



HAL
open science

Conception d'interfaces adaptatives basée sur l'ingénierie dirigée par les modèles pour le support à la coordination

Thomas Altenburger

► To cite this version:

Thomas Altenburger. Conception d'interfaces adaptatives basée sur l'ingénierie dirigée par les modèles pour le support à la coordination. Interface homme-machine [cs.HC]. Université de Lorraine, 2013. Français. NNT : 2013LORR0364 . tel-01238708v2

HAL Id: tel-01238708

<https://theses.hal.science/tel-01238708v2>

Submitted on 8 Jun 2018

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.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Conception d'interfaces adaptatives basée sur l'ingénierie dirigée par les modèles pour le support à la coordination

THÈSE

présentée et soutenue publiquement le 12 Décembre 2013

pour l'obtention du

Doctorat de l'Université de Lorraine

(spécialité informatique)

par

Thomas Altenburger

Composition du jury

Rapporteurs :	Pr. Jean VANDERDONCKT Dr. Sophie DUPUY-CHESSA	de l'Université Catholique de Louvain de l'Université de Grenoble
Examineurs :	Pr. Emmanuel DUBOIS Pr. Eric DUBOIS	de l'Université de Toulouse du Centre de Recherche Public Henri Tudor
Directeur :	Dr. Benoît MARTIN	de l'Université de Lorraine
Co-directeur :	Dr. Annie GUERRIERO	du Centre de Recherche Public Henri Tudor
Invité :	M. Alain VAGNER	du Centre de Recherche Public Henri Tudor

*À mon père,
cet homme qui n'a jamais cessé de me soutenir.
Et à ma mère,
qui aurait voulu que je sois ingénieur chez Arthur-Martin.*

«J'ai toujours rêvé que mon ordinateur soit aussi simple à utiliser que mon téléphone. Mon rêve est devenu réalité car aujourd'hui je ne comprends plus comment utiliser mon téléphone. »

Bjarne Stroustrup

Sommaire

Introduction	1
Chapitre 1 Les IHM dans leurs contextes	11
1.1 Le contexte au centre des IHM	12
1.1.1 Notion de contexte : définition dans le cadre des IHM.....	13
1.1.2 L'interaction en contexte : la <i>context-awareness</i>	14
1.2 La coordination dans le contexte	17
1.2.1 De la collaboration aux outils d'aide à la coordination	17
1.2.2 Pourquoi considérer la collaboration dans le contexte?	20
1.2.3 Les cadres d'observation théoriques de la coordination	23
1.2.4 Se coordonner dans la pratique.....	27
1.3 Du contexte à la génération des IHM adaptatives	36
1.3.1 De la personnalisation des IHM à leur plasticité.....	36
1.3.2 Conception et génération d'IHM adaptatives.....	39
1.3.3 Rapprochement entre coordination et génération d'IHM	48
1.4 Conclusion de chapitre	50
Chapitre 2 Les <i>Model-Based User Interfaces</i> et leur support de la coordination	53
2.1 MBUI pour les collecticiels : un tour d'horizon de l'existant	55
2.1.1 Méthodologies de modélisation de collecticiels pour MBUI.....	55
2.1.2 Génération d'IHM pour la collaboration par MBUI.....	70
2.2 Supporter la coordination à haut niveau.....	78
2.2.1 Métamodèles de l'activité collective	78
2.2.2 Métamodèles de tâches multi-utilisateurs.....	87
2.3 Supporter la coordination à bas niveau.....	96
2.3.1 Métamodèles et interactions pour la conscience de groupe.....	96

2.3.2	Métamodèles et application des flux de travaux	103
2.4	Conclusion de chapitre	108

Chapitre 3 Un cadre de conception d'IHM adaptatives pour la coordination **111**

3.1	Modélisation du contexte de l'activité collective	113
3.1.1	Contribution au métamodèle de la collaboration	114
3.1.2	Choix d'un métamodèle de flux de travaux.....	118
3.1.3	Métamodèles de tâches et du domaine.....	121
3.1.4	Adaptation au contexte de l'activité collective.....	123
3.2	Prise en compte de la conscience de groupe par widgets adaptatifs	130
3.2.1	Modélisation des besoins en conscience de groupe.....	130
3.2.2	Réponse sous forme de widgets adaptatifs	133
3.2.3	Composition d'interfaces et dépôt d'arbres de tâches de widgets . .	136
3.3	Prise en compte des flux de travaux.....	142
3.3.1	Application des flux de travaux.....	142
3.3.2	Génération d'un tableau de bord.....	143
3.4	Intégration dans une méthodologie de conception	147
3.4.1	Étapes de modélisation	147
3.4.2	Itération pour l'identification des besoins en conscience de groupe .	149
3.5	Conclusion de chapitre	154

Chapitre 4 Instanciation des concepts via la suite d'outils ADAGIOS sur un cas d'étude **159**

4.1	La suite d'outils ADAGIOS.....	161
4.1.1	Le générateur et interpréteur d'IHM de GENIUS.....	162
4.1.2	Extensions d'ADAGIOS à GENIUS.....	165
4.2	Cas d'étude : le secteur AIC	173
4.2.1	Détail du secteur	173
4.2.2	Le scénario de la réunion de chantier	176
4.2.3	Application de la méthodologie	178
4.3	Expérimentation sur les besoins en conscience de groupe.....	189
4.3.1	Protocole de test	189
4.3.2	Synthèse des résultats de l'expérimentation	197
4.4	Conclusion de chapitre	211

Conclusion générale	215
<hr/>	
Annexes	225
Annexe A Identification et validation des besoins en conscience de groupe	225
A.1 Identification des besoins en conscience de groupe	226
A.2 Validation des besoins en conscience de groupe.....	227
A.3 Validation des représentations des widgets.....	228
Annexe B Fiches de scénario pour l'expérimentation	231
Bibliographie	237

Introduction

I. Problématique

Les Interfaces Homme-Machine (IHM) représentent un domaine de recherche vaste qui n'a de cesse de révolutionner nos usages de l'**informatique**, depuis les interfaces **Windows-Icons-Menus-Pointer** introduites par PARC¹ avec l'**ordinateur** Xerox Alto en 1973, aux dernières avancées en matière de **Natural User Interfaces**, par exemple. Cette croissance est largement marquée par la problématique de la conception efficace d'IHM toujours plus évoluées et impliquant de nouveaux paradigmes d'interaction.

Un contexte d'interaction complexe

Ces évolutions impliquent notamment que le paysage numérique dans lequel nous vivons est de plus en plus riche et complexe. Les IHM font partie intégrante de notre quotidien et nous accompagnent dans la plupart de nos activités. Depuis quelques années, nous assistons à une métamorphose dans notre manière d'interagir avec elles. Nous ne sommes plus attachés à un seul ordinateur sédentaire, mais à une multitude d'appareils mobiles tels que les téléphones intelligents (**smartphones**), les tablettes tactiles et les montres intelligentes (**smartwatches**), ou encore des appareils à l'usage très spécifique comme les appareils de navigation par GPS, ordinateurs de bord et liseuses numériques. Chaque appareil vient avec ses propres interfaces, fonctionnalités et modalités d'interaction et sera utilisé dans des environnements divers et variés (e.g. au bureau, à la maison, en déplacement, etc.). Se pose alors la question de la continuité de l'interaction : si un utilisateur écrit un document sur son ordinateur à la maison, peut-il ensuite poursuivre son écriture dans le bus sur son smartphone pour le terminer sur son ordinateur portable à son bureau? Un changement d'appareil induit potentiellement un changement de modalités d'interaction et de contraintes techniques, sans compter que les usages sont généralement différents suivant l'environnement. Ces éléments tendent à complexifier les interactions et à potentiellement perdre l'attention de l'utilisateur en chemin, ou à réduire son efficacité. Ainsi les mots de Mark Weiser de 1991 [122] prennent tout leur sens **aujourd'hui**. Selon lui, le futur de l'**informatique** sera ubiquitaire et évanescent. On peut voir ceci comme une réponse à cette complexité, les IHM se faisant le plus utilisables possible, au point que les utilisateurs ne se rendent plus compte de leur présence. Il est donc nécessaire de comprendre l'**impact** du matériel et de l'environnement sur les interactions des utilisateurs

1. Palo Alto Research Center : <http://www.parc.com/>

afin de concevoir des IHM appropriées en toutes conditions, le but étant de masquer la complexité des environnements à **l'utilisateur** afin de maintenir **l'utilisabilité** de **l'IHM**.

En réponse à cela, divers domaines de recherche ont émergé, tels que la sensibilité au contexte (**context-awareness**) introduite par Dey & Abowd [31] qui considère le **contexte d'interaction** comme étant **l'ensemble** des circonstances **d'une** interaction en termes de matériel (e.g. les spécificités techniques du médium **d'interaction**), **d'environnement** physique (e.g. les conditions naturelles, telles que la luminosité, le bruit ambiant, etc.) et **d'utilisateurs** (e.g. leurs particularités, telles que leur expérience, leurs éventuelles déficiences, etc.). Les services sensibles au contexte prennent en compte ces éléments afin de proposer les fonctionnalités répondant le mieux aux besoins des utilisateurs en fonction du contexte. De nos jours, nous retrouvons cette notion dans bon nombre de services populaires, telles que ceux de Google Inc.² qui tiennent notamment compte de la géo-localisation de **l'utilisateur** et de **l'historique d'utilisation** des services (e.g. Google Now³, capable de faire des recommandations sur base de **l'utilisation** des autres services). Les systèmes adaptatifs constituent une spécialisation du concept de sensibilité au contexte dans la mesure où ceux-ci sont capables de reconnaître un contexte donné et **d'adopter** une réaction adaptée [110]. Cependant la sensibilité au contexte **n'offre** aucune garantie en termes de qualité de **l'expérience** utilisateur. **C'est** pour cela que fut introduite la notion de plasticité des IHM [111] qui vise des IHM capables de **s'adapter d'elles-mêmes** au contexte **d'interaction** tout en préservant leur utilisabilité.

Le contexte de l'activité collective

Contrairement à **l'informatique** grand-public où tout service a pour objectif principal les intérêts de **l'individu**, **l'informatique d'entreprise** a pour but de supporter la création de valeur issue de **l'effort** collectif des collaborateurs de cette entreprise. Or pour rester compétitives, de nombreuses entreprises évoluent, fusionnent, établissent des partenariats, produisent des spin-offs dans des écosystèmes de plus en plus complexes. Un modèle émergent **d'organisation d'entreprise** est par exemple **l'Open Innovation** [25], qui prône un décloisonnement de **l'entreprise** et une collaboration accrue avec son écosystème dans le but **d'augmenter** sa capacité **d'innovation**. **L'organisation** des entreprises est donc à **l'heure** actuelle de plus en plus mouvante, ce qui a un impact direct sur les logiciels supportant la collaboration de ses employés. Il nous semble donc opportun de considérer ces éléments dans **l'adaptation de l'interface** des logiciels de **l'entreprise**.

Au-delà des considérations habituelles relatives au contexte **d'interaction** (i.e. le matériel, **l'utilisateur** et **l'environnement** physique), une activité collective ajoute des paramètres non négligeables tels que la hiérarchie de **l'entreprise**, les activités à réaliser et les moyens mis à disposition pour que les utilisateurs puissent collaborer. Nous considérons cet ensemble de facteurs (incluant le contexte de **l'interaction**) comme étant le **contexte de l'activité collective**. Ce contexte-ci est particulièrement hétérogène et variable, même au sein **d'une** seule organisation.

Par exemple, une agence Web de taille moyenne ayant pour objectif la réalisation de sites Internet a une organisation de type structure professionnelle (e.g. une administration

2. <http://www.google.com/about/company/>

3. <http://www.google.com/landing/now/>

bureaucratique avec des divisions plus flexibles) telle que définie chez Mintzberg [75] : les intervenants y ont un domaine **d'expertise** et une certaine liberté dans les procédés. Dans cette organisation, les formes de coordination sont différentes en fonction des départements. Ainsi dans le département administratif, la majorité des procédés de travail sont standardisés. Par contre, le pôle opérationnel, constitué de développeurs, de Web designers et **d'administrateurs** systèmes, est organisé par projet, sous la supervision directe **d'un** chef de projet. Lors de bugs critiques ou de problèmes de production, la résolution de ces problèmes relève **d'ajustements** mutuels entre les intervenants. De plus, certains acteurs sont mobiles, tels que les commerciaux qui se rendent chez les clients tout en communiquant des informations à leur équipe. Cette différence dans les formes de coordination est notamment due à des objectifs différents pour chaque département. Le besoin de collaboration entre les acteurs est prégnant, quelque soit leur contexte **d'usage**, leur position dans la structure organisationnelle ou leurs modes de coordination.

Cependant, les outils qui viennent en support à une collaboration en contexte **n'ont** pas une couverture optimale de la problématique. Des outils tels que les logiciels **d'Enterprise Resource Planning** (ERP) répondent à une standardisation forte des procédés de collaboration mais ne correspondent pas à toutes les pratiques **d'une** entreprise. À **l'inverse**, les **groupware** sont hautement personnalisables, favorisant l'émergence de pratiques, mais sont trop peu structurés pour supporter une bureaucratie (i.e. structure où les procédés sont standardisés). La mobilité des acteurs requiert également pour ces solutions le support de différents contextes, notamment le changement **d'appareils** ou **d'environnements**. Cependant, ceci **n'est** habituellement pas pris en compte dans les solutions actuelles du marché et engendre donc des développements spécifiques **d'IHM** (cette problématique est néanmoins traitée dans le champ de recherche des Distributed User Interfaces, DUI).

Certains auteurs, tels que Shen et al. [101] et Tarpin-Bernard [109], mettent **d'ailleurs** en avant les types de problèmes qui peuvent surgir si le contexte de **l'activité** collective est mal pris en compte dans la conception des IHM : problèmes de communication, dysfonctionnements de coordination, mauvaise compréhension entre les utilisateurs, problèmes de gestion documentaire, etc. Autant de problèmes qui ont un impact direct sur **l'activité**, que ce soit en termes de productivité ou de coûts non maîtrisés. Pour pallier à ces éventualités, ces auteurs prônent la considération du contexte de **l'activité** collective dès la conception des IHM. **C'est-à-dire** en concevant des IHM capables de **s'adapter** aux organisations et à leur structure, aux moyens de coordination, à **l'activité** et de présenter des interactions qui favorisent un bon fonctionnement.

Pour synthétiser, la problématique que nous soulevons est celle de la conception **d'IHM** capables de satisfaire les contraintes du contexte de **l'activité** collective.

II. Orientation

Le contexte de **l'activité** collective est complexe. Il se pose donc la question de la conception **d'interfaces** adaptatives prenant compte de ces éléments.

Concevoir des IHM adaptatives

Concevoir une IHM pour plusieurs contextes donnés est un processus relativement complexe. Si l'on emploie des méthodes classiques d'ingénierie des IHM, et considérant des contextes discrets, on se trouve devant une alternative : soit produire une IHM capable de gérer tous les contextes à la fois, soit développer une IHM pour chaque contexte. Dans les deux cas nous rencontrons des problèmes évidents de coûts, de maintenabilité, de cohérence des interfaces et d'ergonomie. Par exemple, Microsoft Word® existe actuellement en quatre versions, pour Windows®, pour Mac®, pour mobile et tablette, et pour le cloud (Web). Si ces logiciels utilisent une base de code commune, il n'empêche que ces différentes versions ont une IHM pour chaque contexte, Microsoft a dès lors dû investir dans le développement d'une version différente pour chaque contexte.

La solution que nous envisageons à cette problématique est de concevoir des IHM adaptatives via l'approche MBUI (Model-based User Interfaces). L'Ingénierie dirigée par les modèles (IDM) a notamment reconsidéré les MBUI dans les années 2000, dans la mesure où elle leur fournit un cadre structuré et unifié de modélisation en vue de produire un logiciel. Du point de vue de la conception des IHM adaptatives, diverses méthodes de conception et architectures logicielles existent [112, 20] et les MBUI représentent une approche faisant l'objet de nombreuses recherches avec des travaux tels que TERESA [85], UsiXML [70] ou encore MARA [103].

De manière générale, l'IDM présente divers intérêts pour une démarche de conception logicielle. Il s'agit notamment d'abstraire les problématiques d'ordre technologique par des modèles (les *Platform Independent Models* ou PIM), ceci afin d'exprimer toutes les connaissances sur un système en cours de conception dans un même langage de modélisation. Il est dès lors facilement possible d'établir des liens entre les modèles décrivant un même système. Ceci facilite également la communication entre les différents intervenants de la conception via des représentations plus appropriées. L'IDM prône également une séparation des préoccupations, chaque intervenant s'occupant des modèles relatifs à son domaine d'expertise. Enfin l'IDM vise l'automatisation de la production de logiciels via des modèles conçus à cette fin et des transformations permettant de raffiner les modèles et de générer un code exécutable. L'IDM a notamment fait ses preuves en termes d'efficacité, notamment via l'exemple récent de la voiture hybride Volt de Chevrolet, conçue grâce à l'IDM⁴.

Appliquée aux MBUI, et donc à la conception d'IHM adaptatives, l'IDM permet une industrialisation du processus de conception. Ainsi, les concepteurs, analystes et ergonomes se concentrent sur leur domaine d'expertise. Grâce à une génération des IHM, il est possible de prototyper et de tester rapidement une IHM, et donc d'itérer efficacement sur la conception.

Les MBUI trouvent par ailleurs un cadre de référence à travers les travaux sur CAMELEON [21, 113, 23]. Plus particulièrement, ce cadre propose de modéliser les IHM en partant de modèles abstraits décrivant les tâches et leur contexte d'interaction pour aboutir à des modèles concrets décrivant les composants exécutables. CAMELEON met également en avant l'adaptation des IHM à leur contexte d'interaction afin de les rendre

4. <http://www.embedded.com/electronics-blogs/-include/4215057/Ten-million-lines-in-29-months--model-driven-development-on-the-Chevy-Volt>

plus utilisables dans des contextes donnés. Cependant, la considération du contexte de l'**activité** collective représente encore des challenges dans ce paradigme de conception. Cette présente thèse se propose d'explorer ce problème et notamment la difficulté de concevoir des IHM pouvant prendre en compte le contexte de l'**activité** collective.

Considérer le contexte de l'activité collective

Afin d'intégrer les aspects du contexte de l'**activité** collective dans une génération d'IHM par les modèles, il convient tout d'abord de mieux comprendre ce contexte. Lorsque l'on considère le contexte de l'**activité** collective dans les IHM, il est intéressant de prendre un point de vue sur la théorie et la pratique du travail collectif, tel quel pris par Lonchamp [71] :

- **La théorie** regroupe un ensemble de cadres d'analyse qui permettent de structurer et de comprendre la collaboration, tels que la **théorie de la coordination** [73] ou la **théorie de l'activité** [69]. Ces travaux permettent de déterminer les points d'adaptation d'une IHM pour l'**activité** collective et alimentent également les aspects pratiques ;
- **La pratique** représente les outils (ou interactions) permettant aux utilisateurs, à un niveau individuel, de se coordonner, de communiquer et de coopérer. Il s'agit d'outils tels que la conscience de groupe (**group awareness** [55]) i.e. la possibilité de percevoir l'**activité d'autrui** pour mieux comprendre une situation, la communication médiatisée [71] ou encore les flux de travaux (**workflow**) i.e. formalisation de l'**enchaînement** des tâches.

Proposer des IHM pouvant **s'adapter** au contexte de l'**activité** collective passe par la considération de ces deux aspects. L'**orientation** de nos travaux est de tendre vers la définition d'un cadre de conception d'IHM adaptables au contexte de l'**activité** collective sur base de MBUI. Plus précisément, dans l'espace du Travail Coopératif Assisté par Ordinateur (TCAO)⁵, nous prenons position sur la coordination, c'est-à-dire sur la manière dont les utilisateurs synchronisent leur activité pour tendre vers un but commun. Le domaine de la collaboration étant vaste, nous nous concentrons sur cet aspect pour y apporter une réponse pertinente. À ce titre, nous mettons un accent particulier sur les mécanismes qui favorisent ou cadrent la coordination tels que la conscience de groupe (**group awareness**) et les flux de travaux (**workflow**).

Notre analyse de l'**existant** se focalise alors sur la considération des éléments théoriques et pratiques de la coordination dans les travaux gravitant autour des MBUI. Cette revue de la littérature s'opère en trois temps.

Tout d'abord, d'un point de vue d'ensemble, en considérant des méthodologies de conception de collecticiels proches de l'**IDM** telles que TOUCHE [88, 87] ou AMENITIES [43, 45]. Ces travaux sont ceux qui se rapprochent le plus de notre problématique et couvrent la modélisation de nombreuses de ses facettes, voire s'intègrent à des MBUI pour générer des IHM (le cas de TOUCHE par exemple).

5. Le TCAO (en anglais **Computer Supported Cooperative Work**, CSCW) a été défini par Wilson en 1991 [124] comme le terme générique qui combine la compréhension de la façon dont les personnes travaillent en groupe avec les technologies qui supportent ce travail, à savoir les réseaux informatiques, les logiciels, le matériel informatique, les services et les techniques.

Puis, afin **d’approfondir** notre revue, nous considérons les travaux qui modélisent les aspects théoriques du contexte de **l’activité** collective. Nous scindons cette revue suivant les différents éléments qui constituent le contexte. Ceci, en examinant notamment des métamodèles de la collaboration, tels que proposés au **sein d’UsiXML [70]** ou des travaux plus conceptuels comme ceux de De Farias et al. [29]. Nous abordons également la modélisation de tâches collaboratives avec des travaux comme CTML [126, 127, 39].

Enfin, nous prenons le point de vue de la pratique, **c’est-à-dire** la considération **d’outils** comme la conscience de groupe et les flux de travaux dans les MBUI. Nous nous intéressons donc à la modélisation de ces outils. Concernant la conscience de groupe, nous soulignons les travaux sur TOUCHE [88, 87] ainsi que ceux sur WSL [37], alors que du côté des flux de travaux nous abordons le sujet avec FlowiXML de Garcia et al. [51, 42] ou encore UsiWSC [15, 14].

Au terme de cette réflexion, nous émettons les hypothèses de recherche suivante :

- H1. Apporter des métamodèles du contexte de **l’activité** collective en entrée **d’un** processus de génération **d’IHM** permet de produire des IHM adaptatives tenant compte de ces éléments.
- H2. Apporter des métamodèles **d’abstraction** de la pratique de la coordination, notamment de la conscience de groupe et des flux de travaux, permet de générer des IHM mettant en **œuvre** ces principes.
- H3. Une méthodologie de conception tenant compte des hypothèses H1 **et** H2, **c’est-à-dire** des métamodèles et des mécanismes de génération **d’IHM** associés, permet de couvrir **l’ensemble** de la chaîne de conception **d’IHM** (i.e. de l’élucitation des besoins jusqu’à la production) pour **l’activité** collective.

III. Proposition

L’analyse de ces différents éléments nous permet de mettre en lumière les points de contributions sur lesquels cette thèse **s’oriente**. Notre proposition consiste alors à traiter des éléments suivants, en réponse aux hypothèses :

1. **Compléter la modélisation des concepts de haut niveau** : **c’est-à-dire** fournir un ensemble de métamodèles décrivant toutes les facettes du contexte de **l’activité** collective afin de les intégrer dans un processus de génération **d’IHM** adaptatives. Ces éléments servent à la fois à définir des points **d’adaptation** mais alimentent également les outils pratiques (i.e. conscience de groupe et flux de travaux) ;
2. **Établir un support de la coordination dans les IHM générées** : **c’est-à-dire** de permettre la génération **d’IHM** favorisant la conscience de groupe et mettant en application les flux de travaux. Pour cela, il est nécessaire **d’intégrer** ces éléments à différents niveaux du processus de génération **d’IHM**, notamment en passant par une abstraction de la conscience de groupe ;
3. **Étendre les méthodologies de conception de collecticiels** : si les méthodologies étudiées représentent une bonne base de modélisation, il serait toutefois intéressant

d'aller plus loin, à l'image de TOUCHE [88, 87] qui propose de modéliser les IHM. Cependant, aucune approche ne semble aller jusqu'à la génération effective des IHM, ni ne considérer l'adaptation. Nous orientons alors nos travaux vers l'intégration d'un processus de génération d'IHM adaptatives dans ces méthodologies de conception de collecticiels.

IV. Méthodologie de recherche

Dans cette thèse, nous avons suivi une méthodologie basée sur la science du design (design science) [57][123] qui prône la généralisation des processus de design et l'évaluation des artefacts proposés (méthodes, modèles, métamodèles, services, instanciations, etc.). Hevner définit trois cycles pour la science du design : la pertinence, la rigueur et le design en soi. Nous les décrivons ci-après en figure I.1.

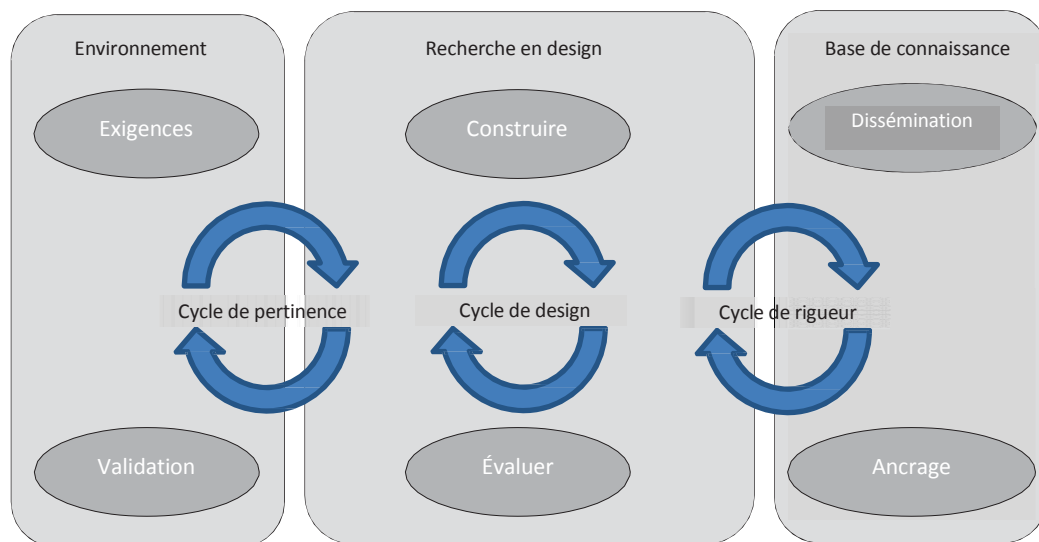


Figure I.1 – Le processus de science du design de Hevner. [57]

Le **cycle de pertinence** alimente la recherche en design avec les exigences des professionnels, les limitations identifiées des approches existantes ainsi que des besoins d'innovation. Ce cycle utilise les artefacts construits via la recherche en design et les valide auprès des professionnels ciblés par ce travail. Dans notre cas, les exigences liées à la conception d'IHM par les modèles ont été identifiées dans le projet de recherche Genius [104], mais aussi validées par l'intermédiaire d'experts métiers spécialisés dans notre domaine d'application sur lesquels les artefacts ont été instanciés, à savoir le secteur de la construction qui met en lumière la complexité du contexte de l'activité collective. L'ensemble de ces exigences nous a permis de bien cerner la problématique de cette thèse.

Le **cycle de rigueur** s'intéresse à l'ancrage de l'activité de recherche en design dans la réutilisation des connaissances scientifiques et des pratiques industrielles et par-là même permet le positionnement de cette thèse. Nous avons donc procédé à un état

de l'art à deux niveaux : tout **d'abord** un état **de l'art** global et multi-perspectives sur les différents domaines (sensibilité au contexte, travail collaboratif assisté par ordinateur et IHM adaptatives), puis nous avons mis le focus sur l'état de **l'art** des artefacts qui nous intéressent (i.e. permettant la prise en compte de la coordination dans les MBI). Pour chaque artefact identifié, nous avons étudié les différentes solutions issues de l'état de **l'art**, que nous avons ensuite sélectionné sur base de besoins provenant du cycle de pertinence et enfin nous avons détecté les manques de ces différentes solutions. Dans le but de participer à **l'enrichissement** de la base de connaissances scientifique globale, une partie des résultats de ces travaux a été publiée dans plusieurs conférences et workshops liés aux thèmes de cette thèse, à savoir IHM [2, 3] et construction [4, 5].

Le cycle de design consiste en la construction et l'évaluation des artefacts sur base de la connaissance acquise dans les cycles de pertinence et de rigueur. À partir des besoins et des manques identifiés, nous avons pu construire notre proposition qui consiste en un cadre de conception **d'IHM** adaptatives pour le support à la coordination, constitué de métamodèles, **d'une** méthode et **d'une** architecture logicielle pour un outil support à la méthodologie de conception.

Les artefacts de cette proposition ont ensuite fait **l'objet d'une** validation de par leur application à une expérimentation sur un cas d'étude ancré dans le secteur de la construction. Nous avons cherché à évaluer **l'applicabilité** :

1. des métamodèles sur un cas concret, afin de valider **l'hypothèse H1** ;
2. de **l'architecture** logicielle, afin de valider **l'hypothèse H2** ;
3. de la méthodologie ainsi que ses principales caractéristiques, afin de valider **l'hypothèse H3**.

Comme la qualité de **l'expérimentation** dépend directement du cas d'étude, nous avons sélectionné un cas d'étude dont les caractéristiques correspondent bien à notre problématique, **c'est-à-dire** dont le contexte de **l'activité** collective est complexe et variable.

Nous avons développé un démonstrateur sur base de **l'architecture** logicielle. Le résultat du développement constitué par ce démonstrateur permet de valider **l'applicabilité** de **l'architecture** logicielle. Ensuite, au sein de notre cas d'étude, nous avons défini un ensemble de scénarios réalistes que nous avons fait valider par des experts métier. Nous avons modélisé ces scénarios suivant nos métamodèles proposés et appliqué la méthodologie en impliquant différents utilisateurs ayant un profil représentatif vis-à-vis du cas d'étude. **L'applicabilité** des métamodèles et de la méthodologie sont donc validés par la réussite de **l'expérimentation**.

Cependant, nous avons souhaité évaluer plus précisément certaines caractéristiques de notre méthodologie, notamment sa capacité à implémenter certains besoins de collaboration au sein des interfaces générées. Pour cela nous avons combiné **l'observation** directe des utilisateurs lors du déroulement de **l'expérimentation** avec **l'utilisation** de questionnaires ciblés sur cette question et permettant une évaluation qualitative de la méthodologie.

V. Plan du manuscrit

Le plan de ce manuscrit reflète cette méthodologie de recherche et **s'articule** autour de quatre chapitres :

1. **Un chapitre d'état de l'art préliminaire**, qui explore toutes les facettes des interactions dans leur contexte et fait office **d'introduction** étendue à la problématique. Nous y mettons une emphase sur la coordination et les approches de conception **d'IHM** pouvant répondre au contexte. Nous aborderons ainsi **l'IDM** avec les MBUI et **l'adaptation** des IHM, en passant par une couverture plus étendue des aspects théoriques et pratiques de la coordination afin de cerner au mieux le contexte de **l'activité collective** ;
2. **Un chapitre d'état de l'art centré sur les MBUI**, en vue **d'apporter** une réponse à notre problématique. Nous explorons ici la littérature en relevant les travaux qui abordent notre sujet sous différents angles. Il **s'agit** ici de passer en revue des travaux généralistes sur les méthodologies de conception de collecticiels pour MBUI qui se rapprochent le plus de notre problématique. Puis, afin **d'approfondir** les composantes du contexte de **l'activité collective**, nous étudions les travaux qui proposent la modélisation des aspects théoriques, ensuite pratiques, du problème de la coordination dans les MBUI. Cet état de **l'art** nous permettra de mettre en lumière les points pouvant faire **l'objet** de contributions ;
3. **Un chapitre de propositions**, où nous détaillons notre vue **d'un** MBUI pour le support à la coordination et sur base des éléments identifiés en état de **l'art**. Nous apporterons des propositions dans la modélisation du contexte de **l'activité collective**, dans la génération **d'un** support à la coordination via la génération **d'interactions** pour la conscience de groupe et les flux de travaux. Nous proposerons également **l'intégration** de ces éléments dans une méthodologie de conception ;
4. **Un chapitre de validation et de résultats**, dans lequel nous instancions nos propositions à travers un démonstrateur afin de valider nos travaux **d'un** point de vue technique, puis à travers un cas d'étude afin de valider les aspects méthodologiques et de confronter nos propositions de métamodèles à une réalité. Pour cela, nous présenterons une expérimentation menée qui met une emphase sur nos propositions en termes de coordination. Ce chapitre se conclut par les résultats obtenus et les perspectives **qu'ils** ouvrent.

Chapitre 1

Les IHM dans leurs contextes

Sommaire

1.1	Le contexte au centre des IHM	12
1.1.1	Notion de contexte : définition dans le cadre des IHM.....	13
1.1.2	L'interaction en contexte : la context-awareness	14
1.2	La coordination dans le contexte.....	17
1.2.1	De la collaboration aux outils d'aide à la coordination.....	17
1.2.2	Pourquoi considérer la collaboration dans le contexte?.....	20
1.2.3	Les cadres d'observation théoriques de la coordination	23
1.2.4	Se coordonner dans la pratique	27
1.3	Du contexte à la génération des IHM adaptatives	36
1.3.1	De la personnalisation des IHM à leur plasticité	36
1.3.2	Conception et génération d'IHM adaptatives.....	39
1.3.3	Rapprochement entre coordination et génération d'IHM	48
1.4	Conclusion de chapitre	50

De nos jours, nous rencontrons dans notre quotidien une multitude **d'IHM**. Nous possédons notamment divers dispositifs qui nous suivent dans nos déplacements : smartphone, tablette tactile, ordinateur portable et bientôt lunettes et montres intelligentes. Autant **d'appareils** avec lesquels nous interagissons dans divers milieux, tels **qu'au** travail, chez soi ou dans les transports en commun. Chaque lieu fait **l'objet** de ses propres usages. Cette situation complexifie les interactions mais aussi la conception de telles IHM qui doivent être utilisables sur tous supports et en tous lieux. Ceci est **d'autant** plus vrai lorsque **l'on** aborde le sujet des interfaces collaboratives, rajoutant le paramètre du groupe dans la boucle. Se pose alors la question de la pertinence de **l'IHM** et surtout, la question qui nous intéresse, celle de la conception **d'IHM** supportant tous ces paramètres (i.e. le matériel, **l'environnement** et la collaboration).

Ce premier chapitre pose les bases de cette question qui nous conduira à identifier un certain nombre de leviers scientifiques. Ainsi, notre cheminement démarrera avec le **contexte d'interaction** qui représente **l'ensemble** des paramètres sus-nommés. **L'importance** de ce contexte vis-à-vis des IHM nous mènera à la conception de systèmes prenant

en compte celui-ci. Nous aborderons ensuite la notion de contexte sous **l'angle** de la coordination. Pour cela, nous relèverons ses particularités et nous essaierons **d'en tirer l'essence** en vue **d'en** tenir compte lors de la conception **d'IHM**. Ceci nous conduira à considérer une méthode de conception **d'IHM** particulière, issue de **l'adaptation** des interfaces et de **l'Ingénierie Dirigée par les Modèles (IDM)**.

