services de suivi sont disponibles (tâche “présenter un plan du système”). À t_2, l'utilisateur s'est rendu dans la zone de diffusion de données et sa région d'exécution $R\mathrm{u}_*$ est liée à la région des données $R\mathrm{d}_1$. À cet instant, l'utilisateur accède à une lecture séquentielle des données statiques des bateaux (c-à-d. aux tâches de l'arbre situé, depuis “détailler les données d'un bateau” jusqu'à “présenter les meta-données”). À t_3, $R\mathrm{u}_*$ et $R\mathrm{p}_1$ sont en relation et l'utilisateur accède aux traitements dispensés par le serveur $C\mathrm{p}_1$. L'information géographique de positionnement de bateau est rendue en temps réel sur une carte (tâches de la branche “suivre la régate”, hormis la tâche “prendre et partager une image”). L'accès aux détails des bateaux n'est plus séquentiel mais direct (e.g., clic sur l'icône d'un bateau). Enfin, à t_4, l'utilisateur est entré dans la zone de course et les régions d'exécution $R\mathrm{u}_*$ et d'intérêt $R\mathrm{s}_1$ sont liées. Le système considère que cet utilisateur est à proximité de la zone de provenance des données et lui permet de prendre et de partager une photo pour le bateau sélectionné (tâche “prendre et partager une image”). À chaque entrée dans un nouveau contexte d'exécution, une barre de progression en bas de l'interface indique à l'utilisateur quel est le niveau fonctionnel du système. Elle représente un indicateur de *qualité de service*, mesurant la proportion de l'arbre des tâches réalisable dans le contexte d'exécution courant (Fig. B.1). Entre t_1 et t_4, l'utilisateur passe progressivement d'un contexte d'exécution

proposant peu de fonctionnalités et qui permet une atteinte très partielle des objectifs fixés par le scénario nominal, à un accès plus complet au système et aux services de suivi en temps-réel.

VI.2 Cas d'étude : système d'information de campus

Le cas d'étude du système d'information de campus propose une illustration des méthodes de personnalisation de systèmes mobiles et distribués par la construction de recommandations. Ce système a été développé d'après le modèle de conception étendu et son implémentation permet de distinguer un ensemble de contextes d'exécution au niveau des plates-formes des utilisateurs.

Les techniques de construction de profil et de personnalisation du contenu et de l'interface présentés sont mises en application par-dessus cet ensemble de contextes d'exécution. Pour chaque contexte donné, un utilisateur produit un profil personnel qui résume ses préférences par rapport au contenu présenté. Si à un moment donné plusieurs utilisateurs se retrouvent dans un même contexte, leurs profils sont combinés et des recommandations dites "de groupe" sont produites au sein d'un profil commun. Les profils produits, soit en groupes soit individuellement, sont appliqués pour adapter le contenu d'un assistant numérique de visite de campus universitaire.

VI.2.1 Contexte du système d'information campus

Le système d'information campus propose à des étudiants de découvrir un campus universitaire avec l'assistance d'un système d'information géographique pour le guidage et la diffusion d'informations à propos des installations de leur entourage. Le système d'information est conçu selon le modèle de distribution des régions d'exécution et certaines zones du campus proposent un accès à un service de documentation sur les bâtiments couverts. Ces aires de service permettent aux clients mobiles de recevoir des informations appropriées selon la localisation des visiteurs, matérialisée par l'aire de service à laquelle ils accèdent. Quand un utilisateur est proche d'un bâtiment particulier, il peut utiliser le système d'information pour se documenter sur l'historique et les fonctions du bâtiment, pour connaître les salles de cours, les départements ou les laboratoires qui composent le bâtiment, ou encore pour identifier sa position par rapport aux accès à ce bâtiment ou aux bâtiments affiliés.

La maquette de l'interface utilisateur du client, proposée en figure VI.9, divise la présentation des informations en deux niveaux :

- le volet des informations géographiques compose la partie gauche de l'interface. Ce volet présente un plan du campus et localise les différents bâtiments à propos desquels l'utilisateur peut se documenter dans un contexte d'exécution donné. La position de l'utilisateur par rapport aux infrastructures du campus est repérée par un marqueur rouge au niveau de la carte numérique. Les actions usuelles de cartographique numérique (zoomer, orienter, placer, sélectionner) sont proposées sous la forme d'un menu de boutons modaux à l'extrême gauche de l'interface ;

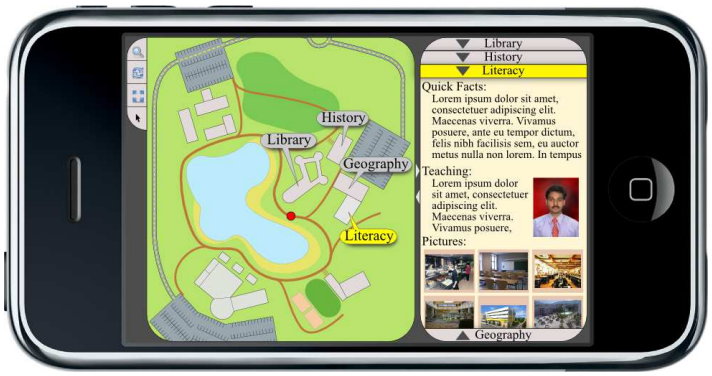


Figure VI.9 – Interface utilisateur du système d'information campus

- Le volet des informations textuelles, sur la droite de l'interface, organise un ensemble de fiches d'identité des bâtiments et les présente à l'utilisateur sous la forme d'une colonne d'onglets. Chaque onglet détaille un élément du plan numérique et fournit les fonctions et des caractéristiques des bâtiments désignés. Une page de contenu HTML, mêlant texte et images miniatures propres à chaque bâtiment, compose la fiche d'information affichée dans les onglets.

Les utilisateurs du système peuvent sélectionner les éléments du plan numérique et obtenir des informations dans le volet de droite à propos du bâtiment ou du département désigné. De la même manière, une action de sélection d'un onglet dans le volet des informations provoque l'affichage de la fiche d'information du département désigné et met à jour la sélection de ce bâtiment au niveau de la carte numérique. Dans l'exemple de la figure VI.9, l'utilisateur a sélectionné l'élément "Literacy". Cette sélection est marquée en jaune sur la carte et l'onglet affiché détaille les informations disponibles à propos du département d'enseignement littéraire.

VI.2.2 Architecture distribuée au sein du campus

Le système d'information campus est déployé sur un ensemble de plates-formes de traitements qui génèrent les éléments présentés à l'utilisateur. Lors de la conception, différents rôles ont été attribués aux composants du système et se retrouvent à l'exécution dans les contextes mesurés par les utilisateurs :

- "p", désigne un rôle de traitement sur des données brutes pour produire les fiches d'information des bâtiments ;
- "d1", "d2" et "d3" désignent les rôles de stockage d'information brute sur les bâtiments des départements à composantes respectivement littéraires, administratives et scientifiques ;
- le rôle "u" identifie les utilisateurs qui accèdent au système.

Les rôles "d1", "d2", "d3" et "p" sont répartis sur six composants. Plus spécifiquement, serveurs de traitement et de stockage sont associés par paires et répartis à trois emplacements du campus, au sein de l'espace désigné par les données stockées. Par exemple, les composants $C\mathrm{d}1_1$ et $C\mathrm{p}_1$ sont localisés au sein de l'espace physique occupé par les bâtiments des départements littéraires, et les données brutes enregistrées et gérées par le composant $C\mathrm{d}1_1$ concernent ces départements.

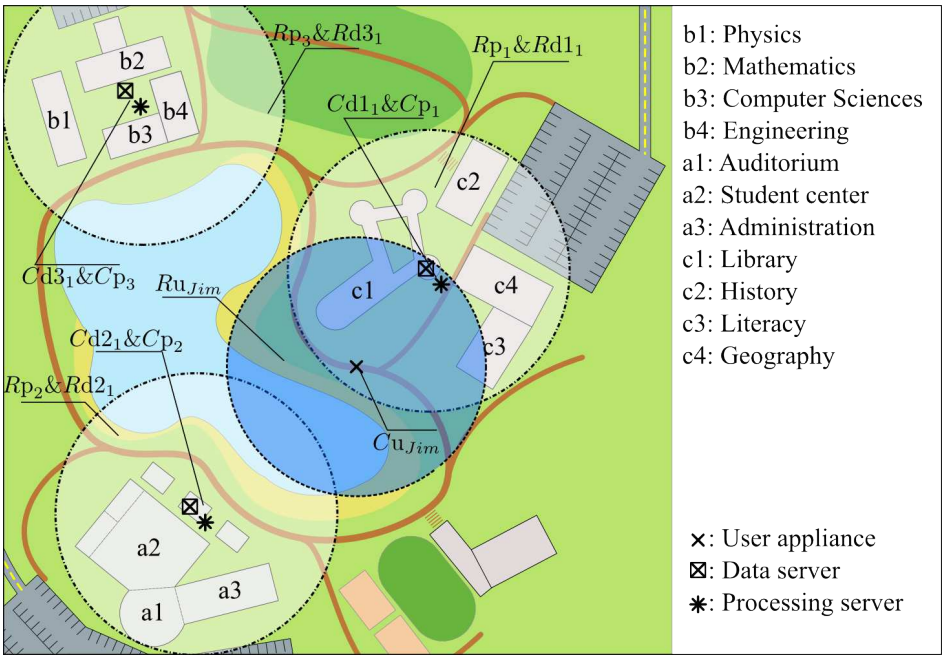


Figure VI.10 – Exemple de répartition des composants et des régions d'exécution du système d'information campus

Chaque paire de composants offre un point d'accès sans fil à un emplacement donné et est capable de diffuser localement (c-à-d. aux alentours d'un ensemble de bâtiments) des informations locales (c-à-d. des données qui concernent ces bâtiments) sous la forme de fiches de renseignements présentées aux utilisateurs du système. Comme un même point d'accès est utilisé pour chaque paire de composants, les régions d'exécution générées par ces composants sont confondues à tout instant. Les aires de ces régions sont établies à partir des capacités de transmission sans fil des points d'accès et forment six régions d'exécution confondues en paires. Les régions $R\mathrm{d}1_1$ et $R\mathrm{p}_1$, $R\mathrm{d}2_1$ et $R\mathrm{p}_2$, $R\mathrm{d}3_1$ et $R\mathrm{p}_3$ représentent les aires de diffusion des services d'information portant sur des départements littéraires, administratifs et scientifiques.

Les utilisateurs du système adoptent le rôle "u" et sont mobiles sur le campus. Leurs plates-formes clientes génériques $C\mathrm{u}_*$ déterminent un ensemble de régions d'exécution $R\mathrm{u}_*$ des utilisateurs. Dans l'exemple proposé en figure VI.10, Jim, un utilisateur du système, est identifié par sa plate-forme cliente $C\mathrm{u}_{Jim}$ qui génère une région d'exécution $R\mathrm{u}_{Jim}$ centrée sur cet utilisateur.

Mis à part les clients des utilisateurs, les autres composants du système qui diffusent leurs informations depuis l'un des trois points d'accès, demeurent fixes à l'exécution et les régions générées par deux composants qui diffusent depuis deux points d'accès différents ne sont jamais en relation. Inversement, les régions générées par des composants diffusant depuis un même point d'accès sont systématiquement en relation à l'exécution. Ces contraintes de fixité des composants et de combinaison des aires de diffusion impliquent une importante limitation du nombre de contextes d'exécution mesurables, et seuls quatre contextes peuvent être distingués (Fig. VI.11).

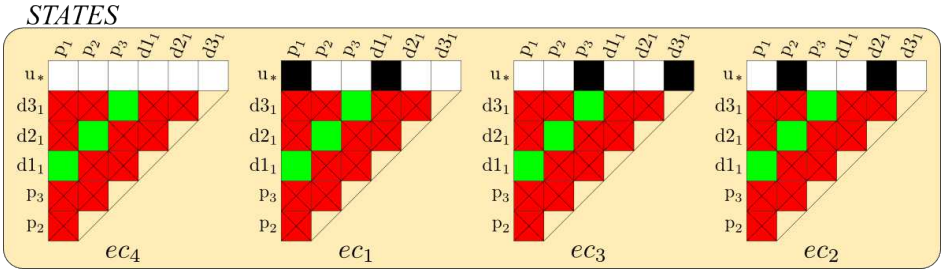


Figure VI.11 – Contextes d'exécution mesurables dans le cas du système de visite de campus

Les contextes ec_1, ec_2 et ec_3 représentent les situations dans lesquelles un utilisateur accède au système par l'intermédiaire de l'un des points d'accès sans fil. Le contexte ec_4 mesure une situation d'exécution où l'utilisateur est "seul", en dehors des régions de diffusion des points d'accès.

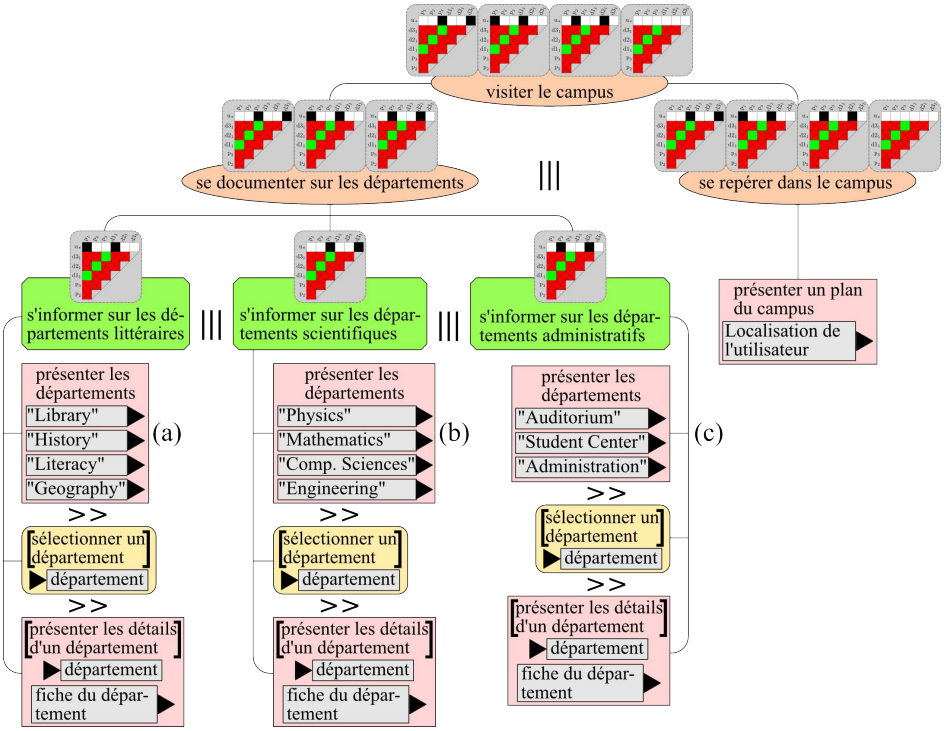


Figure VI.12 – Arbre des tâches situé du système de visite de campus

D'après le modèle de conception étendu, un comportement approprié du système est modélisé sous la forme d'une organisation de tâches pour chaque contexte mesurable (Fig. VI.12). Dans le cas du système de visite de campus, les utilisateurs mesurant un contexte ec_1 ou ec_2 peuvent effectuer les sous-arbres "s'informer sur les départements littéraires" ou "s'informer sur les départements administratifs" qui décomposent une tâche globale "se documenter sur les départements". Lorsqu'un utilisateur se retrouve isolé du reste du système, seule la tâche "se repérer dans le campus" est proposée en compensation de la perte des sources d'information sur les départements. Dans ce contexte, le composant client est chargé de traiter l'affichage et l'interaction avec la carte du campus, comme aucun autre composant

de traitement n'est disponible. L'ensemble des tâches décrites est implémenté au niveau du client et l'interface d'utilisation présente un contenu adapté à la situation rencontrée (Fig. VI.13).

(a) Contexte ec_1 : présentation d'informations sur les départements "littéraires"

(b) Contexte ec_4 : localisation de l'utilisateur dans l'espace du campus

(c) Contexte ec_2 : présentation d'informations sur les départements "Administratifs"

Figure VI.13 – Exemples de variations du contenu et de l'interface présentés en fonction du contexte d'exécution.

VI.2.3 Perspective multi-utilisateurs et adaptations

Le contenu présenté dans les différents contextes ec_1, ec_2 et ec_3 mobilise en sortie des branches de l'arbre des tâches des informations décrivant des sous-ensembles des bâtiments du campus. Chaque tâche système "présenter les départements" sollicite un ensemble important d'informations sous la forme de séries de noms de départements. L'implémentation de chacune de ces tâches provoque l'affichage de trois à quatre éléments au niveau des clients (Fig. VI.12(a, b et c))). Dans l'exemple de la figure VI.9, les quatre départements détaillés constituent des éléments produits en sortie par la sous-tâche "présenter les départements littéraires". D'après le chapitre V, ces sorties peuvent être ordonnées suivant l'intérêt d'un utilisateur et présentées par ordre de préférence. Cette approche est adoptée dans le système de visite de campus. Des scores sont distribués entre chaque élément des listes de départements présentés pour constituer des profils personnels et de groupes.

Il est admis dans la suite que trois utilisateurs, ci-après nommés Jim, Joe et $Jack$, accèdent au système de visite de campus. Ces utilisateurs sont représentés par leurs composants clients $C\mathrm{u}_{Jim}$, $C\mathrm{u}_{Joe}$ et $C\mathrm{u}_{Jack}$ (Fig. VI.14). Entre t_0 et t_5, ils effectuent un parcours qui les mène dans le périmètre du service d'information portant sur les départements littéraires. Les utilisateurs du système de documentation mesurent soit un contexte d'exécution local ec_4 lorsqu'ils restent en dehors de l'espace du service, soit un contexte ec_1 lorsque leur région d'exécution $R\mathrm{u}_{[Jim,\ Joe \text{ ou } Jack]}$ est en relation avec la région de diffusion de service $R\mathrm{p}_1 \& R\mathrm{d}_1$. Lors de leurs parcours, les localisations de ces utilisateurs, leurs pauses et leurs interactions avec les informations présentées sont monitorées et utilisées pour construire des profils personnels. À plus haut niveau, lorsque ces utilisateurs mesurent un même contexte local, les profils personnels sont combinés pour constituer un profil de groupe. Ces deux types de profils, personnels ou de groupe, servent ensuite à adapter la présentation des données au niveau des clients.

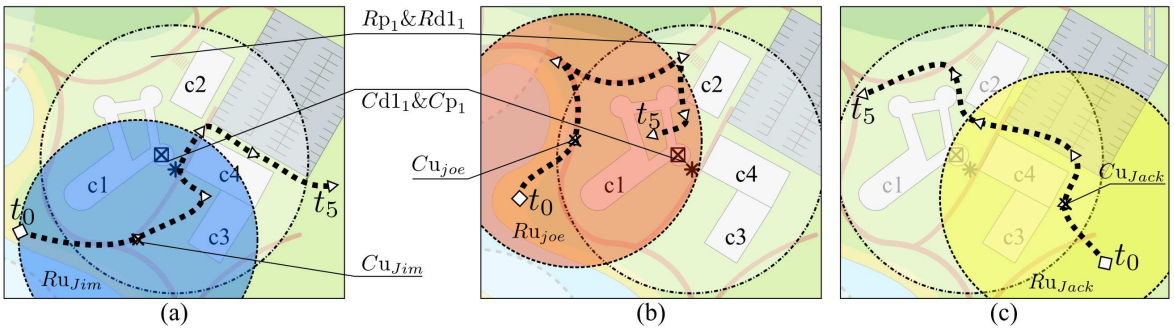


Figure VI.14 – Instantané à t_1 d'une visite effectuée par les trois utilisateurs entre t_0 et t_5

Scores et profils personnels

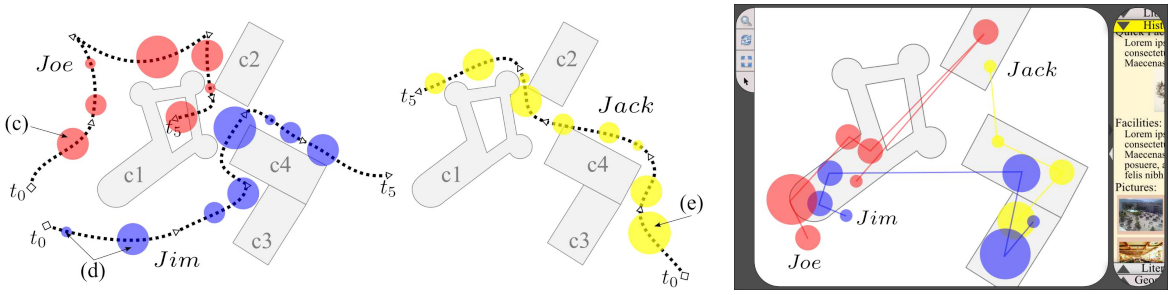
Les éléments présentés sont disponibles dans un contexte d'exécution donné et des scores associés à ces éléments peuvent êtres construits en appliquant l'approche mixte proposée au chapitre V.2. La partie interaction du score d'intérêt d'un élément est construite en enregistrant les sélections et les actions cartographiques. La partie environnementale du score d'intérêt d'un élément est calculée à partir de la proximité physique de l'utilisateur aux différents bâtiments de l'environnement et de la durée de ses sessions d'utilisation du système dans ses déplacements. Plus spécifiquement, les règles suivantes sont appliquées :

- plus la représentation numérique d'un département est la cible des interactions de l'utilisateur, plus le score de ce département est élevé. Parmi les interactions mesurées figurent les sélections sur la carte numérique, les sélections dans le volet des informations textuelles, et l'utilisation des outils de zoom et de centrage ;
- plus les bâtiments d'un département sont géographiquement proches de l'utilisateur, plus le score d'environnement de ce département est élevé. La mesure de distance nécessite de connaître la localisation de l'utilisateur en temps réel. Dans ce cas d'étude, les déplacements des utilisateurs sont simulés par l'artifice d'une interface de saisie des parcours des utilisateurs. En pratique, des clients mobiles en extérieur disposeraient de moyens de géolocalisation (p.ex. GPS intégré, algorithme de triangulation cellulaire ou Wi-Fi) ;
- plus un utilisateur passe de temps à l'utilisation du système à proximité des bâtiments d'un département, plus ce département obtient un score d'environnement élevé. Le temps d'utilisation est mesuré lors d'une pause entre deux déplacements. Une pause entre deux instants d'exécution a lieu lorsque la position de l'utilisateur est invariante sur l'intervalle de temps entre ces deux instants [6].

Dans le cas d'étude proposé, les utilisateurs effectuent des pauses durant leurs parcours de visite (ces pauses peuvent indiquer les endroits où les utilisateurs interagissent avec le service). Les instants et lieux d'arrêt des différents utilisateurs sont représentés en figure

6. Dans le cas d'un parcours simulé, la position de l'utilisateur ne varie pas lors d'une pause. En pratique, une position est considérée invariante entre t_x et t_{x+1} lorsque chaque mesure de localisation effectuée au cours de $[t_x, t_{x+1}]$ appartient à la couverture d'une région circulaire centrée sur la position mesurée moyenne. Le rayon de cette aire correspond au seuil de tolérance en deçà duquel il n'est pas possible de déterminer si un écart entre deux mesures est causé par un déplacement de l'utilisateur ou par un aléa de perception. Par exemple, la précision des mesures de localisation par GPS implique un seuil de tolérance d'une dizaine de mètres autour de la position moyenne.

VI.15a par des cercles bleus pour les pauses de Jim, des cercles rouges pour les pauses de Joe et des cercles jaunes pour les pauses de Jack. Leurs centres désignent les localisations des pauses et leurs rayons quantifient la durée de ces pauses. Pour simplifier la lecture des durées, quatre rayons représentant des pauses de dix à quarante secondes sont discriminés. Les bornes triangulaires le long des parcours rappellent les instants d'exécution t_0 à t_5 auxquels ont lieu les calculs de profils et les éventuelles adaptations. Dans ce cas, entre t_0 et t_1, Joe effectue une pause d'environ trente secondes à l'ouest du bâtiment de la bibliothèque (Fig. VI.15(c)). Lors du même intervalle temporel, Jack s'immobilise une quarantaine de secondes à l'est du bâtiment c3 (Fig. VI.15(e)). Jim, quant à lui, s'arrête une première fois pour une courte durée (c-à-d. environ 10s) puis plus longuement au sud de l'élément c1 (Fig. VI.15(d)).



(a) Parcours et pauses des utilisateurs entre t_0 et t_5

(b) Interactions des utilisateurs entre t_0 et t_1

Figure VI.15 – Primitives captées pour la construction des scores personnels

Les mesures d'interactions de l'utilisateur constituent la deuxième source de données pour construire les scores des éléments. Ces mesures sont déduites de l'enregistrement implicite des actions des utilisateurs sur leurs clients au cours du dernier intervalle d'exécution. La figure VI.15b présente quelles ont été les positions des interactions de chacun des utilisateurs dans l'intervalle $[t_0, t_1]$. Les chemins bleu, rouge et jaune correspondent respectivement aux interactions de Jim, Joe et $Jack$. Dans cette vue, les centres des cercles désignent les endroits des clics des utilisateurs et les rayons des cercles permettent de visualiser le temps d'interaction passé avant que l'utilisateur ne pointe à un autre endroit. Par exemple, entre t_0 et t_1, l'essentiel des interactions de Joe se concentre sur l'élément représentant la bibliothèque à l'écran. Dans le même intervalle, Jim partage son attention entre les représentations numériques de la bibliothèque et du département des lettres. $Jack$ reste principalement focalisé sur les représentations des bâtiments c3 et c4.

Les scores de localisation et d'interaction calculés à t_1 quantifient les intérêts présumés des utilisateurs à un moment donné au moyen de mesures de proximité entre les lieux des actions géographiques ou interactionnelles et les éléments d'un contexte d'exécution (Tab. VI.3a & VI.3b). Par exemple, la pause de $Jack$ entre t_0 et t_1 permet d'attribuer aux bâtiments proches, notamment c3 et c4, des scores d'environnement supérieurs. En proportion, ces éléments représentent plus de 70% du score d'environnement et de l'intérêt supposé de $Jack$ à t_1. De la même manière, les scores d'environnement de Jim et Joe favorisent l'élément c1, proche d'eux entre t_0 et t_1. Les scores d'interaction classent en premier les éléments qui ont été beaucoup sélectionnés par les utilisateurs. Par exemple, les interactions de $Jack$ ont tendance à corroborer son score d'environnement. Pour cet utilisateur, les scores des éléments c3 et c4 constituent trois quarts de ses notes d'interaction. Les sélections de Jim favorisent ces mêmes

éléments à t_1, mais contrairement à $Jack$, les scores d'interaction de Jim ne suivent pas les scores d'environnement. Cet utilisateur est à t_1 proche de l'élément c1, mais se documente plutôt sur les éléments c4 et c3.

Tableau VI.3 – Scores des éléments "c1" à "c4" calculés à t_1.

(a) Scores d'interaction

elem	*c*1	*c*2	*c*3	*c*4
$UIScore$ $(elem, Cu_{Jim}, t_1)$	.19 16%	.07 6%	.51 42%	.45 36%
$UIScore$ $(elem, Cu_{Joe}, t_1)$	.42 40%	.46 44%	.08 8%	.08 8%
$UIScore$ $(elem, Cu_{Jack}, t_1)$	.06 10%	.09 15%	.22 39%	.21 36%

(b) Scores d'environnement

elem	*c*1	*c*2	*c*3	*c*4
$LocScore$ $(elem, Cu_{Jim}, t_1)$	.53 42%	.20 15%	.29 23%	.25 20%
$LocScore$ $(elem, Cu_{Joe}, t_1)$	.27 47%	.11 18%	.10 17%	.11 18%
$LocScore$ $(elem, Cu_{Jack}, t_1)$	.22 14%	.23 14%	.65 41%	.49 31%

(c) Scores combinés et ordonnés (1er, 2ème, 3ème et 4ème)

elem	*c*1	*c*2	*c*3	*c*4
$Score(elem, Cu_{Jim}, t_1)$	28.9%	10.7%	32.3%	28.1%
$Score(elem, Cu_{Joe}, t_1)$	43.3%	31.5%	12.3%	12.8%
$Score(elem, Cu_{Jack}, t_1)$	11.7%	14.9%	39.8%	33.6%

Les scores combinés proposent une mesure unifiée des scores d'interaction et d'environnement. Dans ce cas d'étude, la combinaison des scores est équilibrée et les scores donnés dans le tableau VI.3c sont une moyenne entre les scores d'interaction et d'environnement pour chaque élément et utilisateur. Cette composition du score combiné semble valable lorsqu'il n'y a pas d'opposition entre les intérêts présumés d'après les approches géographique et d'interaction. Dans le cas de Joe à t_1 par contre, les différences entre les scores d'environnement et d'interaction sont lissées par la moyenne lors du calcul des scores combinés. Ce lissage des scores n'est pas sans impact du point du vue de la compréhension de l'intérêt de l'utilisateur et de meilleures combinaisons des scores peuvent probablement être imaginées lors de la définition par les concepteurs de la fonction de balance (Chap. V.2.1). Notamment, en cas d'opposition des scores, la priorité peut être donnée à l'un ou l'autre des scores d'interaction ou d'environnement. L'étude des compositions de scores combinés à partir de sources de mesures hétérogènes demeure un sujet de recherche ouvert dépassant largement le cadre de ce mémoire. Pour les systèmes mobiles et plus particulièrement dans le cas du système de visite de campus, cet aspect du travail de recherche est mené par l'équipe d'informatique de l'University College Dublin [7].

Scores et profils de groupe

Au cours de l'exécution, les scores obtenus pour des utilisateurs partageant un même contexte local peuvent être combinés pour former des scores de groupe. Dans le système de visite de campus, Jim, Jack et Joe se retrouvent régulièrement dans le groupe des utilisateurs qui mesurent le contexte d'exécution local égal à ec_1, et à tout instant t_x au moins deux utilisateurs accèdent au service :

7. http ://www.csi.ucd.ie/

- aux instants t_1, t_3 et t_4, tous trois ont accès au service et attribuent des scores aux éléments c1 à c4 ;
- à t_2, Joe est sorti de l'aire de service et seuls Jim et Jack contribuent aux scores de groupe ;
- à t_5, Jim quitte à son tour l'aire du service et les scores de groupe sont constitués par Joe et Jack.

Deux niveaux de scores multi-utilisateurs sont distingués. D'une part, les valeurs moyennes sont calculées à un instant t_x donné à partir des scores personnels des utilisateurs du groupe à cet instant. D'autre part, les scores de groupe à un instant donné sont constitués par des mesures moyennes des scores individuels auxquelles s'ajoutent une mesure des scores de groupe aux instants précédents. Différentes fonctions de pondération ont été introduites au chapitre précédent et permettent un contrôle fin dans la manière de combiner les scores des différents utilisateurs.

Dans l'exemple du système de visite de campus, ces fonctions $\omega_{usr}(\ldots)$, $\omega_{bal}(\ldots)$, et $\omega_{hist}(\ldots)$ conservent la même définition qu'au chapitre V.2. Dans ce cas :

- les scores d'un utilisateur ayant passé du temps dans un contexte donné seront favorisés lors du calcul des scores moyens instantanés ;
- les scores moyens instantanés et les valeurs historiques contribuent à parts égales au calcul du score de groupe ;
- lors du calcul des scores de groupe à un instant donné, seules deux générations d'historiques sont prises en compte.

Les scores d'intérêt calculés au niveau personnel, puis au niveau d'un regroupement d'utilisateurs, permettent d'adapter le contenu proposé entre chaque étape de l'exécution. La variation des scores entre deux instants va décider de l'importance des modifications d'apparence des informations présentées sur le client. Dans le cas du système de visite, plus les variations des scores sont importantes plus les changements de présentation entre deux instants seront brusques. Dans ce cas, un score calculé entre plusieurs utilisateurs doit répondre à plusieurs contraintes :

- les distances séparant les scores des utilisateurs et les scores du groupement auquel ils contribuent doivent être acceptables pour une majorité du groupe. Autrement dit, un score de groupe calcule un consensus entre les scores des utilisateurs ;
- dans l'optique d'une adaptation de la présentation des éléments, la continuité entre deux scores successifs doit être maximisée pour éviter les ruptures brusques de l'interaction et de l'interface sur le client.

Dans la suite, les variations entre scores multi-utilisateurs sont comparées de manière statistique. Les écarts entre scores de groupe et valeurs moyennes instantanées illustrent l'intérêt de l'intégration des mesures historiques pour assurer une meilleure continuité des scores et profils. Deux interprétations des scores sont proposées et seront utilisées pour adapter la présentation des éléments : 1) les valeurs des scores des éléments sont directement corrélées à leurs dessins sur la carte numérique ; 2) l'ordre des éléments entre eux permet de classer la liste des fiches de renseignements sur les départements. Pour vérifier la compatibilité des scores de groupe avec la contrainte de continuité, les scores des éléments sont comparés par valeur et par rang.

Les tableaux VI.4a et VI.4b résument respectivement les valeurs moyennes instantanées et les scores de groupe de chaque élément disponible dans le contexte ec_1 aux différents instants de l'exécution. Ces deux approches des scores multi-utilisateurs se différencient essen-

Tableau VI.4 – Scores par groupe des éléments “c1” à “c4” calculés à chaque instant entre t_0 et t_5 (ordre des éléments par scores décroissants : 1er, 2ème, 3ème et 4ème)

(a) Valeurs moyennes instantanées (en %)

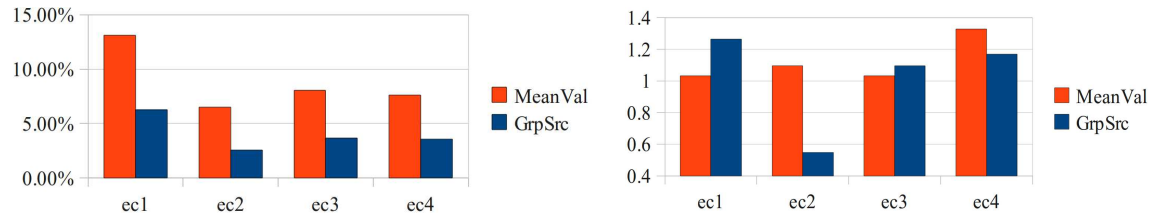
x	0	1	2	3	4	5
$MeanVal$ (c1,ec_1,t_x)	25	28	15	21	24	53
$MeanVal$ (c2,ec_1,t_x)	25	19	12	29	29	19
$MeanVal$ (c3,ec_1,t_x)	25	28	35	19	17	13
$MeanVal$ (c4,ec_1,t_x)	25	25	38	31	30	15

(b) Scores de groupe (en %)

x	0	1	2	3	4	5
$GrpScore$ (c1,ec_1,t_x)	25	26	20	22	23	38
$GrpScore$ (c2,ec_1,t_x)	25	23	18	23	25	21
$GrpScore$ (c3,ec_1,t_x)	25	26	30	25	23	19
$GrpScore$ (c4,ec_1,t_x)	25	25	32	30	29	22

tiellement du point de vue de la volatilité des scores et des rangs des éléments d'un instant au suivant. Notamment, du fait de l'intégration des scores aux instants d'exécution précédents, les scores de groupe varient moins rapidement que les scores moyens instantanés. D'un point de vue quantitatif, l'écart-type des scores des éléments sur l'intervalle $[t_0, t_5]$ fournit un ordre de grandeur de la variation moyenne entre deux instants (Fig. VI.16a). Par exemple, entre deux instants de l'intervalle d'exécution, le score moyen instantané de l'élément c1 varie de près de 15% alors que le score de groupe de ce même élément varie deux fois moins (c-à-d. en moyenne 6% de variation d'un instant à un autre). Tous éléments confondus, les écarts de scores entre deux instants sont deux fois moindres dans le cas des scores de groupe et quel que soit l'élément considéré, la variation des scores qu'il obtient par le calcul de groupe est toujours moindre que dans le cas de la valeur moyenne instantanée.

Les résultats de l'analyse des variations des rangs sont plus nuancés. En moyenne, entre deux instants, l'ordre des éléments varie légèrement moins dans le cas des scores de groupe (i.e. variation d'un rang pour les scores de groupe et de 1.2 rangs pour les valeurs moyennes). Toutefois, dans le détail, les éléments c3 et surtout c1 changent de rang avec plus d'amplitude dans les classements des scores de groupe. Ces variations sont compensées par le très faible écart de rang de l'élément c2 qui ne change qu'une fois de rang au cours de l'intervalle d'exécution et passe de dernier à avant-dernier entre les instants t_2 et t_3 (Fi. VI.16b).