Ce chapitre se conclura par une formalisation de notre question de recherche qui nous permettra **d'identifier** les axes de recherche nécessaires à sa résolution à travers les différentes synthèses de ce chapitre.

Structuration du chapitre

La construction de la question de recherche passe par une exploration pluridisciplinaire afin de cerner au mieux les tenants et aboutissants de la problématique. Ainsi, nous aborderons tout **d'abord** une notion qui se trouve à la base de nos travaux, à savoir celle du contexte **d'interaction**, en section 1.1. Nous nous intéresserons à sa définition et surtout à son rôle dans la conception des systèmes interactifs. En effet, les systèmes dits sensibles au contexte représentent un paradigme qui sied à nos problèmes contemporains de multiplication des IHM en les rendant « intelligentes » dans la mesure où ils permettent **d'en** limiter la complexité.

Nous nous approcherons ensuite du problème de la coordination lorsque **l'on** considère le contexte, en section 1.2. Pour cela, nous soulèverons la complexité du contexte que cela représente. Nous explorerons aussi les facettes théorique et pratique de la coordination en vue **d'en** tirer **l'essence** pour une mise en contexte, notre intention étant **d'identifier** les éléments du contexte **d'interaction** importants pour la conception et **l'utilisation d'IHM** favorisant la coordination.

Enfin, la complexité des IHM et de leurs contextes nous mènera à considérer **l'adaptation** dans les interfaces, en section 1.3. Les IHM adaptatives sont une réponse possible aux interactions en contexte, notamment vis-à-vis de notre problématique de coordination. Nous nous intéressons tout particulièrement au cas de la **plasticité** des IHM, ou plus généralement de la génération **d'IHM** adaptatives. Ainsi, nous couvrirons à la fois les aspects de conception des IHM tenant compte du contexte, mais aussi de leur mise en **œuvre**. Le tout, porté par **l'Ingénierie Dirigée par les Modèles (IDM)**, une méthode **d'ingénierie** qui se prête à notre approche du problème et qui nous conduira en chapitre 2 vers un état de **l'art** plus spécifique au support de la coordination.

1.1 Le contexte au centre des IHM

Dans une situation **d'interaction** complexe, impliquant divers dispositifs, lieux, utilisateurs et autres paramètres, il est nécessaire **d'aborder** les IHM sous un angle particulier. Le problème étant **qu'il** existe un risque de perdre **l'utilisateur** dans cette masse **d'interactions** et **d'informations**. Dès lors, **c'est** aux interfaces de se plier à cette situation et de ne distiller que **l'essentiel** à **l'utilisateur**. **L'interface** doit disparaître, comme le dirait Weiser [122].

Cette section traite ainsi du sujet du contexte **d'interaction**, une notion centrale dans la

démarche qui vise à rendre les IHM intelligentes. La section 1.1.1 introduit cette notion et explicite son importance. Le contexte est un élément qui permet notamment de rendre les systèmes interactifs proactifs, se faisant alors plus discrets, au profit de l'utilisateur. Nous aborderons d'ailleurs ce type de systèmes dits sensibles au contexte, en section 1.1.2. Nous verrons comment ils peuvent répondre à la multiplication des situations d'interaction.

1.1.1 Notion de contexte : définition dans le cadre des IHM

Afin de mieux aborder la complexité d'une situation d'interaction, il convient de prendre du recul et de cadrer une notion clef : celle de **contexte**. Si l'on s'en tient à la définition académique de ce terme telle qu'énoncée dans un dictionnaire⁶, celui-ci représente l'« ensemble des circonstances dans lesquelles se produit un événement, se situe une action. » Dans son usage, on peut noter que cette définition subit une spécialisation suivant les domaines d'application. Par exemple, en littérature, le contexte définit l'entourage d'un mot ou d'un texte, lui donnant son sens précis. En archéologie, le contexte représente les circonstances historiques d'une découverte, replacées dans leur époque d'origine. Le contexte est également relatif à une perspective donnée : on peut étudier celui-ci localement à une action (e.g. l'environnement proche, la pièce dans laquelle on se situe, etc.) ou plus globalement (e.g. le contexte géopolitique). Dans un certain sens, on peut dire que la notion de contexte est elle-même dépendante du métier et de la perspective. Ce fait est notamment relevé par Winograd [125] et Brézillon [18]. Réfléchir en termes de contextes nous est complètement naturel et implicite car cette notion est partie intégrante de nos fonctions cognitives d'êtres humains, nos plus inconscientes décisions étant traitées vis-à-vis de leur contexte.

Dans le cadre des travaux de cette thèse, nous nous intéressons uniquement à son application aux IHM. C'est-à-dire aux circonstances qui entourent une interaction entre un utilisateur et un système informatique. Schilit et al. [95] qualifient par exemple le contexte comme étant l'utilisateur, sa localisation ainsi que la présence d'autres personnes et de ressources à portée. Pascoe et al. [83], quant à eux, le définissent par l'espace de travail de l'utilisateur et sa localisation. De manière plus générale, nous parlerons au fil de cette thèse de **contexte d'interaction**. En ce sens, la définition la plus admise est donnée par Dey & Abowd [31] et s'énonce comme suit, traduite de l'anglais :

Définition. Contexte d'interaction : toutes informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité étant une personne, un lieu, ou un objet qui est considéré comme étant pertinent pour l'interaction entre un utilisateur et une application, incluant l'utilisateur et l'application eux-mêmes.

On peut noter que cette définition est englobante, incluant l'utilisateur et le système. Les auteurs vont d'ailleurs plus loin en scindant la notion de contexte en trois **environnements** :

L'**environnement système** qui représente les caractéristiques de l'application et du matériel dont elle dépend (e.g. le type d'appareil, la taille de l'écran, la puissance du

6. Le Petit Larousse illustré 2010, en l'occurrence.

processeur, la présence **d'un** clavier physique, la version du système **d'exploitation**, etc.);

L'**environnement de l'utilisateur** qui représente les caractéristiques de **l'utilisateur** (e.g. son identité, sa localisation, son âge, son rôle, sa tâche, ses déficiences, etc.);

L'**environnement physique** qui représente les propriétés du monde physique environnant (e.g. la météo, le bruit ambiant, la luminosité, la présence de poussière, etc.).

Dans la littérature, on peut souligner une dimension additionnelle, plus ou moins implicite : le **temps**. Bien que non incluse explicitement dans leur définition, Dey & Abowd [31] synthétisent le contexte comme étant « the **who's, where's, when's and what's** to determine why ». Dès lors, la notion de temps est transverse et la globalité du contexte **est l'ensemble** des états passés et présents des trois environnements susnommés. On parle plus généralement **d'historique du contexte** chez Schmidt et al. [98], voire **d'expériences antérieures** chez Suchman [107], qui détermine **l'appréhension d'une** situation actuelle. À ce titre, nous posons également une définition du terme **situation** tel que nous **l'entendons** à travers ce manuscrit :

Définition. Situation : état précis du contexte **d'interaction** à un instant donné.

Comme nous l'avons souligné en introduction, le contexte **d'interaction** est de nos jours devenu très riche. Il **n'est plus rare qu'un** utilisateur interagisse avec un smartphone, une tablette tactile et un ordinateur portable dans des situations aussi variées que chez soi, au travail ou en déplacement via train, bus ou autre et ce, au sein **d'une** même journée. Ceci a donc un impact non-négligeable sur les usages et **l'IHM** est confrontée à de nombreuses combinaisons contextuelles. On peut donc à juste titre se demander si le contexte impacte **l'interaction** elle-même. La section suivante **s'attèle** à mettre en lumière la place du contexte.

1.1.2 L'interaction en contexte : la *context-awareness*

L'étude de **l'impact** du contexte sur **l'utilisateur** et ses interactions **n'est pas nouvelle**. Cette étude est généralement centrée soit sur **l'utilisateur** (souvent via les sciences cognitives), soit sur le système interactif (plus généralement issue des recherches en informatique).

Vis-à-vis de **l'utilisateur**, Suchman [107] a posé les bases dès 1987 avec ce qui est devenu la **théorie de l'action située**. Dans ses dires, toute action est indissociable de son contexte qui est alors lié aux aspects social, culturel et physique dans **l'instant** de la situation. En ce sens, la façon dont un utilisateur abordera et expérimentera une interaction dépend du contexte.

Du point de vue du système interactif, les mots de Weiser de 1991 [122] font énormément sens **aujourd'hui**. À ce moment, il présageait un avenir proche où les systèmes interactifs seraient si nombreux et intrusifs dans nos vies **qu'un** nouveau paradigme **d'interaction** serait nécessaire. Il introduit ainsi **l'idée d'ubiquitous computing** (francisé informatique ubiquitaire) qui suggère des interactions transparentes. Celles-ci auraient une affordance et une utilisabilité telles que **l'utilisateur** ne se rendrait même plus compte **qu'il**

interagit avec un système. Les interfaces seraient alors cachées aux yeux de **l'utilisateur**, mais non moins omniprésentes.

Ces notions impliquent la même idée sur les systèmes interactifs : ces derniers doivent tenir compte du contexte. Ceci a donc ouvert des perspectives de recherches attribuées au champ de la **context-awareness** (ou dit des systèmes sensibles au contexte). Ce terme, mentionné pour la première fois par Schilit & Theimer [96] définit un système comme étant sensible au contexte si celui-ci repose sur des informations contextuelles (e.g. la localisation de **l'utilisateur** et des personnes **l'entourant**, **l'heure** courante, etc.) et sur la réaction appropriée du système à ces informations dans le but de mieux satisfaire un besoin utilisateur ou **d'améliorer l'efficacité** du système lui-même. **C'est** la définition de Dey & Abowd [31] qui fera consensus en 2001 (traduite de **l'anglais**) :

Définition. Système sensible au contexte : un système est dit sensible au contexte si celui-ci utilise le contexte pour fournir des informations et/ou services pertinents à **l'utilisateur**, où la pertinence est relative à la tâche de **l'utilisateur**.

Dans la pratique, un tel système fait communément appel à des capteurs (e.g. antenne de GPS, capteur de luminosité, microphone, boussole numérique, etc.) ou des ponts logiciels (e.g. horloge numérique, base de données **d'utilisateurs**, introspection sur les propriétés du système, etc.) pour identifier un contexte donné. Après quoi le système infère des décisions dans le but de mener des actions qui amélioreront **l'expérience** utilisateur dans un contexte précis. Ces actions peuvent être de plusieurs formes : **n'afficher** que les données pertinentes, proposer des services adéquats, adapter **l'interface** à la situation, etc. **L'usage** de la sensibilité au contexte trouve de nombreuses applications, dont les suivantes [12][60] :

Les bâtiments intelligents qui consistent à détecter la présence et les actions des utilisateurs à **l'intérieur d'un** bâtiment afin de proposer des services ou de **l'aide**. **L'exemple** typique est la maison médicalisée, où des capteurs sont cachés dans des objets de tous les jours (e.g. pèse-personne intégré au lit, caméra pour la détection de chutes, etc.) afin de suivre un patient tout en lui permettant de garder confort et indépendance chez soi ;

Le commerce en mobilité qui consiste essentiellement à reposer sur la géo-localisation de **l'utilisateur** (i.e. via le capteur de GPS de son smartphone) afin de proposer des services contextualisés (e.g. carte interactive) ou plus généralement **d'afficher** de la publicité en fonction des commerces locaux ;

Le guidage touristique qui utilise la géo-localisation de **l'utilisateur** pour fournir des informations sur les monuments proches et déterminer une route optimale de visite ;

Le traitement et la visualisation de données où le contexte permet de filtrer les informations importantes, **d'inférer** des décisions plus efficaces ou **d'enrichir** des données existantes ;

L'adaptation d'interfaces qui consiste à présenter à **l'utilisateur** des interfaces adaptées à son contexte, passant par des mécanismes de re-modélage ou de re-distribution des interfaces (e.g. un smartphone pourrait ne proposer que des applications de pro-

ductivité si **l'utilisateur** est situé « au travail » ou rejeter tous les appels entrants **s'il** est en réunion).

Ce dernier point nous intéressera plus particulièrement dans la section 1.3 et le chapitre 2.

Concrètement, nous sommes confrontés à de tels systèmes quotidiennement. Les services Google Inc. en sont un bon exemple : les résultats de recherches sont filtrés en fonction de notre localisation et de notre historique de recherches et bon nombre **d'applications** pour smartphone sont contextualisées (e.g. Maps®, FieldTrip®, Now®, etc.).

Résumé de section

Cette section 1.1 se résume par un premier constat :

① Le contexte est un concept important à prendre en compte dans la conception **d'IHM**.

Ainsi, les systèmes sensibles au contexte sont une réponse à la problématique de complexité du contexte, notamment via les interfaces adaptatives.

1.2 La coordination dans le contexte

Notre problématique s'oriente autour de l'**utilisation** des IHM en contexte et plus particulièrement autour de la coordination entre utilisateurs dans une telle situation. Mais nous venons de le voir, la notion de contexte, telle que définie, **n'**englobe pas les aspects **d'une** interaction de groupe. En effet, les différentes dimensions du contexte (i.e. environnements physique, utilisateur et matériel) ne spécifient pas explicitement la situation de l'**utilisateur** au sein **d'une** activité collective. Il est toutefois possible d'étendre la définition du contexte et **d'inclure** la collaboration dans la description de la tâche (englobée dans l'**environnement de l'utilisateur**).

1.2.1 De la collaboration aux outils d'aide à la coordination

Intéressons-nous à la notion de **collaboration**. Par ceci, nous entendons le processus collectif visant à atteindre un objectif commun. En informatique, lorsque l'**on** évoque la collaboration, on se réfère essentiellement au domaine du **Travail Coopératif Assisté par Ordinateur** (TCAO). Pour cadrer les différents aspects de la collaboration, on aborde notamment le sujet suivant le modèle **3C** introduit par Ellis et al. [34] puis, plus récemment, raffiné par Fuks et al. [40] (auxquels nous nous référerons). Ce modèle présente la collaboration comme étant le lien étroit entre les composantes définies comme suit :

Définition. Coopération : action conjointe des intervenants (utilisateurs) consistant à interagir au sein **d'un** espace de travail dans le but de réaliser une tâche. Celle-ci permet, via la **coordination**, **d'atteindre** un objectif plus global. On parle aussi parfois de co-action.

Définition. Coordination : processus qui vise à organiser les actions de coopération. Elle implique de diviser le travail en tâches, **d'ordonner** celles-ci dans le temps et de les répartir parmi les intervenants.

Définition. Communication : moyen servant à échanger et se faire comprendre au sein du groupe. La communication se fait à travers un canal (ou médium) et suit généralement les codes **d'un** langage. Elle est nécessaire à la coopération (que la communication soit directe ou par stigmergie⁷) et permet de fournir les éléments qui composeront l'**acte** de coordination.

Leurs relations sont illustrées en figure 1.1 où l'**on** constate bien le recoupement de ces trois composantes (parfois même représentée sous forme **d'un** diagramme de Venn). Il est à noter que dans la littérature francophone, il ne semble pas y avoir de véritable consensus autour des termes coopération et collaboration, qui se retrouvent souvent comme synonymes (la différence entre les deux étant étymologiquement très mince). Par souci de cohérence, nous nous en tiendrons aux définitions ci-dessus, propres à la littérature TCAO anglophone.

La collaboration est aussi distinguée par deux dimensions importantes : le temps et l'**espace**. À cet titre, la représentation de la **matrice de Johansen** de 1988 [63] fait encore foi

7. Méthode de communication émergente (e.g. le fait **d'observer** une action permet **d'en** inférer des détails que la personne réalisant l'**action** aura indirectement communiqué).

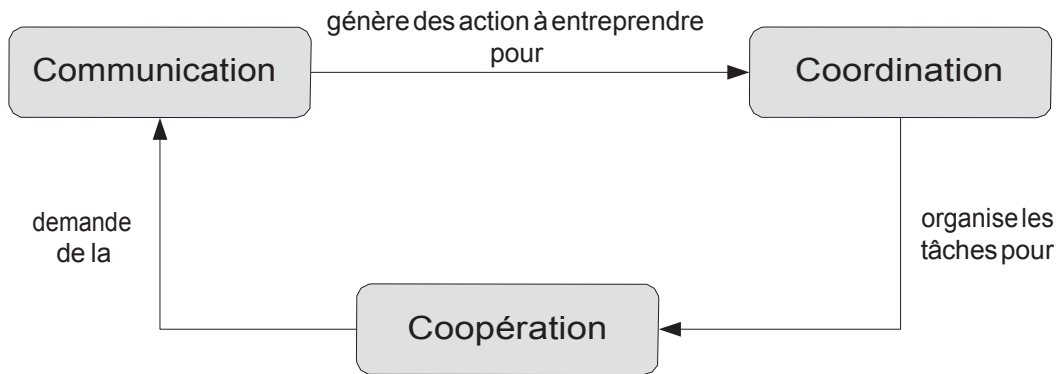


Figure 1.1 – Le modèle 3C vu par Fuks et al. [40]

Table 1.1 – La matrice de Johansen [63] de l'espace et du temps avec des exemples d'outils de TCAO

		Temps	
		Synchrone	Asynchrone
Espace	Colocalisé	Diaporama, support aux réunions, tableau partagé, etc.	Mémos, messagerie, gestion de projet, etc.
	Distant	Vidéo-conférence, discussion instantanée, édition de documents partagée, téléphone, etc.	Collecticiel, courriel, forum, etc.

telle que présentée en table 1.1. Le temps y représente un travail synchrone ou asynchrone entre les utilisateurs, alors que l'espace est lié à la proximité ou non des utilisateurs. Cette matrice fait tout de même l'objet de quelques critiques de nos jours, car des outils de TCAO modernes couvrent souvent plusieurs quadrants.

Dans nos travaux, nous nous concentrons sur la *coordination* bien que, nous le verrons, notre approche répond partiellement aux besoins de communication et de coopération par effet de bord du modèle 3C. Pour autant, il est difficile de situer nos travaux dans la matrice de Johansen, tant les usages sont glissants. Notons cependant que nous ne traitons pas d'IHM relatives aux espaces partagés synchrones (e.g. de type édition de documents partagée) afin de nous focaliser sur l'essence de la coordination dans les IHM.

Plus précisément, nous pouvons considérer la coordination comme le fait de mettre en commun l'état et/ou le résultat d'une tâche avec le reste du groupe afin de converger vers un but d'ensemble et/ou de vérifier la cohérence du travail accompli. Différents outils peuvent d'être utilisés pour supporter la coordination du groupe : les réunions, la messagerie, les processus métiers, les courriels, les tableaux de bord, les outils de plani-

fication, etc. De plus, la coordination peut revêtir différentes formes comme l'a souligné Mintzberg [75] :

La supervision directe où un coordinateur (superviseur, ou supérieur hiérarchique) est chargé de la répartition des tâches et du contrôle des résultats des autres intervenants. Une configuration type serait un groupe chapeauté par un chef de projet qui planifie et gère ses ressources pour répartir les tâches ;

La standardisation des résultats où le but à atteindre est fixé en amont car l'objectif global tend vers une qualité de service ou le maintien d'une performance plutôt qu'une production. La coordination émergera des intervenants qui contribueront localement au résultat. Un fournisseur en télécommunication opérera ainsi ;

La standardisation des procédés où le travail est défini et régulé par des processus à suivre. Ici, la coordination est strictement cadrée par des procédés pré-établis. C'est l'exemple de la bureaucratie ou des chaînes de montage ;

La standardisation des qualifications où ni les procédés, ni les résultats ne peuvent être définis, le travail étant alors spécifié par la profession. La coordination est une conséquence des pratiques métiers. C'est le cas de certaines professions intellectuelles ou libérales ;

L'ajustement mutuel où le travail n'est soutenu par aucune organisation formelle et où la coordination s'opère essentiellement par communication directe. C'est notamment le cas dans le secteur du bâtiment, où sur un chantier chaque intervenant prend individuellement la responsabilité de son travail (ce qui n'est pas nécessairement le cas à d'autres phases d'un projet de construction).

Pour chacune de ces formes de coordination, divers outils pour les supporter peuvent y être associés et sur lesquels nous reviendrons plus en détail en section 1.2.4. Dans nos travaux, nous souhaitons mettre en avant les outils spécifiques à la coordination et nous introduisons le terme d'« outil d'aide à la coordination » que nous définissons comme suit :

Définition. Outil d'aide à la coordination : toute méthode ou système interactif permettant de structurer ou de favoriser les mécanismes de coordination de groupe, un mécanisme étant l'artefact qui médiate la coordination.

Dans la pratique, nous parlons communément de *groupware* (ou collecticiel dans la littérature francophone). Ce type de système a pour but premier de promouvoir le partage de documents et d'informations au sein d'un groupe. Nous préférons parler d'outils d'aide à la coordination qui ont, à nos yeux, un sens plus spécifique que le terme collecticiel qui englobe plus que la coordination. En effet, de nos jours, la notion de collecticiel a évolué en un ensemble d'outils allant jusqu'à la co-édition de documents en temps réel, les espaces partagés, la messagerie, la vidéoconférence, etc. Un collecticiel a donc plus pour vocation d'intégrer l'ensemble du modèle 3C et de la matrice de Johansen, se présentant alors comme le « couteau suisse » de la collaboration. Dès lors, lorsque nous aborderons des travaux relatifs aux collecticiels, nous sous-entendons n'observer que leur composante d'outil d'aide à la coordination.

1.2.2 Pourquoi considérer la collaboration dans le contexte ?

Au vu de notre problématique de conception des outils **d'aide** à la coordination supportant différents contextes, intéressons-nous aux enjeux que cela représente. En ce sens, nous nous baserons sur la littérature propre à la coordination, mais aussi liée aux collectifs, qui englobent les mêmes aspects et nous permet **d'avoir** un recul. Ce recul implique **d'observer** dans un premier temps la collaboration dans son ensemble.

La collaboration représente un contexte particulier qui va au delà des considérations habituelles (i.e. les environnements physique, matériel et utilisateur), notamment lorsque **l'on** aborde le sujet des organisations. Comme le relève Mintzberg [75], une organisation peut être complexe et flexible : les acteurs sont de tous horizons, ont des tâches différentes et leur organisation peut être variable suivant les conditions **d'un** projet. Il identifie cinq types **d'organisations** :

La structure simple où le pôle stratégique est confondu avec le cœur de **l'activité**, parfois coordonné par un supérieur seul. La coordination **s'y** opère essentiellement par ajustement mutuel. Il **s'agit** du type **d'organisation** de **l'artisanat** ou de très petites structures ;

La structure professionnelle où les intervenants ont un métier **d'expertise** et une certaine liberté dans les procédés. La standardisation de ceux-ci **n'est** pas nécessairement forte et les résultats souvent contrôlés. **C'est** le cas général de **l'entreprise** de taille moyenne ;

La bureaucratie où le pôle stratégique est séparé de la production qui, elle, est définie par des procédés stricts. Dans le pôle de production, la coordination est faite via la standardisation des procédés, alors que la stratégie peut se réguler via différentes formes. On peut par exemple parler de taylorisme ;

La structure missionnaire où **l'organisation** est essentiellement un pôle stratégique chapeautant **d'autres** organisations. **C'est** le cas des sièges de grandes structures et la coordination y est généralement dirigée par les résultats et objectifs à atteindre ;

L'adhocratie où les intervenants ont une forte expertise et un haut degré de liberté. Ceci est souvent dû à des objectifs émergents ou lorsque le travail se fait au besoin. Cette structure **n'est** viable que si les intervenants sont hautement qualifiés et responsables. La coordination y est faite par ajustement mutuel. Les **startups** ou certaines grandes entreprises de technologies ont souvent un fonctionnement adhocratique ;

L'arène politique où chaque individu ou sous-structure mène ses propres objectifs en compétition les uns par rapport aux autres. La coordination y est informelle.

Dans sa conception, un collectif doit permettre de supporter ces différents types **d'organisations**. De plus, une organisation est généralement évolutive au fil de sa vie, voire suivant les projets, la culture **d'entreprise** ou le type **d'activité**. Cette structuration dynamique impacte directement les mécanismes de coordination qui ne seront pas les mêmes **d'un** type à **l'autre**. **L'organisation** est donc un facteur déterminant dans la façon dont doivent être abordées les IHM qui soutiendront sa coordination.

En ce sens, Tarpin-Bernard [109] met en avant la nécessité de considérer les usages de la collaboration dès la **conception d'un** système interactif afin de soutenir convenablement

la dynamique de groupe. **L'auteur** va également plus loin en suggérant que la conception de collecticiels doit tendre vers ce **qu'il** appelle la **flexibilité** des IHM. Cette flexibilité doit permettre aux collecticiels de **s'adapter** facilement aux besoins des utilisateurs et de la collaboration. Dans le détail, un système flexible doit pouvoir adapter son IHM au contexte (e.g. en filtrant les informations, en adaptant les dialogues à la situation, etc.) mais aussi à la configuration de **l'activité** collective (e.g. en permettant la définition de procédés, de structurer les accès dans **l'organisation**, etc.). À ce titre, Tarpin-Bernard voit la flexibilité dans les collecticiels à deux niveaux de perspective : le niveau macroscopique (i.e. l'échelle de **l'organisation**) et le niveau microscopique (i.e. l'échelle de **l'interaction** individuelle). Intéressons-nous à ces niveaux du point de vue de **l'IHM** :

Flexibilité macroscopique : niveau de **l'activité** et de **l'organisation**. Le système doit permettre la définition de procédés (e.g. via des **workflow**) tout en permettant une coordination informelle. Ces procédés doivent par ailleurs être évolutifs, une activité étant rarement figée. **L'organisation**, qui est elle aussi dynamique dans le temps et suivant les activités, doit pouvoir être supportée par le système. Par exemple, il est nécessaire de pouvoir structurer le système en groupes **d'utilisateurs**, en rôles, tâches, projets ainsi que supporter la hiérarchie. Il faut également tenir compte de la variabilité des rôles et du cumul de rôles par un même utilisateur. Au niveau macroscopique, on considère également les artefacts produits et consommés (e.g. documents, objets, outils, etc.) et leurs méthodes **d'accès** (régies par les procédés) ;

Flexibilité microscopique : niveau de **l'interaction** et de **l'utilisateur**. Ici, il est question de se focaliser sur les mécanismes qui vont favoriser la collaboration à l'échelle individuelle. Pour cela, il est nécessaire de fournir les informations relatives à **l'activité** et son état à **l'utilisateur**. La flexibilité se situe dans les représentations et les moyens **d'acheminer l'information**. Une même information doit pouvoir être adaptée aux utilisateurs (e.g. un plan de bâtiment avec des symboles électriques fera sens pour un électricien, mais pas pour un plombier) tout en **n'introduisant pas de biais** dans la communication entre utilisateurs sur une même information. L'acheminement de **l'information** doit aussi être adaptatif, en permettant de jongler parmi les différents moyens (e.g. **feedback**, **feed-through**, **awareness**, que nous verrons en section 1.2.4) suivant les situations.

Les nombreux paramètres de la flexibilité complexifient à la fois la conception et **l'utilisation** de ces interfaces. Une surcharge des IHM peut conduire à la perte de **l'attention** de **l'utilisateur** et, par conséquent, introduire un risque de mauvaise coordination, ce qui serait coûteux pour **l'activité**. Le rôle de la flexibilité est alors de réguler **l'information** et de permettre **d'adapter l'IHM** pour maintenir la qualité de la coordination, la performance et permettre aux utilisateurs **d'agir** en contexte (e.g. en tenant compte de leur mobilité et des profils dynamiques).

Shen et al. [101] se concentrent sur les dysfonctionnements de coordination que peut engendrer une mauvaise prise en compte du contexte dans la conception de systèmes. Ils soulèvent notamment des problèmes liés à la communication (e.g. incompréhension entre les utilisateurs lorsqu'ils ne sont pas dans le même référentiel) et la gestion de documents (e.g. problèmes de droits et de politiques **d'accès**). En réponse, ils insistent sur une prise en compte systématique de **l'activité collective** dès la conception du système

mais aussi à l'exécution pour l'utilisateur. Le système doit gérer tous types d'organisations et leurs évolutions (e.g. via des procédés, une gestion polyvalente des droits d'accès, etc.) et présenter la bonne information au bon moment.

Bourguin [16], quant à lui, introduit la notion de *malléabilité* dans les collecticiels qui s'apparente à la flexibilité de Tarpin-Bernard [109]. Il insiste sur la nécessité pour un collecticiel de permettre de s'adapter aux différentes structures d'organisations et d'autoriser utilisateurs et concepteurs à façonner le système à leurs besoins. Dans la pratique, la malléabilité s'opère communément via une approche orientée composants favorisant la modification ou la réutilisation de parties du système. Mais cette notion insiste sur le caractère dynamique d'une activité collective et l'importance de celle-ci dans le contexte d'interaction.

Suivant ces différents auteurs, il nous paraît essentiel de considérer le contexte dans la conception d'outils d'aide à la coordination et d'étendre le contexte à l'activité collective. En ce sens, nous considérons l'activité comme une composante supplémentaire du contexte, à côté des environnements physique, matériel et utilisateur. Bien que cette approche ne soit pas nouvelle, posons une définition de cette composante du contexte telle que nous l'entendons dans nos travaux :

Définition. L'environnement de l'activité : partie du contexte qui représentent la configuration de l'organisation de l'activité collective (e.g. les organisations, leur structure, les processus, les tâches, etc.).

Notons que la tâche de l'utilisateur est déplacée dans l'environnement de l'activité, par consistance des concepts et pour replacer toutes propriétés liées à l'activité au sein d'un même espace. Avec la considération de ce nouvel environnement, nous introduisons le terme de contexte de l'activité collective qui englobe tous les aspects précédents du contexte :

Définition. Contexte de l'activité collective : ensemble des environnements du contexte ayant un impact sur une IHM dans le cadre d'une activité collective. C'est-à-dire les environnements physique, matériel, utilisateur et de l'activité.

Ainsi, en table 1.2 nous récapitulons toutes les composantes du contexte et y replaçons l'activité avec des exemples de propriétés.

Dans cette section, nous avons mis en avant l'importance de considérer l'activité de collective dans le contexte d'interaction. Mais cette étude met aussi en perspective la coordination sous deux angles : une vision théorique à haut niveau (i.e. la description de l'activité et des organisations) et une vision pratique (i.e. les interactions permettant aux utilisateurs de se coordonner). On retrouve cette dichotomie chez Tarpin-Bernard [109] ainsi que d'autres acteurs du TCAO tels que Lonchamp [71]. Dans l'optique de mieux supporter la conception et l'utilisation d'IHM en contexte pour la coordination, nous avons suivi cette même démarche pour bâtir le socle de nos travaux. Une approche qui, nous le verrons en section 1.3.2.1, sied aussi à la méthode de conception envisagée (i.e. l'*Ingénierie Dirigée par les Modèles*). Ainsi, les sections suivantes présentent la coordination sous ces deux points de vue : les théories de la coordination (section 1.2.3) et les outils pratiques (section 1.2.4).

Table 1.2 – Récapitulatif des dimensions du contexte de l'activité collective et leurs propriétés.

Matériel	Physique	Utilisateur	Activité
Taille d'écran, puissance du processeur, présence de clavier physique, version du système d'exploitation, etc.	Température, luminosité, bruit ambiant, localisation, etc.	Age, rôle, compétences, déficiences, etc.	Tâches, organisation, processus, relations, etc.
Temps			
Évolutions matérielles, changement de dispositif, etc.	Historique / prévisions météo, traçabilité de la localisation, etc.	Habitudes, préférences, profilages, etc.	Historique des actions, des documents, de la présence des utilisateurs, des changements organisationnels, etc.

1.2.3 Les cadres d'observation théoriques de la coordination

À un niveau théorique, nous nous intéressons à la structuration d'une activité collective à haut niveau. C'est-à-dire aux organisations, à l'articulation du travail et aux utilisateurs en tant que groupe. À ce titre, il existe un ensemble de travaux visant à étudier la coordination, souvent issus de la psychologie mais aussi de l'ingénierie des outils de TCAO. Nous nous intéresserons plus particulièrement à deux cadres d'observation et d'analyse : la *théorie de la coordination* et la *théorie de l'activité*. Ces théories posent des concepts essentiels et établissent les bases de la conception de systèmes interactifs pour la coordination.

1.2.3.1 La théorie de la coordination

La théorie de la coordination, via Malone et al. [73], est le fruit d'un projet du Massachusetts Institute of Technology qui consistait en l'étude de la coordination dans divers domaines d'application afin d'en tirer une synthèse. Partant de domaines tels que l'informatique, la psychologie, les sciences économiques ou l'étude des organisations (fortement inspirée de Mintzberg [75], voir section 1.2.2), Malone et al. ont isolé un ensemble de concepts et de liens entre eux. Le but de cette synthèse est de proposer un cadre de référence dans le but d'analyser la coordination à un niveau macroscopique en vue de concevoir des systèmes interactifs. Les auteurs de cette théorie distinguent alors les concepts et relations suivants :

Les **tâches** qui constituent une activité et peuvent être composées ;

Les **ressources** produites ou consommées par des tâches ;

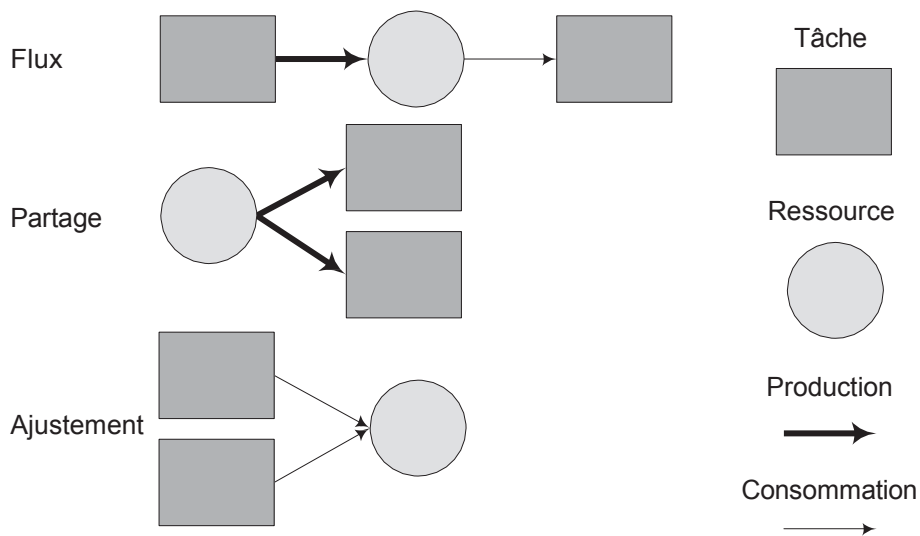


Figure 1.2 – Schéma des types de dépendances élémentaires chez Malone et al. [73]

Les **acteurs** (i.e. utilisateurs) qui réalisent les tâches ;

Les **relations temporelles** qui déterminent la nature de composition des tâches et leur ordonnancement.