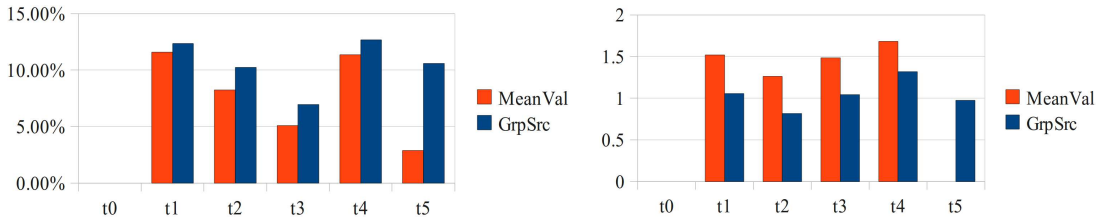


(a) Écarts-types des scores d'éléments sur t_0, t_5 (b) Écarts-types des rangs des éléments sur t_0, t_5

Figure VI.16 – Comparaison des variations des scores de groupe et des valeurs moyennes instantanées

À la fois dans les comparaisons par rang et dans les comparaisons par valeur, les scores de groupe demeurent plus réguliers que les valeurs moyennes instantanées. Les profils générés à partir de ces scores respectent mieux la contrainte de continuité. Pour respecter la contrainte de consensus entre les utilisateurs, la divergence entre ces scores et les scores personnels doit rester limitée.

Dans la suite, les valeurs moyennes instantanées sont considérées comme une "référence" établissant le consensus des intérêts des utilisateurs. L'écart-type de ces valeurs par rapport aux scores personnels est comparé à l'écart-type entre scores de groupe et scores personnels. Lorsqu'à un instant donné, ces écarts ont même ordre de grandeur, les scores de groupe et les valeurs moyennes instantanées respectent la contrainte de consensus avec une efficacité semblable. Lorsque l'écart entre les scores de groupe et les scores personnels sont inférieurs, les profils de groupe représentent un meilleur consensus que les profils composés des valeurs moyennes instantanées.



(a) Écarts des profils par comparaison des valeurs

(b) Écarts des profils par comparaison des rangs

Figure VI.17 – Comparaison des variations entre les scores personnels et les scores multi-utilisateurs

La figure VI.17 représente, lors de chaque instant de l'exécution, ces écarts entre une approche personnelle des scores d'intérêt et les deux approches de combinaisons entre plusieurs utilisateurs, par valeur moyenne puis par score de groupe. Comme précédemment, les lectures par valeur et par rang sont différenciées. Dans le cas des écarts par valeur, les scores personnels des éléments sont en moyenne plus proches des valeurs instantanées que des scores de groupe. A l'exclusion de t_5, ces différences restent toutefois minimes et les scores de groupe ne sont pas franchement plus éloignés des scores personnels que les valeurs moyennes instantanées (Fig. VI.17a). Plus précisément, sur l'intervalle $[t_0, t_5]$, les valeurs de scores de groupe des éléments divergent en moyenne de plus ou moins 10 points par rapport aux scores personnels contre 8.5 points pour les valeurs moyennes. Du point de vue des variations d'ordre des éléments, et toujours à l'exclusion de t_5, les résultats obtenus par l'application des scores de groupe sont encourageants. Sur l'intervalle d'exécution, l'adéquation entre l'ordre des scores personnels et les rangs des éléments dans les scores de groupe est meilleure (Fig. VI.17b). En moyenne, les éléments ordonnés des profils de groupe sont à un rang d'écart par rapport à leur place dans les profils personnels ; tandis que l'ordre issu du classement des valeurs moyennes instantanées diverge plus fortement (environ 1.3 rangs d'écart).

Le tableau VI.5 propose une synthèse des résultats obtenus par analyse statistique. Les scores de groupe sont moins volatiles que les valeurs moyennes instantanées quelle que soit l'interprétation des scores (c-à-d. par valeur ou par rang). Par rapport aux classements des valeurs instantanées, les rangs des scores de groupe approchent davantage les classements déduits des scores personnels. Seules les comparaisons de valeurs donnent l'avantage aux moyennes instantanées pour répondre à la recherche de consensus entre les scores d'utilisateurs.

La généralisation de ces résultats reste cependant difficilement réalisable. Les variables d'entrée (c-à-d. les interactions et localisations des utilisateurs) sont nombreuses. La manière de les combiner pour construire des scores personnels, des valeurs moyennes instantanées et enfin des scores de groupe dépend d'un ensemble de fonctions de pondération. Leurs

Tableau VI.5 – Synthèse des performances comparées des scores de groupe par rapport aux valeurs moyennes instantanées

	Comparaison des valeurs	Comparaison des rangs
Contrainte de continuité	mieux	légèrement mieux
Contrainte de consensus	légèrement moins bon	mieux

définitions sont susceptibles de faire varier du tout au tout les scores de groupe obtenus. Dans le cas du système de visite de campus, il est vérifié que les scores de groupe offrent une solution efficace de représentation des intérêts des utilisateurs 1) avec les définitions choisies pour les fonctions de pondération, 2) avec les interactions et pauses telles que décrites en entrée, et 3) sur l'intervalle d'exécution $[t_0, t_5]$. Dans la suite, cette hypothèse statistique est illustrée qualitativement par l'adaptation de la présentation des éléments aux scores de groupe.

Prototypes et adaptations aux profils générés

Pour tester la génération des profils personnels et de groupe, une chaîne de simulation est implémentée (Fig. VI.18). Cet ensemble de composants simule les conditions d'exécution d'un système cible multi-contextes et multi-utilisateurs pour faciliter la mise au point du calcul des scores en évitant d'avoir à déployer l'application cible. Cette chaîne repose sur une classe de calcul, codée en Java, qui intègre les primitives d'interaction et les parcours dans l'environnement pour un ensemble d'utilisateurs. Cette classe génère les scores personnels, les scores instantanés moyens et les scores de groupe à partir de ces entrées(Fig. VI.18(d)). En début de chaîne, deux interfaces permettent la saisie interactive des scores d'interaction et d'environnement (Fig. VI.18(a & b)). Ces simulateurs sont paramétrés en entrée par une description du système à simuler. Les bornes des régions d'exécution et d'intérêt, les emplacements des composants, et les espaces accessibles, fournis aux interfaces de simulation, déterminent les différents contextes d'exécution au cours du temps. Cette description du système cible est constituée par un ensemble de fichiers de configuration. Une version plus avancée de la chaîne de simulation permettrait le design interactif du système cible préalablement à la saisie des interactions et des déplacements des utilisateurs.

Les données brutes d'interaction et d'environnement issues de l'utilisation des simulateurs constituent à la fois une sauvegarde de session de simulation et l'entrée fournie à la classe de calcul de scores (Fig. VI.18(c)). Cette classe abstraite implémente les méthodes servant au calcul des profils des utilisateurs. Les méthodes du calculateur prennent les mêmes paramètres en entrée et retournent les mêmes types de valeurs que leurs définitions fonctionnelles précisées au chapitre V.2 (Fig. VI.18(d)). Les concepteurs doivent en outre définir une classe concrète qui implémente les méthodes correspondant aux fonctions de pondération (c-à-d. les méthodes `wBal()`, `wUsr()`, `wScore()` et `wHist()`). En sortie de chaîne, le générateur d'interface regroupe les différentes solutions d'adaptation et permet de contrôler le résultat de la méthode de calcul de scores (Fig. VI.18(e)). La boucle de rétro-adaptation du calculateur vers le simulateur d'interaction met à jour la présentation des éléments et de l'interface (Fig. VI.18(f)). La saisie des actions des utilisateurs s'effectue systématiquement à partir d'une présentation adaptée de l'interface du client.

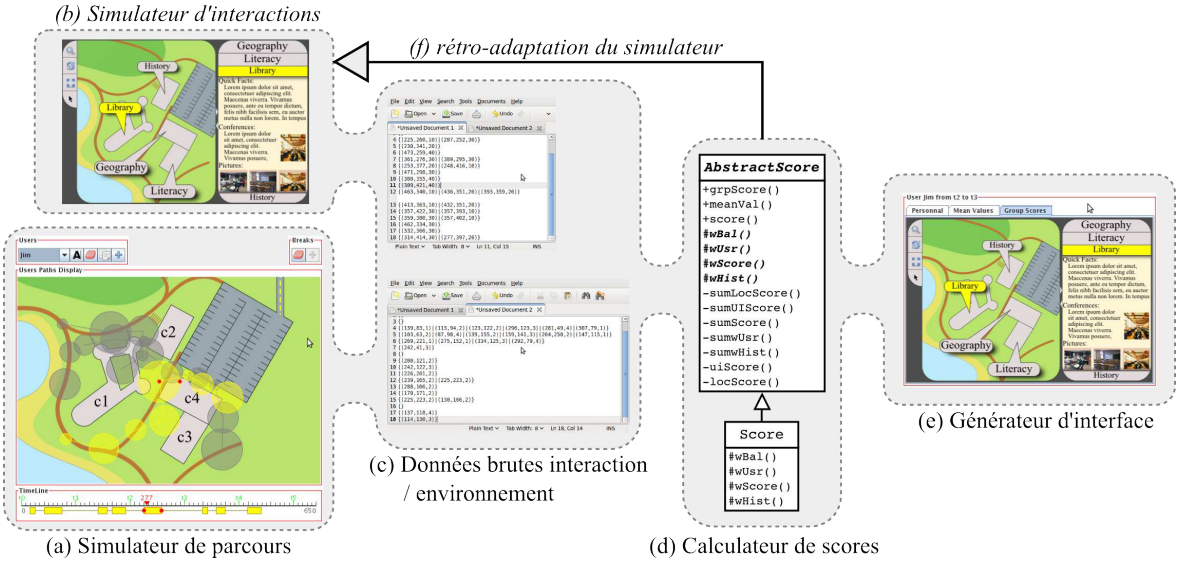


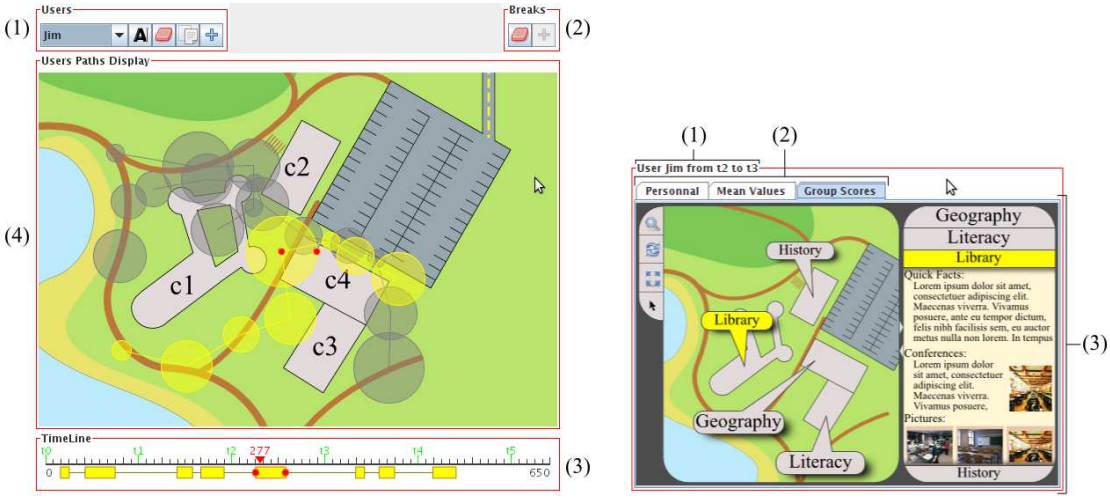
Figure VI.18 – Vue d'ensemble de la chaîne de génération de scores et d'adaptations

Le simulateur de parcours permet l'enregistrement et la modification des utilisateurs du système cible (Fig. VI.19a-1). Les parcours sont saisis en localisant les pauses des utilisateurs lors de l'exécution. Les éléments mesurables de l'environnement sont présentés au niveau de la carte numérique (Fig. VI.19a-4). Chaque cercle représente une pause de l'utilisateur (les cercles jaunes référencent les pauses de l'utilisateur courant). Leurs localisations et durées peuvent être ajustées directement sur la carte. La durée peut être réglée plus précisément en modifiant les bornes supérieure et inférieure des intervalles représentés en séquence temporelle (Fig. VI.19a-3). Les pauses et trajectoires représentées dans la figure VI.19a et saisies dans l'interface de simulation ont été présentées précédemment en exemple du calcul des scores et de l'analyse statistique des profils.

Au niveau de la séquence temporelle, le curseur rouge présente l'instant courant, exprimé en secondes depuis l'initialisation du système. Les graduations vertes rappellent les instants de l'exécution auxquels ont lieu les calculs de scores et les adaptations. Ces graduations sont réparties selon un pas fourni parmi les données de description du système cible. Dans le cas du système de visite de campus, une mise à niveau s'effectue toutes les deux minutes.

La sélection des utilisateurs et les actions sur la séquence temporelle se répercutent sur l'ensemble de la chaîne de simulation. En particulier, lors d'un changement d'utilisateur sélectionné ou lors du passage d'un instant d'exécution à un autre (graduations vertes), les données brutes sont régénérées et le calculateur de scores effectue une mise à niveau sur cette nouvelle base puis déclenche la génération des interfaces et la mise à jour du simulateur d'interactions.

Le générateur d'interface combine dans une même vue trois solutions d'adaptations (Fig. VI.19b-2). Ces solutions sont générées pour l'utilisateur et l'instant d'exécution sélectionnés dans le simulateur de parcours (Fig. VI.19b-1). Chacune d'elle correspond à une implémentation possible du client en se basant alternativement sur les scores personnels, sur les scores moyens instantanés, ou sur les scores de groupe. Plus spécifiquement, ces trois approches interrogent respectivement le calculateur de scores par l'appel des méthodes



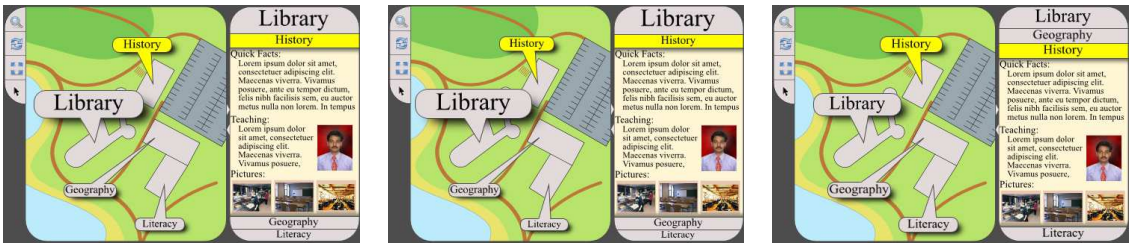
(a) Simulation des déplacements et des pauses des utilisateurs

(b) Interfaces proposées à l'utilisateur

Figure VI.19 – Développements pour les tests du système de navigation campus

publiques `scores()`, `meanval()` ou `grpScore()`. Ces méthodes retournent des scores qui paramètrent la présentation des éléments (Fig. VI.19b-3). Notamment, la taille des labels de la carte ou de la liste est directement indexée sur le score des éléments. L'ordre des fiches de renseignements dans le panel d'information est donné par le rang des scores d'éléments.

Le générateur d'interface permet de se rendre compte des variations induites par le calcul des profils et d'ajuster éventuellement les méthodes de pondération pour augmenter ou atténuer les effets d'une adaptation. Par exemple, les indicateurs statistiques ont montré que la variation de rangs entre les profils personnels et les profils moyens est nulle lors de l'adaptation à t_5 (Fig. VI.17b). De la même manière, les variations des scores sont relativement faibles lors de cette étape d'adaptation (c-à-d. environ 3 points de variation moyenne (Fig. VI.17a)). En termes de présentation, ces similarités de scores se traduisent par des interfaces générées qui demeurent relativement similaires. Uniquement quelques différences de taille de labels sont observables entre l'interface générée à partir des scores de l'utilisateur *Joe* (Fig. VI.20a) et l'interface générée à partir des scores moyens instantanés (Fig. VI.20b).



(a) Interface basée sur les scores personnels

(b) Interface basée sur les scores moyens instantanés

(c) Interface basée sur les scores de groupe

Figure VI.20 – Exemples d'interfaces générées à t_5

Les différences de présentation sont plus marquées entre ces deux interfaces et l'interface générée à partir des scores de groupe (Fig. VI.20c). Dans ce cas, l'impact des scores passés se traduit graphiquement par un amoindrissement de l'importance du label "Library" et un changement de l'ordre de l'élément "History" dans le panel d'information. Ces modifications corrèlent les indicateurs statistiques qui font état à l'étape t_5 d'une divergence plus importante entre les profils de groupe et profils personnels ou instantanés moyens.

VI.3 Conclusion

Deux cas d'étude ont été développés pour illustrer les principes présentés dans les chapitres de la contribution. Dans un premier exemple, l'analyse conceptuelle d'un système de suivi de compétition nautique illustre l'utilisation des outils de conception de systèmes mobiles et distribués présentés aux chapitres III et IV. L'analyse que mènerait l'équipe chargée de la conception et l'implémentation de ce service est détaillée. Les démarches d'analyse contextuelle par l'approche des régions d'intérêt et d'exécution ainsi que la construction des arbres de tâches annotés et situés sont mises en pratique dans l'intention d'augmenter les services du projet Navtrack par un nouvel outil de suivi en temps réel. Cet exemple d'utilisation de la méthodologie proposée par le cadre de conception étendu s'avère efficace pour cadrer les cas d'utilisation du système de suivi, et pour déterminer d'éventuelles alternatives à un fonctionnement nominal.

Le second exemple met en œuvre un système de recommandation de contenu dans le cas d'un service de visite de campus. Les utilisateurs, équipés de clients mobiles légers, peuvent circuler entre les départements du campus et recevoir des informations selon les contextes d'exécution qu'ils mesurent à un instant donné. Le système de recommandation est basé sur la méthode de calcul de profils définie au chapitre V.2. Il distingue différents niveaux d'adaptation, selon que les profils appliqués sont générés par les utilisateurs seuls ou en groupe. Une analyse statistique sur l'exemple proposé montre qu'avec la définition donnée aux fonctions de pondération, les profils de groupe construits disposent de bonnes capacités de représentation des préférences des utilisateurs, et assurent une certaine continuité historique entre deux étapes d'adaptation. L'implémentation d'une chaîne de simulation permet de valider simplement les paramètres de calcul des scores et de contrôler les adaptations et les différences entre les interfaces clientes générées.

Conclusion

Cette thèse étudie les nouveaux enjeux de l'informatique mobile et propose un cadre de réflexion pour l'analyse et la conception de systèmes distribués qui adaptent leur comportement à l'exécution. En complément des avancées technologiques observées dans les domaines des télécommunications, des réseaux et de la géolocalisation en temps-réel, l'approche adoptée dans cette thèse concerne la mise en oeuvre de systèmes d'information mobiles et s'interroge sur les impacts de ces nouvelles technologies pour l'utilisation et la conception de ces systèmes.

Dans une première partie consacrée à l'état de l'art, les recherches menées dans les domaines des systèmes mobiles et distribués sont présentées et soulignent les besoins de méthodes de conception intégratrices. Ces défis conceptuels sont engendrés par les différentes typologies de mobilité, par les moyens technologiques à disposition et par les attentes et les besoins des utilisateurs. Pour adresser ces défis, de nombreux travaux envisagent les notions de contexte, de sensibilité au contexte et de capacité d'adaptation en réponse à la dynamique d'exécution et à la variété des systèmes mobiles. Dans cette thèse, la sensibilité contextuelle pour les systèmes d'information est motivée par son application dans les domaines de l'informatique mobile et distribuée et une approche spatiale pour la définition du contexte est proposée. Une classification des différentes dimensions qui composent un contexte est retenue. Ces dimensions sont discriminées en *contexte d'usage* et en *contexte d'exécution*. Ces deux lectures du contexte permettent de caractériser une situation d'exécution selon le point de vue de l'utilisateur ou celui du système.

L'intégration de cette décomposition contextuelle dans un processus de conception est au coeur de la contribution de cette thèse. Le contexte d'exécution est défini par la description des conditions d'exécution du futur système. Une approche géographique pour modéliser la répartition des éléments du système constitue la base de cette description contextuelle. Elle associe des régions d'influence aux composants et étudie leur mobilité et leurs relations. Les différentes configurations spatiales de ces régions caractérisent l'ensemble des conditions d'exécution du système. Cette modélisation des contextes d'exécution est fournie en entrée d'un processus de conception étendu, centré sur l'analyse des tâches des utilisateurs par l'écriture de scénarios. L'association de l'analyse de tâches et de l'analyse contextuelle définit les espaces et les conditions nécessaires et suffisantes pour la réalisation des objectifs des utilisateurs. Chaque contexte permet un ensemble de tâches et les contextes d'exécution qui n'ont pas été associés à des tâches indiquent aux concepteurs des alternatives aux scénarios d'utilisation initiaux.

Dans un second temps, les impacts de la décomposition géographique du contexte d'exécution pour la mise en oeuvre de la sensibilité au contexte d'usage sont étudiés. Plus spécifiquement, les espaces définis par les couvertures spatiales des contextes d'exécution fournissent, lors de l'exécution, un critère de groupement d'utilisateurs. Ces derniers partagent au sein d'un contexte à la fois une proximité d'environnement et une proximité dans les tâches et les informations qu'ils manipulent. Du point de vue du contexte d'usage, ces groupements permettent la construction d'un tri et d'une présentation de données issues des préférences communes. Un algorithme de recommandation évalue les comportements interactifs et spatiaux des utilisateurs localement à chaque contexte d'exécution. Ces évaluations permettent de classer les éléments présentés dans ce contexte selon des critères de préférences d'un utilisateur seul ou de l'ensemble du groupe. L'algorithme proposé est paramétrable par un ensemble de fonctions de pondération. Les concepteurs peuvent adapter la constitution des scores selon l'usage du système, la dynamique des groupes ou encore l'historique des interactions.

Deux cas d'application illustrent les propositions de cette thèse. Un scénario de modélisation d'un système de suivi de compétition nautique applique les étapes du cadre de conception étendu. En se basant sur l'infrastructure du projet NavTrack pour le suivi en temps-réel de compétition de voile, cette modélisation par l'approche des contextes d'exécution caractérise l'ensemble des situations d'utilisation du service par des spectateurs mobiles. Des scénarios alternatifs complètent les tâches des utilisateurs. Avant l'implémentation, et pour chaque situation future d'exécution, l'usage des ressources et des informations disponibles est optimisé pour favoriser l'atteinte des objectifs des utilisateurs. Le second cas d'étude porte sur l'application de recommandations pour la personnalisation des informations présentées par un système mobile de visite de campus. Ce système associe différents contextes d'exécution aux régions du campus. Dans chaque région les informations présentées concernent les bâtiments des départements alentour. Les interactions des utilisateurs, qu'elles soient de nature physique (déplacements, pauses, itinéraires, etc.) ou numérique (sélections, saisies d'informations, etc.), permettent d'attribuer des scores à ces éléments pour adapter leur présentation. Ce second cas d'étude constitue le support du travail d'implémentation. Une simulation du système de visite de campus a été développée. Elle permet la saisie interactive des parcours d'utilisateurs et des interactions logicielles lors de visites simulées du campus. L'application cliente, et plus particulièrement la cartographie du campus, est adaptée en fonction des préférences inférées à partir des visites simulées. Cette chaîne de simulation constitue un banc de test pour l'ébauche du client et le réglage des paramètres de l'algorithme de construction de profils.

En conclusion, les dynamiques d'environnement de systèmes mobiles et distribués ont été décrites par des ensembles de contextes d'exécution. Ces travaux se sont ensuite attachés à illustrer l'utilisation des contextes aux moments de la modélisation et de l'exécution. Plus spécifiquement, une démarche conceptuelle a été proposée pour associer notre définition du contexte à un cadre de modélisation de système interactif. La mise en oeuvre des contextes ne se limitant pas à la modélisation, cette thèse propose une solution de personnalisation à l'exécution qui s'appuie sur la définition géographique du contexte pour adapter les éléments présentés.

Perspectives de recherche

Les approches et solutions proposées dans cette thèse permettent d'envisager de nombreuses perspectives liées aux thématiques de l'informatique mobile, de la conception d'un système ou de la personnalisation dynamique. L'essentiel de ces perspectives et de ces extensions consiste en un approfondissement de chaque thème abordé, pour explorer d'autres solutions éventuellement plus appropriées, et pour affiner les modèles proposés. Ces pistes sont déclinées selon les trois thèmes développés dans ce mémoire : modélisation contextuelle de l'environnement, approche intégratrice pour la conception et perception implicite de préférences. L'ensemble des perspectives retenues permettrait à moyen ou à court terme de généraliser les contributions décrites dans ce mémoire pour étendre l'espace d'application des modèles proposés.

Description contextuelle de l'exécution du système

En premier lieu, certaines améliorations du modèle géographique de l'environnement sont envisagées. Les relations entre régions, qui sont au coeur de la définition des contextes d'exécution, fonctionnent sur un mode binaire : une région peut être liée à une autre ou pas. La formalisation de relations qualitatives entre régions (p. ex. "*proche* de", "*presque* disjointe", etc.) permettrait d'affiner la granularité des contextes et enrichirait la palette d'expressions contextuelles à disposition des concepteurs du système. Ces relations géographiques plus discriminantes seraient associées à des comportements fonctionnels distincts : un client situé au centre d'une région disposerait d'outils différents d'un client proche de la frontière, etc.

La détermination des relations entre régions, et plus particulièrement la formulation de relations conditionnées constituerait un enrichissement important du modèle contextuel. Dans cette perspective, il serait possible d'énoncer des règles d'inclusion ou d'exclusion de la forme "Si une relation $R_x = Relate(\ldots)$ entre deux régions est *vraie*, alors telle autre relation $R_y = Relate(\ldots)$ est *vraie/fausse*".

Une classification des contextes en familles, selon la nature des situations et des fonctionnalités qu'ils décrivent, pourrait être proposée aux concepteurs. Une telle classification alimenterait une description ontologique des contextes qui permettrait d'analyser les transitions possibles d'un contexte vers un autre. La situation du système pourrait dépendre à la fois du contexte à l'instant d'exécution et du chemin contextuel passé des utilisateurs et des composants.

Intégration de la description contextuelle à un cadre de conception

L'approche d'intégration des contextes d'exécution dans une démarche de conception fait appel aux propriétés d'équivalence entre contextes. Ces équivalences permettent de guider les concepteurs dans leur sélection de sous-ensembles fonctionnels utiles aux clients. Le choix des propriétés d'équivalence reste subjectif et repose sur le bon sens des concepteurs. En développant davantage l'idée des équivalences entre contextes, il serait possible de déterminer, s'il existe, l'ensemble minimal des propriétés à appliquer quelles que soient les caractéristiques

du système développé. Des propriétés d'équivalence supplémentaires, appliquées au cas par cas selon les systèmes développés, pourraient constituer une boîte à outils et les concepteurs pourraient choisir d'appliquer l'un ou l'autre des ensembles de règles en fonction du système qu'ils modélisent.

Le modèle de conception étendu enrichit le *cadre de référence unifié* par l'ajout d'annotations contextuelles. Le choix du cadre de référence unifié comme point de départ tient à la nature des systèmes qui illustrent cette étude. Ce modèle de conception, centré sur l'analyse des tâches des utilisateurs, est adapté à la description de systèmes interactifs. Dans une approche plus générale des systèmes d'information, il peut être envisagé de combiner d'autres méthodes de conception à la description contextuelle. A l'instar du modèle de tâches proposé dans le cadre de référence unifié, les modélisations de concepts sous la forme d'*entités-associations* (p. ex. méthodes Merise ou MADS), ou sous la forme d'objets (p. ex. méthode UML), pourraient être "situées" en conditionnant les relations entre objets/entités par des contextes d'exécution. Il serait possible dans cette perspective de concevoir, par exemple, des systèmes de gestion de données distribués et mobiles proposant différents niveaux d'accès aux informations en fonction de contextes d'exécution.

Personnalisation implicite au sein des contextes d'exécution

La proposition de personnalisation par la perception implicite des actions spatiales et interactives des utilisateurs mérite d'être débattue. La démarche de quantification de scores mise en oeuvre pourrait être associée à une approche plus qualitative, qui laisserait certaines initiatives de la personnalisation aux utilisateurs. L'élaboration de profils par cette démarche moins calculatoire pourrait s'avérer un meilleur compromis dans l'objectif d'adaptations basées sur les préférences. D'autre part, les comportements mobiles des utilisateurs, leurs localisations, leurs parcours, leurs pauses constituent en l'état des suggestions d'indicateurs d'intérêt. Leur qualité et leur pertinence restent à démontrer avant de pouvoir les considérer aussi fiables que les indicateurs issus d'interactions Humain-machine.

Au-delà de ces efforts de validation, une autre extension du système de personnalisation consiste à partager les profils en dehors de contextes particuliers. Dans l'algorithme de construction de profils proposé, les préférences des utilisateurs ayant déterminé un profil de groupe sont perdues lorsque ces utilisateurs quittent leur contexte commun. La diffusion des informations de profil dans tout l'espace du système permettrait de limiter cette perte d'information. Plus spécifiquement, les utilisateurs et les composants mobiles peuvent véhiculer les scores constitués dans un contexte vers un autre et les partager sur un mode pair à pair. Par exemple, des utilisateurs qui ont dans leurs parcours traversé des profils semblables, pourraient, s'ils se rejoignent dans un contexte commun, échanger les préférences inférées lors de leurs passages dans des contextes passés. De cette manière, des profils associés à un contexte particulier peuvent être partagés à distance de ce contexte. Les informations de préférences peuvent être conservées tant qu'il reste des pairs (c-à-d. des composants actifs) mobiles pour transporter ces profils .

Bilan et généralisation

Les questionnements sur les thèmes des systèmes mobiles d'information, de la modélisation contextuelle et de la personnalisation précisés en introduction de ce mémoire, ont constitué un fil conducteur durant la réalisation de cette thèse. Après avoir proposé des solutions, mis en oeuvre des algorithmes, imaginé des situations d'utilisation et des cas d'application, il est souhaitable de revenir sur ces interrogations initiales afin d'examiner leur pertinence et l'à-propos des travaux initiés pour y répondre.

Quelle définition donner au concept de mobilité dans le cas des systèmes d'information ? En quoi la nature mobile des systèmes distribués impacte-t-elle leur conception ? La notion de mobilité des systèmes d'information, et les sous-jacents techniques qu'elle recouvre, sont en perpétuelle évolution. Au début de ces travaux de thèse, il était encore pertinent de s'interroger sur les moyens d'accès à l'information en mobilité. Aujourd'hui, les supports réseaux deviennent omniscients et les problèmes d'accès aux constituants du système se posent de moins en moins au moment de la conception. Notamment, depuis 2005, les technologies EDGE ou UMTS se sont très fortement développées et de nombreux projets logiciels considèrent la connectivité réseau comme acquise et permanente. À mesure que les composants d'un système pourront être liés à un support réseau global, les régions et contextes basés sur la portée des composants qui illustrent ce mémoire pourront paraître anachroniques. Toutefois, l'idée de régionalisation de l'espace d'exécution demeure pertinente pour la conception de systèmes mobiles. Dans une perspective plus générale, les régions d'exécution pourraient être formées à partir de critères distincts des capacités de transmission. Ces définitions des couvertures de régions permettraient d'aborder d'autres questions centrales de la mobilité des systèmes concernant en particulier la sécurité de l'information et l'accès aux données privées.

Comment appréhender l'environnement d'un système mobile ? Quelles sont les dimensions et les éléments qui caractérisent une situation d'exécution ? Est-il possible lors de la conception de catégoriser les situations d'exécution que le système rencontrera ? La démarche de recherche des contextes d'exécution à partir des régions de diffusion des composants mobiles met en lumière un ensemble de situations d'exécution. Dans cette perspective, une partie au moins des futurs contextes peut être *listée* au moment de la conception du système. Cependant, l'acceptation courante de "situation d'exécution" est sans doute beaucoup plus large que l'interprétation d'un contexte basé sur des relations de régions. Dans ce mémoire, la question de leur catégorisation est résolue en définissant notre propre notion de situation d'exécution. Notamment, la décomposition du contexte en parties d'usage et d'exécution a permis de simplifier, peut-être artificiellement, la variété des situations d'exécution. Il demeure délicat d'unifier, voire simplement de déterminer chaque type de contexte qui participe à la définition d'une situation d'exécution. Les démarches conceptuelles existantes, et à plus forte raison celle présentée dans cette thèse, sont contraintes à des réponses partielles de ce point de vue. Notre proposition d'une définition simple d'un contexte d'exécution, basé uniquement sur des relations de composants et de régions, permet déjà d'enrichir une approche conceptuelle par de nouvelles situations d'exécution. Toute dimension supplémentaire donnée au contexte d'exécution (p. ex. niveau

d'énergie d'un composant, charge mémoire, etc.) engendrerait des situations potentielles à prendre en compte par les concepteurs. Dans ce cas, les contextes seraient groupés par des équivalences qui ne sont plus seulement fondées sur des relations spatiales, mais qui dépendent de relations selon chaque dimension du contexte. Enoncer ces règles d'équivalence deviendrait une limite de la démarche de génération exhaustive des contextes.

Comment rendre compte de ces situations d'exécution dans une démarche de conception ? Quel impact provoque leur intégration pour la modélisation du système ? Pour intégrer notre décomposition du contexte, le choix s'est porté sur l'enrichissement d'un modèle conceptuel existant. Reconnu et utilisé pour le développement de logiciels interactifs, le *cadre de référence unifié* reste suffisamment ouvert pour pouvoir être facilement adapté à une prise en compte d'un contexte d'exécution variable. Le *cadre de conception étendu*, proposé dans cette thèse, doit être considéré comme un premier guide de développement pour mettre en oeuvre différentes situations d'exécution. Il existe toutefois une différence importante entre ces deux démarches de modélisation. Le cadre de conception unifié part de spécifications à haut niveau d'abstraction pour arriver vers la plate-forme. Les concepteurs définissent les besoins, analysent les tâches, structurent leur implémentation et déploient une plate-forme capable de supporter l'exécution de leur implémentation. Dans le cadre de conception étendu, cette méthodologie "top-down" de la conception de systèmes est mise à mal. La plate-forme constitue dans ce modèle une primitive au même titre que les spécifications des besoins. Dans la mesure où le cadre de référence unifié considère la distribution des composants parmi les dernières étapes de la conception, il est paradoxal d'utiliser cette décomposition en entrée d'un modèle qui découle de ce cadre de référence. Résoudre ce paradoxe amènerait probablement à décorréler les deux modèles. Les perspectives de recherche portant sur l'étude d'autres modèles de conception vont aussi dans ce sens.

Parmi les situations prévues à la conception, comment offrir une latitude d'exécution suffisante pour améliorer la valeur du système ? L'adaptation du système et la personnalisation des informations présentées ont été retenues comme moyens d'améliorer la valeur du système. Dans ces travaux, nous avons illustré quelques possibilités d'approches multi-utilisateurs pour personnaliser la présentation d'informations cartographiques. Avec le recul, l'originalité principale de la méthode de personnalisation proposée réside probablement davantage dans l'association des profils aux contextes d'exécution que dans la technique de construction des profils. Dans la plupart des méthodes de personnalisation présentées en bibliographie, les utilisateurs (individus ou groupements) constituent la cible d'adaptation. Dans ces systèmes, les préférences peuvent être appliquées indépendamment du contexte. Selon notre proposition, non seulement les profils sont construits entre utilisateurs, mais ces utilisateurs participent d'un même contexte d'exécution. Cette notion de profil local à un contexte définit un critère non-ambigu pour la participation d'un utilisateur à la construction d'un profil de groupe. Il serait possible de s'interroger davantage sur la signification de ces "profils de contexte" en généralisant la démarche au-delà des contextes d'exécution. Peut-être existe-t-il aussi des profils de contexte d'usage pouvant servir à améliorer la valeur ou l'utilité du système.

Relations entre le titre de thèse et le contenu développé. Le titre retenu pour cette thèse –*Approche spatiale pour la caractérisation du contexte d'exécution d'un système d'information ubiquitaire*– sous-entend que l'idée d'une approche spatiale est posée comme solution au problème de la caractérisation d'un contexte d'exécution. Les développements de ces notions dans le manuscrit montrent qu'il est possible d'imaginer que des contextes d'exécution soient déduits de relations entre des régions. Ces définitions de l'approche spatiale et des contextes d'exécution relèvent de nos propositions. Elles ont été préférées à la réutilisation de décompositions contextuelles ou d'approches de caractérisation existantes. En ayant défini ces deux notions clés d'une décomposition contextuelle, il est cohérent que les cas d'application présentés tirent bénéfice de l'approche proposée. En ayant comme point de départ une construction conjointe entre approche spatiale et contexte d'exécution, il est difficile de traiter par la suite ces deux propositions de façon indépendante. Rétrospectivement, cette problématique se traduit par deux questions ouvertes :

- l'approche spatiale pourrait-elle être étendue au-delà de la caractérisation de contextes d'exécution ?
- la définition proposée pour les contextes d'exécution aurait-elle un sens si ceux-ci n'étaient pas définis spatialement ?