Cette théorie établit également des liens élémentaires de consommation et de production de ressources. Ces liens, illustrés en figure 1.2, sont de trois types :

Les **flux** où une ressource produite est également consommée dans une même activité ;

Les **partages** où différentes tâches consomment une même ressource ;

Les **ajustements** où de multiples tâches s'associent pour produire une même ressource.

D'après Malone et al. [73], ces liens élémentaires permettent d'établir des relations plus complexes de temporalité (e.g. une tâche ne pouvant démarrer si une ressource **n'a** pas été produite) ou de distribution entre les tâches. Ceci permet de coller de manière générique à tous types **d'organisations**.

Cependant, Malone et al. **n'offrent qu'une** vue sur la structuration des tâches et de **l'organisation**, la caractérisation des mécanismes ou outils médiatisant la coordination étant laissée libre (ceci **n'ayant pas été l'objectif** initial des auteurs). En effet, la théorie de la coordination, vise plus à schématiser les types **d'organisations** identifiés par Mintzberg [75] plutôt que de caractériser les mécanismes de la coordination. Cette théorie inclut toutefois des notions clés comme les ressources ainsi que **l'interdépendance** des tâches qui siéent bien à une vue **d'ensemble** de **l'activité collective**.

1.2.3.2 La théorie de l'activité

La théorie de **l'activité** représente probablement le cadre de référence le plus utilisé pour qualifier la collaboration, ou du moins celui dont découle le plus de travaux. Issue

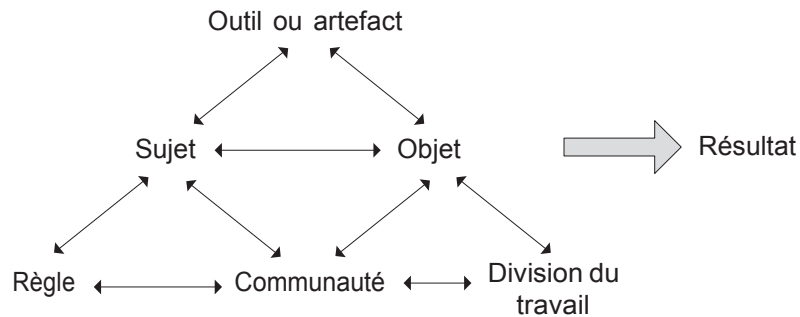


Figure 1.3 – Représentation d'un système d'activité suivant la théorie de l'activité

du champ de la psychologie moderne, cette théorie formalisée par Leontiev en 1978 [69] a surtout été mise en perspective dans le TCAO par Nardi [79] et Kuutti [68] en 1996. Cette approche **s'oriente** autour des concepts **d'activité** (le travail collaboratif) et **d'objet** (le but de la collaboration) afin de produire un **résultat**. Tout comme la théorie de la coordination, elle représente surtout un cadre **d'observation** en vue de comprendre et **d'analyser l'activité** collective. Étudier cette dernière passe alors par la **définition d'un système d'activité** composé des concepts suivants (et illustré en figure 1.3) :

Un **objet** qui représente la nature de **l'activité**, son but, et dont la réalisation tend vers la production **d'un** résultat. Le traitement de **l'objet** est soutenu par des outils et peut être réalisé par plusieurs sujets à travers une division du travail. **L'objet** définit les tâches;

Un **sujet** qui est un individu dont le travail consiste en le traitement de **l'objet** via des outils. Dans le champ du TCAO, il **s'agit** des utilisateurs du système interactif qui outille **l'activité**;

Une **communauté** qui est **l'ensemble** des sujets **œuvrant** pour **l'objet** ou qui en dépendent. Les sujets **d'une** communauté sont régis par des règles et se répartissent le travail pour un même objet. Une communauté est assimilable à une organisation;

Des **règles** qui sont les liens structurels qui déterminent **l'organisation** des sujets au sein **d'une** communauté. Les règles forment la hiérarchie des organisations ou les contraintes qui régissent les liens entre les sujets;

Une **division du travail** qui détermine les tâches pour la réalisation de **l'objet** et répartit le travail dans la communauté. Cette division englobe la coordination du groupe;

Un **outil** (ou artefact) qui médiatise la réalisation de **l'objet**. Dans le TCAO, il **s'agit** du système interactif en lui-même, ou les services **qu'il** propose à **l'utilisateur** pour réaliser les tâches et partager les informations.

Un système **d'activité** donné est lié à l'étude **d'un** seul utilisateur (ou plutôt type **d'utilisateur**, par exemple un rôle dans une organisation) et un seul objet. Le but étant **d'isoler** et **d'analyser** les aspects sociaux et techniques autour **d'un** individu. Il convient

donc d'étendre **l'analyse** à plusieurs systèmes **d'activité** imbriqués pour couvrir toutes les facettes **d'un** travail de groupe. Cette représentation permet de mettre en lumière les interactions sociales, **l'importance** du groupe et la médiatisation de la coordination et des tâches à travers les artefacts (i.e. outils ou mécanismes de coordination).

Vis-à-vis de la conception de systèmes interactifs, l'étude par la théorie de **l'activité** permet de clairement identifier les outils nécessaires à la réalisation **d'une** tâche ainsi que les mécanismes de coordination à mettre en **œuvre** (notamment en termes de communication et d'échanges). **D'un** point de vue concret, les travaux en conception de systèmes reposant sur la théorie de **l'activité** passent souvent par une formalisation sous forme de modèles. Ce point nous intéressera plus particulièrement en section 2.2.1, mais soulignons tout de même des auteurs tels que Bourguin [16] et Garrido et al. [44] qui vont en ce sens.

1.2.3.3 Autres travaux

Au rang des autres cadres d'étude de la collaboration, nous pouvons citer des approches plus orientées vers la psychologie cognitive telles que **l'ergonomie cognitive** [47] ou la **cognition distribuée** [61].

La cognition distribuée [61] est un modèle qui consiste à considérer la collaboration comme étant **l'agrégat** des processus cognitifs de chaque individu impliqué. À l'échelle **d'un** seul individu, la réalisation **d'une** tâche revient à comprendre **l'objet** de celle-ci (via cognition en entrée) et à produire une action (en sortie). **L'ensemble** de la collaboration est alors la **chaîne d'entrées/sorties** entre individus. Par exemple, si **l'on** considère une interaction simple entre une personne A et une personne B avec pour objet de saler un plat, la **chaîne** de cognition pourrait être la suivante : 1) A analyse la situation et demande par voix orale le sel à B ; 2) B analyse la requête et répond gestuellement en présentant le sel à A ; 3) A analyse le geste de B et réagit en prenant possession du sel, puis en salant le plat. La cognition distribuée met alors en valeur les actions, les canaux de communication et **l'enjeu** des outils.

Plutôt **qu'un** cadre propre à la collaboration, **l'ergonomie cognitive** [47] est un pan de la psychologie visant l'étude du travail dans sa globalité. Cette étude est essentiellement centrée sur **l'utilisateur et** vise à faire émerger des critères qui influenceront la qualité des conditions de travail. Ceci passe par **l'analyse** de **l'activité**, des tâches et des systèmes interactifs qui entrent en jeu, le but étant alors **d'améliorer l'ergonomie** de ces derniers pour réduire la charge cognitive de **l'utilisateur**. **L'ergonomie cognitive** met en avant les notions **d'activités**, de tâches, **d'artefacts** (i.e. les outils informatiques) et insiste sur les canaux de coordination, notamment via la communication.

D'un point de vue plus TCAO, la **théorie de l'action/interaction** de Fitzpatrick et al. [38] met en avant les concepts de trajectoires, actions, acteurs, communautés et technologies. Une trajectoire est, dans un sens, un concept similaire à **l'enchaînement** des actions dans la cognition distribuée. Il **s'agit** de la séquence **d'actions** nécessaires à la réalisation **d'une** tâche. Les communautés, quant à elles, sont empruntées à la théorie de **l'activité**, tout comme les technologies qui représente en fait les artefacts médiatisant la tâche.

1.2.3.4 Une synthèse des concepts

Afin de supporter convenablement la coordination dans notre approche, il convient de tirer une synthèse des éléments essentiels. La théorie de **l'activité** [79, 68] pose des bases largement admises, mais l'étude **d'autres** cadres **d'analyse** permet de renforcer cette synthèse et de mieux cerner les tenants et aboutissants de la coordination. Nous notons donc les concepts suivants comme étant essentiels :

Les activités qui représentent les objectifs et la décomposition du travail en tâches ainsi que leur ordonnancement pour **l'obtention d'un** résultat ;

Les acteurs qui sont les individus qui se coordonnent ou coopèrent pour la réalisation **d'une** tâche ;

Les organisations qui représentent les relations entre les acteurs, leur hiérarchie et les règles qui régissent la communication et la coordination des acteurs ;

Les ressources qui représentent les éléments nécessaires à la réalisation **d'une** tâche et qui peuvent être le résultat produit par **d'autres** tâches en amont ;

Les outils qui médiatisent **l'exécution** des tâches et/ou la coordination. Il **s'agit** généralement du système interactif en lui-même.

Nous pouvons également soulever **l'importance** des liens entre ces différents concepts et différencier deux angles dans le support de la coordination : la structuration de **l'activité** (e.g. le type **d'organisation**, la répartition du travail, les points de synchronisation, etc.) et la médiatisation pour **l'utilisateur** (e.g. les outils et canaux pour que les acteurs se coordonnent). Ces éléments nous permettent de fixer un socle théorique au support de la coordination, éléments qui serviront de base aux aspects pratiques, traités en section 1.2.4.

1.2.4 Se coordonner dans la pratique

La seconde facette de la coordination représente le niveau pratique. **C'est-à-dire** le niveau des interactions et des utilisateurs. Cette section s'intéresse par conséquent aux aspects pratiques de la coordination via les outils (ou artefacts) permettant sa mise en **œuvre**. Ces outils permettent généralement **d'instrumenter** les éléments théoriques vus en section 1.2.3. Ainsi, parmi les trois instruments les plus significatifs [71] nous traiterons les deux derniers : la **communication médiatisée**, la **conscience de groupe** et les **flux de travaux**.

1.2.4.1 La conscience de groupe

La **conscience de groupe**, de l'anglais **group awareness**, est un mécanisme cognitif de base dans toutes activités collectives. Énoncée en ces termes pour la première fois par Dourish & Bellotti [33], la conscience de groupe se définit comme suit (traduction depuis l'anglais reprise de chez Lonchamp [71]) :

Définition. Conscience de groupe : compréhension de **l'activité** des autres, qui procure un cadre pour votre propre activité. Ce contexte est utilisé pour **s'assurer** que les contributions individuelles **s'insèrent** dans **l'activité** globale

du groupe, et pour évaluer les actions individuelles au regard des buts et de l'évolution du groupe.

En **d'autres** termes, avoir une vue sur **l'activité** des autres acteurs **d'un** groupe permet de mieux cerner sa propre activité au sein de la dynamique globale. Par exemple, dans une IHM, avoir la possibilité de savoir si un autre acteur est en train d'éditer un document donné permet d'inférer des informations quant à l'état **d'une** activité en lien avec ce document. Comme expliqué par Schmidt [99], la conscience de groupe est une interaction implicite qui consiste à afficher et surveiller les informations disponibles sur le groupe. **C'est** alors le savoir de **l'utilisateur** sur les informations disponibles qui va influencer sur des décisions de coordination.

À la base de ce mécanisme : le gestaltisme. Il **s'agit d'un** courant psychologique qui théorise que la cognition se fait en deux temps : **d'abord** via une analyse **d'ensemble**, puis via une analyse des éléments et de leurs liens. Prenons en guise **d'exemple** une **œuvre** de Giuseppe Arcimboldo, comme **Les Saisons** en figure 1.4. De prime abord, nous percevons des visages de profil, avant de nous attarder sur les détails qui les composent. Le gestaltisme appliqué à la coordination, ou plus particulièrement à **l'interaction** en contexte, a notamment été soulevé par Endsley qui a posé consensus avec la **situation awareness** [35] (conscience de la situation) **qu'elle** définit comme étant « la perception des éléments des environnements à **l'intérieur d'un** espace temps et volumique, la compréhension de leur signification, et la projection de leur état dans un futur proche. » Dans ses travaux, appliqués au secteur de **l'aviation militaire**, cette définition se traduit par **l'appréhension d'une** situation par un acteur vis-à-vis de la situation des membres de son groupe afin de prendre des décisions.

D'un point de vue pratique, nous pouvons prendre **l'exemple** de certains clients de discussions instantanées, tels que Lotus Sametime® **d'IBM** en figure 1.5. Ici, il est possible de **connaître** l'état du groupe, via des icônes décrivant la disponibilité des acteurs, mais aussi leur rôle dans **l'organisation**, leur localisation et éventuellement leur tâche courante. Ces informations ne sont souvent pas critiques pour **l'activité** de **l'utilisateur**, mais par gestaltisme en observant la liste des acteurs connectés, il est possible **d'obtenir** de **l'information** de manière indirecte. Par exemple, si **l'on** édite un document et que **l'on** constate **qu'un** de nos collaborateurs est en réunion au sujet de ce même document, **l'utilisateur** peut choisir **d'attendre** les retours de la réunion pour terminer son document.

Dans le domaine du TCAO, **c'est** surtout Gutwin & Greenberg [52, 53, 54] qui feront suite à Dourish & Bellotti [33] en spécifiant davantage la conscience de groupe pour les collecticiels. Ils décrivent notamment la nature de la conscience de groupe et les mécanismes à travers lesquels elle **s'opère** :

La stigmergie (consequential communication) qui est le fait de voir et entendre les autres acteurs, les informations sont alors transmises implicitement par le décodage de leurs actions (i.e. voir une personne réaliser une action et interpréter ses gestes) ;

Les feedback et feedthrough⁸ qui sont les signaux qui renseignent de l'état **d'un** artefact et la conséquence **d'une** action. **Lorsqu'un** artefact est manipulé, il génère des informations sur son état qui permettent à **l'utilisateur** de le percevoir (e.g. le son de

8. Feedthrough **n'a** pas d'équivalent dans la langue française.



Figure 1.4 – *Les Saisons* de Giuseppe Arcimboldo, une bonne illustration de la cognition par gestaltisme : à première vue des personnages de profil puis, dans le détail, des végétaux de chaque saison.



Figure 1.5 – Lotus Sametime® d'IBM affiche l'état des collaborateurs (i.e. disponible, absent, occupé ou en réunion) via des icônes, fournissant des informations sur la *présence*, l'*identité*, l'*incarnation*, la *localisation* et l'*action* des acteurs du groupe.

papier froissé lorsque l'on vide la corbeille sous un système Windows® nous informe du bon déroulement de l'opération). On distingue les notions de feedback et feedthrough suivant l'origine de l'action en amont. Un feedback est conséquence d'une manipulation de l'utilisateur, alors qu'un feedthrough est conséquence d'un événement non-initié par l'utilisateur tel que l'action d'autrui. En ce sens, un feedthrough est un feedback indirect sur l'activité ;

La communication directe (*direct communication*) qui véhicule tous types d'informations. Qu'elle soit orale, gestuelle, écrite ou autre, il est possible de surprendre une conversation (intentionnellement ou non), d'observer des *outlouds* (i.e. explicitation d'action, par exemple « mais où est ce fichu bouton?! ») ou tout simplement d'être un participant actif.

Gutwin et al. [53, 55] caractérisent également les types d'informations qui peuvent être véhiculés par gestaltisme. Pour cela, ils regroupent les informations générées par le set de questions *qui, quoi, où comment* et *quand*. Chaque catégorie est composée de plusieurs éléments qui étayent ces questions. Cette catégorisation est reprise et synthétisée en table 1.3. Pour illustrer, si nous reprenons l'exemple de Lotus Sametime® en figure 1.5, l'IHM met en avant les propriétés de *présence, d'identité, d'incarnation, de localisation* et *d'action*. Nous pouvons également faire le parallèle avec la vue de Dey & Abowd [31] sur le contexte de l'interaction qui qualifient les questions de *qui, où quand* et *pourquoi*. Ainsi, Dey & Abowd concentrent leur vue sur les interactions avec le système alors que Gutwin et al. se penchent sur l'interaction entre les membres du groupe.

L'espace problème de la conscience de groupe se distingue communément en deux sous-ensembles, un orienté vers la dynamique de groupe et l'autre en lien avec la coordination directe en espace partagé :

La conscience sociale (*social awareness*) [33, 6, 52] qui représente un mécanisme de coordination tacite. Bardram & Hansen [6] la définissent par le fait que les acteurs, « de manière tacite et non importune, alignent et intègrent leurs activités d'une façon transparente et sophistiquée sans s'interrompre les uns les autres. » C'est en tirant des informations sur l'état du groupe par gestaltisme que les acteurs peuvent coordonner leur propre activité individuelle avec celles du reste du groupe. La conscience sociale est donc propre à des tâches individuelles, n'impliquant pas de coopération ;

La conscience de l'espace de travail (*workspace awareness*) [52, 53, 54] qui s'affilie à la coopération synchrone en espace de travail partagé. Gutwin & Greenberg [53] la décrivent comme étant « la compréhension dans l'instant des interactions des autres acteurs au sein de l'espace commun. » Ici, les informations sont essentiellement véhiculées par feedback / feedthrough via des objets ou vues partagés.

Pour illustrer cette distinction, la figure 1.6 expose deux IHM. En (a), une liste des acteurs connectés qui fournit des informations sociales. En (b), un espace de travail partagé qui affiche les pointeurs de souris de tous les acteurs présents sur l'espace (i.e. télépointeurs) et par conséquent, une activité de coopération synchrone propre à la conscience de l'espace de travail. Notons également que cet exemple, issu des travaux de Redondo & Bravo [90], respecte un code couleur afin de maintenir une homogénéité à travers les

Table 1.3 – Caractérisation des informations véhiculées par conscience de groupe suivant Gutwin et al. [53, 55].

Catégories	Éléments	Questions spécifiques
Qui	Présence	Qui est connecté?
	Identité	Qui est-ce?
	Propriété	Qui réalise cette action?
	Incarnation	À quoi ressemble-t-il?
Qui (passé)	Historique des présences	Qui était connecté et quand?
Quoi	Action	Que fait-il? Est-il disponible?
	Intention	Quel est son but, son action courante?
Quoi (passé)	Artefact	Quels objets sont manipulés?
	Historique des actions	Qu'a-t-il fait?
Où	Localisation	Où est-il?
	Regard	Que regarde-t-il?
	Vue	Que peut-il voir?
	Portée	Jusqu'où peut-il voir?
Où (passé)	Historique des localisations	Quel est son trajet?
Comment	Historique des actions	Comment ceci s'est -il produit?
	Historique des artefacts	Quelles modifications a subi cet élément?
Quand	Historiques des événements	Quand cette action s'est -elle produite?

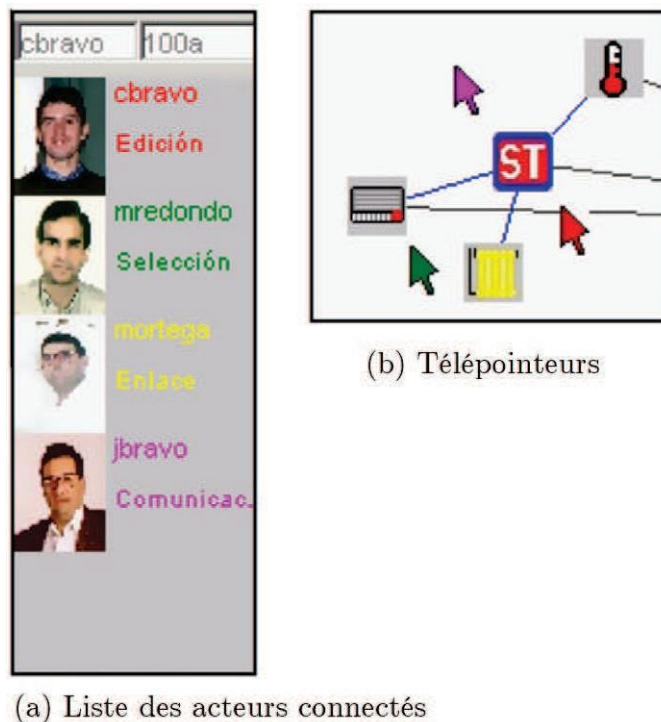


Figure 1.6 – Exemples d’IHM pour la conscience de groupe provenant de DomoSim-TCP [90] : (a) les acteurs connectés donnant une conscience sociale ; (b) pointeurs de souris partagés donnant une conscience de l’espace de travail. Un code couleur maintient l’homogénéité entre les deux. Les captures sont reprises depuis Gallardo et al. [41].

informations de groupe fournies. La conscience de l’espace de travail peut également s’apparenter au concept du *What You See Is What I See* (WYSIWIS) de Stefik et al. [105].

Pour autant, la frontière entre conscience sociale et de l’espace de travail est souvent mince. Si la conscience sociale est bien délimitée, Dourish & Bellotti [33] mettent en avant le fait que la conscience de l’espace de travail contribue également à la coordination sociale. Par exemple, une IHM qui indiquerait quels acteurs consultent un document n’est pas nécessairement liée à une tâche de coopération, mais l’indication peut jouer un rôle social de coordination. Vis-à-vis de nos travaux, nous parlerons donc de conscience du groupe sans véritablement faire cette distinction. Notre problématique s’oriente davantage vers la coordination et la communication que la coopération. Nous ne traitons par conséquent pas de la conscience de l’espace de travail dans le cas des espaces partagés.

La conscience de groupe est une composante importante de la coordination dans les IHM. Comme souligné par Schmidt [99], elle est implicite et exploite des informations déjà présentes dans les systèmes interactifs. En ce sens, c’est un outil qui permet d’optimiser la coordination et qui permet parfois de se substituer à une coordination informelle (i.e. ajustement mutuel).

Il est également intéressant de faire le parallèle entre IHM sensibles au contexte et conscience de groupe, que Heath et al. [56] définissent comme étant « une vue sur le contexte des autres membres du groupe. » Sous cet angle, la conscience de groupe peut alors être vue comme une sensibilité au contexte distribuée, renforçant la pertinence du

contexte dans les IHM.

1.2.4.2 Les flux de travaux

Les flux de travaux (ou *workflow* en anglais), sont un ensemble d'outils et de formalismes servant à décrire et mettre en œuvre des procédés métiers (*business process*). Ils sont l'instrumentation des procédés telle que définie chez Mintzberg [75] (voir section 1.2.2) et ont pour vocation de structurer et organiser le travail [71]. La standardisation des procédés métiers est venue des besoins de l'industrie d'optimiser la productivité et la qualité de production, trouvant ainsi son origine dans le taylorisme. Les flux de travaux s'appliquent bien aux activités répétitives ou règlementées, comme c'est le cas dans l'industrie ou les administrations. Cet outil permet notamment de délimiter la structuration de l'organisation et de séquencer le travail dans le temps et à travers les acteurs.

Une activité est scindée en plusieurs tâches atomiques (en termes de rôle et d'instant) qui sont reliées à travers un enchaînement temporel. Les flux peuvent être soumis à des points de contrôles qui déterminent par exemple si des tâches peuvent être menées en parallèle ou les conditions nécessaires au début d'une tâche. On distingue communément les concepts suivants (et illustrés en figure 1.7) :

Pool (ou piscine) qui représente une activité de manière globale. Elle contient des *swimlanes* ;

Swimlane (ou couloir) qui représente la séquence de tâches propre à un acteur, ou type d'acteur (généralement par rôle) ;

Task (ou tâche) qui représente une tâche atomique au sein de l'activité et associée à une *swimlane* ;

Flow (ou flux) qui relie les tâches entre elles de façon unidirectionnelle dans le temps ;

Event (événement) qui signalent les conditions de début d'une tâche ou celles activées par la fin d'une autre ;

Gateway (porte) qui contrôle l'enchaînement logique des tâches (e.g. tâches parallèles, exclusives, etc.).

Les flux de travaux peuvent être représentés via une multitude de formalismes, que ce soit via des diagrammes d'états-transitions ou d'activité UML⁹, des réseaux de Petri [119], des *flowcharts* ou via des langages de modélisation plus spécifiques [71, 118]. Toutefois, ce sont deux standards industriels qui font aujourd'hui référence dans le domaine : *Business Process Model and Notation*¹⁰ (BPMN) qui est un standard maintenu par l'*Object Management Group* (OMG) et *XML Process Definition Language*¹¹ (XPDL) maintenu par la *Workflow Management Coalition*. Ces deux standards sont toutefois compatibles l'un avec l'autre, XPDL étant une sérialisation XML de BPMN qui lui est avant tout une représentation graphique (voir figure 1.7). Nous pouvons également mentionner le langage *Business Process Execution Language*¹² (BPEL) standardisé auprès de l'*Organization for*

9. <http://www.uml.org/>

10. <http://www.bpmn.org/>

11. <http://www.wfmc.org/xpdl.html>

12. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>

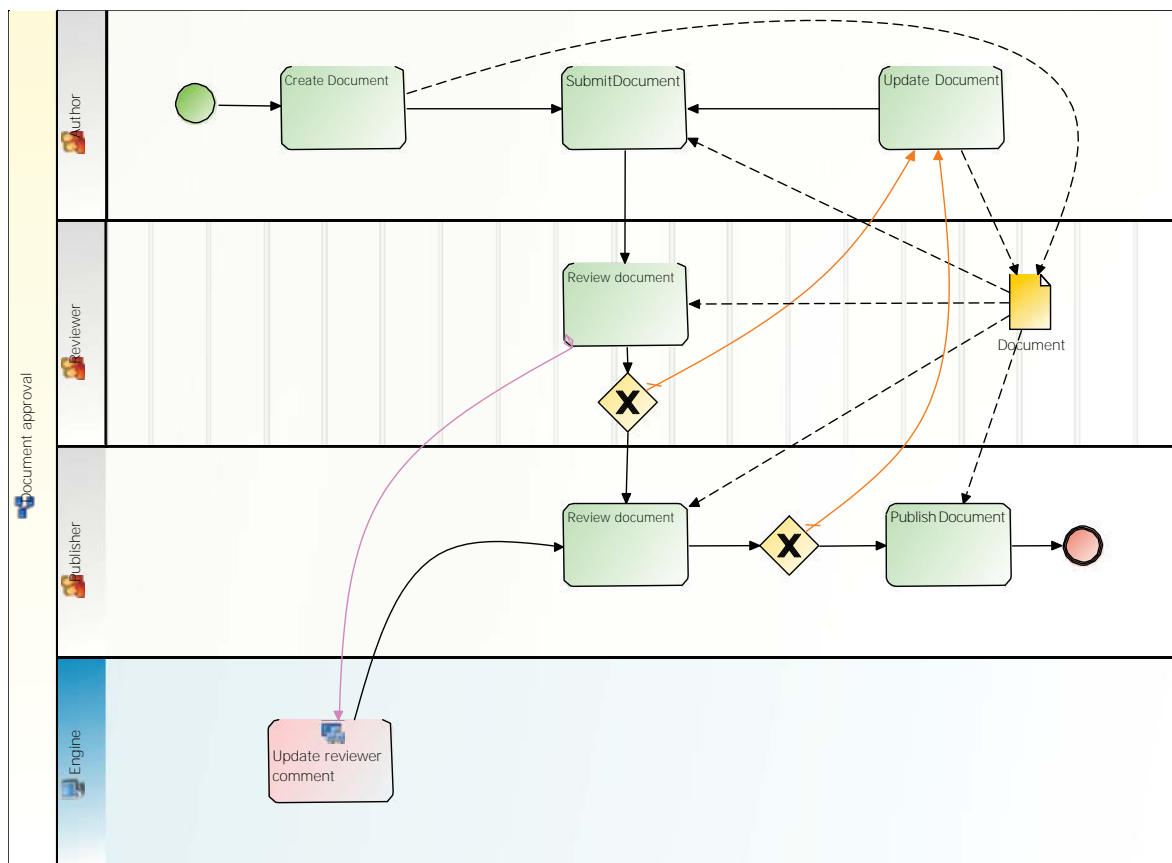


Figure 1.7 – Exemple de flux de travaux exprimé en BPMN. Ici, un procédé d’approbation de documents mettant en jeu une *pool* et quatre *swimlanes*.

the Advancement of Structured Information Standards (OASIS) qui permet la définition de flux de travaux avec une emphase sur l’orchestration de services web.

Au-delà de la représentation, l’intérêt des flux de travaux est qu’ils peuvent être automatisés par des systèmes qui en font l’interprétation. À la manière d’une machine à état, un système interactif peut tirer parti de flux pour gérer l’accès aux IHM suivant les rôles et l’état de l’activité. Ils ont aussi un rôle de consultation de cet état et permettent donc aux décideurs d’avoir une vue instantanée sur une situation. Ils ont aussi l’intérêt de favoriser l’interopérabilité inter-organisationnelle via les standards susnommés.

Dans la pratique, les flux de travaux sont souvent supportés directement dans les collecticiels, comme c’est le cas dans la solution EGroupware¹³. Cet outil étant très formel, il ne convient toutefois pas à tous les types d’activités collectives ou organisations. Les flux ne permettent par exemple pas de soutenir l’ajustement mutuel qui, par nature, est très informel.

13. <http://www.egroupware.org/index.php>

Résumé de section

Cette section 1.2 se résume par un second constat :

② La collaboration représente un contexte bien plus complexe que les considérations habituelles du contexte **d'interaction** en IHM et il est essentiel de pouvoir adapter **l'IHM** en fonction du contexte, dont **l'activité** collective, pour supporter au mieux le travail de groupe.

En vue de considérer la coordination dans **l'adaptation** au contexte, un rapprochement entre théorie et pratique nous permet de mieux définir les paramètres qui entrent en compte. Cette perspective est résumée en figure 1.8 et sera reprise au cours de ce document car celle-ci sied à **l'approche** de conception envisagée.

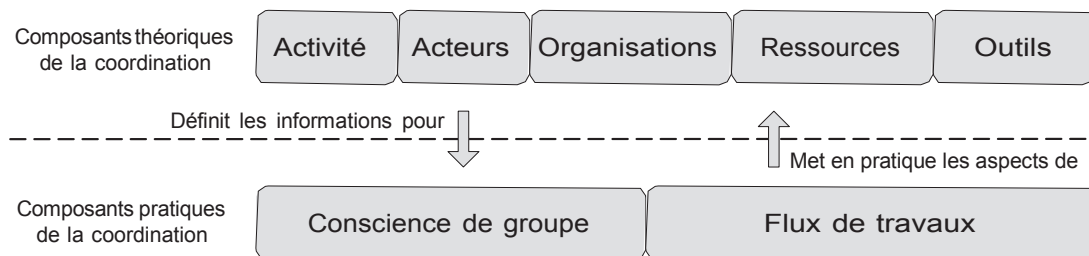


Figure 1.8 – Résumé de la vue par dichotomie sur la coordination : les aspects théoriques et pratiques.

1.3 Du contexte à la génération des IHM adaptatives

Nous l'avons vu dans les sections précédentes, le contexte **d'interaction d'une** IHM joue un rôle capital, que ce soit lors de la phase de conception ou lors de l'interaction elle-même. La section 1.1 pose les bases de l'interaction en contexte et met en avant les possibilités ouvertes aux IHM. En section 1.2, nous avons relevé le fait **qu'en** milieu collaboratif, le contexte est une notion plus complexe. Certains auteurs [109, 32, 48, 101, 16] mettent en avant la nécessité de prendre en compte l'**activité** dans le contexte, voire de proposer des IHM flexibles, adaptables ou adaptatives.

Se pose alors la question de la conception et de l'**exécution d'IHM** qui tiennent compte du contexte, notamment celui de l'**activité** collective. Cette section met en avant un type **d'IHM** qui répond à ces besoins, soit les IHM adaptatives, ainsi que leurs méthodes de conception associées.

1.3.1 De la personnalisation des IHM à leur plasticité

Lorsque nous avons décrit les systèmes sensibles au contexte en section 1.1, nous avons souligné une possibilité particulière de ces derniers : l'**adaptation** des interfaces. L'**adaptation** des IHM est un besoin qui remonte au début de l'**informatique** moderne et des premiers systèmes grand public. Les IHM se faisant de plus en plus complexes et utilisées par des publics de plus en plus variés, il est dès lors nécessaire **d'en** améliorer l'**utilisabilité** afin de compenser cette complexification.

1.3.1.1 Les systèmes adaptatifs

Une des premières méthodes visant à améliorer l'**utilisabilité** des IHM et à prendre en compte les différences entre utilisateurs fut l'**introduction** des IHM **adaptables** (ou personnalisables suivant la littérature). Un des premiers exemples **d'IHM grand public** adaptable est le logiciel Photoshop® **d'Adobe** (voir figure 1.9) où chaque élément de l'**interface** peut être manipulé et ajusté à la guise de l'**utilisateur**. Cependant, les travaux sur ces IHM ont rapidement évolué vers des interfaces capables de se personnaliser **d'elles-mêmes**, comme notamment décrites par Greenberg et al. [49] ou Trevelyan & Browne [116]. Ces systèmes plus évolués, dits capables **d'adaptabilité**, représentent la faculté des logiciels de **s'auto-personnaliser** au lancement suivant un ensemble de paramétrages, ou une configuration spécifique (voir Stephanidis et al. [106]). Ils ont précédé l'**adaptivité** des IHM qui a apporté une personnalisation à l'**exécution**, permettant ainsi une adaptation fine et automatique suivant les habitudes de l'**utilisateur** (e.g. des menus rétractables suivant les options les plus utilisées, des contrôles adaptés suivant le dimensionnement donné par l'**utilisateur**, etc.) [106, 28, 100]. Pour expliciter ces différentes possibilités, Totterdell & Rautenbach [115] et Browne et al. [19] posent les définitions suivantes (traduites par Thevenin [110]) pour qualifier un système interactif :

Définition. Adaptable : se dit d'un système personnalisable sur intervention explicite de l'**utilisateur** qui peut agir sur les paramètres fixés par le concepteur.

Définition. Adaptatif : se dit d'un système doué de discrimination. Il sait reconnaître la situation (parmi plusieurs déclencheurs fixés par le concepteur)

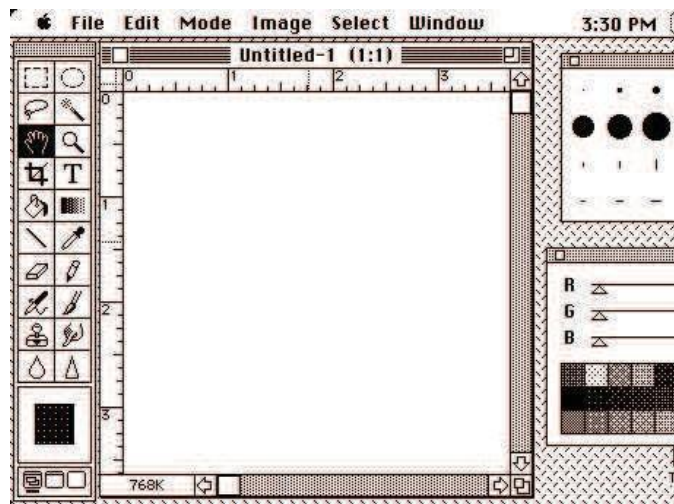


Figure 1.9 – L’interface d’Adobe Photoshop® 1.0, parmi les premières IHM personnalisables grand public. Chaque élément peut être déplacé, redimensionné ou masqué.

et adopte la réaction (recommandation) fixée elle aussi par le concepteur pour cette situation. Mais le système est incapable d’évaluer l’effet de sa réaction.