Posées en début de thèse, ces questions auraient potentiellement orienté les travaux dans des directions distinctes. Plus spécifiquement, en s'attachant à étendre l'approche spatiale au-delà de la définition de contextes, cette thèse se rapprocherait des sciences de l'information géographique. Dans ce cas, les parcours des utilisateurs, les relations topologiques entre régions, les composants et leur mobilité permettraient d'identifier des situations ou des séquences clés plus complexes et riches que les contextes d'exécution. À l'inverse, en cherchant de nouvelles manières de définir la notion de contexte d'exécution, les travaux s'orienteraient plus vers le domaine de la conception des systèmes et prendraient un angle très informatique. Dans ce cas, les contextes d'exécution seraient définis non plus uniquement en fonction de relations de régions, mais aussi par exemple en fonction de liens de communication, de droits d'échange entre composants, etc. D'autres règles de généralisation et de groupement seraient imaginées et permettraient d'énoncer des ensembles de contextes sans doute très différents. Préférablement à un choix entre ces orientations, cette thèse conserve un équilibre entre description spatiale et modélisation contextuelle.

Annexes

ANNEXE

A Détails des calculs de profil de groupe

A.1 Calcul du score moyen instantané : cas d'étude 1

$$
\begin{aligned}
MeanVal(\mathrm{TaP}, ec_1, t_7) &= \underbrace{\sum_{j=1}^{\left|\substack{Users \\ (ec_1,t_7)}\right|} \left(\frac{\omega_{usr}(elm_i, usr_j, t_7)}{Sum_{\omega_{usr}}(elm_i, t_7)} \times \frac{Score(elm_i, usr_j, t_7)}{Sum_{Score}(usr_j, t_7)} \right)}_{\text{as at least } C\mathrm{u}_1 \text{ or } C\mathrm{u}_2 \text{ undergoes } ec_1 \text{ and } Users(ec_1,t_7)\neq\emptyset} \\
&= \sum_{j=1}^{|\{C\mathrm{u}_2, C\mathrm{u}_1\}|} \left(\underbrace{\frac{\omega_{usr}(\mathrm{TaP}, usr_j, t_7)}{\sum_{k=1}^{|\{C\mathrm{u}_2,C\mathrm{u}_1\}|} \omega_{usr}(\mathrm{TaP}, usr_k, t_7)}}_{\substack{\text{as at least } C\mathrm{u}_2\text{'s or } C\mathrm{u}_1\text{'s} \\ \text{weight } \omega_{usr}(\mathrm{TaP},..,t_7)>0}} \times \underbrace{\frac{Score(\mathrm{TaP}, usr_j, t_7)}{\sum_{l=1}^{|\{\mathrm{Sem},\mathrm{TaP}\}|} Score(elm_l, usr_j, t_7)}}_{\substack{\text{as at least Sem or TaP} \\ \text{score } Score(..,usr_j,t_7)>0}} \right) \\
&= \frac{2}{2+3} \times \frac{Score(\mathrm{TaP}, C\mathrm{u}_2, t_7)}{\left(\substack{Score(\mathrm{TaP},C\mathrm{u}_2,t_7)+ \\ Score(\mathrm{Sem},C\mathrm{u}_2,t_7)}\right)} + \frac{3}{2+3} \times \frac{Score(\mathrm{TaP}, C\mathrm{u}_1, t_7)}{\left(\substack{Score(\mathrm{TaP},C\mathrm{u}_1,t_7)+ \\ Score(\mathrm{Sem},C\mathrm{u}_1,t_7)}\right)} \\
&= \left(\frac{2}{5} \times \frac{.9}{(.9+.1)}\right) + \left(\frac{3}{5} \times \frac{.45}{(.45+.55)}\right) = \frac{63}{100} = 0.63
\end{aligned}
$$

A.2 Calcul du score moyen instantané : cas d'étude 2

$$
\begin{aligned}
MeanVal(\mathrm{Sem}, ec_1, t_4) = MeanVal(\mathrm{TaP}, ec_1, t_4) &= \underbrace{\frac{1}{|Elems(ec_1, t_4)|}}_{\text{comme } Users(ec_1,t_4)=\emptyset} \\
&= \frac{1}{|\{\mathrm{Sem}, \mathrm{TaP}\}|} = \frac{1}{2}
\end{aligned}
$$

A.3 Calcul du score de groupe : cas d'étude 1

$$
\begin{aligned}
GrpScore(\mathrm{TaP},ec_1,t_2) &= \big(1-\omega_{bal}(\mathrm{TaP},ec_1,t_2)\big)\times MeanVal(\mathrm{TaP},ec_1,t_2)+\omega_{bal}(\mathrm{TaP},ec_1,t_2)\times \\
&\qquad \sum_{j=1}^{2}\left(\frac{\omega_{hist}(\mathrm{TaP},ec_1,t_{2-j},t_2)}{\sum_{j=1}^{2}\omega_{hist}(\mathrm{TaP},ec_1,t_{2-j},t_2)}\times GrpScore(\mathrm{TaP},ec_1,t_{2-j})\right) \\
&= (1-0)\times 0.26 + 0\times\left(\frac{\omega_{hist}(\mathrm{TaP},ec_1,t_1,t_2)\times GrpScore(\mathrm{TaP},ec_1,t_1)}{\omega_{hist}(\mathrm{TaP},ec_1,t_1,t_2)+\omega_{hist}(\mathrm{TaP},ec_1,t_0,t_2)}+\frac{\omega_{hist}(\mathrm{TaP},ec_1,t_0,t_2)\times GrpScore(\mathrm{TaP},ec_1,t_0)}{\omega_{hist}(\mathrm{TaP},ec_1,t_1,t_2)+\omega_{hist}(\mathrm{TaP},ec_1,t_0,t_2)}\right) \\
&= 0.26 + 0\times\left(\frac{1\times 0.25}{1+1/2}+\frac{1/2\times 0.5}{1+1/2}\right) = 0.26
\end{aligned}
$$

ANNEXE

B Mesure de qualité de service

.

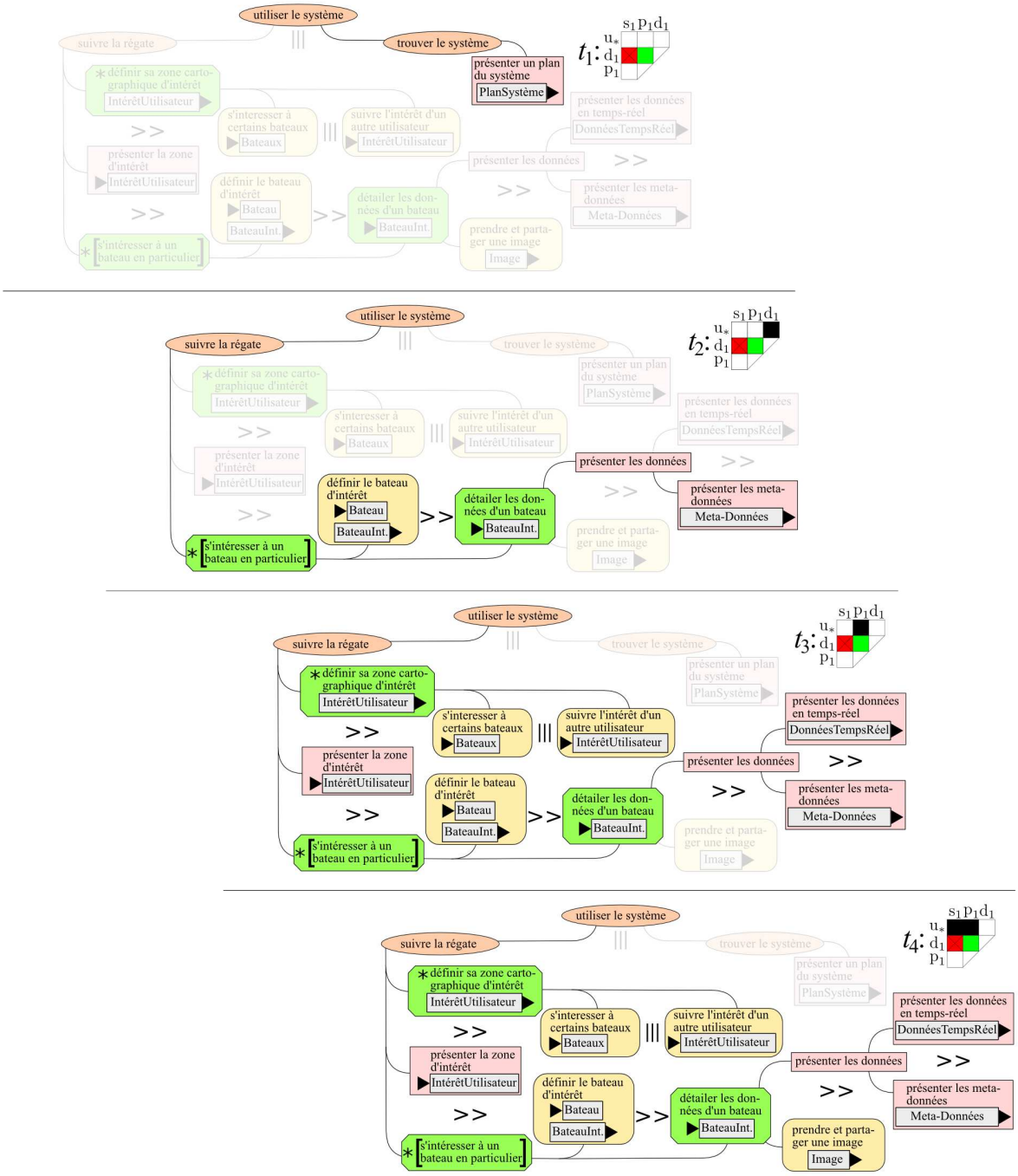


Figure B.1 – Proportion des tâches réalisables en fonction du contexte d'exécution rencontré

ANNEXE

C Notation matricielle du contexte d'exécution

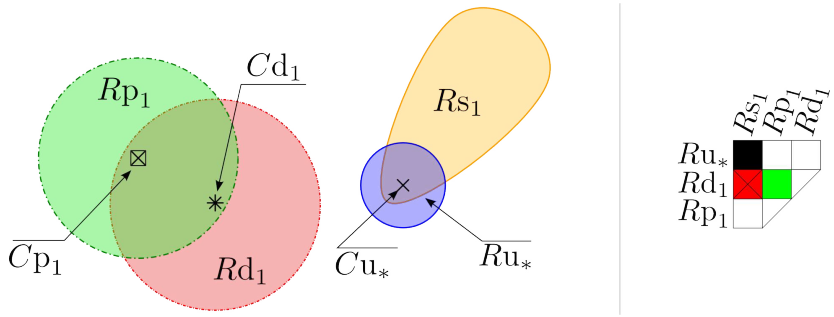
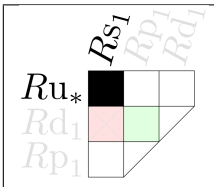
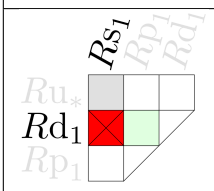
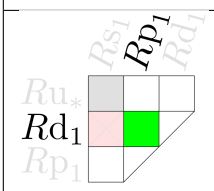
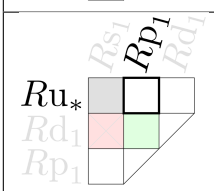


Figure C.1 – Exemple de contexte d'exécution d'un système distribué : empreinte spatiale des régions et notation matricielle des relations entre régions

Soit un système d'information distribué en quatre régions et quatre composants à t_x et la représentation spatiale et matricielle du contexte d'exécution à cet instant (Fig. C.1). La notation matricielle du contexte représente différentes propriétés des relations entre régions :

	Relation de régions : la relation de deux régions à un instant donné est représentée par une cellule noire. Dans le cas présenté en exemple, les régions Rs_1 et Ru_* sont liées à t_x (c-à-d. $Relate(Rs_1, Ru_*, t_x) = 1$)
	Relation exlue entre régions : une relation impossible entre deux régions quel que soit l'instant de l'exécution est représentée par une cellule rouge barrée d'une croix. Dans le cas présenté en exemple, les régions Rd_1 et Rs_1 ne sont jamais liées à l'exécution (c-à-d. $\forall\ t_x \in T\ : Relate(Rs_1, Rd_1, t_x) = 0$)
	Relation systématique entre régions : une relation permanente entre deux régions quel que soit l'instant de l'exécution est représentée par une cellule verte. Dans le cas présenté en exemple, les régions Rd_1 et Rp_1 sont systématiquement liées à l'exécution (c-à-d. $\forall\ t_x \in T\ : Relate(Rp_1, Rd_1, t_x) = 1$)
	Non-relation de régions : l'absence de relation entre deux régions à un instant donné est représentée par une cellule blanche. Dans le cas présenté en exemple, les régions Ru_* et Rp_1 ne sont pas liées à l'instant t_x de l'exécution (c-à-d. $Relate(Rp_1, Ru_*, t_x) = 0$)

ANNEXE D Liste de publications

D.1 Articles de revues

M. Petit, C. Ray, et C. Claramunt. Caractérisation de l'environnement d'exécution pour la conception d'un système d'information mobile et distribué. *Ingénierie des Systèmes d'Information*, 14(5) :11–34, October 2009.

E. Mac Aoidh, G. McArdle, M. Petit, C. Ray, M. Bertolotto, C. Claramunt, et D. Wilson. Personalization in adaptive and interactive gis. *Annals of GIS*, 11(1) :23–33, June 2009.

M. Petit, C. Ray, et C. Claramunt. An adaptive interaction architecture for collaborative GIS. *Cartographic and Geographical Information Sciences. Special issue on Modeling and Visualization for Spatial Decision Support*, 35(2) :91–102, April 2008.

D.2 Actes de conférences

M. Petit, C. Ray, C. Claramunt, et G. Calvary. A design process for the development of an interactive and adaptive GIS. In M. Bertolotto, X. Li, et C. Ray, editeurs, *Proceedings of the 8th International Symposium on Web and Wireless Geographical Information Systems*, number 5373 in LNCS, pages 100–111. Springer, December 2008.

M. Petit, C. Ray, et C. Claramunt. A user context approach for adaptive and distributed GIS. In M. Wachowicz et S. Fabrikant, editeurs, *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, LNG&C, pages 121–133. Springer, May 2007.

C. Claramunt, T. Devogele, S. Fournier, V. Noyon, M. Petit, et C. Ray. Maritime GIS : From monitoring to simulation systems. In V. Popovitch, M. Schrenk, et K. Korolenko, editeurs, *Proceedings of the 3rd International Workshop on Information Fusion and Geographical Information System*, LNG&C, pages 34–44. Springer, May 2007.

M. Petit, C. Ray, et C. Claramunt. A contextual approach for the development of GIS : Application to maritime navigation. In J. Carswell et T. Tekuza, editeurs, *Proceedings of the 6th International Symposium on Web and Wireless Geographical Information Systems*, number 4295 in LNCS, pages 158–169. Springer, December 2006. Best student paper award.

D.3 Autres communications

M. Petit, C. Ray, et C. Claramunt. Algorithme de recommandation adaptable pour la personnalisation d'un système mobile. In *Actes des 6èmes Journées Francophones Mobilité et Ubiquité : UBIMOB'10*, 4 pages, June 2010. URL `http://ubimob10.liris.cnrs.fr/index.php?n=Main.Programme`.

M. Petit et C. Ray. Approche conceptuelle d'un système mobile et distribué : de l'environnement dans le design. Poster Session - 21th international french-speaking conference on Human-Computer Interaction : IHM'09, September 2009. URL `http://www.aromate.org/papers/poster_ihm09.pdf`.

C. Ray, T. Devogele, V. Noyon, M. Petit, S. Fournier, et C. Claramunt. Gis technology for maritime traffic systems. *Ercim News : Special Theme on Traffic Planning and Logistics*, 68 :41–42, January 2007.

Bibliographie

G.D. Abowd, C.G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide : A mobile context-aware tour guide. *Wireless Networks*, 3(5) :421–433, 1997. [21, 25, 28]

G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *HUC'99 : Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, volume 1707 of *Lecture Notes In Computer Science*, pages 304–307, London, UK, 1999. Springer. [26]

G.D. Abowd, A.K. Dey, R. Orr, and J. Brotherton. Context-awareness in wearable and ubiquitous computing. *Virtual Reality*, 3(3) :200–211, September 1997. [58]

T. Alatalo and M.T. Siponen. Towards the OWLA methodology for development of open, Web/Wireless and adaptive hypermedia information systems. In *Poster session at the 12th ACM Conference on Hypertext and Hypermedia*, page 5pp, New York, NY, USA, 2001. ACM. [65]

J. Anke and D. Sundaram. *Human Computer Interaction : Concepts, Methodologies, Tools, and Applications*, volume 1, chapter Personalization Techniques and Their Applications, pages 168–176. Information Science Reference, Hershey, PA, USA, October 2008. [63]

M. Balabanović and Y. Shoham. Fab : content-based, collaborative recommendation. *Communications of the ACM*, 40(3) :66–72, 1997. [141, 142, 143, 156]

M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4) :263–277, 2007. [ix, 38, 39, 42, 44, 49, 52, 56, 57, 58, 62]

M. Banâtre, C. Bryce, P. Couderc, and F. Weis. *Informatique diffuse : des concepts à la réalité (Collection informatique)*, chapter Emergence de l'informatique diffuse, pages 35–48. Hermes-Lavoisier, 2007. [ix, 12, 13, 35, 39, 41, 42, 60, 61, 62, 66]

L. Barkhuus and P. Dourish. Everyday encounters with context-aware computing in a campus environment. In *Proceedings of the 6th International Conference on Ubiquitous Computing, UbiComp'04*, volume 3205 of *Lecture Notes in Computer Science*, pages 232–249. Springer, September 2004. [15, 16, 23, 24, 28]

J.J. Barton, S. Zhai, and S.B. Cousins. Mobile Phones Will Become The Primary Personal Computing Devices. In *WMCSA'06 : Proceedings of the Seventh IEEE Workshop on Mobile*

Computing Systems & Applications, pages 3–9, Washington, DC, USA, April 2006. IEEE. [14]

V.R. Basili, R.W. Selby, and D.H. Hutchens. Experimentation in software engineering. *IEEE Transactions on Software Engineering*, 12(7) :733–743, 1986. [27]

R. Bastide, D. Navarre, and P. Palanque. A tool-supported design framework for safety critical interactive systems. *Interacting with Computers*, 15(3) :309–328, 2003. [111]

J.M.C. Bastien and D.L. Scapin. Evaluating a user interface with ergonomic criteria. *International Journal of Human-Computer Interaction*, 7(2) :105–121, 1995. [114]

E. Beck, M. Christiansen, J. Kjeldskov, N. Kolbe, and J. Stage. Experimental evaluation of techniques for usability testing of mobile systems in a laboratory setting. In *Ozchi'03 : Proceedings of Computer-Human Interaction Special Interest Group (CHISIG) of the Ergonomics Society of Australia*, pages 106–115, November 2003. [22, 28]

B.B. Bederson, A. Clamage, M.P. Czerwinski, and G.G. Robertson. Datelens : A fisheye calendar interface for pdas. *ACM Transactions on Computer-Human Interaction*, 11(1) :90–119, 2004. [v, 25, 28]

M. Belatar and F. Poirier. Uniglyph : une méthode universelle pour la saisie de texte sur dispositifs mobiles. In *IHM'07 : Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine*, pages 111–118, New York, NY, USA, 2007. ACM. [24, 28]

G. Bell and P. Dourish. Yesterday's tomorrows : notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing*, 11(2) :133–143, 2007. [7]

Rudi Belotti, Corsin Decurtins, Moira C. Norrie, Beat Signer, and Ljiljana Vukelja. Experimental platform for mobile information systems. In *MobiCom'05 : Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 258–269, New York, NY, USA, 2005. ACM. [21, 24, 26, 28]

D. Billsus and M.J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2) :147–180, 2000. [143]

W. Birmingham, R. Dannenberg, and B. Pardo. Query by humming with the vocalsearch system. *Communication of the ACM*, 49(8) :49–52, 2006. [25, 28]

S. Björk, L.E. Holmquist, J. Redström, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzén. West : a web browser for small terminals. In *UIST'99 : Proceedings of the 12th annual ACM symposium on User interface software and technology*, pages 187–196, New York, NY, USA, 1999. ACM. [36]

B. Boehm. A spiral model of software development and enhancement. *SIGSOFT Software Engineering Notes*, 11(4) :14–24, 1986. [112]

C. Bolchini, C. Curino, F.A. Schreiber, and L. Tanca. Context integration for mobile data tailoring. In *MDM'06 : Proceedings of the 7th International Conference on Mobile Data Management*, pages 5–13, Washington, DC, USA, May 2006. IEEE. [50]

C. Bolchini, C.A. Curino, E. Quintarelli, F.A. Schreiber, and L. Tanca. A data-oriented survey of context models. *SIGMOD Record*, 36(4) :19–26, 2007. [49, 56]

A. Bouju. *Systèmes d'Information Géographique Mobiles : Architecture et optimisation.* PhD thesis, Univerité de La Rochelle, December 2008. [14]

B. Bouzy and T. Cazenave. Using the object oriented paradigm to model context in computer go. In *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context*, pages 279–289, February 1997. [50]

S. Brewster. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing*, 6(3) :188–205, 2002. [25, 28]

S. Brewster, F. Chohan, and L. Brown. Tactile feedback for mobile interactions. In *CHI'07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–162, New York, NY, USA, 2007. ACM. [25]

A. Brodt, D. Nicklas, S. Sathish, and B Mitschang. Context-aware mashups for mobile devices. In *WISE'08 : Proceedings of the 9th international conference on Web Information Systems Engineering*, volume 5175 of *Lecture Notes In Computer Science*, pages 280–291, Heidelberg, Germany, 2008. Springer Verlag. [36]

G. Broll, H. Hubmann, G.N. Prezerakos, G. Kapitsaki, and S. Salsano. Modeling context information for realizing simple mobile services. In *Proceedings of the 7th Mobile and Wireless Communications Summit*, pages 1–5, Washington, DC, USA, July 2007. IEEE. [51]

D. Brosset, C. Claramunt, and E. Saux. An ACS cooperative learning approach for route finding in natural environment. In H. Samet, C. Shahabi, and O. Wolfson, editors, *ACM-GIS'08 : proceedings of the 16th International Symposium on Advances in Geographic Information Systems*, pages 1–8, New York, NY, USA, November 2008. ACM. [44]

B. Brown and M. Chalmers. Tourism and mobile technology. In *ECSCW'03 : Proceedings of the eighth conference on Computer Supported Cooperative Work*, pages 335–354, Norwell, MA, USA, 2003. Kluwer Academic Publishers. [21, 25, 28]

M.G. Brown. Supporting User Mobility. In J.L. Encarnação and J.M. Rabaey, editors, *Proceedings of IFIP World Conference on Mobile Communications*, IFIP Conference Proceedings, pages 69–77. Chapman & Hall, 1996. [58]

P.J. Brown. Triggering information by context. *Personal and Ubiquitous Computing*, 2(1) :18–27, March 1998. [58]

P.J. Brown, J.D. Bovey, and X. Chen. Context-Aware Applications : From the Laboratory to the Marketplace. *IEEE Personal Communications*, 4(5) :58–64, 1997. [37, 58]

T. Brugère and A. Mollard. *Mathématiques à l'Usage des Informaticiens*, chapter Logique des Propositions & Logique des Prédicats, pages 99–158. Ellipses, Paris, France, 2003. [49]

P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User modeling and user-adapted interaction, Special issue on adaptive hypertext and hypermedia*, 6(2-3) :87–129, 1996. [65]

P. Brusilovsky. *Intelligent Hypertext : Advanced Techniques for the World Wide Web*, volume 1326 of *Lecture Notes in Computer Science*, chapter Efficient techniques for adaptive hypermedia, pages 12–30. Springer Verlag, Heidelberg, Germany, April 1997. [45]

R. Burke. Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370, 2002. [ix, 143]

R.D. Burke, K.J. Hammond, and B.C. Young. FindMe approach to assisted browsing. *IEEE Expert : Intelligent Systems and Their Applications*, 12(4) :32–40, July 1997. [143]

J. Burrell and G.K. Gay. E-graffiti : Evaluating Real-World Use of a Context-Aware System. *Interacting with Computers*, 14(4) :301–312, 2002. [27, 28, 39]

L. Butts and A. Cockburn. An evaluation of mobile phone text input methods. *Australian Computer Science Communications*, 24(4) :55–59, 2002. [25, 28]

G. Calvary. *Plasticité des Interfaces Homme-Machine, (Habilitation à Diriger des Recherches)*. PhD thesis, Université Joseph Fourrier, November 2007. [43, 47, 111]

G. Calvary and J. Coutaz. *Information, Interaction, Intelligence. Actes des 2e Assises nationales du GdR I3*, volume 3, chapter Plasticité des Interfaces : une Nécessité, pages 247–261. Cépaduès, 2002. [40, 41]

G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3) :289–308, 2003. [111]

S. Card, T.P. Moran, and A Newell. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates, 1983. [115]

J.M. Carroll. *Making use : scenario-based design of human-computer interactions*. MIT Press, Cambridge, MA, USA, September 2000. [110]

D. Chalmers, M. Sloman, and N. Dulay. Map adaptation for users of mobile systems. In *WWW'01 : Proceedings of the 10th international conference on World Wide Web*, pages 735–744, New York, NY, USA, 2001. ACM. [46]

G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Dept. of Computer Science, Dartmouth College, 2000. [9, 15, 26, 27, 28, 62]

H. Chen. *An Intelligent Broker Architecture for Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore, MD, USA, January 2003. [53, 55, 56]

H. Chen, F. Perich, T.W. Finin, and A. Joshi. Soupa : Standard ontology for ubiquitous and pervasive applications. In *MobiQuitous'04 : Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems, Networking and Services*, pages 258–267, Washington, DC, USA, August 2004. IEEE. [51, 57]

M.C. Chen, J.R Anderson, and M.H Sohn. What Can a Mouse Cursor Tell Us More ? Correlation of Eye/Mouse Movements on Web Browsing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'01)*, pages 281–282, Seattle, WS, USA, 2001. ACM Press. [150]

K. Cheverst, K. Mitchell, and N. Davies. Design of an object model for a context sensitive tourist guide. *Computers & Graphics*, 23(6) :883–891, 1999. [50]

C. Claramunt, T. Devogele, S. Fournier, V. Noyon, M. Petit, and C. Ray. Maritime GIS : From Monitoring to Simulation Systems. In *IF&GIS'07 : Proceedings of the 3rd International Workshop on Information Fusion and Geographic Information Systems*, Lecture Notes on Geoinformation and Cartography, pages 34–44, Heidelberg, Germany, May 2007. Springer Verlag. [169]

M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. Technical Report WPI-CS-TR-99-16, Worcester Polytechnic Institute, Computer Science Department, Worcester, MA, USA, June 1999. [143]

M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *IUI'01 : Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, January 2001. ACM. [144]

G. Cockton. Some Critical Remarks on Abstractions for Adaptable Dialogue Managers. In *People and Computer III : proceedings of the 3rd Conference of the British Computer Society, Human-Computer Interaction Specialist Group*, pages 325–343, Cambridge, UK, September 1987. British Information Society. [66, 67]

G. Cockton. From quality in use to value in the world. In *CHI'04 : Proceedings of the 22nd Conference on Human Factors in Computing Systems : extended abstracts on Human factors in computing systems*, pages 1287–1290, New York, NY, USA, April 2004. ACM. [47]

J.R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine, and W.A.S. Buxton. Evolution of a reactive environment. In *CHI'95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 170–177, New York, NY, USA, May 1995. ACM Press/Addison-Wesley Publishing Co. [58]

A. Cornuéjols, L. Miclet, and Y. Kodratoff. *Apprentissage artificiel. Concepts et algorithmes.* Collection Algorithmes. Eyrolles, Paris, France, August 2002. [142]

A.L. Cox and M.M. Silva. The Role of Mouse Movements in Interactive Search. In *Proceedings of the 28th Annual CogSci Conference*, pages 1156–1162, Vancouver, Canada, July 2006. [150]

F. Cuppens and N. Cuppens-Boulahia. Modeling contextual security policies. *International Journal of Information Security*, 7(4) :285–305, 2008. [46]

N. Davies, G.S. Blair, K. Cheuerst, and A. Friday. Supporting adaptive services in a heterogeneous mobile environment. In *WMCSA'94 : Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, pages 153–157, Washington, DC, USA, 1994. IEEE. [41]

R. De Virgilio and R. Torlone. A general methodology for context-aware data access. In *MobiDE'05 : Proceedings of the 4th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 9–15, New York, NY, USA, 2005. ACM. [49]

M. Derntl and K.A. Hummel. Modeling context-aware e-learning scenarios. In *PERCOM'05 : Workshop Proceedings of the 3rd International Conference on Pervasive Computing and Communications*, pages 337–342, Washington, DC, USA, March 2005. IEEE. [51]

G. Desvignes, G. Lucas de Couville, E. Peytchev, T. Devogele, S. Fournier, and C. Claramunt. The share-loc project : a wap-based maritime location system. In *WISE'02 : Proceedings of the Third International Conference on Web Information Systems Engineering*, pages 88–94, Washington, DC, USA, December 2002. IEEE. [18, 19, 20]

P. Deutsch. The eight fallacies of distributed computing. May 2009. [8, 57]

A.K. Dey. Context-Aware Computing : the CyberDesk Project. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54. AAAI Technical Reports, 1998. [37, 39, 58]

A.K. Dey, G.D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2) :97–166, 2001. [27, 37, 38, 39, 40, 41, 42, 53, 57, 59, 60, 61]

A.K. Dey, G.D. Abowd, and A. Wood. Cyberdesk : a framework for providing self-integrating context-aware services. In *IUI'98 : Proceedings of the 3rd international conference on Intelligent user interfaces*, pages 47–54, New York, NY, USA, January 1998. ACM. [58]

A.K. Dey, D. Salber, G.D. Abowd, and M. Futakawa. The Conference Assistant : combining context-awareness with wearablecomputing. In *Proceedings of the 3rd International Symposium on Wearable Computers*, pages 21–28, Washington, DC, USA, October 1999. IEEE. [26, 28, 37, 58]

H. Dieterich, U Malinowski, T. Kühme, and M. Schneider-Hufschmidt. *Adaptive User Interfaces : Principles and Practice*, chapter State of the art in adaptive user interfaces, pages 13–49. Number 10 in Human Factor in Technology. Elsevier, Amsterdam, The Netherlands, 1993. [v, 63, 64, 65]

A. Dix, D. Ramduny, T. Rodden, and N. Davies. Places to stay on the move : software architectures for mobile user interfaces. *Personal and Ubiquitous Computing*, 4(2-3) :171–181, June 2000. [17, 18, 19, 20]

P. Dourish. Seeking a foundation for context-aware computing. *Human-Computer Interaction*, 16(2) :229–241, 2001. [37]

T. D'Roza and G. Bilchev. An overview of location-based services. *BT Technology Journal*, 21(1) :20–27, 2003. [35]

M. Dunlop and S. Brewster. The challenge of mobile devices for human computer interaction. *Personal and Ubiquitous Computing*, 6(4) :235–236, 2002. [21, 27]

W.W. Eckerson. Three tier client/server architecture : achieving scalability, performance, and efficiency in client/server applications. *Open Information Systems*, 10(1) :1–12, January 1995. [8, 79]

E.A. Edmonds. *Computing skills and the user interface*, chapter Adaptive man-computer interfaces, pages 389–426. Academic Press, London, UK, April 1981. [63]

W.K. Edwards, V. Bellotti, A.K. Dey, and M.W. Newman. The challenges of user-centered design and evaluation for infrastructure. In *CHI'03 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 297–304, New York, NY, USA, April 2003. ACM. [15]

J. Eisenstein, J. Vanderdonckt, and A. Puerta. Adapting to mobile contexts with user-interface modeling. In *Proceedings of the 3rd workshop on Mobile Computing Systems and Applications*, pages 83–92, Los Alamitos, CA, USA, July 2000. IEEE. [v, 67]

J. Eisenstein, J. Vanderdonckt, and A. Puerta. Applying model-based techniques to the development of uis for mobile computers. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 69–76, New York, NY, USA, January 2001. ACM. [111]

S. Elrod, G. Hall, R. Costanza, M. Dixon, and J. Des Rivières. Responsive office environments. *Communications of the ACM*, 36(7) :84–85, July 1993. [58]

P.L. Emiliani and C. Stephanidis. Universal access to ambient intelligence environments : opportunities and challenges for people with disabilities. *IBM Systems Journal*, 44(3) :605–619, 2005. [23, 24, 28]

C. Endres, A. Butz, and A. MacWilliams. A survey of software infrastructures and frameworks for ubiquitous computing. *Mobile Information Systems*, 1(1) :41–80, 2005. [17]

B. Esselink. *A practical guide to localization*, chapter Introduction, pages 1–24. John Benjamins Publishing Company, Amsterdam, The Netherlands, 2000. [62]

P. Fahy and S. Clarke. Cass – a middleware for mobile context-aware applications. In *Proceedings of the MobiSys'04 Workshop on Context Awareness*, June 2004. [50]

S. Fickas, G. Kortuem, and Z. Segall. Software organization for dynamic and adaptable wearable systems. In *ISWC'97 : Proceedings of the 1st IEEE International Symposium on Wearable Computers*, pages 56–63, Washington, DC, USA, 1997. IEEE. [58]

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context : the concept revisited. *Transactions on Information Systems*, 20(1) :116–131, 2002. [39]

G. Fischer. User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11(1-2) :65–86, 2001. [ix, 63, 64]

G.H. Forman and J. Zahorjan. The challenges of mobile computing. *Computer*, 27(4) :38–47, 1994. [15, 16, 24, 28, 41]

D. Garlan and M. Shaw. An introduction to software architecture. *Advances in Software Engineering and Knowledge Engineering*, 2 :1–40, December 1993. [8]

Antonietta Grasso, Alain Karsenty, and Marco Susani. Augmenting paper to enhance community information sharing. In *DARE'00 : Proceedings of DARE 2000 on Designing augmented reality environments*, pages 51–62, New York, NY, USA, 2000. ACM. [22, 28]

S. Greenberg. Context as a dynamic construct. *Human-Computer Interaction*, 16(2) :257–268, 2001. [37]

S. Greenberg and D. Marwood. Real time groupware as a distributed system : Concurrency control and its effect on the interface. In *CSCW'94 : Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 207–217, New York, NY, USA, 1994. ACM. [41]

C. Greenhalgh, S. Benford, A. Drozd, M. Flintham, A. Hampshire, L. Oppermann, K. Smith, and C. von Tycowicz. Addressing Mobile Phone Diversity in Ubicomp Experience Development. In *Proceedings of the 9th International Conference on Ubiquitous Computing, UbiComp'07*, volume 4717 of *Lecture Notes in Computer Science*, pages 447–464. Springer, September 2007. [15]

M.S. Grewal, L.R. Weill, and A.P. Andrews. *Global positioning systems, inertial navigation, and integration*, chapter Fundamentals of Satellite and Inertia Navigation, pages 9–29. John Wiley & Sons, 2001. [12]

W.G. Griswold, P. Shanahan, S.W. Brown, R. Boyer, M. Ratto, R.B. Shapiro, and T.M. Truong. Activecampus : Experiments in community-oriented ubiquitous computing. *Computer*, 37(10) :73–81, 2004. [23]

B. de Groot and M. Van Welie. Leveraging the context of use in mobile service design. In *MobileHCI'02 : Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*, pages 334–338, Heidelberg, Germany, September 2002. Springer. [40, 41, 42]

M. Grossniklaus, M.C. Norrie, B. Signer, and N. Weibel. Putting location-based services on the map. In *Proceedings of the 6th International Symposium Web and Wireless Geographic Information Systems*, volume 4295 of *Lecture Notes in Computer Science*, pages 1–11. Springer, December 2006. [22, 28]

J. Grudin. Utility and usability : research issues and development contexts. *Interacting with computers*, 4(2) :209–217, August 1992. [48]

T. Gu, H.K. Pung, and D.Q. Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1) :1–18, 2005. [51, 57]

R.M. Gustavsen. Condor–An Application Framework for Mobility-Based Context-Aware Applications. In *Proceedings of the Workshop on Concepts and Models for Ubiquitous Computing*, 2002. [39]

C. Gutwin and C. Fedak. Interacting with big interfaces on small screens : a comparison of fisheye, zoom, and panning techniques. In *GI'04 : Proceedings of Graphics Interface 2004*, pages 145–152, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society. [35]

K. Hafner. *Where Wizards Stay Up Late : The Origins Of The Internet.* Simon & Schuster, 1998. [8]

M. Hampe and V. Paelke. Adaptive maps for mobile applications. In *MobileHCI'05 : Electronic Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, Workwhop on interactivity and usability of map-based mobile services*, New York, NY, USA, 2005. ACM. [24, 28]

A. Held, S. Buchholz, and A. Schill. Modeling of context information for pervasive computing applications. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, July 2002. [50]