Définition. Automodificateur : se dit d’un système doué de généralisation qui, par ses méta-connaissances (n.d.a. les données du contexte accessible au système), a la capacité d’apprendre de nouveaux déclencheurs et de nouvelles réactions.

Définition. Autorégulateur : se dit d’un système doué de facultés d’auto-évaluation. Non seulement il reconnaît la situation parmi les déclencheurs prévus par le concepteur, mais utilise, pour effectuer son choix de réaction, une fonction de feedback sur lui-même par essai-erreur. Autrement dit, il utilise une métrique qui évalue l’effet de la réaction retenue précédemment pour ce même déclencheur.

Définition. Automédiateur : se dit d’un système qui a les caractéristiques d’autorégulation, mais en plus, est capable de résoudre le problème de l’adaptation par planification et peut évaluer a priori l’effet d’une réaction (alors qu’un système autorégulateur n’évalue l’effet qu’a posteriori).

Ces différents types de systèmes considèrent généralement l’utilisateur et ses habitudes comme seuls paramètres de contexte dans l’adaptation. Cependant, depuis les années 2000 avec l’arrivée des systèmes sensibles au contexte [31], des recherches autour de la prise en compte de toutes les composantes du contexte dans les IHM adaptatives se sont multipliées. Ceci, notamment via l’introduction de la notion de plasticité des IHM [110, 111] qui nous intéresse plus particulièrement dans ces travaux de thèse.

Une IHM plastique peut, suivant les approches, se comporter comme un système automodificateur, autorégulateur, voire automédiateur. Telle que définie dans son cadre de référence [22], la plasticité n’explicite pas l’usage systématique d’une auto-évaluation de l’IHM mais laisse la voie ouverte.



Figure 1.10 – Exemple d’une IHM plastique. Ici, un système de contrôle de chauffage adapté à trois cibles différentes, notamment impacté par les spécificités matérielles. Figure adaptée depuis [113].

1.3.1.2 La plasticité des IHM

La **plasticité** des IHM est un domaine de recherche qui se présente au croisement des systèmes adaptatifs et des systèmes sensibles au contexte. Introduit par Thevenin & Coutaz [110, 111, 113, 114], cette notion met en avant la capacité **d’une** IHM à **s’adapter d’elle-même** au contexte (tel que défini par Dey [31]) tout en préservant son utilisabilité. La plasticité est définie comme suit [21, 20] :

Définition. Plasticité : une IHM est dite plastique pour un ensemble de contextes C si, donné une propriété P, elle peut **s’adapter** à tous changements dans C en préservant P.

Cette approche promeut une considération systématique du contexte **d’interaction**, et ce, dès la conception de **l’IHM**. Pour cela, un contexte est représenté sous la forme **d’un** triplet **<utilisateur, environnement physique, matériel>**, soit une représentation basée sur la définition de Dey [31]. Ce triplet représente une **cible** que **l’IHM** devra supporter, plusieurs cibles formant alors **l’ensemble C** de la définition. Ainsi, la plasticité se base sur des ensembles **d’utilisateurs** et de situations stéréotypés, et ne tient pas compte des micro-changements du contexte. Cette approche permet **d’avoir** un contrôle poussé sur **l’impact** du contexte sur **l’IHM** en maîtrisant un ensemble de cas définis a priori, et donc **d’assurer** le respect de la propriété P (qui est généralement associée à des critères de qualité ou **d’utilisabilité** tels que ceux établis par Nielsen [80] ou Bastien & Scapin [94]).

Ainsi, pour une IHM et des cibles données, la plasticité permet la production **d’interfaces** adaptées à chaque cible et permet aussi la transition de **l’IHM d’une** cible à **l’autre**. La figure 1.10 met en avant une IHM plastique qui **s’adapte** à **l’espace** disponible (images de gauche et du milieu) et au matériel (image de droite).

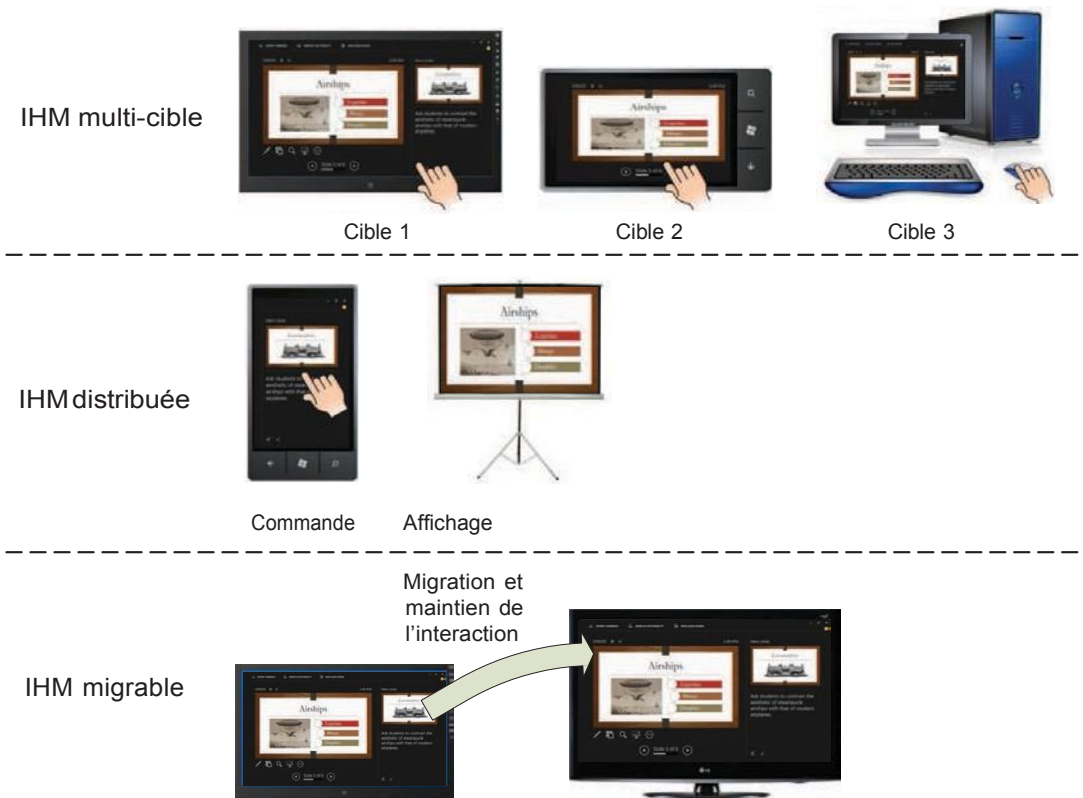


Figure 1.11 – Les trois facettes de la plasticité des IHM : multi-ciblage, distribution et migration.

Afin de répondre au paradigme **d’informatique** ubiquitaire énoncé par Weiser [122], la plasticité produit trois grands types **d’interfaces** adaptatives :

Interfaces multi-cibles : interfaces capables de **s’adapter** à plusieurs triplets <utilisateur, environnement physique, matériel> ;

Interfaces distribuables : interfaces capables de se re-distribuer à travers les appareils proches identifiés dans une cible ;

Interfaces migrables : interfaces capables de migrer **d’un** appareil à un autre en cours **d’utilisation**.

Le paradigme de la plasticité consiste alors, pour une interface, à pouvoir suivre **l’uti-**lisateur à travers toutes les cibles et les appareils concernés comme illustré en figure 1.11.

1.3.2 Conception et génération d’IHM adaptatives

Dans cette section, nous détaillons davantage la notion de plasticité et plus particulièrement la génération **d’IHM** adaptatives via **l’Ingénierie Dirigée par les Modèles (IDM)**. Celle-ci véhicule des concepts clés pour la proposition de cette thèse et qui feront écho à

notre vue sur la coordination. Nous nous intéressons ensuite à son lien avec la plasticité en introduisant la notion de génération **d'IHM** et les différents niveaux **d'abstraction d'un** cadre de référence lié à la plasticité, à savoir CAMELEON [21, 113, 23]. Enfin, nous referons un passage par la coordination pour remettre la vue par dichotomie de la coordination (voir figure 1.8, page 35) en parallèle de **l'approche** générative des IHM.

1.3.2.1 Ingénierie dirigée par les modèles

L'Ingénierie Dirigée par les Modèles (IDM), ou *Model-Driven Engineering* (MDE) en anglais, est une méthode de conception logicielle qui consiste à représenter les concepts informatiques via des modèles et des métamodèles en lieu et place des spécifications techniques (e.g. algorithmes, programmation, etc.) [36]. En utilisant des représentations accessibles à tous les experts liés à une production logicielle (e.g. chef de projet, développeurs, concepteurs, utilisateurs, etc.), **l'IDM** entend faciliter la communication entre les différentes parties et à réduire l'écart entre conception et production. Pour cela, **l'IDM** définit un ensemble de relations et **d'opérations** entre modèles (depuis Favre et al. [36]) :

Système : un système est un mot abstrait pour référer à ce qui est actuellement à l'étude ;

Modèle : un modèle est une abstraction **d'un** système et est construit dans un but particulier. Un modèle doit être viable vis-à-vis des questions que **l'on** se pose sur le système modélisé ;

Relation « est une représentation de » : lie un modèle à un système, un modèle étant une représentation **d'un** système. Un système peut être un modèle représentant un autre système (dans ce cas, le système est lui-même un modèle). Un système **n'est** pas nécessairement un modèle ;

Relation « est un élément de » : des systèmes peuvent être regroupés sous forme **d'ensemble**. Les systèmes sont alors des éléments de cet ensemble. Un ensemble peut être un élément **d'un** autre ensemble. Les éléments **d'un** ensemble ont une existence propre en dehors de celui-ci ;

Relation « est inclus dans » : à la manière des éléments, un système peut être inclus dans un ensemble. Dans le cas **d'une** inclusion, si **l'ensemble n'est** plus, ses constituants disparaissent ;

Métamodèle et relation « est conforme à » : un métamodèle est un modèle qui est une représentation **d'un** ensemble de modèles. Un modèle est dit conforme à un métamodèle si il est un élément de **l'ensemble qu'il** représente. En termes plus généraux, un métamodèle **décrit l'ensemble** des règles qui permettent de construire un modèle qui lui est conforme ;

Relation « décomposé en » : un système peut être décomposé en plusieurs systèmes ;

Transformation : une transformation peut être assimilée à une fonction qui produit un système, donné un autre système ;

Relation « est transformé en » : un système A peut être transformé en un système B via une transformation. Il est alors dit de B **qu'il** est une instance de la transformation appliquée ;

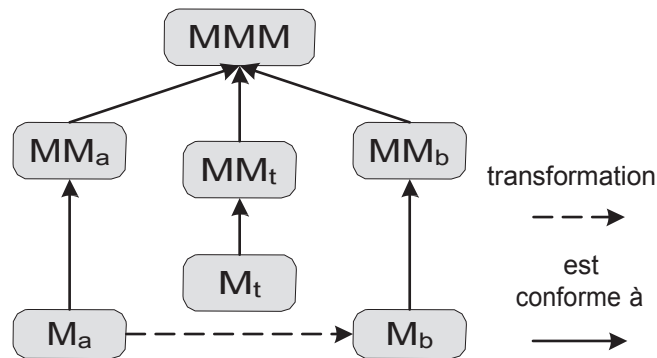


Figure 1.12 – Les liens entre modèles (M), métamodèles (MM), langage (MMM) et transformation dans l’IDM.

Modèle de transformation : un modèle de transformation est un modèle qui est la représentation d’une transformation.

Ces différentes relations permettent à l’IDM de définir des langages et des grammaires pour décrire tous types de systèmes. Dans notre cas, nous nous intéressons surtout aux notions de modèles, de métamodèles et de transformations qui sont à la base de la génération d’IHM et de la plasticité. Les transformations permettent notamment de passer de représentations abstraites à des représentations plus spécifiques, jusqu’à production d’une IHM utilisable. La conception d’une IHM revient alors à une chaîne de transformations de modèles. La figure 1.12 met en lumière les relations entre ces concepts. On distingue la nécessité d’avoir un métamodèle commun à tout le système, que l’on appelle aussi méta-métamodèle, ou langage de modélisation (MMM sur la figure 1.12). De ce langage, peuvent alors être dérivés des métamodèles (MM_a et MM_b) ainsi que des métamodèles de transformation (MM_t). Chaque métamodèle permet de définir un ensemble de modèles (M_a , M_b et M_t). L’avantage d’un langage commun (MMM) est de faciliter la formalisation des transformations, car ainsi elles manipulent le même ensemble de règles pour transformer des modèles conformes à différents métamodèles.

La définition de modèles n’est toutefois pas indépendante et ne prend de sens qu’à travers un langage de modélisation (MMM sur la figure 1.12). Nous pouvons par exemple citer l’*Unified Modeling Language*¹⁴ (UML) ou le *Meta-Object Facility*¹⁵ (MOF) comme représentants courants de langages de modélisation.

Les intérêts de l’IDM sont multiples. Ils permettent notamment de réduire le biais entre conception et implémentation, les modèles produits lors des phases de conception pouvant être exploités sans ré-interprétation (contrairement à une méthode d’ingénierie logicielle par programmation). Ceci permet également de garder une trace des concepts de haut niveau et donc de ne pas perdre d’information (une propriété qui nous sera importante pour ne perdre aucune information sur le contexte lors de la génération d’une IHM).

14. <http://www.uml.org/>

15. <http://www.omg.org/mof/>

Un autre intérêt de l'IDM est de capitaliser les savoirs et les savoirs-faire, les méta-modèles représentant une synthèse du savoir-faire des concepteurs et les modèles qui en sont les instances expriment le savoir. Dans cette même logique, le paradigme de modèle facilite la ré-utilisation et le partage des concepts modélisés.

1.3.2.2 Cadre de référence CAMELEON pour la génération d'IHM adaptatives

Comme nous l'avons souligné en section 1.3.1, les travaux sur la plasticité des IHM prennent généralement le paradigme de l'IDM. Par ailleurs, ce processus a fait l'objet de la formalisation d'un cadre de référence pour la conception d'IHM plastiques : CAMELEON¹⁶ (*Context Aware Modelling for Enabling and Leveraging Effective interaction*). Ce cadre, défini par Calvary et al. [21, 113, 23], fait aujourd'hui l'objet d'un effort de standardisation¹⁷ auprès du *World Wide Web Consortium*¹⁸ (W3C).

De manière plus générale, CAMELEON fait ainsi partie de la famille des *Model-Based User Interface* (MBUI). Le principe des MBUI reprend le paradigme de l'IDM qui consiste à focaliser la production sur des représentations abstraites qui seront spécialisées en IHM. Ces représentations concernent le plus souvent le cœur fonctionnel de l'interaction, c'est-à-dire une abstraction des tâches que l'IHM doit pouvoir permettre de réaliser. Parmi les premiers systèmes de conception d'IHM basés sur les tâches, nous pouvons citer TA-DEUS [97], DIANE+ [108], TRIDENT [120], TEALLACH [50] ou encore ADEPT [64].

Le cadre de référence CAMELEON s'articule autour des types de modèles suivants :

Modèle du contexte d'interaction qui permet de définir des cibles sous le triplet <utilisateur, environnement physique, matériel> ;

Modèle du domaine qui représente les concepts manipulés par le système (e.g. date, propriété, objet, etc.). Il consiste communément en un modèle relationnel de données (e.g. entités-associations) ;

Modèle de tâches qui représente l'enchaînement des interactions nécessaires pour réaliser une tâche et les concepts du domaine qu'il manipule. Il s'agit des aspects fonctionnels de l'IHM. Cette représentation prend généralement la forme d'un arbre de décomposition où chaque nœud est une interaction élémentaire sur un concept du domaine (e.g. entrer une date, cliquer sur un bouton, etc.). Les feuilles et branches d'un arbre de tâches sont également liées temporellement (e.g. tâches parallèles, séquentielles, etc.). A ce titre, plusieurs formalismes sont couramment utilisés, comme ConcurTaskTrees (CTT) de Patterno et al. [84] ou MAD de Scapin et al. [93]. Ces deux formalismes ont la même expressivité concernant la modélisation de tâches individuelles et utilisent tous deux des opérateurs LOTOS [117] pour définir les contraintes temporelles et logiques existant entre deux branches d'un arbre de tâches. La figure 1.13 donne l'exemple d'un modèle de tâche ConcurTaskTrees ;

16. <http://giove.isti.cnr.it/projects/cameleon.html>

17. <http://www.w3.org/2005/Incubator/model-based-ui/XGR-mbui-20100504/>

18. <http://www.w3.org/>

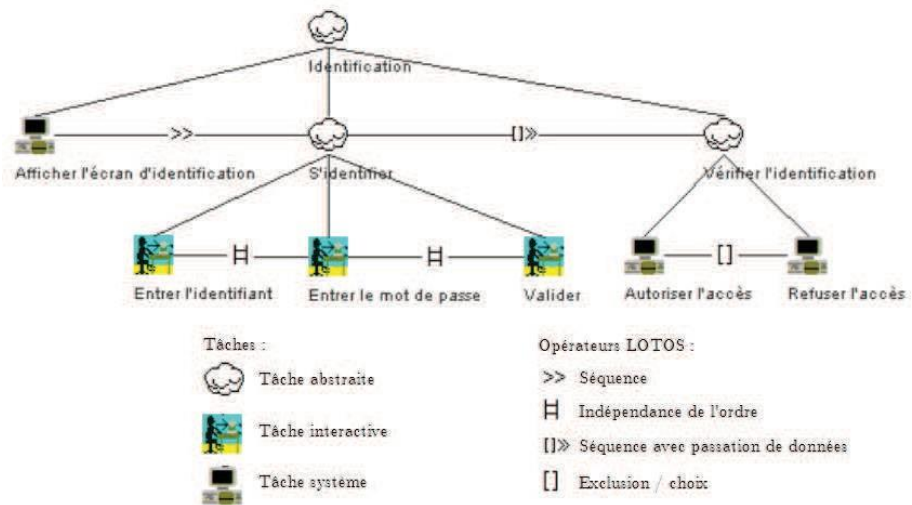


Figure 1.13 – Exemple de modèle de tâche réalisé en ConcurTaskTrees [84]. Ici, la tâche d'identification d'un utilisateur par identifiant et mot de passe.

Modèle d'interface abstraite ou *Abstract User Interface* (AUI), qui consiste en la présentation de l'IHM de manière abstraite, indépendamment de la cible, de tout dispositif et de contrôle. En somme, il s'agit d'une représentation de l'agencement général de l'IHM et de son séquençage (temporel et spatial) vis-à-vis de la tâche supportée. Par exemple, une tâche peut être répartie à travers plusieurs fenêtres et sous-espaces. La figure 1.14 donne l'AUI correspondant au modèle de tâches en figure 1.13 ;

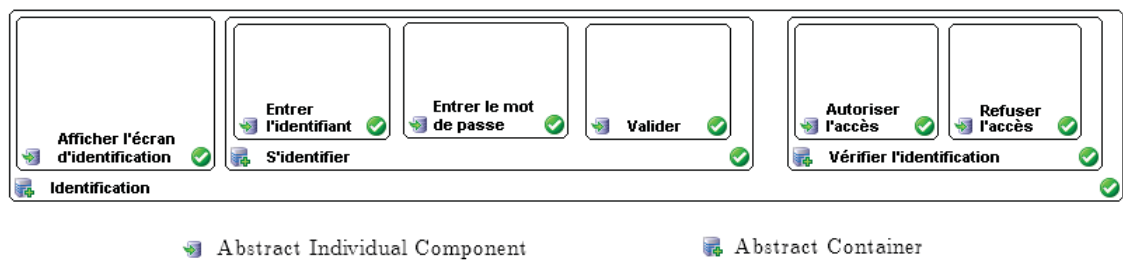


Figure 1.14 – Exemple de modèle d'AUI tel que modélisé dans l'outil IdealXML [77].

Modèle d'interface concrète ou *Concrete User Interface* (CUI), qui représente l'IHM et l'ensemble de ses composants et dialogues, mais indépendamment de toutes technologies d'exécution. La figure 1.15 illustre une CUI définie par l'exemple précédent ;

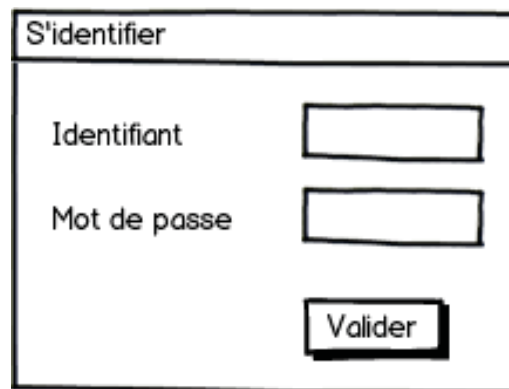


Figure 1.15 – Exemple de modèle de CUI représenté sous la forme d'une maquette.

Modèle d'interface finale ou *Final User Interface* (FUI), qui est l'interface utilisable et exécutable. Elle est liée à une technologie d'exécution et au dispositif matériel de la cible concernée. La figure 1.16 illustre deux FUI adaptées à des cibles différentes.



Figure 1.16 – Exemple de FUI sur deux cibles différentes. À gauche avec un matériel de type smartphone et à droite avec un matériel de type tablette tactile.

La conception d'une IHM à travers ces modèles se fait alors via une succession de transformations de modèles. En premier lieu, il convient de définir les différentes cibles à supporter via la définition de modèles de contexte, puis du modèle de tâches et de domaine. Partant de ces modèles abstraits, les spécifications fonctionnelles de l'IHM sont ainsi définies. La production de l'IHM se fait alors par transformations vers des modèles toujours plus spécifiques. L'arbre de tâches est transformé en AUI qui sera transformée en CUI. L'interface concrète donne alors lieu à l'interface finale, propre à une cible. Lors de chacune de ces transformations, les modèles de contexte influent sur le processus afin de produire une IHM pour chaque cible, qui sera alors adaptée. Lorsque l'on évoque ce processus de transformation qui conduit à l'IHM finale, nous parlons communément de *génération d'interface*. CAMELEON et son processus de génération sont repris en figure 1.17.

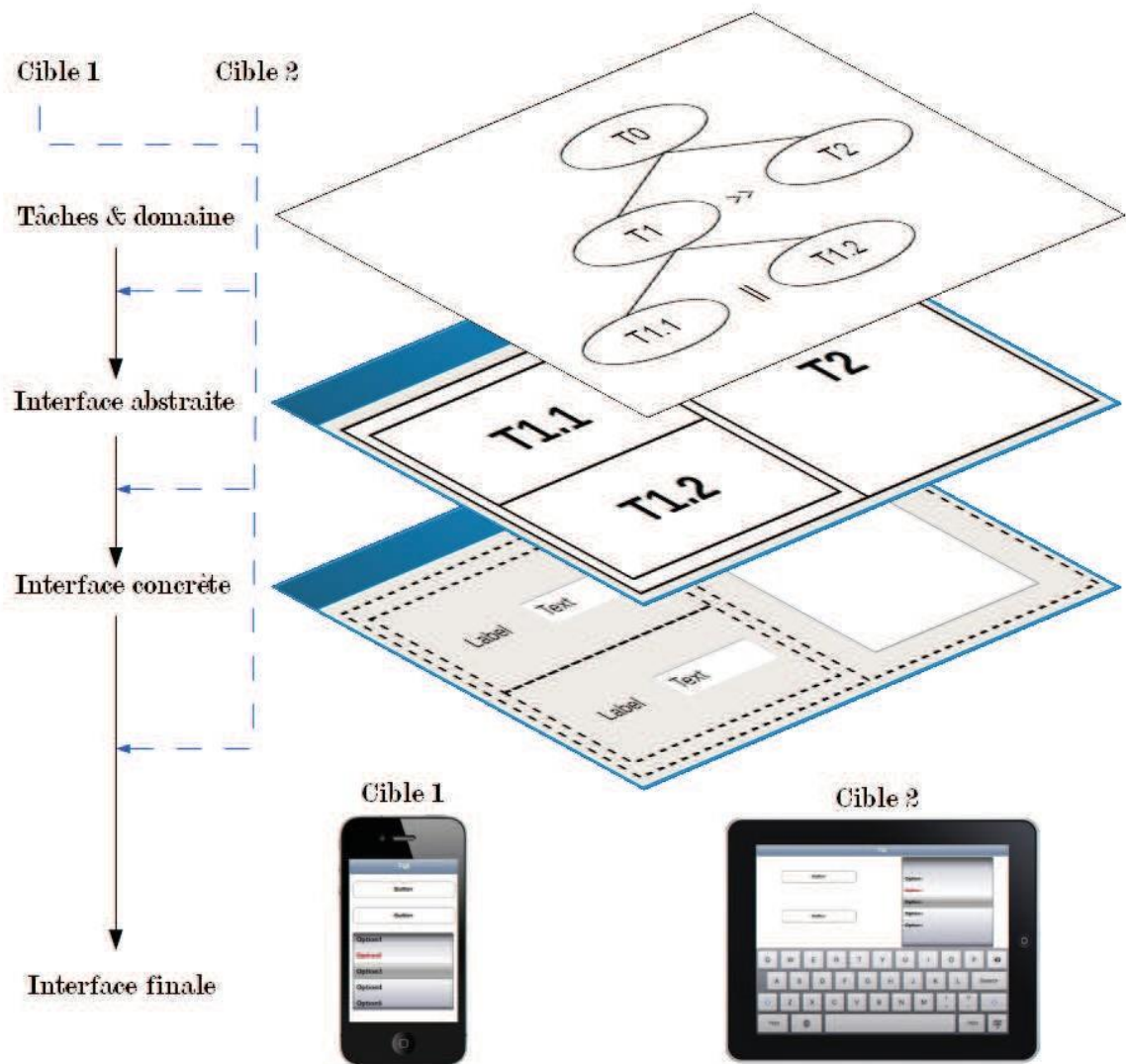


Figure 1.17 – Illustration du cadre de référence CAMELEON [21, 113, 23].

Vis-à-vis de l'**adaptation** de l'interface, le cadre CAMELEON se focalise sur trois aspects, calqués sur le modèle Arch [7] :

L'adaptation de la présentation physique qui consiste à adapter l'**apparence** seule de l'interface, notamment lors du passage **d'un** espace technologique à un autre (e.g. passage **d'une** technologie Java Swing à de l'**HTML5**). Il **s'agit** de l'**adaptation** faite lors du passage à la FUI, l'espace technologique y étant lié. Les changements ne sont généralement que **d'ordre** cosmétique ;

L'adaptation de la présentation logique qui consiste à adapter le type des composants pour une interaction donnée. Par exemple, pour une tâche de navigation, celle-ci pourrait se faire via un menu latéral, ou via une navigation par onglet. Le but de l'**adaptation** logique est de choisir le type de composant qui convient le mieux pour la cible concernée. Ce type **d'adaptation** se réalise lors de la transformation en CUI où les types de tâche présents dans l'AUI sont spécifiés en composants. Ce choix se fait en fonction de la cible (i.e. modèle du contexte **d'interaction**) ;

L'adaptation des contrôles qui agit sur la nature de l'interaction et influe sur la structuration du modèle de tâches. Ainsi, certaines branches/feuilles de l'**arbre** de tâches peuvent être **d'importance** moindre ou supérieure suivant la cible. Le rôle de l'**adaptation** des contrôles est alors d'élaguer l'**arbre** ou de renforcer un chemin afin **d'optimiser** la tâche pour une cible particulière. Par exemple, sur la cible **d'un** dispositif mobile, seule les tâches de consultation pourraient être pertinentes, l'**adaptation** se chargeant alors d'élaguer les tâches d'édition pour optimiser l'interface au paradigme du *touch*. Ce type **d'adaptation** peut se faire en transformant le modèle de tâches en un autre modèle de tâches intermédiaire à une cible, ou en générant directement l'AUI en tenant compte de la cible.

Il est également à noter que CAMELEON ne spécifie pas formellement le passage de la CUI à la FUI, laissant ainsi le choix du traitement de la CUI. Une interface concrète peut alors être traitée de deux manières :

Génération de code où la transformation de CUI vers FUI consiste en une génération de code ciblant la technologie **d'exécution** liée au matériel cible via des transformations particulières de modèles vers texte (i.e. code). L'**intérêt d'une** génération de code réside dans les performances **d'exécution**, le code étant proche et spécifique à la plateforme cible ;

Interprétation à l'exécution où le modèle de CUI **n'est** pas transformé mais interprété tel quel par une application tierce (i.e. un interpréteur). L'**avantage d'un** interpréteur par rapport à une génération de code est de garder le modèle de CUI à l'**exécution** et ainsi profiter **d'une** traçabilité des éléments (e.g. savoir à quel élément **d'un** modèle de tâches est lié un composant de CUI) ainsi que de la possibilité de modifier les modèles à l'**exécution**. Vis-à-vis de l'interprétation de modèle, nous pouvons citer les travaux sur MARA [103] ou les travaux de Blumendorf et al. [11].

CAMELEON étant un cadre de référence, son implémentation technique est laissée ouverte. À ce titre, de nombreuses implémentations existent avec leur particularités propres, chaque travaux de recherche ayant un but spécifique. Parmi les recherches qui implémentent ou dérivent CAMELEON, nous retrouvons notamment les travaux suivants :

TERESA [85] qui est un MBI pour la conception et le développement d'IHM multi-modales et multi-appareils. Les travaux autour de TERESA ont par la suite conduit à MARIA [86], un langage de description d'IHM qui consolide les travaux précédents et met en avant une architecture de type **Service-Oriented Architecture** dans la génération des IHM;

UsiXML [70] qui est un langage de modélisation pour la conception et la génération d'IHM suivant le principe « $\mu 7$ » pour multiple appareils, plateformes, utilisateurs, langages, organisations, contextes et modalité. UsiXML est un langage bien supporté par la communauté scientifique et fait l'objet d'un effort de standardisation auprès du **World Wide Web Consortium**¹⁹;

MARA [103] qui met en avant une architecture pour l'usage des modèles à l'exécution. Celle-ci permet, par transformations dynamiques de modèles, l'adaptation des IHM à l'exécution;

SERENOA [24] qui est un environnement de développement de MBI visant à produire des IHM sensibles au contexte et hautement utilisables pour des services. SERENOA met une emphase sur la qualité des outils de conception et d'exécution des IHM afin de générer des interfaces adaptées de manière optimale;

Malai [9] qui se concentre sur des modèles d'interaction afin de séparer leur description (comprenant les feedbacks) du reste de l'IHM, ceci afin de tendre vers une factorisation et ré-utilisation des interactions dans d'autres systèmes;

UseML [74] qui instancie CAMELEON au cas particulier de la conception de **useware**, un paradigme de conception qui place l'analyse cognitive des interactions au centre du processus.

Notons enfin que l'espace problème de CAMELEON est vaste [20]. Le cadre de référence fixant l'architecture générale et l'écosystème de modèles, de nombreuses recherches gravitent autour pour aborder des problèmes spécifiques. La figure 1.18 expose les différents domaines associés et concernent les facettes suivantes :

Les moyens d'adaptation qui dénotent de la possibilité que l'IHM puisse se redistribuer et/ou se remodeler;

La granularité des composants qui représente la plus petite portion d'IHM pouvant s'adapter;

La récupération de l'état de l'IHM qui traite de la reprise de l'interaction après adaptation. L'interaction pouvant être reprise au début de la session (i.e. la tâche est perdue après adaptation), au niveau de la tâche (i.e. l'état du modèle de tâches est conservé) ou au niveau de l'action (i.e. l'état du modèle de tâches et des actions réalisées dans l'IHM sont conservés);

Le déploiement de l'IHM qui détermine si l'adaptation se fait sur des cibles statiques définies a priori ou si l'IHM est capable de s'adapter à l'exécution indépendamment de modèles de cibles;

La couverture du contexte qui dénote de la capacité de l'IHM à répondre aux facettes du contexte d'interaction définies par Dey & Abowd [30];

19. <http://www.w3.org/2005/Incubator/model-based-ui/wiki/UsiXML>

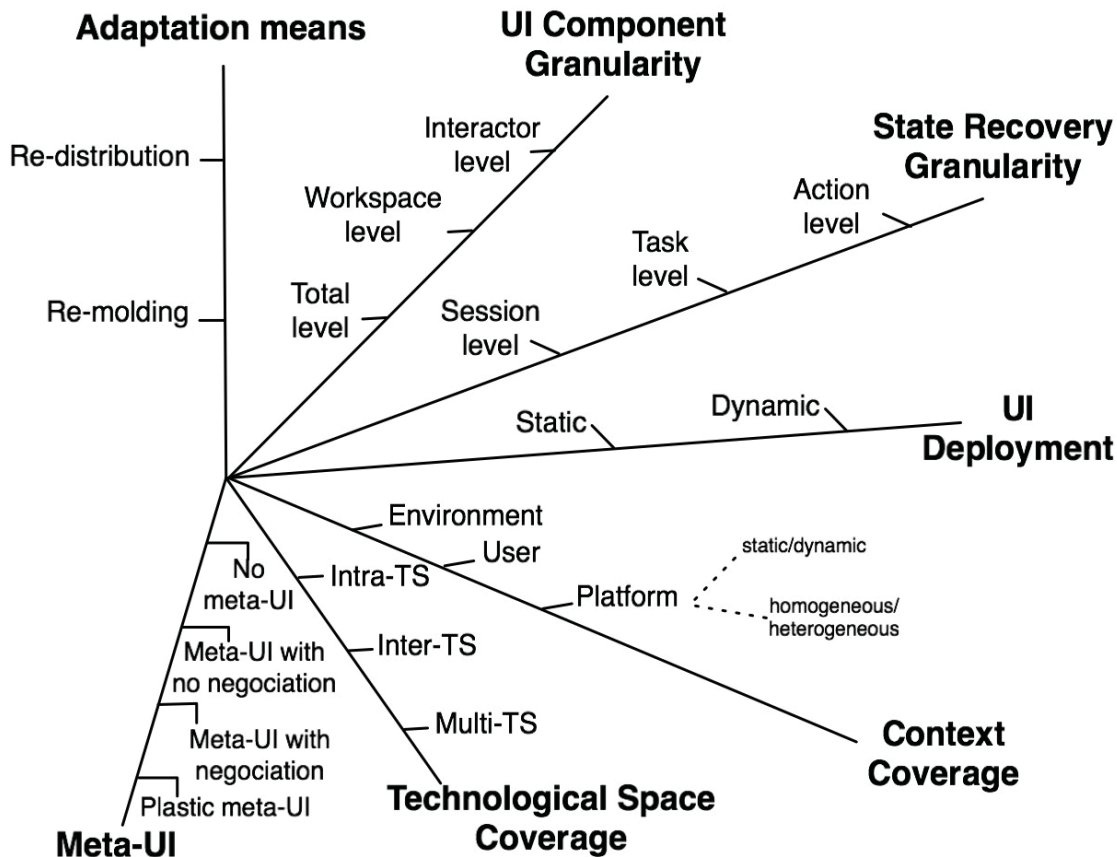


Figure 1.18 – L’espace problème de CAMELEON (en anglais), repris de Coutaz et al. [26].

La **couverture de l’espace technique** qui détermine si l’IHM est capable de s’exécuter dans un seul espace technique ou plusieurs;

La **méta-IHM** qui donne accès à l’utilisateur au paramètre d’adaptation afin de lui permettre de contrôler et d’évaluer le processus.

1.3.3 Rapprochement entre coordination et génération d’IHM

Nous venons de couvrir le cas de la génération d’IHM et il est intéressant de faire un parallèle avec notre approche de la coordination (voir section 1.2 et figure 1.8). Pour rappel, le support de la coordination se fait communément à deux niveaux : à un niveau théorique (i.e. niveau de l’activité et des organisations) et à un niveau pratique (i.e. niveau de l’utilisateur et des interactions). Du côté des MUI, ceux-ci partent d’un haut niveau d’abstraction (qui décrit les tâches) pour aller à un bas niveau d’abstraction (qui décrit l’interaction). Une mise en parallèle de ces deux domaines peut alors être faite.

En effet, la vue pratique représente les interactions concrètes qui vont porter la coordination du groupe et les flux de travaux, soit des interfaces concrètes/finales. Comme

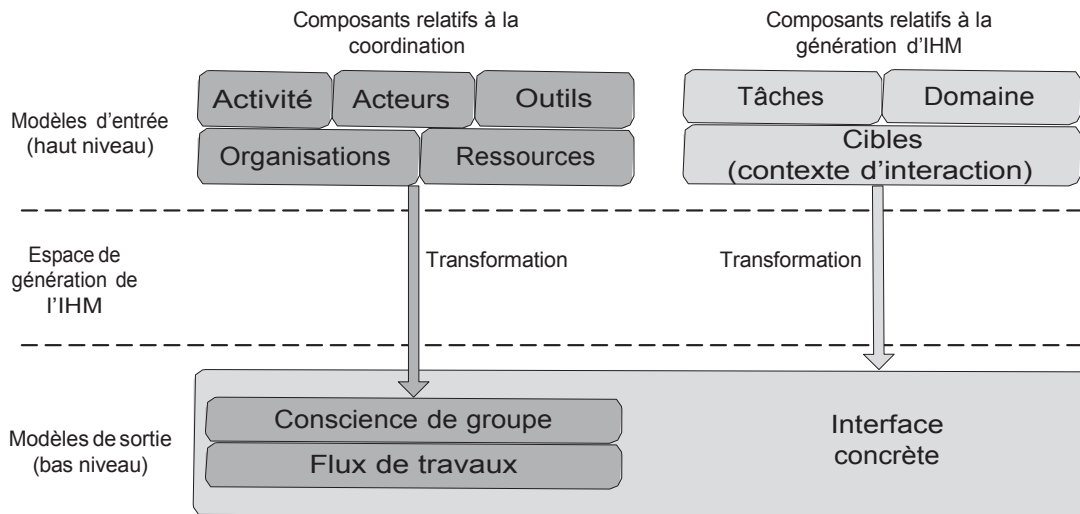


Figure 1.19 – Superpositions de la vue par dichotomie de la coordination et des concepts de génération d’IHM par CAMELEON [21, 113, 23].

nous l’avons vu, ces interfaces en support à coordination sont alimentées par les données de la vue théorique (e.g. la structuration de l’organisation et la description de l’activité fournissent des informations exploitables par des mécanismes de conscience de groupe). En parallèle, CAMELEON exploite les modèles de haut niveau lors des transformations afin de ne pas perdre d’information. Les modèles théoriques de la coordination peuvent alors être considérés comme des modèles d’entrée dans un processus de génération d’IHM afin de produire des interfaces pour la conscience de groupe et les flux de travaux.

Ainsi, l’approche de notre problème de conception d’IHM adaptatives pour le support à la coordination peut se voir comme une mise en parallèle de ces deux domaines. La figure 1.19 met cela en perspective. Cette vue nous permet de déterminer l’angle d’approche de notre problématique qui consiste donc à établir ce pont. Les différentes questions soulevées par ce problème seront explicitées en conclusion de chapitre (section 1.4) en vue d’établir la base de la structuration du cœur de l’état de notre recherche.

Résumé de section

Cette section 1.3 se résume par un troisième constat :

③ L’approche par *Model-Based User Interface*, avec une génération d’IHM adaptatives, sied à notre problématique de support à la coordination en contexte.

Cette mise en relation est résumée en figure 1.19 et nous permet, en conclusion de ce chapitre, de délimiter les angles d’approche de cette problématique.

1.4 Conclusion de chapitre

Dans ce premier chapitre, nous avons abordé le sujet des IHM dans leur contexte. En introduisant ce problème, nous avons mis en lumière le cas particulier de la collaboration et, plus précisément, de la coordination dans les IHM. Notre intention étant **d'amener** le sujet vers la prise en compte du contexte dans la conception **d'outils d'aide** à la coordination.

Partant de la complexité des situations quotidiennes que les utilisateurs rencontrent, notamment à travers la multiplication des appareils et la mobilité des acteurs, nous avons introduit la notion de contexte **d'interaction** (section 1.1.1). Nous avons vu **qu'il** était possible pour un système interactif **d'en** tenir compte avec les systèmes sensibles au contexte (section 1.1.2). Ces systèmes ont la possibilité d'être proactifs vis-à-vis de **l'utilisateur** en adaptant leurs services, et notamment leurs interfaces, en fonction de la situation.

Nous avons ensuite introduit la coordination dans le contexte en justifiant de son intérêt et aussi de **l'importance d'une** certaine flexibilité dans les IHM pour la coordination via la considération de ce que nous appelons le contexte **de l'activité** collective. En vue de supporter ces aspects du contexte nous avons vu la question de la coordination en deux temps via une vue théorique et une vue pratique. Cette dichotomie nous a permis de mettre en corrélation ces deux vues.

Enfin, la prise en compte du contexte dans les IHM nous a mené à considérer **l'adaptation** des interfaces. Ce champ de recherche vise à tirer parti du contexte pour fournir des IHM qui correspondent davantage à la situation de **l'utilisateur**. Nous avons plus particulièrement mis notre attention sur la plasticité des IHM qui aborde ce problème sous **l'angle** de la génération **d'interfaces** et **l'IDM**. Ce paradigme, mis en parallèle avec les aspects de la coordination, sied à notre problématique de conception **d'outils d'aide** à la coordination.

La synthèse des sections précédentes nous a conduit à **l'enchaînement** de constats suivant :

- ① Le contexte est un concept important à prendre en compte dans la conception **d'IHM**.
- ② La collaboration représente un contexte bien plus complexe que les considérations habituelles en IHM et il est essentiel de pouvoir adapter **l'IHM** en fonction du contexte pour supporter au mieux le travail de groupe.
- ③ **L'approche par *Model-Based User Interface***, avec une génération **d'IHM** adaptatives, sied à notre problématique de support à la coordination en contexte.

Ce qui nous a mené à mettre en parallèle les fondamentaux de la coordination avec le cadre de référence CAMELEON. Cette comparaison nous permet de formaliser les hypothèses de recherche suivantes afin de traiter du problème de la conception **d'IHM** capables de satisfaire les contraintes du contexte de **l'activité** collective :

H1. Apporter des métamodèles du contexte de **l'activité** collective en entrée **d'un** processus de génération **d'IHM** permet de produire des IHM adaptatives tenant compte de ces éléments.

→ Cette première hypothèse considère le contexte de **l'activité** collective et notamment de pouvoir **s'y** adapter. Son enjeu est de déterminer le processus de génération

d'IHM (i.e. les transformations de modèles) et les métamodèles du contexte de l'**activité collective** qui paramètrons ce processus pour produire des IHM adaptatives.

H2. Apporter des métamodèles **d'abstraction** de la pratique de la coordination, notamment de la conscience de groupe et des flux de travaux, permet de générer des IHM mettant en **œuvre** ces principes.

→ Cette seconde hypothèse considère la matérialisation à un niveau concret des mécanismes de coordination. Il **s'agit** de définir le processus de génération **d'IHM** qui permettra de produire ces mécanismes. Afin de savoir dans quelles circonstances ceux-ci doivent être générés, il est nécessaire de paramétrer ce processus avec des métamodèles qui sont à définir.

H3. Une méthodologie de conception tenant compte des hypothèses H1 et **H2**, c'est-à-dire des métamodèles et des mécanismes de génération **d'IHM** associés, permet de couvrir l'**ensemble** de la chaîne de conception **d'IHM** (i.e. de l'élicitation des besoins jusqu'à la production) pour l'**activité collective**.

→ Cette troisième hypothèse a pour but **d'unifier** les précédentes afin de répondre au problème de la conception **d'IHM** pour le contexte de l'**activité collective** dans son ensemble. Le but est de couvrir la modélisation des éléments **d'entrée** du processus de génération **d'IHM** (i.e. les modèles du contexte de l'**activité collective**) jusqu'à la production **d'IHM** mettant en **œuvre** des mécanismes de coordination, en passant pas la modélisation des éléments nécessaires à cela.

Définition des axes de recherche

En explicitant ces hypothèses, nous pouvons dès lors identifier les axes de recherche pour y restreindre l'état de l'**art** en chapitre 2. De manière générale, nous chercherons à évaluer le support de la coordination dans les MBUI. Ainsi, nos critères **s'orienteront** autour des points suivants :

La modélisation à haut niveau de la coordination (e.g. organisations, flux de travaux, collaboration, etc.), c'est-à-dire les aspects théoriques ;

La modélisation à bas niveau des IHM pour la coordination (e.g. conscience de groupe), c'est-à-dire les aspects pratiques ;

L'architecture du MBUI pour supporter la génération **d'IHM** à partir des deux points précédents (e.g. modèles à l'exécution, composition **d'IHM**, etc.) ;

La méthodologie de conception générale (e.g. modéliser les besoins métiers, intégration avec les divers intervenants, etc.).

Chapitre 2

Les *Model-Based User Interfaces* et leur support de la coordination

Sommaire

2.1	MBUI pour les collecticiels : un tour d' horizon de l' existant	55
2.1.1	Méthodologies de modélisation de collecticiels pour MBUI	55
2.1.2	Génération d'IHM pour la collaboration par MBUI	70
2.2	Supporter la coordination à haut niveau	78
2.2.1	Métamodèles de l'activité collective	78
2.2.2	Métamodèles de tâches multi-utilisateurs	87
2.3	Supporter la coordination à bas niveau	96
2.3.1	Métamodèles et interactions pour la conscience de groupe	96
2.3.2	Métamodèles et application des flux de travaux	103
2.4	Conclusion de chapitre	108

Ce second chapitre plonge dans le détail de notre problématique qui est de concevoir et de générer des IHM adaptées au contexte de **l'activité** collective. Nous avons notamment vu **qu'il** est question de pouvoir modéliser ce contexte par des concepts de haut niveau (i.e. les modèles **d'entrée**) que sont **l'activité**, les acteurs, les organisations, les ressources et les outils. Via des MBUI, ces éléments devront alors être concrétisés en interaction favorisant la coordination, que ce soit à travers une conscience de groupe ou **l'application** de flux de travaux (i.e. les modèles de sortie).

Au-delà des considérations purement techniques de **l'IDM** et des problèmes de génération **d'IHM** associés, il est tout aussi important de **s'intéresser** à la manière dont sont construits les modèles, notamment à partir de **l'analyse** des besoins métiers. Transposer ces besoins issus **d'une** réalité vers leur représentation en modèles permet de confronter ces derniers à des cas réels de conception **d'IHM**. Car en IDM, il ne suffit pas **d'abstraire** des concepts, encore faut-il que les modèles choisis soient pertinents et utilisables dans une démarche de conception. Nous aborderons donc la problématique avec un point de vue **d'ensemble**, **c'est-à-dire** en partant de la construction des modèles **d'entrée** à la mise en **œuvre** concrète.

Pour reprendre les points marquants de notre problématique, les éléments suivants feront l'objet de notre attention à travers ce chapitre :

La méthodologie générale : c'est-à-dire l'élicitation des besoins métiers vers des modèles exploitables pour la génération d'IHM adaptatives. Il est souvent question de la génération d'IHM d'un point de vue technique ou conceptuel, mais pas toujours dans l'ensemble d'une chaîne de conception. Des modèles ne sont utiles que s'ils sont productifs pour l'IDM et que leur construction représente une image réaliste ;

Le support des concepts de haut niveau : c'est-à-dire la modélisation du contexte de l'activité collective composé des aspects théoriques que sont l'activité, les acteurs, les organisations, les ressources et les outils ;

La génération d'IHM : c'est-à-dire une approche IDM systématique de bout en bout de la chaîne de conception et d'exécution ;

L'adaptation des IHM : c'est-à-dire de permettre la génération d'IHM adaptées au contexte de l'activité collective ;

Le support des concepts bas niveau : c'est-à-dire mettre en œuvre la conscience de groupe et les flux de travaux. Il se pose notamment la question de la modélisation de ces concepts et de leur abstraction en vue d'une génération d'IHM les supportant.

Structuration du chapitre

Nous aborderons ce chapitre en trois temps via des parties (voir figure 2.1) qui se feront écho les unes par rapport aux autres et qui suivent nos hypothèses de recherche (voir page 50). Tout d'abord, nous passerons en revue les travaux qui couvrent au mieux notre problématique en section 2.1. Nous étudierons notamment les méthodologies de modélisation de collecticiels, qui partent de l'élicitation des besoins et couvrent la construction des modèles. Elles sont essentielles car elles déterminent la validité des représentations et métamodèles choisis en les confrontant à des besoins métiers réels. À la suite de ces méthodologies, nous aboutirons à la phase de conception, avec généralement des arbres de tâches en entrée. Ici nous passerons en revue les cadres de génération d'IHM qui visent des interfaces pour supporter la coordination et leur capacité d'adaptation. Nous mettrons en avant les avantages et limites de ces approches.

Afin d'anticiper une réponse à ces limites, nous segmenterons notre recherche pour nous focaliser sur des aspects particuliers de notre problématique. Nous étudierons donc les approches de modélisation des concepts de haut niveau en section 2.2, soit les métamodèles de la collaboration (qui englobent la plupart des facettes comme l'organisation et l'activité) ainsi que les métamodèles de tâches collaboratives (qui sont plus spécifiques aux MBUI et non moins importants). Enfin, de l'autre côté, en section 2.3, nous étudierons les éléments propres aux aspects pratiques de bas niveau, plus particulièrement la modélisation et la mise en œuvre de la conscience de groupe et des flux de travaux.

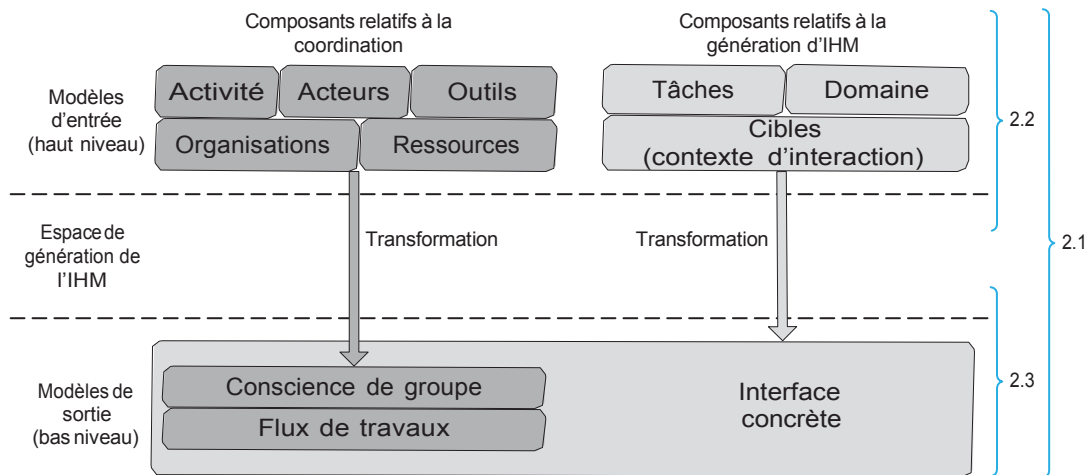


Figure 2.1 – Structuration du chapitre vis-à-vis de notre espace problème.

2.1 MBUI pour les collecticiels : un tour d'horizon de l'existant

Dans cette section, nous étudions dans un premier temps notre problématique dans son ensemble. C'est-à-dire en établissant un état de l'art des approches qui couvrent l'entièreté de la chaîne de conception et de génération d'IHM pour la coordination. Les approches couvrant tout le spectre de la modélisation et de la génération de collecticiels étant rares, nous abordons le sujet en deux temps pour étayer nos propos. Nous étudions tout d'abord les méthodologies de modélisation de collecticiels pour MBUI en section 2.1.1 pour ensuite nous concentrer sur les aspects plus techniques de la génération d'IHM en section 2.1.2. Dans ces deux sous-parties, nous nous intéressons plus particulièrement aux points relevés en figure 2.1, soit la modélisation des concepts de haut niveau dans les méthodologies ainsi que le support des outils bas niveau en génération.

2.1.1 Méthodologies de modélisation de collecticiels pour MBUI

Nous nous intéressons ici à l'étude de méthodologies de conception de collecticiels, plus particulièrement celles qui ont une démarche proche de notre problématique, c'est-à-dire qui partent de l'identification du contexte de l'activité collective pour tendre vers des spécifications d'IHM (voire de supporter une démarche d'IDM). Ces méthodologies permettent de partir des besoins métiers et/ou de collaboration et s'arrêtent le plus souvent à la définition de modèles d'interaction (tels que des modèles de tâches), laissant ainsi la production effective de l'IHM à des cadres de références techniques (ou plus simplement à des développeurs).

En l'occurrence, nous étudions principalement trois méthodologies proches du champ de l'IDM, soit CIAM [76] (en section 2.1.1.1), AMENITIES [44, 45] (en section 2.1.1.2) et TOUCHE [88, 87] (en section 2.1.1.3).

L'étude de ces méthodologies nous apporte en sus des éléments quant à la modélisation des aspects de l'activité collective. Dans cette section, nous nous attarderons essentiellement sur les points méthodologiques, la construction des modèles depuis les exigences et leur apport dans les procédés de conception. Leurs détails ainsi que leur impact pour la génération d'IHM sont couverts dans les sections suivantes de ce chapitre.

2.1.1.1 CIAM

Collaborative Interactive Applications Methodology (ou CIAM) est une méthodologie de modélisation introduite par Molina et al. [76] et qui est partie du constat **qu'il n'existait pas** de notation ni de méthodologie qui permettent de faire le pont entre besoins métiers et spécifications des IHM pour la collaboration. CIAM se propose alors de concilier les points de vue des domaines de l'IHM, du génie logiciel et du TCAO afin **d'aboutir** à une méthodologie de spécification des IHM pour la collaboration à partir des besoins métiers.

La méthodologie se veut également être un vecteur d'échanges entre les différents intervenants du processus de conception (e.g. analystes, concepteurs, experts métiers, etc.), le but étant **d'aboutir** à une spécification des IHM tenant compte des besoins métiers et collaboratifs. Cette spécification **s'arrête** à un niveau abstrait de l'interaction, sous forme **d'arbres** de tâches exprimés via CTT de Paterno et al. [84]. Avant **d'arriver** au niveau des tâches, CIAM repose sur divers métamodèles et se décompose en plusieurs étapes de modélisation comme suit :

1. **Définition du sociogramme de l'activité** : cette première étape consiste à identifier les acteurs et leur organisation. CIAM permet **d'exprimer** les rôles, les individus et de les regrouper sous forme de groupes/catégories ou d'équipes de travail. Ce métamodèle, dont un exemple **d'instance** est donné ci-dessous en figure 2.2, permet également de formaliser la hiérarchie avec des liens **d'héritage** ou **d'association**. **L'exemple** donné est celui de l'encadrement de projets scolaires, partant de leur définition à leur évaluation. Les étudiants, dirigés par un directeur, définissent un projet qui sera soumis à un comité, puis validé par un examinateur ;

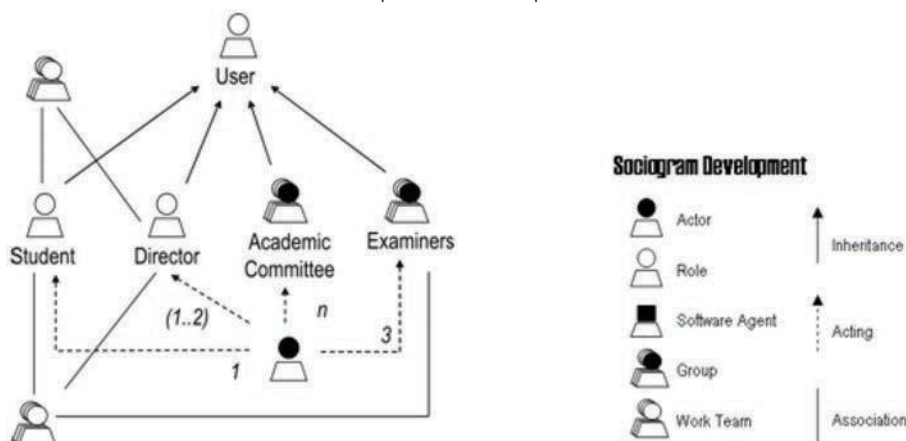










Figure 2.2 – Exemple de modèle de sociogramme dans CIAM [76].

2. **Définition des responsabilités/tâches** : une fois le sociogramme défini, CIAM propose de déterminer les tâches que chaque rôle doit remplir, soit définir les res-

2.1. MBUI pour les collecticiels : un tour d'horizon de l'existant

responsabilités de chacun. Cette étape de modélisation se présente sous la forme d'un tableau à double entrée tâches/rôles, comme illustré en figure 2.3. Il est également spécifié à ce niveau le type de la tâche du point de vue de la collaboration (i.e. tâche individuelle, de coopération ou de collaboration synchrone). Au-delà de l'attribution des responsabilités, cette étape permet pré-définir l'interdépendance des tâches/rôles (e.g. en figure 2.3, on aperçoit que trois rôles sont en lien avec la tâche *Post-DEP Procedure* qui est de nature coopérative) ;

Tasks \ Roles	Roles				Type
	Student	Director	Academic Committee	Examiners	
Draft Plan Writing	X	X			
Examining Board Proposal	X	X			
Request	X				
Academic Committee Valuation			X		
Suggest Changes			X		
DEP Development	X				
Post-DEP Procedure	X	X		X	
Examination	X			X	




Task Types	Icon
Individual Tasks	
Cooperative Tasks	
Collaborative Tasks	

Figure 2.3 – Exemple de tableau de définition des responsabilités dans CIAM [76].

3. **Définition de l'interdépendance des tâches** : cette étape consiste à expliciter la dépendance temporelle entre les tâches sur base de la pré-définition faite en étape précédente. Dans le concept, un parallèle avec les flux de travaux (voir section 1.2.4.2, page 33) peut être fait, à la différence que CIAM propose de factoriser les rôles par tâches. En pratique, ce modèle (dit **d'Inter-Action**) reprend le paradigme des diagrammes d'états-transitions d'UML en y ajoutant une spécialisation des transitions via des opérateurs LOTOS [117] (que l'on retrouve notamment dans la notation CTT [84]). La figure 2.4 illustre un tel modèle ;

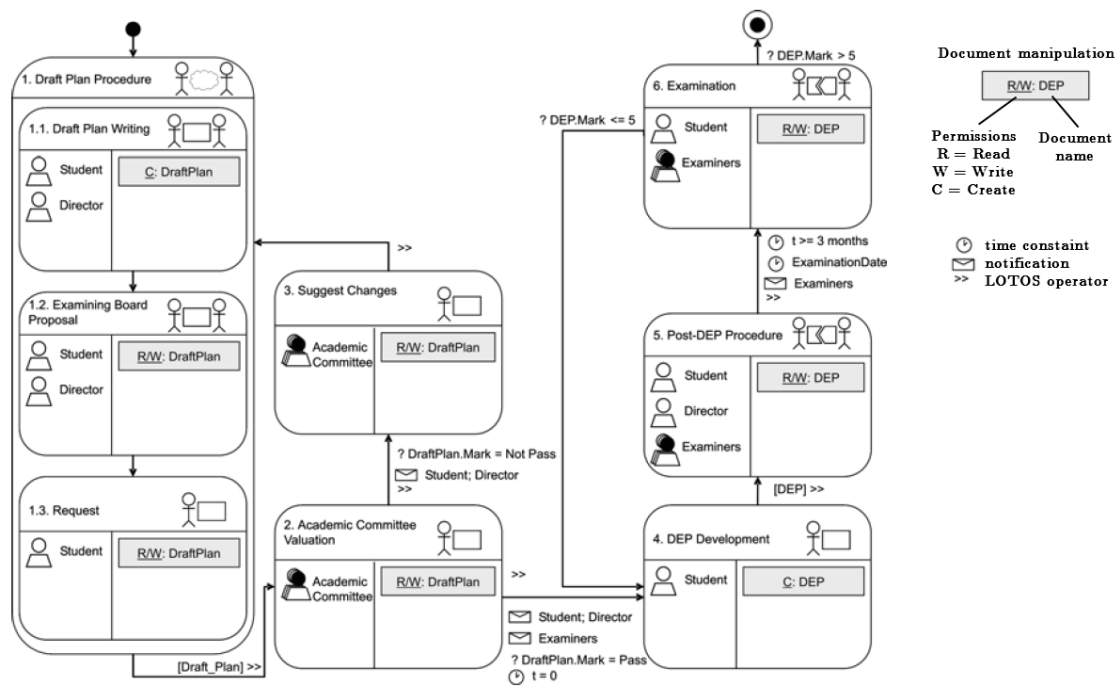


Figure 2.4 – Exemple de modèle *d’Inter-Action* dans CIAM [76], qui reprend les symboles du sociogramme et du tableau de responsabilités sur base d’un diagramme d’états-transitions UML.

4. **Définition de la collaboration** : reprenant le modèle *d’Inter-Action*, cette étape se focalise uniquement sur les tâches où plusieurs rôles interviennent. Le but étant alors de détailler la collaboration entre les rôles en décomposant la tâche en sous-tâches avec le même paradigme que l’étape précédente. CIAM propose alors de faire la distinction entre les tâches relevant de la coopération (i.e. interdépendance de tâches individuelles entre rôles) et la collaboration synchrone (e.g. le cas des espaces de travail partagés). Ce dernier type de tâches est davantage spécifié et précise la manière dont seront partagées les ressources (e.g. accès exclusif à une ressource partagée, synchronisation des vues, etc.). La figure 2.5 illustre ces deux types de tâches décomposées;

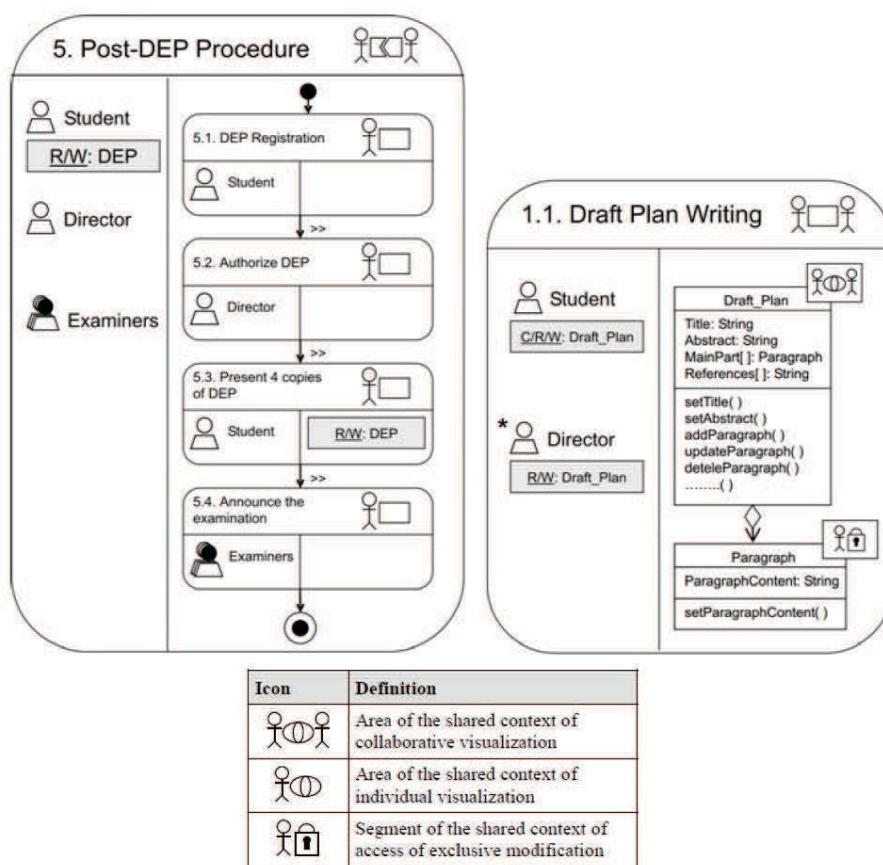


Figure 2.5 – Exemple de modèle de coopération (à gauche) et de collaboration synchrone (à droite) dans CIAM [76].

5. **Définition de l'interaction** : cette dernière étape de CIAM consiste à spécifier les IHM qui doivent supporter les sous-tâches identifiées à l'étape précédente. Pour cela, CIAM propose **d'employer** le formalisme de CTT [84] en mettant en avant la possibilité de reprendre ce type de modèles dans un générateur pour produire les IHM. CIAM **s'arrête** cependant à la définition des arbres de tâches.

CIAM met surtout en avant des métamodèles graphiques pour la spécification **d'IHM** collaboratives et est, par conséquent, plus un cadre **d'analyse** que de conception. La méthodologie **n'est** alors pas vraiment orientée vers une opérationnalisation des modèles et **n'est** donc pas une approche MDE systématique, aucun lien n'étant assuré entre les éléments **d'un** modèle à un autre.

Cependant, CIAM a **l'avantage** de proposer des représentations unifiées. On retrouve par exemple une constance dans **l'usage** des opérateurs LOTOS [117], des concepts empruntés à UML et des symboles propres à CIAM à travers les différents métamodèles. Cette homogénéité assure la cohérence de la méthodologie et la lisibilité des modèles, un atout important vis-à-vis des phases **d'analyse** et de conception. Ces bonnes représentations associées à une approche de modélisation par construction font de CIAM un point **d'entrée** pertinent pour établir le contexte de **l'activité**.

En bout de chaîne, CIAM ouvre la porte à CTT, un formalisme qui (si employé tel quel) ne permet pas la définition de tâches collaboratives, mais uniquement individuelles.

La caractérisation des tâches de coopération ou de collaboration synchrone faites dans CIAM à l'étape 4 est alors perdue, voire non modélisable via CTT seul. Les auteurs ne précisent pas cette transition, qui est alors plus vue comme une ouverture vers des processus de génération d'IHM basés sur les tâches. Nous pouvons néanmoins noter le fait que CIAM est supporté par des outils de modélisation intégrés à l'environnement Eclipse et dans le format Ecore, un format bien adapté à l'IDM.

2.1.1.2 AMENITIES

Introduit par Garrido et al. [44, 45], AMENITIES est une méthodologie issue du champ de l'ingénierie des exigences (*Requirements Engineering*) visant à modéliser les spécificités d'un système collaboratif en vue de le développer. Les auteurs mettent en avant que la difficulté de concevoir de tels systèmes nécessite une méthodologie dédiée tenant compte des particularités du domaine.

En substance, AMENITIES prône une méthodologie de modélisation unifiée avec pour objectif de proposer des modèles cohérents et appropriés afin de faciliter la communication entre les intervenants autour des exigences. Pour cela, AMENITIES se présente comme un pont entre le génie logiciel et le TCAO en empruntant des concepts à chacun d'entre eux, notamment via une spécialisation d'UML.

Afin de tirer l'essence des systèmes collaboratifs, AMENITIES propose des métamodèles basés sur les théories de la collaboration les plus pertinentes, telles que la théorie de l'activité et la cognition distribuée (que nous avons traité en section 1.2.3 de manière générale et en section 2.2.1 vis-à-vis de leur modélisation). Le métamodèle qui en résulte, visible en figure 2.6, est une synthèse de la collaboration et sert à AMENITIES de trame de fond dans sa méthodologie.

Dans la pratique, cette méthodologie repose intégralement sur l'usage d'UML en fournissant quelques métamodèles spécialisés dans l'expression des exigences de la collaboration. L'instanciation de ces métamodèles se fait par étapes suivant une méthodologie générale illustrée en figure 2.7 et décomposée comme suit :

1. **Modélisation des exigences (*Requirement Models*)** : cette première étape, partant des exigences déjà élicitées, consiste à modéliser les exigences fonctionnelles et les aspects sociaux de la collaboration (i.e. les rôles et l'organisation). Pour se faire, la modélisation repose sur des diagrammes de cas d'utilisation UML qui peuvent éventuellement être décomposés en sous-diagrammes suivant la complexité des cas. D'un point de vue méthodologique, ces diagrammes sont élaborés à travers des méthodes d'ethnographie non précisées ;
2. **Modélisation de la collaboration (*Cooperative Model*)** : sur base des diagrammes précédents, cette étape propose de détailler rôles, organisation et tâches dans un formalisme plus concret. Pour cela, Garrido et al. [43, 45] ont introduit un profil d'UML : COMO-UML [43]. À partir de diagrammes d'activité UML, cette notation permet d'explicitier les rôles, l'organisation et les liens entre tâches et rôles (de manière similaire à un flux de travaux). La figure 2.8 donne des exemples de ces trois modélisations ;

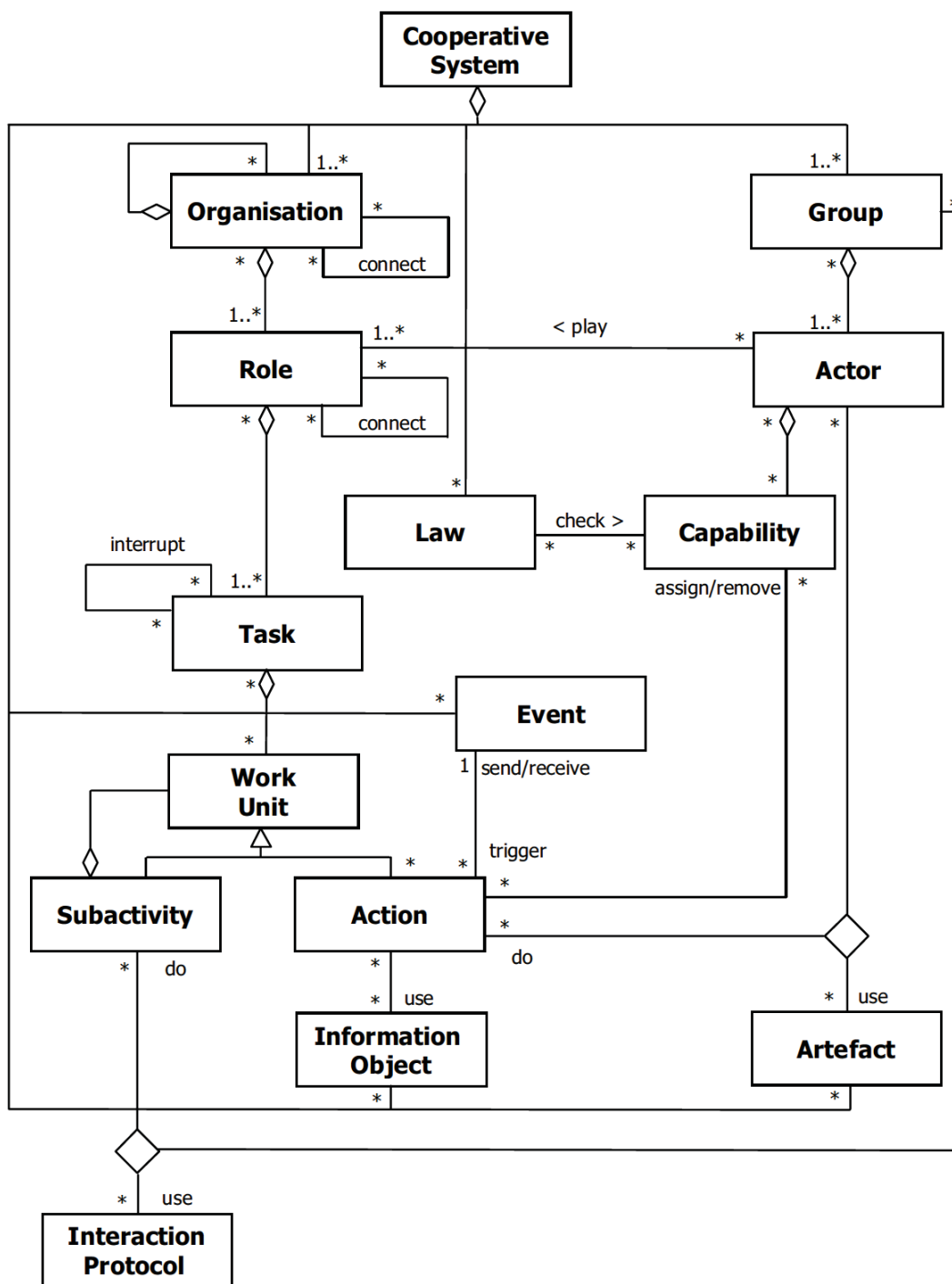


Figure 2.6 – Métamodèle de la collaboration d'AMENITIES [44]. Représenté sous la forme d'un diagramme de classes UML.

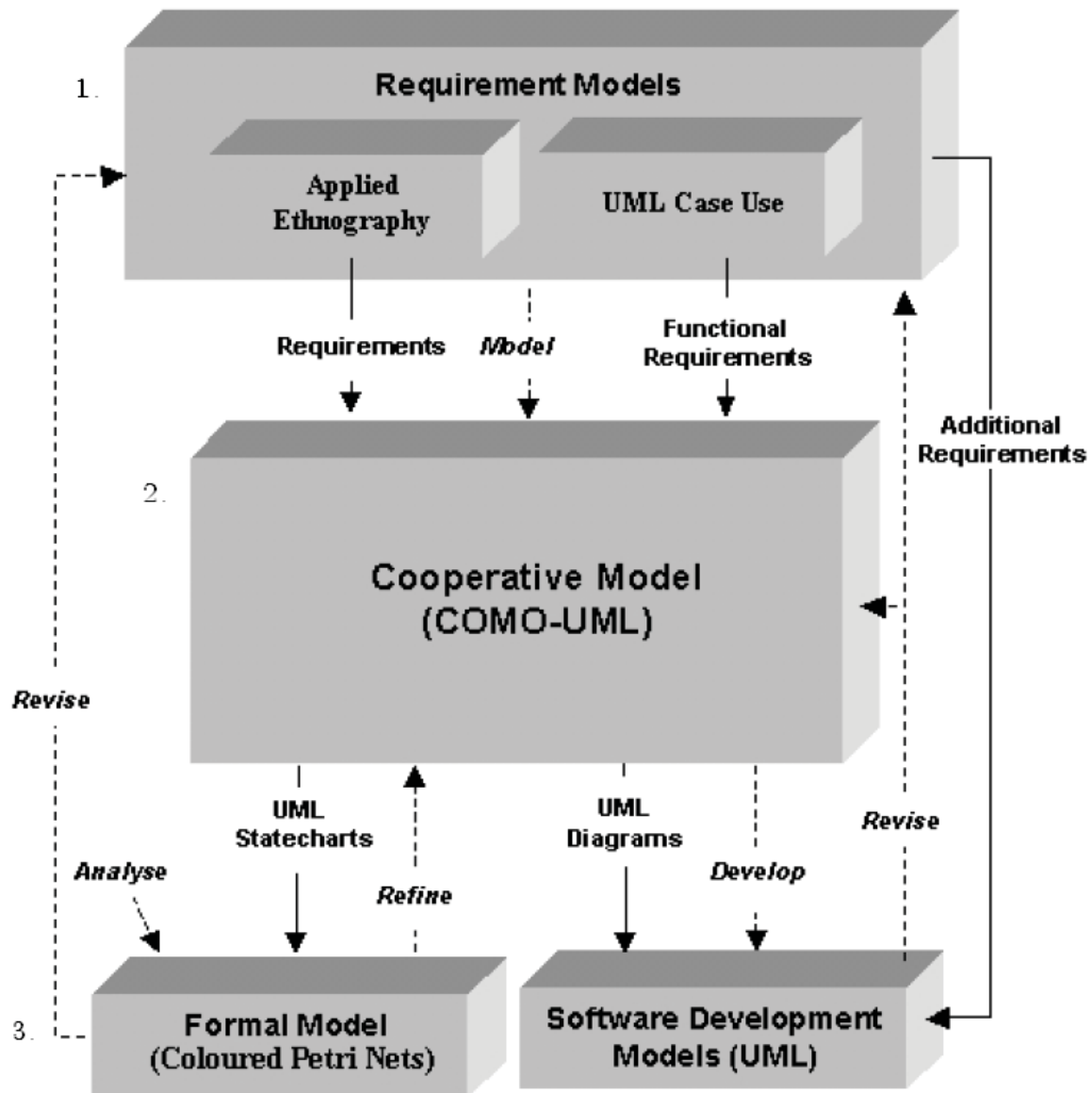


Figure 2.7 – Méthodologie générale d'AMENITIES [44].

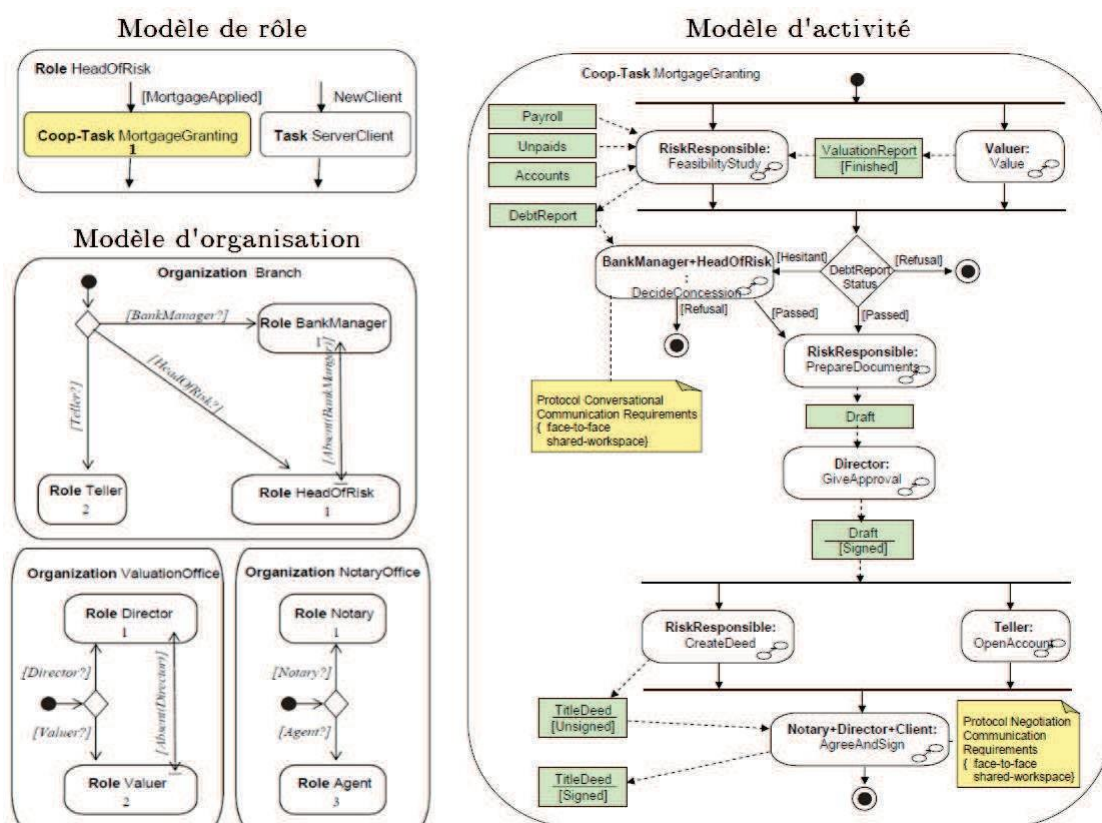


Figure 2.8 – Exemple de modèles COMO-UML, repris de Noguera et al. [81].

3. **Spécifications formelle et technique (*Formal Model & Software Development Models*)** : les aspects collaboratifs définis, cette dernière étape consiste à produire des modèles pour l'implémentation technique de l'application. Ici, ce sont des formalismes typiques du génie logiciel qui sont employés, tels que des diagrammes de classe, d'états-transitions, ou plus généralement UML. La particularité d'AMENITIES est de transposer le modèle d'activité de l'étape précédente sous forme de réseaux de Petri en vue de pouvoir simuler l'activité (et donc de valider les exigences). Notons aussi que, sans le spécifier formellement, les auteurs laissent la porte ouverte vers d'autres formalismes comme CTT [84] pour spécifier la partie technique.

AMENITIES profite d'une très bonne homogénéité dans le choix des représentations, notamment via l'usage systématique d'UML et son profil dédié à la description de l'activité, COMO-UML.

Tout comme CIAM, les modèles d'AMENITIES et leur construction ne sont pas portés par une démarche IDM. Les modèles sont essentiellement contemplatifs et orientés vers une bonne communication entre les intervenants. Néanmoins, bien qu'AMENITIES pointe vers CTT, la méthodologie prône davantage un usage d'UML pour spécifier la partie technique des IHM. En ce sens, AMENITIES est plus proche du génie logiciel classique que d'une approche orientée tâches & IDM comme CAMELEON [21, 113, 23], ne cadrant alors pas au mieux avec notre problématique de génération d'IHM.

Enfin, notons que le métamodèle de collaboration (figure 2.6, page 61), ne sert à

AMENITIES que de trame de fond. Chaque étape de la méthodologie consiste à modéliser un aspect particulier basé sur ce métamodèle. Il **n'est** en soi pas repris, ni formellement spécifié dans sa globalité en bout de méthodologie. Il aurait été intéressant, **d'un** point de vue IDM, **d'avoir** un modèle assimilant **l'ensemble** des concepts en fin de cycle afin de disposer **d'une** vue uniforme et cohérente. Un tel modèle représenterait une bonne vue sur le contexte de **l'activité** qui pourrait servir à **l'exécution**, ce que nous verrons dans les sections suivantes.

2.1.1.3 TOUCHE

Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments (ou TOUCHE), introduit par Penichet et al. [88, 87], est un cadre **d'analyse** et de modélisation pour la conception de collecticiels. À la différence de CIAM [76] et AMENITIES [44], TOUCHE se veut être une approche plus orientée **vers l'IDM**, et ce, dès les premières phases **d'analyse**. En ce sens, les auteurs qualifient TOUCHE de *process model*²⁰, et non de méthodologie, en contraignant les bonnes pratiques à travers un outil de modélisation logiciel (nommé TOUCHE CASE Tool). Par ailleurs, TOUCHE couvre l'élicitation des exigences et va jusqu'à la description de la structure et du comportement du collecticiel (à travers **l'emploi d'UsiXML** [70] et suivant les principes de CAMELEON [21, 113, 23]), ce **qu'AMENITIES** et CIAM ne font pas.

Le déroulement de TOUCHE est similaire à ses parents CIAM & AMENITIES. Son procédé est segmenté en étapes visant à modéliser en premier lieu les acteurs et leur organisation, pour ensuite se focaliser sur les tâches et leurs liens. TOUCHE repose pour cela sur deux types **d'outils** : des modèles²¹ de documents à remplir (afin d'éliciter les exigences) et des IHM **d'aide** à la modélisation (afin de traduire en diagrammes les exigences et de manipuler des modèles). Le procédé de TOUCHE se décompose via les étapes suivantes :

1. **Identification des rôles** : TOUCHE demande en entrée de déterminer quels sont les acteurs et **d'identifier** leurs rôles. Pour cela, les rôles doivent être décrits via des documents types qui conditionnent les informations attendues, telles que les responsabilités (i.e. tâches), les compétences requises et les permissions dont le rôle dispose. Une fois les documents établis, TOUCHE CASE Tool prend le relais afin de modéliser la hiérarchie des rôles et les responsabilités, **d'une** manière relativement proche de CIAM et ses sociogrammes. En effet, TOUCHE permet de modéliser les groupes **d'acteurs** ainsi que leurs rôles et les liens entre ces rôles (i.e. par héritage). La figure 2.9 illustre un sociogramme dans TOUCHE CASE Tool et qui décrit **l'organisation d'un** système de publication où des auteurs peuvent écrire des brouillons de documents qui seront passés en revue par des examinateurs pour publication ;

20. *Process model* n'a pas d'équivalent en français et représente un procédé formel de méthodologie. En un sens, un *process model* est à une méthodologie ce que les flux de travaux sont aux pratiques métiers. Les avantages **d'un process model** sont **qu'il** contraint le bon déroulement **d'une** méthodologie et que sa description formelle peut être facilement comparée et validée.

21. Modèles au sens *templates* ou patrons de documents.

2.1. MBI pour les collecticiels : un tour d'horizon de l'existant

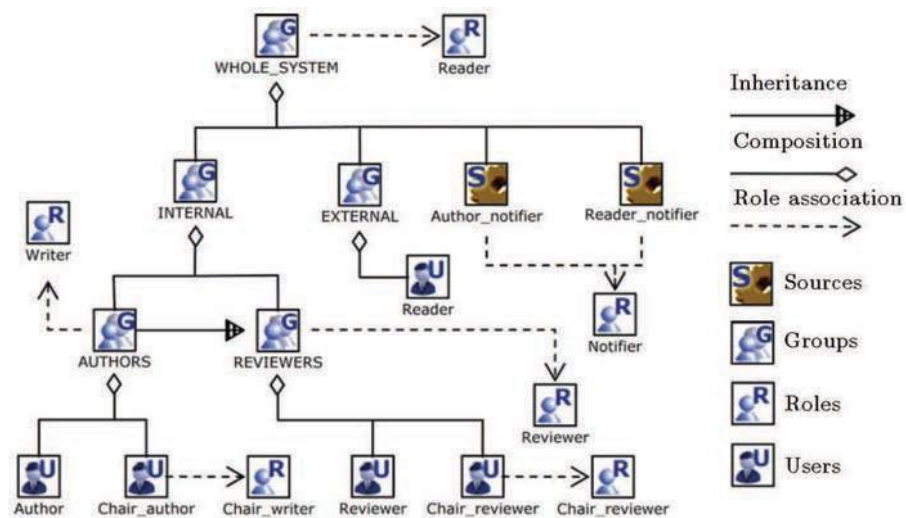


Figure 2.9 – Exemple de modèle de sociogramme dans TOUCHE [87].

- Raffinement des responsabilités** : toujours de manière comparable à CIAM, cette étape de TOUCHE propose de modéliser une matrice de correspondance entre les rôles et les tâches (voir figure 2.3 pour l'équivalent de CIAM). Cette matrice, combinée au modèle de sociogramme de l'étape précédente, est alors transformée par TOUCHE CASE Tool en un modèle dit de co-interaction entre les tâches et rôles (i.e. comparable à un flux de travaux) comme représenté en figure 2.10. TOUCHE met également l'accent sur la traçabilité des éléments, l'outil gardant des liens depuis les modèles d'origine (qui sont notamment portés à travers les transformations de modèles). Par exemple, dans un diagramme de co-interaction, il est possible de tracer l'origine d'un rôle vers le sociogramme. Ce modèle de co-interaction permet aussi de définir les besoins en collaboration des tâches en spécifiant quelles parties du modèles 3C (voir section 1.2 la tâche doit couvrir, mais aussi sa répartition espace-temps (voir table 1.1). Ces informations doivent être spécifiées manuellement par le concepteur car ce type de savoir ne peut être véhiculé à travers la transformation d'un sociogramme. Elles permettront ensuite de spécialiser les IHM au moment de leur conception ;

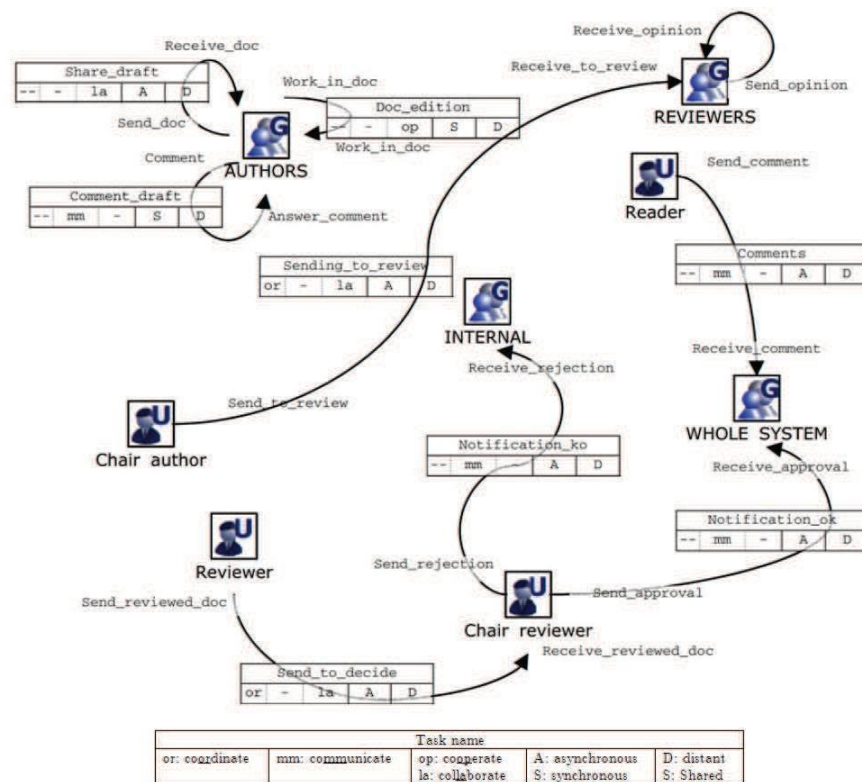


Figure 2.10 – Exemple de modèle d'organisation (co-interaction) des tâches dans TOUCHE [87].

3. **Analyse du comportement** : cette étape consiste à déterminer le comportement du collectif et pour cela TOUCHE demande de décrire les tâches identifiées précédemment via CTT [84] (avec l'appui d'UsiXML [70]) ;
4. **Conception des IHM** : partant des arbres de tâches CTT de l'étape précédente, TOUCHE CASE Tool permet de dériver ces modèles sur le principe de CAMELEON via le support d'UsiXML. Ainsi, TOUCHE transforme les tâches en interfaces abstraites (i.e. **Abstract User Interface** de CAMELEON). Au passage, TOUCHE spécialise les conteneurs abstraits suivant les propriétés de collaboration identifiées dans le diagramme de co-interaction. Celles-ci permettent de déterminer si un conteneur abstrait doit proposer certains services, tels que de la conscience de groupe, ou permettre une coopération synchrone [89]. Cette spécialisation de l'AUI pour la collaboration est décrite plus en détail en section 2.3.1.

Si l'outil TOUCHE CASE Tool s'arrête à la définition des AUI, ses auteurs [89] vont toutefois au bout de leur démarche en passant à des interfaces concrètes, puis à leur implémentation technique. Le procédé n'est pas complètement automatique, certaines transformations étant manuelles, mais TOUCHE se rapproche ainsi d'une démarche IDM de bout en bout.

La volonté de TOUCHE de vouloir opérationnaliser les modèles via des standards plus techniques a toutefois un revers : les représentations n'ont pas l'homogénéité de CIAM ou AMENITIES. En effet, si ces derniers reposent beaucoup sur les symboles UML,

2.1. MBUI pour les collecticiels : un tour d'horizon de l'existant

TOUCHE emprunte à différents standards, tantôt UsiXML (i.e. définition de l'IHM), tantôt en étant inspiré d'UML (i.e. sociogramme) ou tantôt en étant spécifique à un modèle (i.e. co-interaction). Les éléments qui sont repris à travers certains modèles ont cependant les mêmes symboles, notamment les rôles et groupes (important dans la traçabilité du procédé).

L'intérêt de TOUCHE repose surtout dans sa forme de *process model* et son rapprochement avec CAMELEON. Il paraît alors intéressant de se poser la question de la validité de TOUCHE pour la conception d'interfaces adaptatives. En l'état, TOUCHE n'est pas orienté vers la génération d'IHM adaptées au contexte. Cependant, Garrido, Penichet et al. [46] ont récemment posé l'idée d'étendre le procédé pour générer des collecticiels adaptés, se rapprochant alors d'une problématique proche de la nôtre²².

Nous noterons également que TOUCHE est davantage une aide à l'analyse et à la conception de collecticiels qu'une démarche IDM systématique. Une fois la phase d'analyse faite (qui s'arrête à l'étape 3), la plupart des informations modélisées à haut niveau (i.e. modèles de co-interaction et de sociogramme) ne sont plus considérées dans la génération de l'IHM. En effet, seuls les arbres de tâches CTT comptent à partir de l'étape 4. Bien que les arbres de tâches découlent des modèles précédents, certaines informations telles que l'organisation sont perdues aux niveaux de modélisation plus concrets.

2.1.1.4 Synthèse

La table 2.1 synthétise notre analyse de ces méthodologies de conception pour collecticiels. Vis-à-vis de leur intégration à l'IDM, en s'orientant vers la définition des tâches, ces méthodologies représentent un point d'entrée idéal pour des processus de génération d'IHM. TOUCHE est d'ailleurs ce qui se rapproche le plus de notre problématique en proposant des métamodèles qui s'intègrent à UsiXML.

AMENITIES et CIAM, de leur côté, consistent davantage à concilier TCAO et génie logiciel (au sens phase de pré-conception et d'analyse via UML), mais gagnent en homogénéité et pertinence des représentations. TOUCHE, lui, concilie TCAO et IDM. Ces travaux apportent également des éléments essentiels à notre problématique, notamment en proposant des métamodèles permettant de décrire la collaboration et le contexte de l'activité.

Néanmoins, en dehors de TOUCHE, ces méthodologies ne sont pas des approches d'IDM systématiques et peu de transformations automatiques de modèles sont impliquées (même dans le cas de TOUCHE). Ceci peut résulter en une possible perte d'information dans le procédé, ce que nous verrons en section suivante. Les représentations utilisées sont également judicieuses car homogènes et empruntant à des domaines à la fois techniques et métiers. Les métamodèles favorisent donc ainsi des approches pluridisciplinaires de co-construction des modèles avec l'ensemble des acteurs de la chaîne de conception. Vis-à-vis de leur couverture de la collaboration, AMENITIES, bien que reposant sur une métamodèle général (voir figure 2.6, page 61) pour les aspects d'analyse, ne propose pas en sortie des modèles caractérisant la collaboration dans les IHM, mais uniquement l'enchaînement des tâches (à la manière de flux de travaux). CIAM complète cette modélisation

22. L'avancée sur ces travaux reste à paraître au moment de l'écriture de ce chapitre.

en typant les tâches utilisateurs suivant la nature de la collaboration (e.g. coopération, coordination, etc.). TOUCHE va au bout de la démarche en proposant une modélisation de **l'activité** qui tient compte de **l'ensemble** du modèle 3C [34] (i.e. le type de la collaboration) et la matrice de Johansen [63] (i.e. la répartition espace/temps des tâches). Dans le cas de **l'outillage** et des éditeurs supportant les méthodologies, TOUCHE, via **l'outil TOUCHE Case Tool** couvre **l'ensemble** des éléments à modéliser avec tantôt des transformations automatiques ou semi-automatiques entre les étapes afin de reposer sur les éléments déjà modélisés. CIAM dispose **d'un** outillage de modélisation intégré au sein **d'Eclipse Modeling Framework** afin de supporter l'édition de modèles, cependant les éventuelles transformations entre les étapes ne sont pas spécifiées. AMENITIES de son côté ne détermine pas la présence **d'outils** logiciels dans les travaux passés en revue. Par ailleurs, nous notons que ces méthodologies ne portent, pour ainsi dire, pas du tout sur **l'adaptation** des IHM. En effet, il **n'est** pas question **d'identifier** les différents contextes **d'utilisation** dans lesquels les IHM pourraient être utilisées par les acteurs. Ces méthodologies laissent par contre la porte ouverte à des processus de génération **d'IHM** en employant des formalismes souvent proches de ConcurTaskTrees, et donc éventuellement de se rapprocher de cadres de référence comme CAMELEON qui permettent une adaptation. En synthèse, ces méthodologies de conception permettent une très bonne construction de modèles avec une approche par étapes qui se définissent les unes sur les autres. Ces étapes sont par ailleurs similaires **d'une** approche à **l'autre** et il est possible **d'en** tirer une segmentation générale pour ce qui est **d'aborder** la modélisation de **l'activité** :

1. La modélisation des rôles et de leur hiérarchie (i.e. le sociogramme de **l'organisation**) ;
2. La modélisation des tâches et de leur attribution aux rôles/acteurs ;
3. La modélisation des interdépendances entre les tâches et **l'accès** aux ressources (i.e. **l'activité** dans son ensemble) ;
4. La modélisation des tâches utilisateurs en détail.

2.1. MBUI pour les collecticiels : un tour d'horizon de l'existant

Table 2.1 – Récapitulatif des méthodologies de conception de collecticiels étudiées. Légende : **9** très adéquat à notre problématique, **Q** adéquat avec des réserves, **Q** présente des limitations significatives.

Critères d'évaluation	Méthodologies		
	CIAM [76]	AMENITIES [44]	TOUCHE [88, 87]
Intégration à l'IDM	Q Non explicite, mais éditeurs de modèles intégrés à EMF	Q Méthodologie de génie logiciel pure	9 Intégré à l'IDM en reposant sur UsiXML [70]
Éléments en sortie	Q Description des tâches suggérée via CTT	Q Spécification technique UML	9 Une IHM finale via transformation des AUI Q , mais sur des tâches individuelles
Représentation graphique	9 Uniforme sur l'ensemble des modèles via UML et LOTOS	9 Uniforme sur l'ensemble des modèles via UML et COMO-UML	Q Cohérence de symboles mais représentation tantôt UML, UsiXML ou spécifique
Couverture de la collaboration	Q Distinction entre tâches coopératives, de coordination ou individuelles et modélisation de l'activité (i.e. flux de travaux)	Q Modélisation de l'activité (i.e. flux de travaux)	9 Positionnement relatif au modèle 3C et la matrice Johansen
Outillage	9 Éditeurs de modèles intégrés à EMF	Q Non précisé	9 TOUCHE Case Tool qui couvre la chaîne de modélisation jusqu'à l'AUI
Considération de l'adaptation	N/A	N/A	Q Néant, mais perspective dans l'utilisation d'UsiXML [70]

2.1.2 Génération d'IHM pour la collaboration par MBUI

En sortie des méthodologies de modélisation vues précédemment, nous avons généralement un ensemble de modèles plus ou moins adéquats pour entrer dans un processus de génération d'IHM comme les implémentations de CAMELEON [21]. Dans la pratique, les approches qui font suite à ces méthodologies sont peu nombreuses. En effet, la littérature se concentre généralement sur des aspects précis de la chaîne de conception/production des IHM, comme la prise en compte de modèles de collaboration ou le processus de génération en lui-même.

Dans cette section, nous nous intéressons uniquement aux approches de génération d'IHM qui font suite, ou qui font partie des méthodologies de conception pour la collaboration. Cette vue nous permet d'avoir un regard global d'un bout à l'autre de la chaîne. Nous avons donc étudié les travaux tirant parti du trio CIAM [76], AMENITIES [44] et TOUCHE [88, 87, 89], avec une emphase sur la capacité de ces approches à considérer les modèles décrivant la collaboration et à considérer l'adaptation des IHM.

2.1.2.1 Boîtes à outils et *widgets*

Dans la pratique, la génération d'une IHM finale pour la coordination est souvent abordée sous l'angle du génie logiciel, notamment via l'introduction de boîtes à outils dédiées telles que GroupKit [92], Multi-User Awareness UI [58, 59] (MAUI), Groupware Toolkit [27] (GT) ou encore les travaux de Kadri et al. [82]. Ces boîtes à outils viennent en complément de boîtes existantes bien connues des développeurs d'interfaces comme Java Swing ou WinForms. Ces différents travaux apportent des composants synchronisés entre les différentes instances d'une IHM ainsi que l'architecture réseau permettant leur fonctionnement. Par exemple, un composant « bouton » peut représenter les clics d'autres utilisateurs, les barres de défilement afficher la position des autres utilisateurs, etc. C'est notamment le cas dans MAUI, illustré en figure 2.11.

Dans ces boîtes à outils, la conscience de groupe est aussi représentée sous forme de *widgets*, soit un ensemble de composants pré-assemblés qui procurent une vue particulière sur les informations de groupe. C'est notamment le cas de listes d'utilisateurs connectés, d'historiques de modifications ou vues radars partagées comme nous l'avons vu en section 1.2.4. Ces widgets représentent des interactions souvent génériques d'un collectif à l'autre.

Suivant le vocabulaire employé en génie logiciel, les termes *composant* et *widget* peuvent parfois être confondus. Nous tenons donc à définir le terme *widget* comme suit dans le reste de ce manuscrit :

Définition. *Widget* : composant d'interface réutilisable (potentiellement lui-même composé d'autres composants atomiques) qui est associé à une ou plusieurs tâches particulières.

2.1.2.2 TOUCHE et génération d'IHM

TOUCHE de Penichet et al. [88, 87, 89] représente une approche cohérente prenant en compte toute la chaîne de conception. Nous avons vu précédemment que leur méthodologie



Figure 2.11 – Exemples d’interfaces conçues avec MAUI de Hill & Gutwin [58]. À gauche, l’interaction d’un utilisateur et, à droite, une autre instance de l’interface mettant en transparence les actions d’autrui.

permet **d’aboutir** à un modèle **d’organisation** des tâches (voir figure 2.10) qui précise notamment la nature de la collaboration des tâches. Par exemple, une tâche peut être qualifiée suivant son besoin en communication, en coopération, en coordination ou si elle requiert une collaboration synchrone ou un espace partagé (soit un positionnement relatif au modèle 3C [34, 40] et à la matrice de Johansen [63], voir section 1.2 page 17).

TOUCHE propose alors de considérer certains aspects de la collaboration lors de la génération **d’IHM**, comme la conscience de groupe. Pour cela, dans la conception des AUI, TOUCHE étend les métamodèles **d’UsiXML** [70] afin **d’abstraire** la conscience de groupe (voir figure 2.12). Penichet et al. [89] introduisent ainsi un nouveau type de conteneur abstrait, soit les **Abstract Workspace Awareness Container** (AWAC). Un AWAC est créé en lieu et place **d’un Abstract Container** standard pour refléter la caractérisation **d’une** tâche lorsque celle-ci requiert de la conscience de groupe. Les AWAC sont ensuite transformés en composants concrets dans les CUI, ou définissent plutôt des groupes de composants qui devront fournir une conscience de groupe (voir figure 2.13). Pour finir, lorsque des composants sous l’égide **d’un** tel conteneur sont rendus en FUI, TOUCHE fait appel à une **boîte** à outils dédiée à la conscience de groupe. Cette **boîte** à outils, relative à **d’autres** travaux tels que GroupKit [92], Multi-User Awareness UI [58, 59] (MAUI), Groupware Toolkit [27] (GT), propose des composants synchronisés entre chaque instance de **l’IHM**. Par exemple, un composant de type menu issu **d’un** AWAC et utilisant cette **boîte** à outils permet **d’afficher** les éléments du menu que les autres utilisateurs sont en train de manipuler. Nous verrons notamment plus en détail ces **boîtes** à outils en section 2.3.1.

TOUCHE permet donc de couvrir toute la **chaîne** de production **d’un** collectif, en partant de l’élucidation des besoins, la structuration de la collaboration pour aboutir la génération **d’IHM** tenant compte de la conscience de groupe. Notons cependant que TOUCHE **n’est** pas une approche transformationnelle systématique. En effet, certaines étapes ne peuvent être automatisées par transformation de modèles. **C’est** notamment le cas de la définition des arbres de tâches, dont la description de **l’interaction** ne peut être inférée des modèles **d’organisation**. Par ailleurs, TOUCHE ne spécifie pas si le formalisme ConcurTaskTrees (CTT) [84] est utilisé tel quel alors que celui-ci ne permet pas de véhiculer les besoins de conscience de groupe ni les interactions collaboratives. Enfin, certaines informations sont perdues lors du processus de génération **d’IHM**, telles que **l’organisation** et les flux de travaux. Il **n’y** a ainsi pas de mécanisme de synchronisation des tâches.

Du point de vue de **l’adaptation** des IHM, TOUCHE ne traite pas du sujet. Cependant, sa parenté à UsiXML [70] lui confère des propriétés qui pourraient être exploitées à ces fins. En l’état, TOUCHE ne cible **qu’une** plate-forme technique et un seul contexte. Des travaux plus récents de Garrido, Penichet et al. [46] exposent toutefois **l’idée d’utiliser** la démarche de TOUCHE pour produire des collectifs capables **d’adaptation** fonctionnelle, **c’est-à-dire** de présenter des services différents suivant la situation.

2.1.2.3 *Widget Specification Language (WSL)*

Vellis et al. [121] abordent en partie le sujet de **l’adaptation** des IHM en introduisant des « widgets polymorphes. » Leurs recherches portent sur la génération **d’IHM** collaboratives multi-plateforme via UsiXML [70]. Par multi-plateforme, les auteurs entendent différents espaces techniques, que ce soit matériel (e.g. ordinateur, téléphone mobile, etc.)

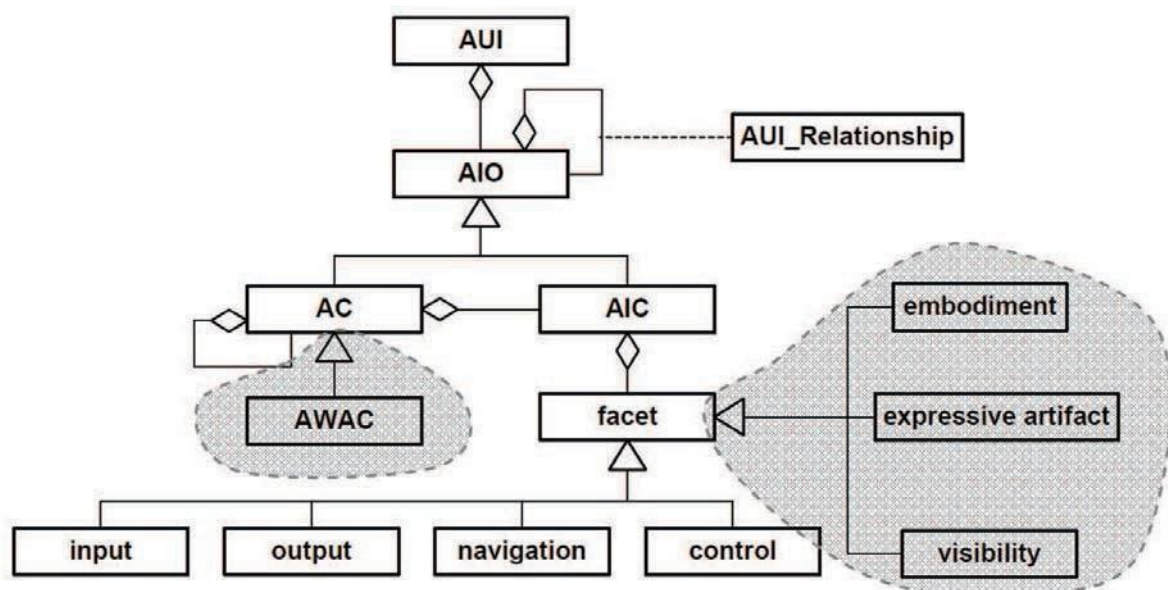


Figure 2.12 – Extensions (entourées) au métamodèle d’AUI d’UsiXML [70] par Penichet et al. [89] (TOUCHE). Représenté sous la forme d’un diagramme de classes UML.

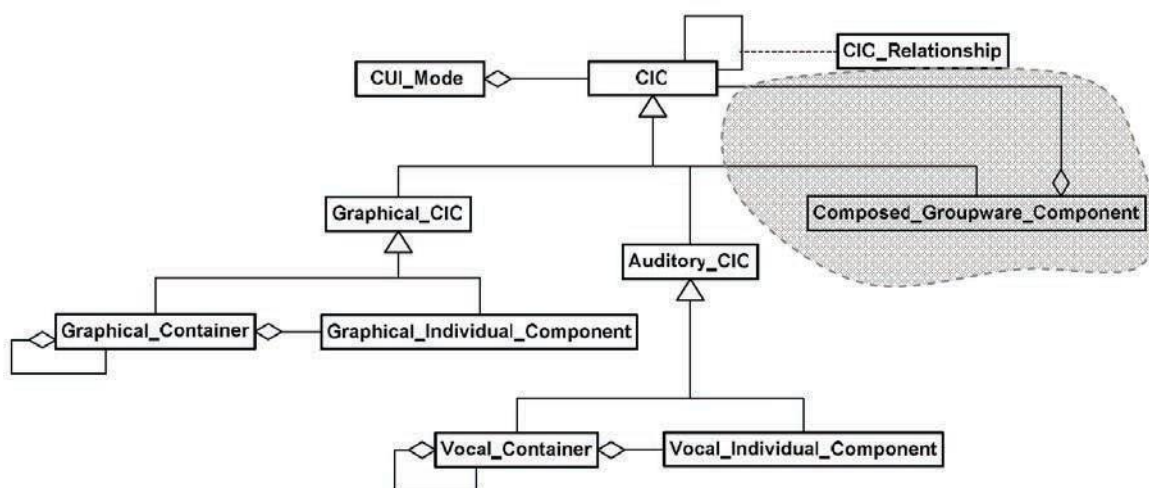


Figure 2.13 – Extensions (entourées) au métamodèle de CUI d’UsiXML [70] par Penichet et al. [89] (TOUCHE). Représenté sous la forme d’un diagramme de classes UML.

ou logiciel (e.g. Windows, Linux, etc.). Pour cela, les auteurs proposent une extension à UsiXML permettant de décrire des widgets multi-plateformes, ou plutôt ayant de multiples représentations (polymorphisme). Cette extension, **Widget Specification Language** (WSL), se propose de décrire les widgets suivant trois aspects :

Un modèle comportemental qui décrit la logique du widget, via des automates finis qui détaillent les états possibles. Il **s'agit** en fait de la spécification technique du widget ;

Un modèle d'abstraction qui définit les propriétés générales du widget, telles que sa capacité à fournir une conscience de groupe, sa dénomination, sa **boîte** à outils cible, etc. ;

Un modèle de cohérence qui sert à synchroniser différentes instances **d'un** même widget à travers les instances **d'IHM**. Ce modèle détermine quelles propriétés sont partagées entre les instances.

WSL est alors un formalisme XML qui permet de lier ces trois aspects. Chaque représentation **d'un** widget dispose de son propre modèle comportemental et **d'abstraction**, puisque ses caractéristiques sont directement dépendantes de la plate-forme de la représentation. WSL intervient au niveau des CUI en remplaçant les composants UsiXML habituels. **C'est** lors de la transformation vers FUI que le widget correspondant à la plate-forme sera utilisé.

Spécifier un widget polymorphe revient alors à décrire chaque représentation possible de ce widget et de ses caractéristiques. Ce qui revient au final à faire une spécification pour chaque plate-forme ciblée, limitant **l'ampleur** de **l'adaptation**. De plus, la prise en compte de la conscience de groupe **n'est** pas réalisée suivant des besoins identifiés, mais sera présente uniquement si la **boîte** à outils de la plateforme cible le permet (**qu'il** y ait un besoin ou non). WSL ne répond donc **qu'à** la question de la transition entre CUI et FUI, sans vraiment tenir compte de besoins amont, les travaux ne détaillant pas si des modèles de la collaboration ont été utilisés pour conduire aux CUI.

2.1.2.4 AMENITIES et génération d'IHM

Rodriguez et al. [91] proposent de produire des IHM à partir **d'AMENITIES** [44]. Pour rappel, la méthodologie **d'AMENITIES** aboutit en sortie à des modèles décrivant **l'activité** via COMO-UML. Les auteurs proposent de partir de ces modèles pour concevoir des IHM, sous une approche relative aux MBUI avec plusieurs modèles pour décrire une IHM (voir figure 2.14) :

Les modèles d'AMENITIES à savoir le modèle de **l'activité**, le modèle **d'organisation** et les modèles de spécification ;

Un modèle de présentation qui décrit les composants de **l'IHM**, à la manière **d'un** modèle **d'interface** concrète **d'UsiXML** [70] ;

Un modèle logique qui décrit le comportement de **l'IHM** sur base des modèles de spécification. Ce modèle utilise un formalisme dédié, MOORE (**Model Oriented to Objects, Relationships and Manipulative Process**), qui permet de représenter la logique

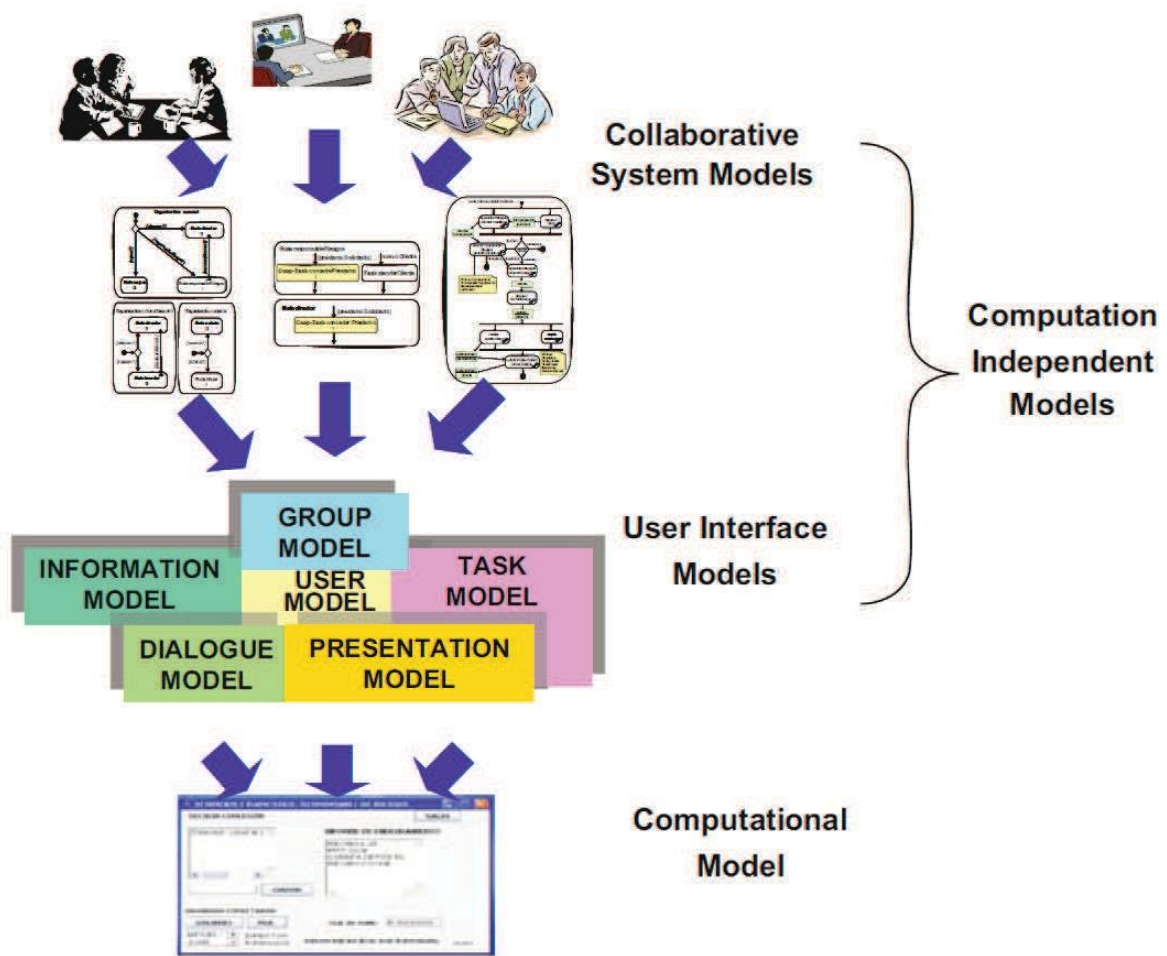


Figure 2.14 – Approche de conception d’IHM pour la collaboration de Rodriguez et al. [91].

de l’IHM sous forme de prédicats appliqués au modèle de présentation et les propriétés des composants.

Ces modèles définis, les auteurs suggèrent d’utiliser des boîtes à outils dédiées à la conscience de groupe comme MAUI [58, 59] afin de réaliser l’implémentation technique.

L’approche IDM de Rodriguez et al. [91] n’est toutefois pas transformationnelle, les transitions entre chaque étape étant manuelles.

2.1.2.5 Synthèse

Parmi les différentes méthodologies de modélisation de collecticiels étudiées, nous notons que seul TOUCHE [89] représente une approche couvrant la majorité des aspects de notre problématique. Vis-à-vis de la génération d’IHM, TOUCHE propose des extensions simples et pertinentes aux AUI et CUI pour la conscience de groupe [89]. Cependant, l’utilisation de CTT [84] (dans sa forme originelle) ainsi que des modèles de haut niveau (comme l’organisation) non considérés lors de la génération de l’IHM rendent l’approche

incomplète. Le support de flux de travaux à l'exécution n'est par exemple pas spécifié. Concernant les travaux de Rodriguez et al. [91], qui font suite à AMENITIES [70], nous notons essentiellement leur contribution au niveau de la description logique de l'IHM via le langage MOORE [91], un aspect souvent sous-représenté dans les MBUI. En dehors des différences en termes de modèles, les interfaces finales chez Rodriguez et al. [91] ou Penichet et al. reposent sur des boîtes à outils spécifiques à la conscience de groupe.

Concernant l'adaptation, seuls les travaux sur WSL [121] tentent d'y répondre, mais cependant uniquement vis-à-vis de l'espace technologique et d'une manière peu élégante mais concrète, concevoir un widget polymorphe revenant à concevoir toutes ses représentations au préalable. Les auteurs de WSL ne se posent également que la question de la représentation au niveau de la CUI et du passage à l'IHM finale, ne détaillant pas de quelle manière les widgets peuvent répondre à des besoins de collaboration.

Résumé de section

En synthèse de cette section, nous **notons l'approche** de modélisation pertinente des méthodologies de conception pour collecticiels. Celles-ci revêtent une approche par étapes :

1. La modélisation des rôles et de leur hiérarchie (i.e. **l'organisation**);
2. La modélisation des tâches et de leur attribution aux rôles/acteurs;
3. La modélisation des interdépendances entre les tâches et **l'accès** aux ressources (i.e. **l'activité** dans son ensemble);
4. La modélisation des tâches utilisateurs en détail.

Vis-à-vis **de l'IDM** et de la **génération d'IHM adaptatives**, ces approches globales présentent des limitations :

La représentation conceptuelle de la collaboration **n'est** pas toujours propice à **l'IDM** alors **qu'il s'agit d'une** notion importante du contexte, et donc essentielle dans **l'adaptation** des IHM;

Le formalisme de description des tâches CTT [84] ne suffit pas à décrire convenablement des tâches collaboratives;

La conscience de groupe **n'est** pas traitée à tous les niveaux **d'abstraction** et ne tire pas partie des informations modélisées en amont de la génération;

Les flux de travaux, bien que modélisés en amont, ne sont pas mis en application de manière opérationnelle.

2.2 Supporter la coordination à haut niveau

Comme nous venons de le voir en section 2.1, les approches couvrant tous les aspects de la génération **d'IHM** ne traitent pas systématiquement tous les aspects que nous avons mis en avant au chapitre précédent. Après avoir adopté un point de vue général, nous étudions dans cette section et les suivantes les concepts essentiels au cas par cas. Cette présente section est dédiée aux concepts de haut niveau et surtout à leur modélisation. Cet aspect est important car, comme nous l'avons vu, ces modèles de haut niveau (i.e. éléments théoriques, qui décrivent la collaboration dans son ensemble) viennent en entrée des processus de génération **d'IHM** et la pertinence des IHM ainsi produites dépend de leur expressivité.

Dans ce support à haut niveau, nous ne nous intéressons pas en détail à certains concepts qui sont le contexte **d'interaction** (i.e. le trio utilisateur/matériel/environnement physique), le domaine et les outils (i.e. les IHM en tant que telles, déjà décrites par les modèles **d'AUI/CUI**) que nous jugeons déjà adéquats dans des approches comme UsiXML [70]. En section 2.2.1, nous nous intéressons donc aux métamodèles de **l'activité collective** qui englobent généralement **l'activité**, les acteurs, **l'organisation** et les ressources. Nous étudierons aussi un aspect important pour la génération **d'IHM** en section 2.2.2 : la modélisation de tâches collaboratives.

2.2.1 Métamodèles de l'activité collective

Disposer **d'un** métamodèle général de **l'activité collective** est important. Celui-ci permet **d'avoir** un cadre général **d'analyse** du contexte de **l'activité** en regroupant tous les concepts nécessaires. De plus, la construction **d'une** instance **d'un** tel métamodèle permet **d'aider** à formaliser les exigences de conception **d'un** système interactif. Il **s'agit** également **d'un** support à la communication inter-disciplinaire utile lors des phases **d'analyse** et de conception du système.

Du point de vue de **l'IDM**, il est important que les modèles les plus abstraits véhiculent des informations ayant une sémantique forte et complète car il **s'agit** des concepts qui seront transformés en modèles plus concrets. Nous l'avons vu en section 1.2, le lien entre modèles théoriques et pratiques est fort, un manque **d'informations** dans les premiers peut résulter en une qualité moindre des seconds.

Lors de **l'analyse** de ces métamodèles, nous nous sommes intéressés aux concepts essentiels identifiés en section 1.2.3, soit la modélisation de **l'activité**, des **acteurs** & leur **organisation**, des **ressources** et des **outils**. Nous verrons que ces métamodèles généraux proposent souvent un niveau **d'abstraction** fort, peu opérationnalisable, ce qui nous conduira dans les sections suivantes à considérer des métamodèles plus spécialisés sur certains aspects (notamment autour du concept **d'activité**).

2.2.1.1 Métamodèles théoriques

L'approche la plus directe pour modéliser la collaboration est sans nul doute de regarder du côté des théories fondamentales, que nous avons abordé en section 1.2.3. Ces théories visent à formaliser et comprendre la manière dont fonctionne la collaboration

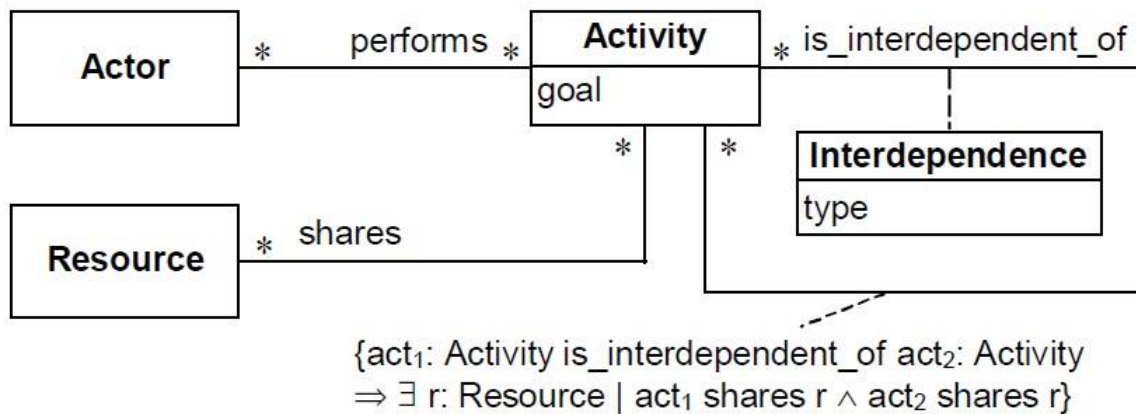


Figure 2.15 – Métamodèle de la théorie de la coordination [73] suivant De Farias et al. [29]. Représenté sous la forme d'un diagramme de classes UML.

et/ou la façon dont **s'organisent** les acteurs. Bien que ces théories **n'explicitent** pas forcément de modèle **d'analyse**, elles sont suffisamment formelles pour être exprimées sous forme de métamodèle.

À ce titre, De Farias et al. [29] ont étudié la modélisation des théories les plus dominantes et proposent des métamodèles UML de celles-ci. Nous relèverons surtout la modélisation des théories auxquelles nous avons fait référence au chapitre précédent (section 1.2.3), soit la théorie de la coordination [73], la théorie de **l'activité** [79, 68] et la théorie de **l'action/interaction** [38].

Le métamodèle de la théorie de la coordination, illustré en figure 2.15 est relativement simple et fait écho à la figure 1.2 qui schématisait les mécanismes de coordination au chapitre précédent. Pour rappel, cette théorie est centrée sur la notion de tâches (**activity** en figure 2.15) qui produisent ou consomment des ressources. Ces dernières peuvent être consommées par plusieurs tâches (i.e. partage de ressources), être le fruit de plus **d'une** tâche (i.e. ajustement des tâches) ou produites/consommées de manière séquentielle. Ce métamodèle de De Farias et al. correspond bien à la théorie mais est toutefois limité vis-à-vis de notre analyse. En effet, telle que définie par Malone et al. [73], la théorie de la coordination **n'est qu'un** noyau minimaliste de représentation de la coordination, son but n'étant pas de décrire la collaboration dans son ensemble. Par conséquent, ce métamodèle ne permet pas **d'exprimer** tous les concepts (tels que **l'organisation** et les rôles), voire uniquement une forme minimale de **l'activité** (i.e. relations entre les tâches) et des ressources.

La théorie de **l'activité** est plus complète et est généralement un point de référence dans la conception de collecticiels. Son métamodèle, en figure 2.16, reprend les concepts de communauté régie par des règles, de sujets (i.e. **participants**), **d'objet** (i.e. le but à atteindre), **d'outils** et de division du travail.

La théorie de **l'action/interaction**, modélisée en figure 2.17, utilise le concept de trajectoire qui, dans leur approche [29], représente **l'enchaînement** des actions (i.e. tâches) nécessaires à la réalisation **d'une** activité. En un sens, une trajectoire est un but commun à atteindre. Cependant, cette approche de **l'activité** est plutôt linéaire dans la mesure

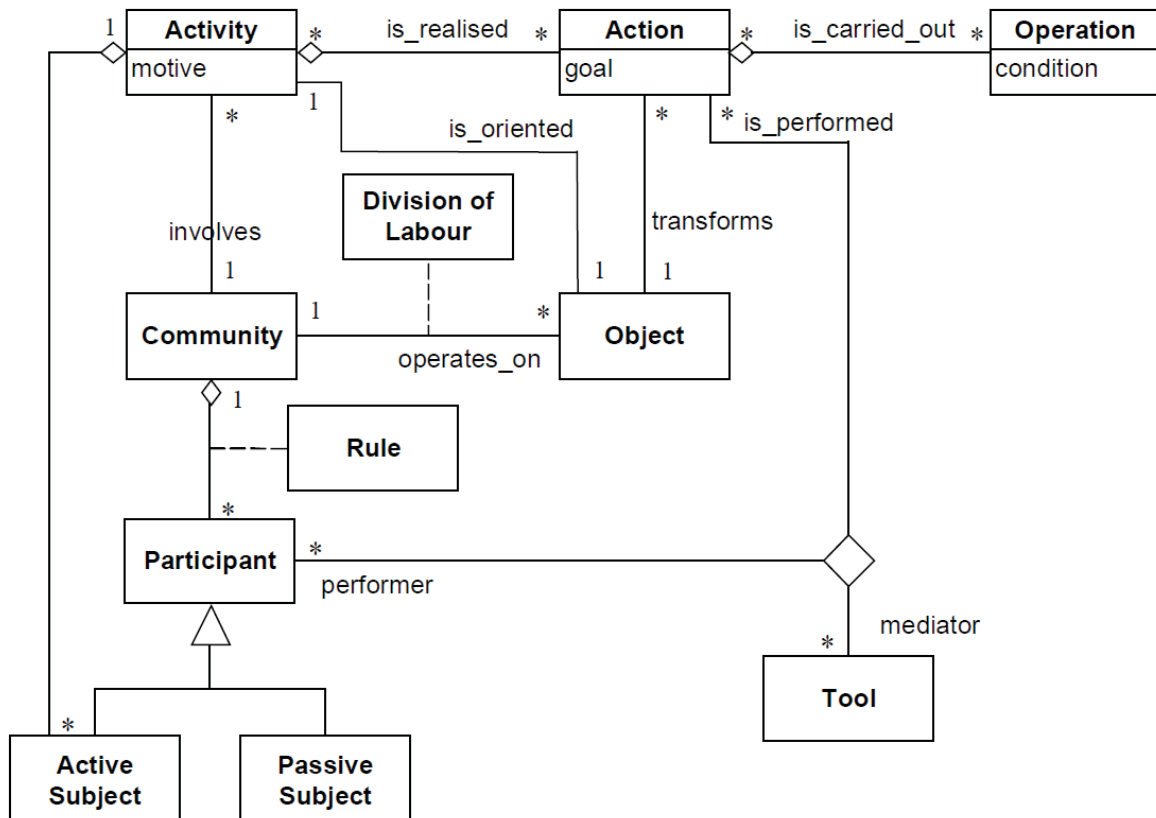


Figure 2.16 – Métamodèle de la théorie de l'activité [79, 68] suivant De Farias et al. [29]. Représenté sous la forme d'un diagramme de classes UML.

où elle ne permet de modéliser que des séquences simples sans synchronisation (i.e. plusieurs tâches sont requises avant le début **d'une** autre) ni parallélisme (i.e. plusieurs tâches peuvent démarrer en même temps). Il est toutefois intéressant de noter que cette théorie est divisée en deux parties : une relative aux trajectoires et **l'autre** relative aux aspects sociaux (i.e. **social world**). Cette deuxième partie détermine le « contexte social » des membres **d'un** groupe lorsqu'ils entreprennent une action, soit **l'outil/l'IHM** médiatisant leur tâche ainsi que leur localisation. Il **s'agit** donc **d'une** vue théorique plutôt intéressante lorsque **l'on** considère la conscience de groupe et le **contexte d'interaction**. Nous soulevons toutefois un manque **d'expressivité** quant à la définition **d'une** organisation qui se limite à des groupes **d'acteurs** sans plus de précisions concernant leurs relations.

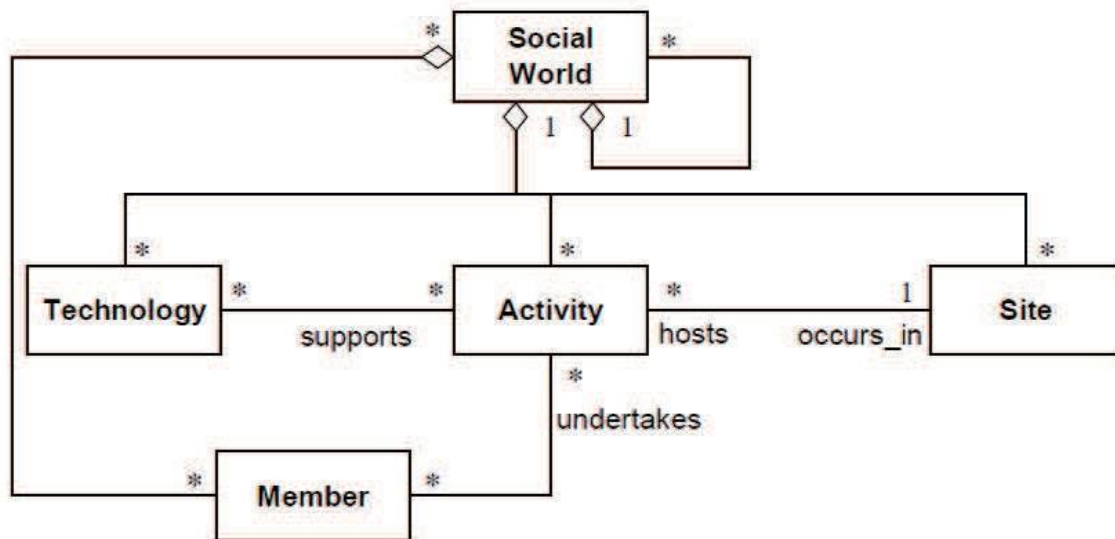
2.2.1.2 Métamodèles de synthèse du TCAO

En dehors de **l'interprétation** des théories de la collaboration, **l'autre** facette consiste à regarder du côté des métamodèles de synthèse du TCAO. Ces métamodèles cherchent à caractériser la collaboration dans son ensemble afin de servir de cadre de conception pour les systèmes collaboratifs.

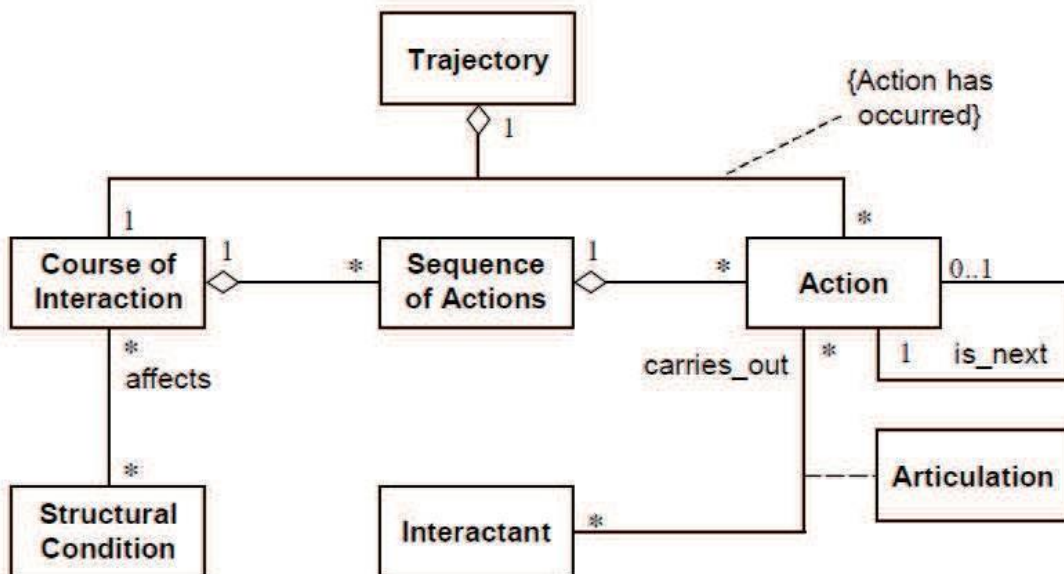
De Farias et al. [29], sur base des théories susmentionnées, proposent **d'ailleurs** leur propre synthèse visible en figure 2.18. Nous y retrouvons **l'ensemble** des concepts, organisés autour des tâches (**activity**), mais dans une forme synthétique. Il est intéressant de voir que ce métamodèle fait la distinction entre les outils (**service**) qui médiatisent les tâches, qui peuvent être informatiques ou non. Ceci permet **d'inclure** dans la modélisation les tâches qui ne relèvent pas toutes **d'une** IHM, une notion importante dans le cas des flux de travaux. Néanmoins, ce métamodèle réduit la notion de communauté par une association simplifiée entre acteurs et activités qui spécifie les rôles et les règles de coordination. Cette simplification se reflète également dans la division du travail proche de la théorie de la coordination. De Farias et al. proposent donc un métamodèle synthétique plutôt **qu'exhaustif**.

AMENITIES [44] est une méthodologie de conception pour systèmes collaboratifs dont le but est de cadrer **l'analyse** des exigences métiers afin de mieux concevoir de tels systèmes (décrite en section 2.1.1.2, page 60). Ces travaux ont **l'intérêt** de reposer sur un métamodèle **d'ensemble** qui cadre les aspects de la collaboration. Ce métamodèle, visible en figure 2.6 (page 61), est basé en partie sur la théorie de l'activité. On y retrouve donc des concepts et un agencement similaire mais tantôt plus détaillés. **C'est** notamment le cas de la notion de communauté, ici plus à même de décrire des structures complexes avec les notions de groupes, de rôles et **d'organisations** qui peuvent être composés. La division du travail est également mieux représentée avec **l'introduction** d'événements et de protocoles qui permettent une définition de **l'activité** proche **d'un** flux de travaux. AMENITIES propose donc un métamodèle plus proche **d'une** réalité pratique, ce qui en soi peut paraître convenu dans la mesure où la méthodologie vise à concevoir des systèmes concrets, et non à analyser la collaboration de manière comportementale.

UsiXML [70] (implémentation de CAMELEON [21, 113, 23]) propose également un métamodèle de collaboration à travers son métamodèle de tâches. **Comme nous l'avons** vu au chapitre précédent (voir section 1.3.2.2, page 42), UsiXML permet de modéliser toute la **chaîne** de génération **d'IHM** suivant le cadre de CAMELEON (i.e. en partant de modèles



(a) Social world concepts



(b) Action/interaction concepts

Figure 2.17 – Métamodèle de la théorie de l’action/interaction [38] suivant De Farias et al. [29]. Représenté sous la forme d’un diagramme de classes UML.

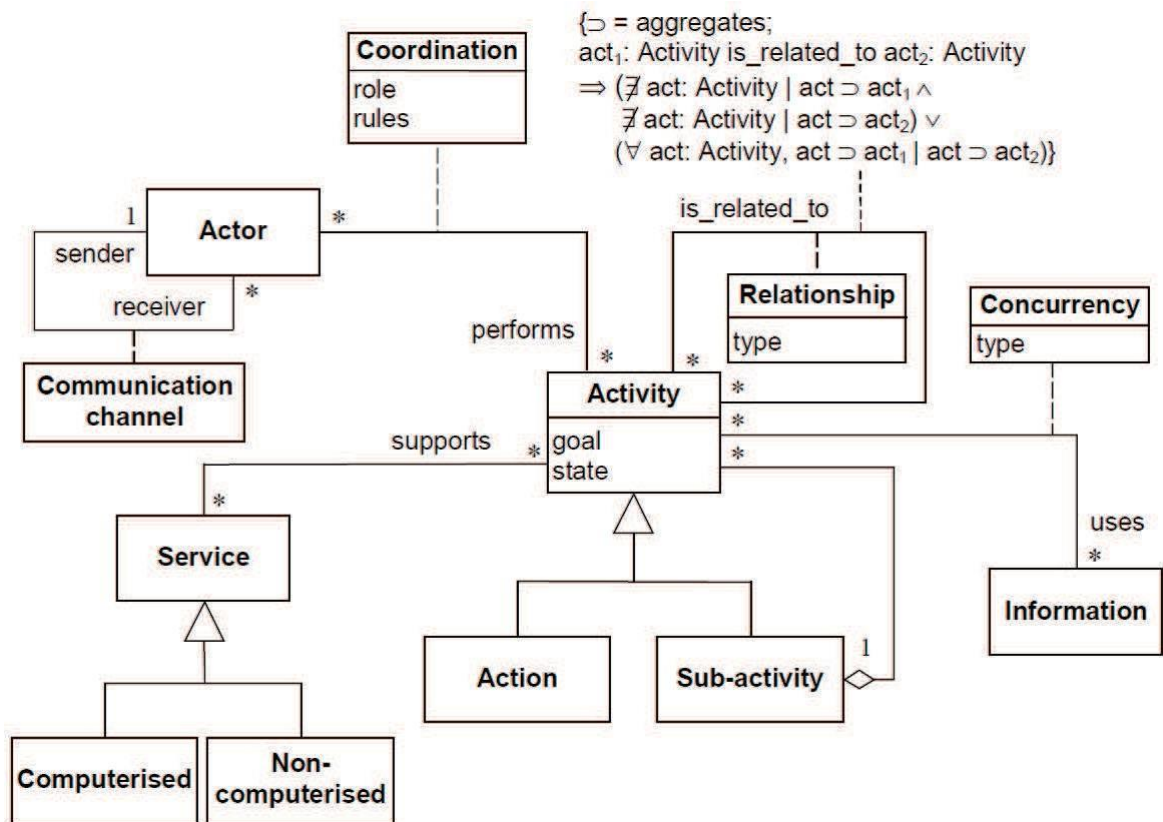


Figure 2.18 – Métamodèle de synthèse de la collaboration de De Farias et al. [29]. Représenté sous la forme d'un diagramme de classes UML.

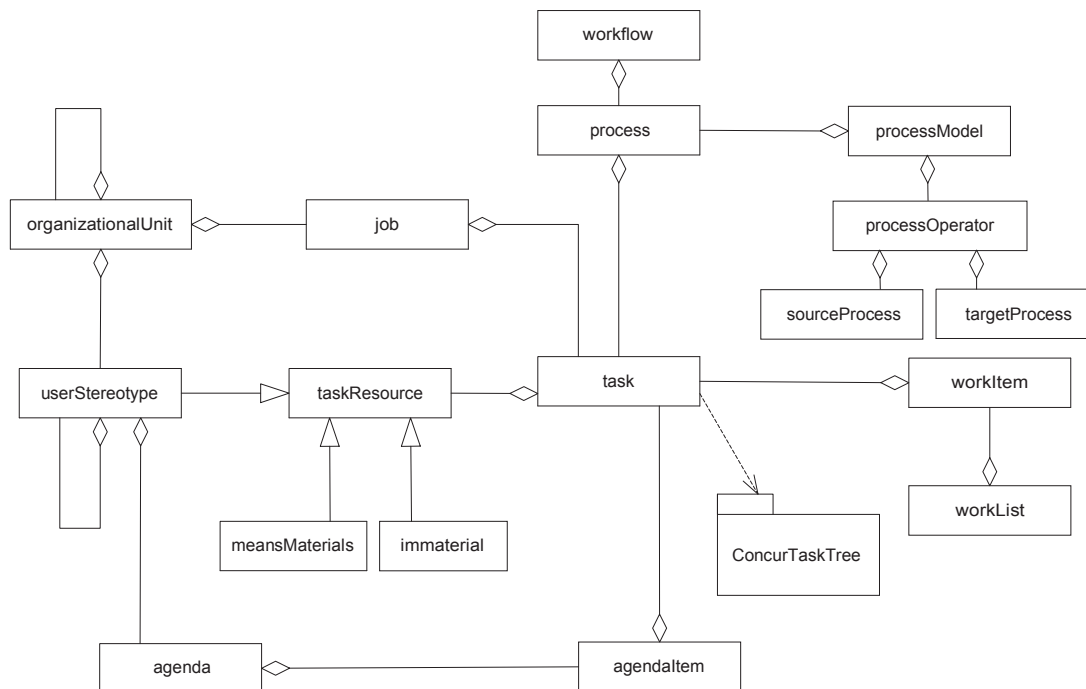


Figure 2.19 – Métamodèle de tâches d’UsiXML [70, 51], représenté sous la forme d’un diagramme de classes UML. Les éléments détaillés provenant de CTT [84] ont été édulcorés pour une meilleure lisibilité.

de tâches, domaines et contextes pour aller vers des IHM concrètes). Cependant, UsiXML couvre des aspects au-delà de CAMELEON et se veut notamment multi-organisations et multi-utilisateurs. En ce sens, son métamodèle de tâches, illustré partiellement en figure 2.19, présente des notions en plus de la description ConcurTask Trees [84]. UsiXML permet notamment la modélisation des organisations [51], des rôles, des ressources et de la division du travail (i.e. **agenda** et **agendaItem**). Le langage permet aussi de définir des flux de travaux (i.e. **process**) pour organiser temporellement les tâches. Cependant, le concept d’acteur est limité car UsiXML se focalise sur des groupes de stéréotypes d’utilisateurs et n’identifie pas les acteurs individuellement alors que la division du travail et l’attribution des rôles le requièrent. Le langage manque par ailleurs de sémantique sur la nature des liens entre organisations et entre utilisateurs (ou stéréotypes). UsiXML est un langage qui a une vocation technique et non contemplative. Ce qui en fait un langage plus propice à l’implémentation qu’à l’analyse et la conception.

Jeners & Prinz [62] proposent un autre métamodèle conceptuel dont le but est de construire des collectifs polyvalents mais aussi de pouvoir comparer des systèmes entre eux. Leur métamodèle, en figure 2.20, est relativement générique via l’emploi de la notion de conteneur (**container**) qui permet de modéliser des groupes d’utilisateurs et d’outils (**artefact**) mais aussi d’y associer un ensemble d’attributs qui peuvent être des règles de coordination ou des rôles. Ce patron de conteneur n’est toutefois pas repris au niveau des

tâches/procédés qui aurait alors permis de modéliser des flux de travaux complexes.

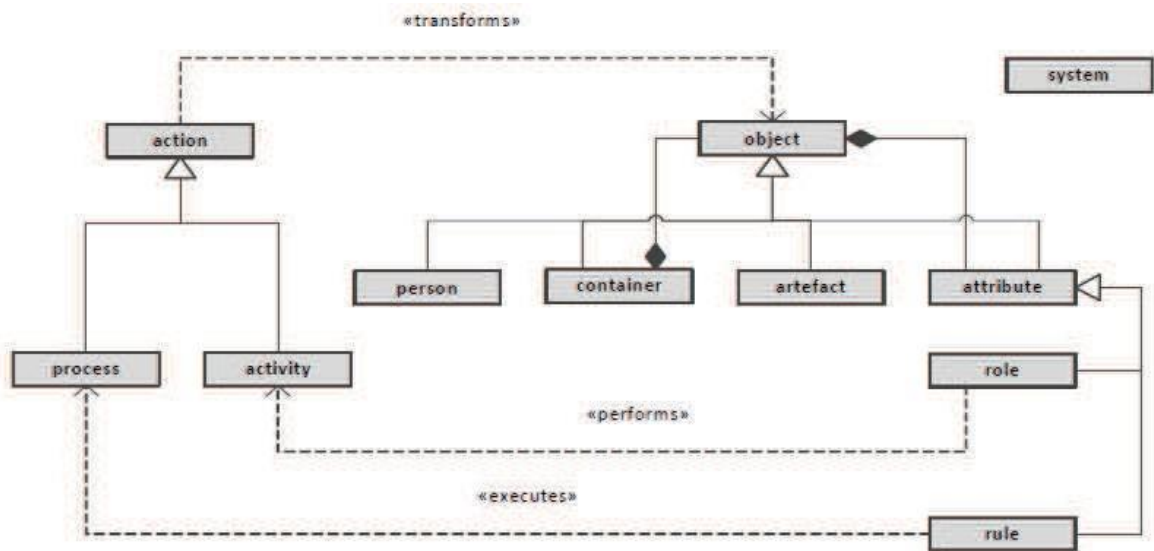


Figure 2.20 – Métamodèle de système coopératif de Jeners & Prinz [62]. Représenté sous la forme d'un diagramme de classes UML.

2.2. Supporter la coordination à haut niveau

Table 2.2 – Récapitulatif du support des concepts de base dans les métamodèles de la collaboration étudiés. Légende : **9** très adéquat à notre problématique, **Q** adéquat avec des réserves, **Q** présente des limitations significatives.

Métamodèle	Concepts				
	Activité	Acteurs	Organisations	Ressources	Outils
Th. de la coordination [29]	Q Interdépendance des tâches minimale	Q Pas de notion de rôle	N/A	Q Représentation minimale	N/A
Th. de l'activité [29]	Q Division du travail pertinente Q mais manque de temporalité	Q Pas de notion de rôle	Q Non multi-organisationnel	Q Représentation minimale	Q Représentation minimale
Th. de l'action/interaction [29]	Q Interdépendance des tâches linéaire	Q Pas de notion de rôle, 9 mais localisation	Q Notion de groupes, Q mais pas de hiérarchie/règles	N/A	Q Notion de technologie vague
De Farias et al. [29]	Q Interdépendance des tâches minimale	9 Représentation adéquate	Q Limité aux règles entre acteurs et tâches	Q Représentation minimale	9 Notion d'outils informatiques ou non
AMENITIES [44]	9 Représentation adéquate	9 Représentation adéquate	9 Représentation adéquate	Q Représentation minimale	Q Représentation minimale
UsiXML [70]	9 Représentation adéquate	Q Limité à des stéréotypes	Q Représentation minimale	9 Représentation adéquate	9 Représentation adéquate
Jeners & Prinz [62]	Q Interdépendance réduite	9 Représentation adéquate	9 Représentation adéquate	Q Représentation minimale	Q Représentation minimale

2.2.1.3 Synthèse

Pour résumer ces différentes approches, nous présentons en table 2.2 la couverture des concepts essentiels de ces métamodèles. Ainsi, AMENITIES [44] et UsiXML [70] proposent des métamodèles plutôt complets et surtout pratiques dans la mesure où ces travaux sont orientés vers une conception effective, et non à une analyse macroscopique de la collaboration comme c'est le cas pour les autres métamodèles. Aucun des métamodèles ne convient parfaitement à notre problématique mais un rapprochement entre la modélisation **d'UsiXML** et **d'AMENITIES** nous semble approprié. Soulignons aussi que le métamodèle de De Fariaset al. [29] de la théorie de **l'action/interaction** propose une vision intéressante en scindant **l'activité** en une partie opérationnelle (i.e. les tâches) et une partie sociale (i.e. le groupe et son organisation) qui sied à la conscience de groupe.

Vis-à-vis des métamodèles proposés dans les méthodologies de conception comme CIAM [76] et TOUCHE [88, 87], ceux-ci ne définissent pas de métamodèle général, mais partent plutôt sur des modèles spécialisés et segmentés, qui **d'un** point de vue méthodologique est plus judicieux dans leur construction (un modèle par étape). AMENITIES **n'utilise d'ailleurs** pas son métamodèle général pour la conception effective, mais repose dessus à la manière **d'une checklist**.

Enfin, il est à noter **qu'un** métamodèle générique de **l'ensemble** de collaboration est difficile à concevoir. Différentes facettes entrent en jeu et les métamodèles dans la littérature ont souvent des niveaux **d'abstraction** différents car ayant des buts différents. Il est donc judicieux de construire un métamodèle propre à nos besoins, ce que nous proposerons au chapitre suivant (section 3.1).

2.2.2 Métamodèles de tâches multi-utilisateurs

Comme nous venons de le voir, un métamodèle **d'ensemble** est souvent trop abstrait pour être précis au niveau des tâches vis-à-vis **d'une** approche IDM générative telle que CAMELEON [21, 113, 23]. Il est donc nécessaire de reposer sur un formalisme ayant une expressivité suffisamment précise pour pouvoir générer des IHM. Dans les implémentations de CAMELEON telles que UsiXML [70], le langage le plus couramment utilisé est ConcurTaskTrees (CTT) [84]. Or, ce langage a avant tout été conçu pour décrire des tâches à un niveau individuel.

Nous avons par conséquent étudié des alternatives de modélisation des tâches. Pour cela, nous passerons en revue dans un premier temps des dérivés de CTT qui visent la génération **d'IHM** collaboratives. Puis, en second temps, nous aurons un regard tourné davantage vers le génie logiciel qui propose des métamodèles de tâches sur base **d'UML**.

2.2.2.1 Dérivés de ConcurTaskTree

Mori et al. [78] (dont Paternò, auteur de CTT [84]) ont introduit une extension au formalisme **d'origine** afin de pouvoir modéliser des tâches collaboratives nommée Collaborative ConcurTaskTrees (CCTT). Pour ce faire, les auteurs proposent de modéliser un arbre de tâches par rôle et par tâche, puis de modéliser un arbre englobant toutes les tâches et rôles confondus (voir figure 2.21). Cet arbre de synchronisation («cooperative part» en

figure 2.21) permet de définir de manière globale la façon dont est synchronisée l'activité via les mêmes opérateurs temporels LOTOS [117] que CTT. CCTT se rapproche alors d'un flux de travaux qui synchroniserait des tâches individuelles. Cependant, cette séparation pose quelques problèmes. Il n'est par exemple pas possible d'exprimer dans une tâche individuelle (du moins de manière graphique) qu'une sous-tâche dépend temporellement d'une autre sous-tâche d'un autre acteur. Ce qui rend les tâches individuelles peu lisibles sans l'arbre de synchronisation et d'autant plus complexes à considérer pour générer des IHM via ce formalisme. De plus, Mori et al. mettent en avant le modèle 3C [34, 40], tandis que CCTT ne permet pas de représenter une coopération synchrone en espace partagé.

La notation *Collaborative and MultiModal* (COMM) de Jourde et al. [66, 65] se présente comme un dérivé de CTT avec une emphase sur les interactions multi-modales et multi-utilisateurs. COMM repose ainsi sur le même paradigme, soit une décomposition en arbre et des opérateurs temporels hérités de LOTOS. Sur les aspects multi-utilisateurs, COMM permet de décorer une sous-tâche avec un rôle et arbore une approche similaire à l'arbre de synchronisation de CCTT. À la différence de CCTT, COMM n'utilise pas d'arbres individuels par rôle mais prône la définition d'un seul modèle intégrant l'ensemble (voir figure 2.22). Pour cela, COMM ne mélange pas les sous-tâches de plusieurs rôles dans un nœud de l'arbre, mais prône plutôt un nœud par rôle et des liens entre nœuds. COMM y gagne alors en lisibilité par rapport à CCTT. Les auteurs ajoutent également un nouveau type de tâches : les tâches collaboratives qui impliquent plusieurs rôles. Ces tâches seront alors décomposées en nœuds pour chaque rôle impliqué et synchronisées à travers de nouveaux opérateurs temporels inspirés des travaux d'Allen [1] (voir figure 2.23) qui viennent en complément à LOTOS. COMM comble ainsi le manque d'expressivité sur la collaboration synchrone de CCTT mais n'est toutefois pas explicite sur le partage d'espace (bien que l'on puisse supposer la nécessité d'un espace partagé si un même concept du domaine doit être manipulé par plusieurs rôles en parallélisme/coïncidence).

2.2.2.2 Métamodèles de tâches en génie logiciel

Les approches basées sur des arbres, dont CTT [84] et MAD [93] font partie, sont régulièrement critiquées vis-à-vis de leur opérationnalisation. Si ces formalismes sont adéquats dans les domaines de l'analyse de la tâche, de la simulation ou de l'ergonomie, ils peuvent montrer leurs limites pour une implémentation technique réelle (dont la génération d'IHM via IDM). En effet, les arbres permettent difficilement d'obtenir une navigation complexe, notamment sur les tâches qui doivent prendre le relais en cas d'annulation ou de retour arrière. Plusieurs auteurs mettent cela en exergue [13, 67] et proposent des alternatives ou des compléments issus du génie logiciel.

Luyten et al. [72] partent de ce constat en vue de générer des IHM. Ils se posent notamment la question de la représentation des transitions entre les tâches afin de générer des dialogues proposant de telles transitions. En complément de CTT, les auteurs suggèrent l'emploi d'automates finis²³ pour modéliser les transitions de tâches non explicites de CTT (voir figure 2.24). Ces automates peuvent notamment être représentés via des diagrammes d'états-transitions UML.

23. Traduction de *finite-state machine*, dont la traduction littérale « machine à états finis » est souvent

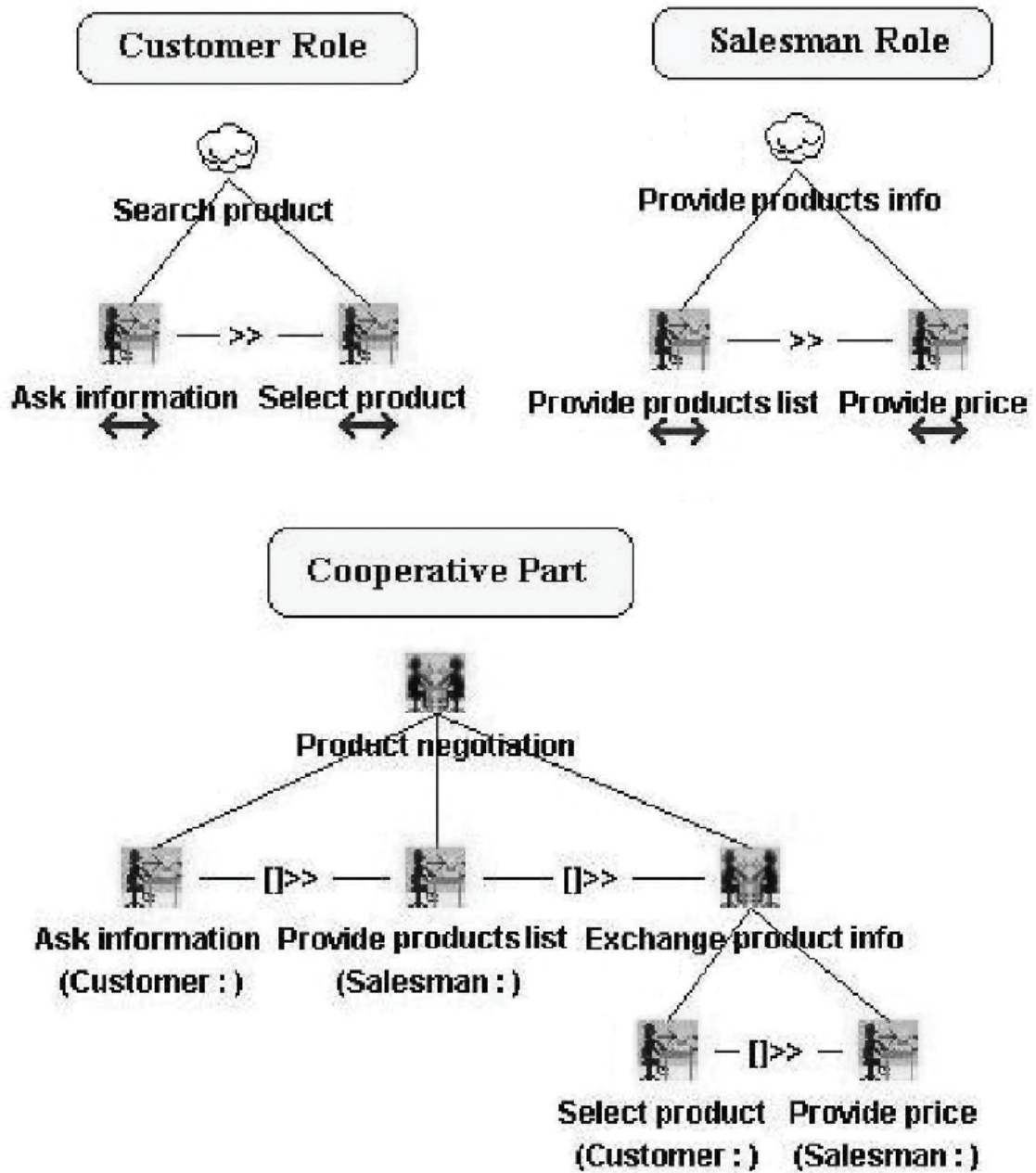


Figure 2.21 – Exemple de tâches collaboratives modélisées avec CCTT [78].

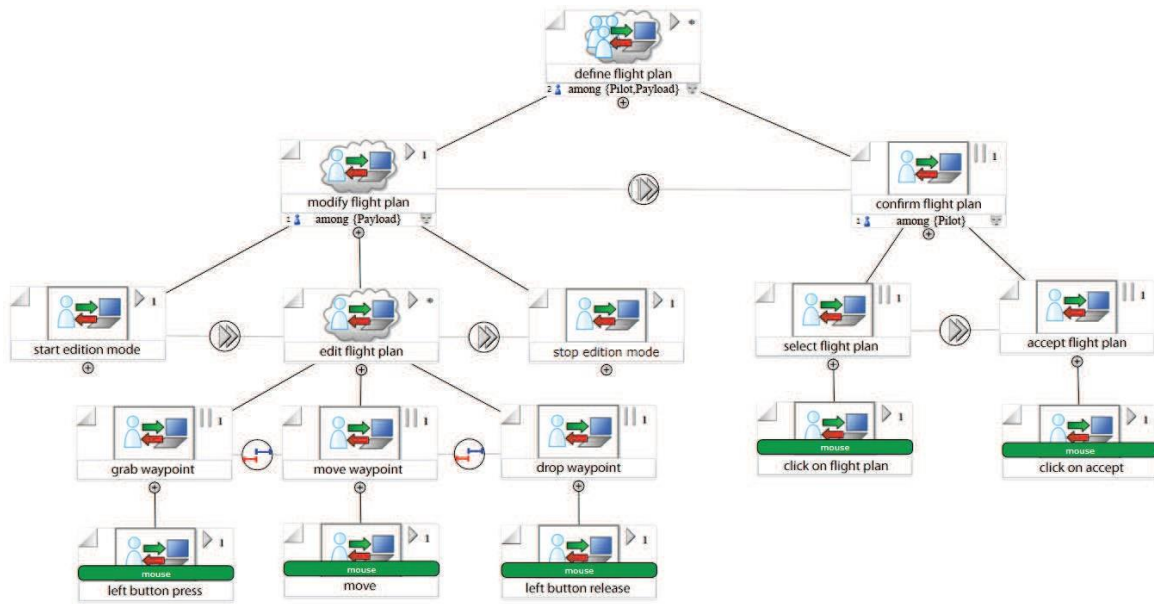


Figure 2.22 – Exemple de tâches collaboratives modélisées avec COMM [66, 65].

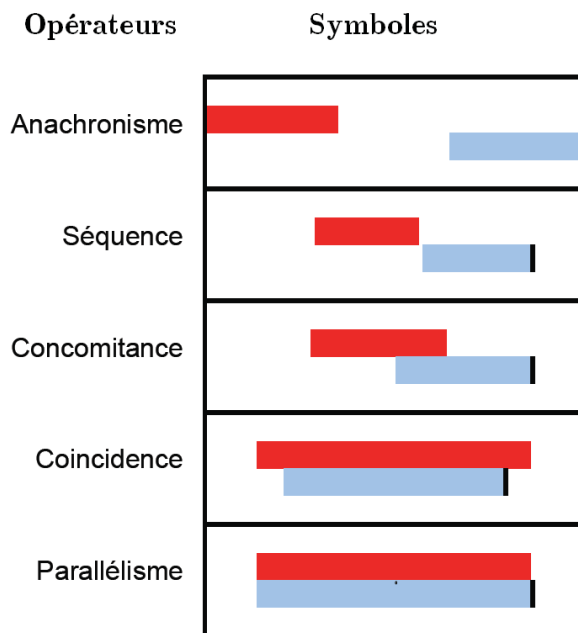


Figure 2.23 – Les opérateurs temporels d’Allen [1] (issus de la recherche en intelligence artificielle) utilisés dans COMM [66, 65].

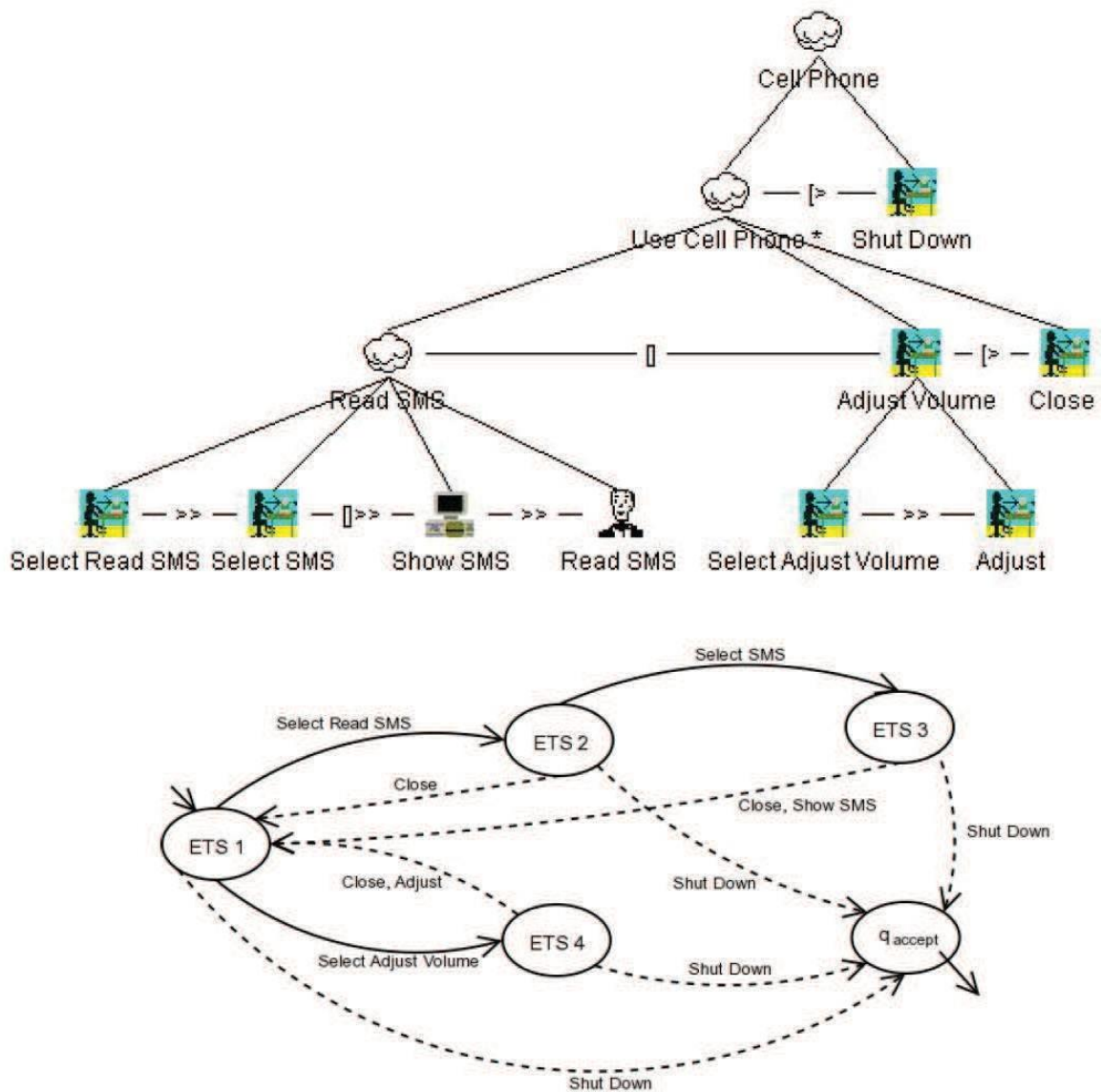


Figure 2.24 – Modélisation d'une tâche selon Luyten et al. [72]. Au dessus, une spécification CTT [84] classique, et au dessous l'automate fini qui spécifie son comportement.

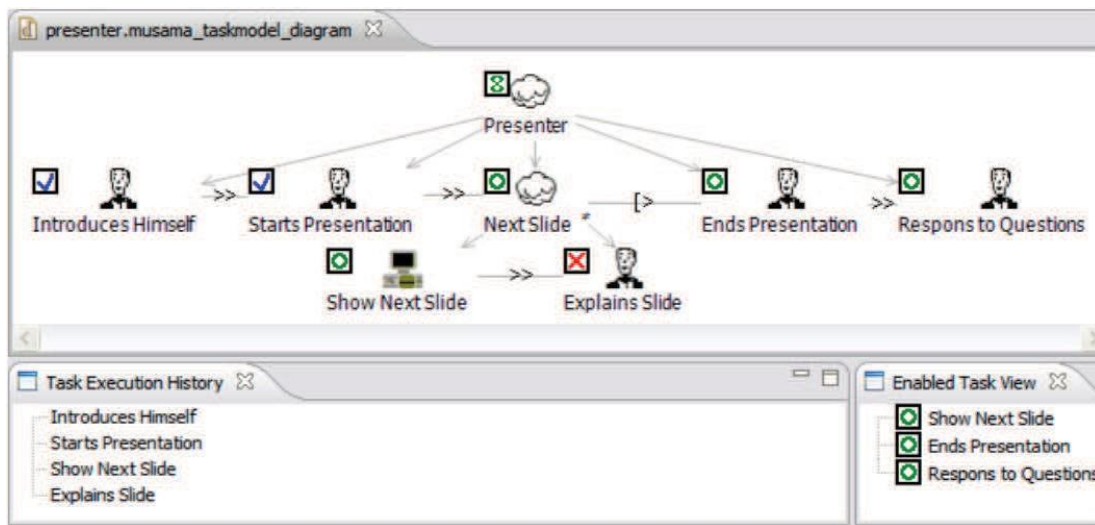


Figure 2.25 – Exemple de modèle de tâches CTML [126, 127, 39] tel que interprété dans l’environnement de simulation du modèle. Les icônes sur les **nœuds** représentent l’état de l’**automate** associé à chaque sous-tâche.

Dans une philosophie proche, Wurdel et al. [126, 127, 39] introduisent Collaborative Task Modelling Language (CTML), illustré en figure 2.25. Leur approche consiste, à la manière de CCTT, à modéliser un arbre de tâches par rôle sans toutefois reposer sur un arbre « chef d’orchestre. » En complément, chaque tâche et sous-tâche se voit attribuer un automate fini qui renseigne son état (e.g. active, inactive, suspendue, annulée, complétée, etc.) mais également les transitions d’un arbre à un autre, voire entre deux instances d’un même arbre. Ceci permet de représenter la collaboration entre les différentes instances de rôles. De plus, CTML est une approche IDM systématique. À travers son éditeur, les modèles sont implémentés par Eclipse Modeling Framework (EMF)²⁴ ce qui permet de les initialiser et de les interpréter pour simulation, puis génération d’IHM. Dans la même logique, les pré/post-conditions des tâches sont réalisées en Object Constraint Language (OCL)²⁵ qui permet d’exprimer des contraintes ou des requêtes sur des modèles. CTML se présente ainsi comme l’approche la plus concrète pour l’IDM, ou tout du moins la plus transparente sur sa réalisation technique. Notons tout de même que CTML ne semble pas pouvoir exprimer d’interaction en espace partagé, chaque instance de rôles ayant son propre état distinct.

2.2.2.3 Métamodèles d’analyse

En dehors des métamodèles pour l’IDM, certaines méthodologies de conception de collectifs offrent un regard intéressant sur la modélisation des tâches collaboratives. Bien que ces métamodèles aient plus pour vocation de servir de support d’analyse des besoins, et sont donc plus contemplatifs qu’opérationnels, ils ont l’avantage d’être proches d’une

utilisée, bien que fausse.

24. <http://www.eclipse.org/modeling/emf/>

25. <http://www.omg.org/spec/OCL/>

réalité pratique. **C'**est notamment le cas de CIAM [76] et AMENITIES [44]. Ces deux méthodologies proposent une étape de modélisation des tâches dérivée de diagrammes d'états-transitions UML.

Dans le cas de CIAM (dont un exemple de modélisation est visible en figure 2.4, page 58), chaque tâche est associée aux rôles intervenants et est décorée **d'un** attribut spécifiant la nature de collaboration (i.e. collaboration synchrone ou coopération simple). De plus, les transitions reprennent la notation LOTOS qui fait écho à CTT [84].

Deson côté, AMENITIES [44] introduit un métamodèle dédié qui étend UML, à savoir COMO-UML [43]. Cette notation (illustrée en figure 2.8, page 63) permet de modéliser les dépendances entre les tâches mais aussi **l'**intervention des ressources produites ou consommées.

Ces métamodèles **n'**ont pas la granularité de formalismes tels que CTT et ses dérivés mais mettent en exergue la pertinence **d'un** rapprochement avec le génie logiciel dans la modélisation des tâches.

2.2.2.4 Synthèse

La table 2.3 reprend nos critères **d'analyse**. Du côté des dérivés de CTT [84], nous identifions une certaine constance dans les approches : **d'une** part, la définition **d'arbres** de tâches individuels par rôle et par tâche, et **d'autre** part, un moyen de synchroniser ces arbres entre eux. Dans le cas de CCTT [78] et COMM [66, 65], un arbre **d'ordre** supérieur se charge de faire le lien entre les rôles et dans le cas de CTML [126, 127, 39], un langage permet **d'exprimer** ces contraintes. CCTT a **l'inconvénient** de présenter une certaine redondance entre les arbres individuels et **l'arbre** de synchronisation, ce que COMM corrige avec une représentation unifiée. COMM contribue également à une notation qui permet de décrire la nature de la collaboration entre les tâches (e.g. interaction synchrone, séquentielle, etc.). Néanmoins, cette approche « pyramidale » des tâches collaboratives nous apparaît trop séquentielle. CCTT et COMM semblent vouloir se rapprocher de la définition de flux de travaux, sans pour autant en avoir **l'expressivité** en termes de mécanismes de synchronisation (e.g. parallélisme, exclusion mutuelle ou synchronisation des tâches) et de transitions entre les tâches (e.g. brancher vers une tâche quelconque, retour arrière, annulation, etc.), soit des défauts courants hérités de CTT.

CTML nous paraît être **l'approche** la plus intégrée à **l'IDM** et donc la plus pertinente pour la génération **d'IHM**. De plus CTML comble certains manques de CTT en faisant un pont avec le génie logiciel, notamment en spécifiant l'état et les transitions des tâches via des automates finis et un langage de contraintes spécifique à **l'IDM** (OCL). Une approche également mise en avant par Luyten et al. [72] et surtout embrassée par les métamodèles plus analytique qui prônent un usage **d'UML** (via des diagrammes d'états-transitions) comme CIAM [76] ou COMO-UML [43].

Une tendance générale que nous pouvons souligner est le rapprochement de ces métamodèles avec les flux de travaux. CTML se présente même comme une alternative aux standards en place. Nous verrons par la suite **qu'il s'agit** en fait **d'une** notion transverse, qui dessert à la fois les aspects théoriques de la coordination (pour décrire **l'activité**) et les outils pratiques (pour coordonner **l'activité**).

Table 2.3 – Récapitulatif des métamodèles de tâches collaboratives étudiés. Légende : **9** très adéquat à notre problématique, **Q** adéquat avec des réserves, **Q** présente des limitations significatives.

Métamodèles	Critères			
	Description des tâches	Synchronisation des tâches	Nature de la collaboration	Lien avec l'IDM
CCTT [78]	9 Un arbre CTT par rôle/tâche	Q Arbre de synchronisation	Q Séquentielle uniquement	Q Conceptuel uniquement
COMM [66, 65]	9 Types de tâches dédiés à la collaboration, gestion des rôles	Q Arbre de synchronisation	9 Via opérateurs dédiés, complet	Q Conceptuel uniquement
CTML [126, 127, 39]	9 Un arbre CTT par rôle/tâche	9 Contraintes gérées via automates finis et OCL	Q Séquentielle uniquement	9 Intégré à EMF et exécutable
CIAM [76]	N/A	9 Métamodèle de l'activité (proche de flux de travaux)	9 Spécifiée dans le métamodèle de l'activité, complet	9 Intégré à EMF
COMO-UML [43]	N/A	9 Métamodèle de l'activité (proche de flux de travaux)	Q Séquentielle uniquement	9 Intégré à EMF

Résumé de section

Les