K. Henricksen, J. Indulska, and A. Rakotonirainy. Modeling context information in pervasive computing systems. In *Pervasive'02 : Proceedings of the First International Conference on Pervasive Computing*, volume 2414 of *Lecture Notes In Computer Science*, pages 167–180, Heidelberg, Germany, August 2002. Springer-Verlag. [51]

K. Henricksen, J. Indulska, and A. Rakotonirainy. Generating context management infrastructure from high-level context models. In *Industrial Tracks Proceedings of the 4th International Conference on Mobile Data Management*, pages 1–6, January 2003. [51]

J. Hiipakka and G. Lorho. A Spatial Audio User Interface for Generating Music Playlists. In *ICAD'03 : Proceedings of the 9th International Conference on Auditory Display*, pages 267–270. Boston University Publications Production Department, 2003. [25, 28]

Y. Hijikata. Implicit user profiling for on demand relevance feedback. In *IUI'04 : Proceedings of the 9th international conference on Intelligent user interfaces*, pages 198–205, New York, NY, USA, January 2004. ACM. [144]

K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz. Sensing techniques for mobile interaction. In *UIST'00 : Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 91–100, New York, NY, USA, November 2000. ACM. [15]

T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger. Context-awareness on mobile devices - the hydrogen approach. In *HICSS'03 : Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, volume 9, pages 292–302, Washington, DC, USA, 2003. IEEE. [39, 50, 58]

Eve Hoggan and Stephen Brewster. Designing audio and tactile crossmodal icons for mobile devices. In *ICMI'07 : Proceedings of the 9th international conference on Multimodal interfaces*, pages 162–169, New York, NY, USA, 2007. ACM. [22, 28]

P. Holleis, F. Otto, H. Hussmann, and A. Schmidt. Keystroke-level model for advanced mobile phone interaction. In *CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1505–1514, New York, NY, USA, 2007. ACM. [25]

J.I. Hong and J.A. Landay. An infrastructure approach to context-aware computing. *Human-Computer Interaction*, 16(2) :287–303, 2001. [54, 55]

K. Hornbæk. Current practice in measuring usability : Challenges to usability studies and research. *International journal of human-computer studies*, 64(2) :79–102, 2006. [22]

R. Hull, P. Neaves, and J. Bedford-Roberts. Towards situated computing. In *ISWC'97 : Proceedings of the 1st IEEE International Symposium on Wearable Computers*, pages 146–153, Washington, DC, USA, October 1997. IEEE. [58]

J. Häkkilä. *Usability with context-aware mobile applications : Case studies and design guidelines*. PhD thesis, Departement of electrical and information engineering, Faculty of technology, University of Oulu, Oulu, Finland, January 2007. [27, 28, 29]

J. Indulska and P. Sutton. Location management in pervasive systems. In *ACSW Frontiers'03 : Proceedings of the Australian Information Security Workshop*, pages 143–151, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. [38]

Working Group 2.7 (User Interface Engineering) International Federation for Information Processing. *Design Principles for Interactive Software*, chapter External Properties : the User's Perspective, pages 25–50. Chapman & Hall, London, UK, July 1996. [47, 48]

F. Jambon. User Evaluation of Mobile Devices : In-Situ versus Laboratory Experiments. *International Journal of Mobile Computer-Human Interaction*, 1(2) :56–71, 2009. [23]

B.E. John and D.E. Kieras. The goms family of user interface analysis techniques : comparison and contrast. *ACM Trans. Comput.-Hum. Interact.*, 3(4) :320–351, 1996. [25]

M. Jones, G. Marsden, N. Mohd-Nasir, K. Boone, and G. Buchanan. Improving web interaction on small displays. In *WWW'99 : Proceedings of the 8th international conference on World Wide Web*, pages 1129–1137, New York, NY, USA, 1999. Elsevier. [35]

M Kaenampornpan, E. O'Neill, and B. Ba Ay. An integrated context model : Bringing activity to context. In *UBIComp'04 : Proceedings of the 6th International Conference on Ubiquitous Computing, Workshop on Advanced Context Modelling, Reasoning and Management*, September 2004. [51]

L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *POLICY'03 : Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 63–74, Washington, DC, USA, June 2003. IEEE. [57]

M. Kakihara and C. Sørensen. Expanding the 'mobility' concept. *SIGGroup Bulletin*, 22(3) :33–37, 2001. [10, 21, 28]

T. Kallio and A. Kaikkonen. Usability testing of mobile applications : A comparison between laboratory and field testing. *Journal of Usability Studies*, 1 :4–16, 2005. [23, 28]

P.B. Kantor, E. Boros, B. Melamed, V. Me nkov, B. Shapira, and D.J. Neu. Capturing human intelligence in the net. *Communications of the ACM*, 43(8) :112–115, August 2000. [144]

C. Karagiannidis, A. Koumpis, and C. Stephanidis. Deciding what, when, why and how to adapt in intelligent multimedia presentation systems. In G.P. Faconti and T. Rist, editors, *Proceedings of the 12th European Conference on Artificial Intelligence, Workshop Towards a Standard Reference Model for Intelligent Multimedia Presentation Systems*, volume 96, page 4pp. John Wiley & Sons, August 1996. [62]

L. Kärkkäinen and J. Laarni. Designing for small display screens. In *NordiCHI'02 : Proceedings of the second Nordic conference on Human-computer interaction*, pages 227–230, New York, NY, USA, 2002. ACM. [35]

A.K. Karlson, B.B. Bederson, and J. SanGiovanni. Applens and launchtile : two designs for one-handed thumb use on small devices. In *CHI'05 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 201–210, New York, NY, USA, 2005. ACM. [24, 28]

L. Karsenty, S. Sire, M. Causse, and N. Deherly. Quel impact de l'entrée vocale sur la conception graphique d'un service mobile ? In *IHM 2005 : Proceedings of the 17th international conference on Francophone sur l'Interaction Homme-Machine*, pages 299–306, New York, NY, USA, 2005. ACM. [25, 28]

R.H. Katz. Adaptation and Mobility in Wireless Informations Systems. *IEEE Personal Communications*, 1(1) :6–17, 1994. [10, 24, 28, 35]

R.H. Katz and R.A. Brewer. *Mobile Computing*, chapter The Case for Wireless Overlay Networks, pages 621–650. Kluwer Academic Publishers, Norwell, MA, USA, 1996. [46]

D. Kelly and J. Teevan. Implicit feedback for inferring user preference : a bibliography. *SIGIR Forum*, 37(2) :18–28, 2003. [143]

S.H. Kim, S.W. Kim, and H.M. Park. Usability challenges in ubicomp environment. In *Proceeding of International Ergonomics Association (IEA'03)*, pages 24–29, 2003. [27, 28]

J. Kjeldskov and C. Graham. A review of mobile HCI research methods. In *Proceedings of the 5th International Symposium on Human-Computer Interaction with Mobile Devices and Services : MobileHCI'03*, volume 2795 of *Lecture Notes in Computer Science*, pages 317–335. Springer, September 2003. [22, 27, 28]

J. Kjeldskov and J. Stage. New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, 60(5-6) :599–620, 2004. [23, 25, 28]

Leonard Kleinrock. Nomadic computing—an opportunity. *SIGCOMM Computer Communication Review*, 25(1) :36–40, 1995. [24, 28]

A. Kobsa, J. Koenemann, and W. Pohl. Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2) :111–155, 2001. [65]

K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darrell. Activity zones for context-aware computing. In *Proceedings of the 5th International Conference on Ubiquitous Computing, UbiComp'03*, volume 2864 of *Lecture Notes in Computer Science*, pages 90–106. Springer, September 2003. [8]

J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. Grouplens : applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3) :77–87, March 1997. [143]

T. Kopomaa and T. Snellman. *The city in your pocket : birth of the mobile information society.* Gaudeamus, Helsinki, Finland, 2000. [10]

G. Kortuem, Z. Segall, and M. Bauer. Context-aware, adaptive wearable computers as remote interfaces to 'intelligent' environments. In *ISWC'98 : Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 58–67, Washington, DC, USA, 1998. IEEE. [58]

S. Kristoffersen and F. Ljungberg. Mobile use of IT. In T.K. Käkölä, editor, *Proceedings of the 22nd Information Systems Research Seminar in Scandinavia : "Enterprise Architectures for Virtual Organisations"*, volume 2 of *Computer Science and Information Systems Reports*,

pages 271–284. Department of Computer Science and Information Systems, University of Jyväskylä, August 1999. [v, ix, 11, 14, 15, 16, 34, 35]

J. Krogstie, K. Lyytinen, A.L. Opdahl, B. Pernici, K. Siau, and K. Smolander. Research areas and challenges for mobile information systems. *International Journal of Mobile Communications*, 2(3) :220–234, 2004. [11, 16]

T. Kuflik, C. Callaway, D. Goren-Bar, C. Rocchi, O. Stock, and M. Zancanaro. Non-intrusive user modeling for a multimedia museum visitors guide system. In *UM'05 : Proceedings of the 10th International Conference on User Modeling*, volume 3538 of *Lecture Notes in Computer Science*, pages 236–240, Heidelberg, Germany, July 2005. Springer Verlag. [143]

W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *SIGIR'96 : Proceedings of the 19th international conference on Research and development in information retrieval*, pages 317–325, New York, NY, USA, August 1996. ACM. [142, 156, 162]

J.A. Landay and T.R. Kaufmann. User interface issues in mobile computing. In *Proceedings of the 4th workshop on Workstation Operating Systems*, pages 40–47, Washington, DC, USA, November 1993. IEEE. [21, 28]

J-M. Le Yaouanc, E. Saux, and C. Claramunt. A semantic and language-based model of landscape scenes. In *SeCoGIS'08 : Proceedings of the Second International Workshop on Semantic and Conceptual Issues in Geographic Information Systems*, pages 334–343, Heidelberg, Germany, 2008. Springer-Verlag. [44]

D. Leutner. *Curriculum, plans, and processes in instructional design*, chapter Instructional design principles for adaptivity in open learning environments, pages 289–308. Interactional Perspectives. Routledge (Taylor & Francis Group), Florence, KY, USA, April 2004. [65]

S. M. Lewandowski. Frameworks for component-based client/server computing. *ACM Computing Survey*, 30(1) :3–27, 1998. [9]

H. Lieberman. A multi-scale, multi-layer, translucent virtual space. In *IV'97 : Proceedings of the IEEE Conference on Information Visualisation*, pages 124–131, Washington, DC, USA, 1997. IEEE. [25, 28]

S. Long, R. Kooper, G.D. Abowd, and C.G. Atkeson. Rapid prototyping of mobile context-aware applications : the cyberguide case study. In *MobiCom'96 : Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 97–107, New York, NY, USA, 1996. ACM. [26, 28]

K. Luyten, T. Van Laerhoven, K. Coninx, and F. Van Reeth. Runtime transformations for modal independent user interface migration. *Interacting with Computers*, 15(3) :329–347, 2003. [111]

E. Mac Aoidh, M. Bertolotto, and D.C. Wilson. Analysis of implicit interest indicators for spatial data. In *GIS'07 : Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–4, New York, NY, USA, 2007. ACM. [144]

E. Mac Aoidh, M. Bertolotto, and D.C. Wilson. Understanding geospatial interests by visualizing map interaction behavior. *Information Visualization, special issue of the ICA Commission on Geovisualization of dynamics, movement and change : key issues and developing approaches in visualization research*, 7(3-4) :275–286, 2008. [144, 146, 150]

J. Magee, N. Dulay, S. Eisenbach, and J. Kramer. Specifying distributed software architectures. In W. Shäfer and P. Botella, editors, *Proceedings of the Fifth European Software Engineering Conference*, volume 989 of *Lecture Notes in Computer Science*, pages 137–153. Springer, September 1995. [9]

M. Maguire. Methods to support human-centred design. *International Journal of Human Computer Studies*, 55(4) :587–634, 2001. [110]

J.U. Mahmud, Y. Borodin, and I.V. Ramakrishnan. Csurf : a context-driven non-visual web-browser. In *WWW'07 : Proceedings of the 16th international conference on World Wide Web*, pages 31–40, New York, NY, USA, 2007. ACM. [36]

T. Makimoto and D. Manners. *Digital nomad*, chapter The Nomadic Opportunity, pages 1–24. John Wiley & Sons, 1997. [10]

B. Martin. Virhkey : a virtual hyperbolic keyboard with gesture interaction and visual feedback for mobile devices. In *MobileHCI'05 : Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 99–106, New York, NY, USA, 2005. ACM. [v, 24, 25, 28]

J. McCarthy. Notes on formalizing context. In R. Bajcsy, editor, *IJCAI'93 : Proceedings of 13th International Joint Conference on Artificial Intelligence*, volume 13, pages 555–560. Morgan Kaufmann, 1993. [50]

R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *DL'00 : Proceedings of the 5th ACM conference on Digital libraries*, pages 195–204, New York, NY, USA, June 2000. ACM. [143]

D. Moore and J. Hebeler. *Peer-to-Peer : Building Secure, Scalable, and Manageable Networks*. McGraw-Hill / Osborne, November 2002. [9]

J. Mooser, L. Wang, S. You, and U. Neumann. An Augmented Reality Interface for Mobile Information Retrieval. In *Proceedings of the 7th IEEE International Conference on Multimedia and Expo*, pages 2226–2229, Washington, DC, USA, July 2007. IEEE. [22, 28]

G. Mori, F. Paterno, and C. Santoro. CTTE : support for developing and analyzing task models for interactive system design. *Transactions on Software Engineering*, 28(8) :797–813, aug 2002. [116]

G. Mori, F. Paterno, and C. Santoro. Design and development of multidevice user interfaces through multiple logical descriptions. *Transactions on Software Engineering*, 30(8) :507–520, 2004. [112]

M. Mouly and M.B. Pautet. Current evolution of the GSM systems. *IEEE Personal Communications*, 2(5) :9–19, 1995. [20]

F. Mueller and A. Lockerd. Cheese : tracking mouse movement activity on websites, a tool for user modeling. In *CHI'01 : Proceedings of the Conference on Human Factors in Computing Systems : extended abstracts*, pages 279–280, New York, NY, USA, March 2001. ACM. [144]

M.J. Muller and S. Kuhn. Participatory design. *Communications of the ACM, Special issue on graphical user interfaces : the next generation*, 36(6) :24–28, 1993. [110]

A.F. Newell and P. Gregor. "user sensitive inclusive design"— in search of a new paradigm. In *CUU'00 : Proceedings on the 2000 conference on Universal Usability*, pages 39–44, New York, NY, USA, 2000. ACM. [24, 28]

J. Nielsen. *Usability engineering*, chapter What is Usability. Morgan Kaufmann, San Francisco, CA, USA, 1994. [47]

A.M. Nivala and L.T. Sarjakoski. Need for context-aware topographic maps in mobile devices. In Kirsi Virrantaus and Havard Tveite, editors, *Prodeedings of the 9th Scandinavian Research Conference on Geographical Information Science*, pages 15–29, Hut, Finland, June 2003. Helsinki University of Technology. [66]

B. Noble. System support for mobile, adaptive applications. *IEEE Personal Communications*, 7(1) :44–49, 2000. [19, 20, 24, 28]

B.D. Noble and M. Satyanarayanan. Experience with adaptive mobile applications in Odyssey. *Mobile Networks and Applications*, 4(4) :245–254, 1999. [19]

D.A. Norman and S.W. Draper. *User Centered System Design : New Perspectives on Human-Computer Interaction.* L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1986. [110]

R.J. Orr and G.D. Abowd. The smart floor : a mechanism for natural user identification and tracking. In *CHI'00 : Extended Abstracts on Human factors in Computing Systems*, pages 275–276, New York, NY, USA, 2000. ACM. [39]

A.M. Ouksel. In-context peer-to-peer information filtering on the web. *SIGMOD Record*, 32(3) :65–70, 2003. [50]

S.R. Page, T.J. Johnsgard, U. Albert, and C.D. Allen. User customization of a word processor. In *CHI'96 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 340–346, New York, NY, USA, April 1996. ACM. [62]

B. Pan, H. Hembrooke, G. Gay, L. Granka, M. Feusner, and J. Newman. The Determinants of Web Page Viewing Behavior : An Eye Tracking Study. In *Proceedings of the 2004 Symposium on Eye Tracking Research and Applications (ETRA)*, pages 147–154, San Antonio, Texas, USA, 2004. ACM Press. [150]

J. Pascoe. Adding generic contextual capabilities to wearable computers. In *ISWC'98 : Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 92–99, Washington, DC, USA, October 1998. IEEE. [37, 38, 58, 59, 60, 61, 62]

F. Paterno and C. Santoro. A unified method for designing interactive systems adaptable to mobile and stationary platforms. *Interacting with Computers*, 15(3) :349–366, 2003. [111]

M.J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5) :393–408, 1999. [143]

C.E. Perkins and D.B. Johnson. Mobility support in ipv6. In *Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 27–37, New York, NY, USA, 1996. ACM. [13]

M. Petit, C. Ray, and C. Claramunt. A user context approach for adaptive and distributed GIS. In M. Wachowicz and S. Fabrikant, editors, *Proceedings of the 10th AGILE International Conference on Geographic Information Science*, LNG&C, pages 121–133. Springer, 10 May 2007. [200]

D. Petrelli, E. Not, C. Strapparava, O. Stock, and M. Zancanaro. Modeling context is like taking pictures. In *Electronic Proceedings of the CHI'00 Workshop "The What, Who, Where, When, Why and How of Context-Awareness"*, April 2000. [49]

T.J. Pingel and K.C. Clarke. Assessing the usability of a wearable computer system for outdoor pedestrian navigation. In *Proceedings of the 13th AutoCarto conference*, page 16pp, 2005. [22, 28]

E. Pitoura and B. Bhargava. Building information systems for mobile environments. In *Proceedings of the third international conference on Information and knowledge management*, pages 371–378. ACM, 1994. [17, 18, 19, 20, 46]

F. Poirier and I. Schadle. État de l'art des méthodes de saisie de données sur dispositifs nomades : typologie des approches. In *IHM 2004 : Proceedings of the 16th conference of the Association Francophone d'Interaction Homme-Machine*, pages 133–140, New York, NY, USA, 2004. ACM. [24, 28]

S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf. Crumpet : creation of user-friendly mobile services personalised for tourism. In *Proceedings of the 2nd International Conference on 3G Mobile Communication Technologies*, pages 28–32. Institution of Electrical Engineers, 2001. [60]

D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere. Towards an extensible context ontology for ambient intelligence. In *Proceedings of the 2nd European Symposium on Ambient Intelligence*, volume 3295 of *Lecture Notes In Computer Science*, pages 148–159, Heidelberg, Germany, November 2004. Springer-Verlag. [51]

J. Pruitt and T. Adlin. *The persona lifecycle : keeping people in mind throughout product design.* Morgan Kaufmann, San Francisco, CA, USA, 2006. [115]

J. Raskin. *The Humane Interface : New Directions for Designing Interactive Systems*, chapter Quantification, pages 71–98. ACM/Addison-Wesley, New York, NY, USA, 2000. [48]

T. Reichenbacher. *Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices.* PhD thesis, Institut für Photogrammetrie und Kartographie, Technischen Universität München, München, Germany, November 2003. [v, 62, 65]

T. Reichenbacher. *Multimedia Cartography*, chapter Adaptation in mobile and ubiquitous cartography, pages 383–297. Springer Verlag, Heidelberg, Germany, 2007. [67]

G. Reitmayr and D. Schmalstieg. Collaborative augmented reality for outdoor navigation and and information browsing. In *In Proceedings of the 2nd Symposium on Location Based Services and TeleCartography*, pages 31–41. John Wiley & Sons, January 2004. [22, 28, 69]

J. Rekimoto, Y. Ayatsuka, and K. Hayashi. Augment-able reality : Situated communication through physical and digital spaces. In *ISWC'98 : Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 68–75, Washington, DC, USA, October 1998. IEEE. [58]

G. Rey. *Contexte en Interaction Homme-Machine : le Contexteur*. PhD thesis, Université Joseph Fourier, 2005. [40, 47]

B.J. Rhodes. The wearable remembrance agent : A system for augmented memory. *Personal and Ubiquitous Computing*, 1(4) :218–224, 1997. [14]

E. Rich. Users are individuals : individualizing user models. *International journal of man-machine studies*, 18(3) :199–214, 1983. [142]

D.C. Robbins, E. Cutrell, R. Sarin, and E. Horvitz. Zonezoom : map navigation for smartphones with recursive view segmentation. In *AVI'04 : Proceedings of the working conference on Advanced visual interfaces*, pages 231–234, New York, NY, USA, 2004. ACM. [v, 25, 28]

V. Robin and E. Brangier. Etude de l'utilisabilité de terminaux nomades de type pda pour accéder à l'information dans un réseau de transport en commun : première évaluation en laboratoire. Technical report, Labpsylor, Université Paul Verlaine, 2006. [23, 24, 28, 29]

T. Rodden, K. Chervest, N. Davies, and A. Dix. Exploiting Context in HCI Design for Mobile Systems. In *MobileHCI'98 : Proceedings of the First International Symposium on Mobile Human-Computer Interaction*, page 11pp, 1998. [40, 41, 42, 46, 48]

A. Rotem-Gal-Oz. Fallacies of distributed computing explained. May 2009. [8, 9, 16, 19, 80]

A. Roudaut and E. Lecolinet. Un espace de classification pour l'interaction sur dispositifs mobiles. In *IHM'07 : Proceedings of the 19th International Conference of the Association Francophone d'Interaction Homme-Machine*, pages 99–106, New York, NY, USA, 2007. ACM. [24]

I. Roussaki, M. Strimpakou, N. Kalatzis, M. Anagnostou, and C. Pils. Hybrid context modeling : A location-based scheme using ontologies. In *PERCOMW'06 : Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 2–7, Washington, DC, USA, March 2006. IEEE. [51]

E.M. Royer and C.K. Toh. A review of current routing protocols for ad hoc mobile wireless-networks. *IEEE Personal Communications*, 6(2) :46–55, 1999. [12]

C. Ryan and P. Rossi. Software, performance and resource utilisation metrics for context-aware mobile applications. In *METRICS'05 : Proceedings of the 11th IEEE International Software Metrics Symposium*, pages 12–22, Washington, DC, USA, September 2005. IEEE. [46]

N. Ryan, J. Pascoe, and D. Morse. Enhanced Reality Fieldwork : the Context Aware Archaeological Assistant. In *Archaeology in the Age of the Internet : Proceedings of the 25th Anniversary Conference on Computer Applications and Quantitative Methods in Archaeology*, page 269. British Archaeological Reports, April 1997. [37]

D. Salber, A.K. Dey, and G.D. Abowd. The context toolkit : Aiding the development of context-enabled applications. In *CHI'99 : Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 434–441, New York, NY, USA, 1999. ACM. [49, 53, 55, 58]

M. Samulowitz, F. Michahelles, and C. Linnhoff-Popien. Capeus : An architecture for context-aware selection and execution of services. In *Proceedings of the 3rd International Conference on New Developments in Distributed Applications and Interoperable Systems*, pages 23–40, Deventer, The Netherlands, September 2001. Kluwer Acamedic Publisher. [49]

M. Satyanarayanan. Fundamental challenges in mobile computing. In *PODC'96 : Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, pages 1–7, New York, NY, USA, May 1996. ACM. [16, 18, 19, 41]

M. Satyanarayanan. Pervasive computing : Vision and challenges. *IEEE Personal Communications*, 8(4) :10–17, 2001. [v, 36, 51]

D.L. Scapin and J.M.C. Bastien. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, 16(4) :220–231, 1997. [48]

A.I. Schein, A. Popescul, and L.H. Ungar. Methods and Metrics for Cold-Start Recommendations. In *SIGIR'02 : Proceedings of the 25th international conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA, August 2002. ACM. [142, 143, 156]

B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *WMCSA'94 : Proceedings of the First Workshop on Mobile Computing Systems and Applications*, pages 85–90, Washington, DC, USA, 1994. IEEE. [ix, 26, 28, 37, 39, 49, 58, 59, 60, 61]

A. Schmidt, M. Beigl, and H.W. Gellersen. There is more to context than location. *Computers & Graphics*, 23(6) :893–901, 1999. [27, 28, 48, 50]

C. Schurgers, V. Raghunathan, and M.B. Srivastava. Power management for energy-aware communication systems. *ACM Transactions on Embedded Computing Systems*, 2(3) :431–447, 2003. [15]

J. Scott and M. Hazas. User-friendly surveying techniques for location-aware systems. In *Proceedings of the 5th International Conference on Ubiquitous Computing, UbiComp'03*, volume 2864 of *Lecture Notes in Computer Science*, pages 44–53. Springer, September 2003. [15]

S. Searby. Personalisation—an overview of its use and potential. *BT Technology Journal*, 21(1) :13–19, January 2003. [62]

A. Shahi, V. Callaghan, and M. Gardner. Introducing Personal Operating Spaces for Ubiquitous Computing Environments. In *Proceedings of the workshop on Pervasive Mobile*

Interaction Devices, in conjunction with The 3rd International Conference on Pervasive Computing, pages 10–14, May 2005. [8]

S. Shrestha. Mobile web browsing : usability study. In *Mobility'07 : Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, pages 187–194, New York, NY, USA, 2007. ACM. [35]

C. Simons and G. Wirtz. Modeling Context in Mobile Distributed Systems With the UML. *Journal of Visual Languages and Computing*, 18(4) :420–439, 2007. [51]

B. Smyth and P. Cotter. A personalized television listings service. *Commun. ACM*, 43(8) :107–111, August 2000. [143]

S. Soltysiak and B. Crabtree. Knowing me, knowing you : practical issues in the personalisation of agent technology. In *Proceedings of the 3rd international conference on the practical applications of intelligent agents and multi-agent technology*, pages 467–484, Blackpool, UK, March 1998. Practical Application Co Ltd. [45]

J-S. Sottet, G. Calvary, J-M. Favre, and J. Coutaz. *Human-Centered Software Engineering*, chapter Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces : MEGA-UI, pages 173–200. Human-Computer Interaction Series. Springer-Verlag, Heidelberg, Germany, first edition, 2009. [43]

R.W. Soukoreff and I.S. MacKenzie. Metrics for text entry research : an evaluation of msd and kspc, and a new unified error metric. In *CHI'03 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2003. ACM. [25, 28]

H. Sridharan, H. Sundaram, and T. Rikakis. Computational models for experiences in the arts, and multimedia. In *ETP'03 : Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence*, pages 31–44, New York, NY, USA, 2003. ACM. [50]

S. Staab, H. Werthner, F. Ricci, A. Zipf, U. Gretzel, D.R. Fesenmaier, C. Paris, and C. Knoblock. Intelligent systems for tourism. *IEEE Intelligent Systems*, 17(6) :53–64, 2002. [35]

T.E. Starner. Attention, memory, and wearable interfaces. *IEEE Pervasive Computing*, 1(4) :88–91, 2002. [22, 28]

S. Steiniger, M. Neun, and A. Edwardes. Foundations of location based services. Technical report, Department of Geography, University of Zürich, 2006. Lecture Notes on Location Based Services. [13]

M.F. Story. Maximizing usability : the principles of universal design. *Assistive technology journal*, 10(1) :4–12, 1998. [23, 28]

T. Strang and C. Linnhoff-Popien. A context modeling survey. In *UbiComp'04 : Proceedings of Ubiquitous Computing Workshop on Advanced Context Modelling, Reasoning and Management*, page 8pp, 2004. [49, 51, 52, 56]

D. Svanaes. Context-aware technology : a phenomenological perspective. *Human-Computer Interaction*, 16(2) :379–400, 2001. [38]

S. Tamminen, A. Oulasvirta, K. Toiskallio, and A. Kankainen. Understanding mobile contexts. *Personal and Ubiquitous Computing*, 8(2) :135–143, 2004. [27, 28, 37]

Telecommunication Development Bureau. *Yearbook of Statistics - Telecommunication Services (Chronological Time Series 1997-2006)*. Telecommunication Development Bureau, 34 edition, 2008. [13]

The UIMS tool developers workshop. A metamodel for the runtime architecture of an interactive system. *SIGCHI Bulletin*, 24(1) :32–37, 1992. [66]

D. Thévenin. *Adaptation en Interaction Homme-Machine : le Cas de la Plasticité*. PhD thesis, Université Joseph Fourier, 2001. [v, 38, 40, 41, 42, 47, 63, 64]

D. Thévenin. From the workstation to the mobile phone : multi-target interface design. In *IHM'2003 : Proceedings of the 15th French-speaking conference on human-computer interaction*, pages 166–173, New York, NY, USA, November 2003. ACM. [v, 66]

D. Thévenin and J. Coutaz. Adaptation des ihm : taxonomies et archi. logicielle. In *IHM'02 : Proceedings of the 14th French-speaking conference on Human-computer interaction*, pages 207–210, New York, NY, USA, November 2002. ACM. [66]

P. Totterdell and P. Rautenbach. *Adaptive User Interfaces*, chapter Adaptation as a problem of design, pages 59–84. Academic Press Ltd., London, UK, 1990. [63]

T. Tran and R. Cohen. Hybrid recommender systems for electronic commerce. In *Proceedings from the AAAI Workshop on Knowledge-Based Electronic Markets*, volume 4 of *IAAA Technical Reports*, pages 78–83. IAAA Press, 2000. [143]

M. Van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137, pages 235–244, Heidelberg, Germany, August 2004. Springer Verlag. [61]

M. Van Welie and G. de Ridder. Designing for mobile devices : A context-oriented approach. In *Proceedings of the IBC Conference on Usability for Mobile Devices*, pages 9–11, 2001. [40]

Various Authors. *Human-Computer Interaction, Special Issue on Context-Aware Computing*, volume 16. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, USA, 2001. [38]

U. Varshney and R. Vetter. A framework for the emerging mobile commerce applications. In *HICSS'01 : Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, volume 9, page 10pp, Washington, DC, USA, January 2001. IEEE. [18, 19, 20]

W. Viana, J. Bringel Filho, J. Gensel, M. Villanova-Oliver, and H. Martin. Photomap : from location and time to context-aware photo annotations. *Journal of Location Based Services*, 2(3) :211–235, 2008. [61]

K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. Veijalainen, A. Katanosov, and H. Tirri. Developing gis-supported location-based services. In *Proceedings of the Second International Conference on Web Information Systems Engineering*, volume 2, pages 66–75, Washington, DC, USA, December 2001. IEEE. [27]

D. Vogel and P. Baudisch. Shift : a technique for operating pen-based interfaces using touch. In *CHI'07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 657–666, New York, NY, USA, 2007. ACM. [24, 28]

D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *ISWC'03 : Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pages 127–135, Washington, DC, USA, 2005. IEEE. [22, 28]

W. Wahlster and A. Kobsa. Dialogue-based user models. *Proceedings of the IEEE*, 74(7) :948–960, July 1986. [45]

R. Want, A. Hopper, V. Falc ao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1) :91–102, 1992. [25, 26, 28, 62]

R. Want, B.N. Schilit, N.I. Adams, R. Gold, K. Petersen, D. Goldberg, J.R. Ellis, and M. Weiser. The PARCTAB ubiquitous computing experiment. Technical Report CSL-95-1, Xerox Palo Alto Research Center, March 1995. [25, 26, 28, 59, 61, 84]

A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5) :42–47, October 1997. [58]

M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7) :75–84, 1993. [7, 23, 28, 35]

M. Weiser. The world is not a desktop. *interactions*, 1(1) :7–8, 1994. [25, 26, 28]

M. Weiser and J.S. Brown. Designing calm technology. *PowerGrid Journal*, 1(1) :75–85, 1996. [14]

T. Winograd. Architectures for context. *Human-Computer Interaction*, 16(2) :401–419, 2001. [52, 55, 56]

J. Wobbrock, B. Myers, and B. Rothrock. Few-key text entry revisited : mnemonic gestures on four keys. In *CHI'06 : Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 489–492, New York, NY, USA, 2006. ACM. [24, 28]

J.O. Wobbrock, J. Forlizzi, S.E. Hudson, and B.A. Myers. Webthumb : interaction techniques for small-screen browsers. In *UIST'02 : Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 205–208, New York, NY, USA, 2002. ACM. [35]

J.O. Wobbrock, B.A. Myers, and J.A. Kembel. Edgewrite : a stylus-based text entry method designed for high accuracy and stability of motion. In *UIST'03 : Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 61–70, New York, NY, USA, 2003. ACM. [v, 24, 25, 28]

W.W. Wu, E.F. Miller, W.L. Pritchard, and R.L. Pickholtz. Mobile satellite communications. *Proceedings of the IEEE*, 82(9) :1431–1448, 1994. [12]

H. Yan and T. Selker. Context-aware office assistant. In *IUI'00 : Proceedings of the 5th international conference on Intelligent user interfaces*, pages 276–279, New York, NY, USA, 2000. ACM. [26, 28]

Y. Yang and C. Claramunt. User preference elication and refinement in web geographical information systems. In *Proceedings of the 4th International Workshop on Web and Wireless GIS (W2GIS'08)*, pages 27–44, Heidelberg, Germany, 2004. Springer Verlag. [144]

J. Zheng and M.J. Lee. Will IEEE 802.15.4 make ubiquitous networking a reality ? - A discussion on a potential low power, low bit rate standard. *IEEE Personal Communications*, 42(6) :140–146, 2004. [12]

M. Ziat, O. Gapenne, J. Stewart, and C. Lenay. Haptic recognition of shapes at different scales : A comparison of two methods of interaction. *Interacting with Computers*, 19(1) :121–132, 2007. [25, 28]

APPROCHE SPATIALE POUR LA CARACTÉRISATION DU CONTEXTE D'EXÉCUTION D'UN SYSTÈME D'INFORMATION UBIQUITAIRE

RÉSUMÉ : Les nouvelles technologies en matière d'accès à l'information, de communication sans-fil et de localisation d'informations ouvrent la voie à des innovations majeures dans l'utilisation des systèmes informatiques. Ces avancées permettent d'imaginer de nouveaux usages informatiques dont la mise en œuvre motive le développement de méthodes de conception appropriées. Plus particulièrement, l'informatique mobile combine désormais des enjeux conceptuels tels que l'accès à des services par un utilisateur en mobilité, la généralisation d'outils de localisation, la compréhension de systèmes complexes par un public non-expert ou l'importante variabilité des situations d'exécution. Cette thèse propose un cadre de conception qui adresse certains des enjeux pour la mobilité des systèmes d'information. Dans cette perspective, les attentes des utilisateurs et les contraintes technologiques inhérentes à la mobilité des constituants du système définissent un espace contextuel dont les dimensions sont prises en compte dès les premières étapes de la conception. Le modèle proposé établit comme point d'entrée une description de l'espace géographique du système pour différencier un ensemble de contextes d'exécution. L'énoncé de ces contextes permet de corréler les attentes des utilisateurs avec les capacités techniques de la plate-forme et d'offrir le niveau fonctionnel le plus acceptable dans une situation donnée. Dans un second temps, les différentes fonctionnalités et les données mises en œuvre dans chaque contexte d'exécution peuvent être ordonnées ou filtrées afin d'optimiser la présentation des informations aux utilisateurs. Ces recommandations sont produites par l'analyse conjointe des préférences d'utilisateurs selon leurs interactions et leurs comportements spatiaux. Deux cadres expérimentaux viennent illustrer les propositions du modèle. La conception d'un système mobile de suivi de compétition nautique prend en compte différents contextes d'exécution et adapte un niveau de service à des situations dégradées. Un système de documentation et de visite de campus illustre les algorithmes de recommandation et affine la présentation d'informations localement à chaque contexte d'exécution.

Mots clés : Modélisation spatiale, Conception de systèmes interactifs, Systèmes mobiles et distribués, Adaptation contextuelle, Recommandations multi-utilisateurs.

A SPATIAL APPROACH TO EXECUTION-CONTEXT MODELING IN UBIQUITOUS INFORMATION SYSTEMS

ABSTRACT : Technology driven developments in wireless information access, power consumption, or real-time localization bring nomadic computing into being and go hand in hand with software design innovation. This thesis qualifies emerging challenges raised by mobile and distributed computing uses and examines their impacts on system design. A literature review of existing mobile systems frameworks shed light on desirable design methodologies improvements. An enriched design methodology is depicted to encompass these improvements. Such framework integrates a description of several geographic-based execution contexts with the early stages of the design. Along with user tasks analysis, this contextual description allows designers to relate user needs to system capabilities. On a higher level, the proposed environment-centered context description provides an input to user categorization. Users sharing same context are likely to access similar information and processes. Together, they may recommend and classify the data they interact with. These recommendations, derived from the users' spatial and interactive behaviours, enable group-based data and interface adaptations to avoid information overload in a mobile context. The testbed of the thesis is twofold. From the environment and context descriptions to the sketching of the client interface, the conceptual framework is at first applied to the design of a regatta tracking service. Next, a campus visit aid illustrates the benefits of multi-user recommendations and adaptations.

Keywords : Spatial modelling, Interactive systems design, Mobile and distributed systems, Context adaptation, Multi-user recommendations.

ARTS ET MÉTIERS ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY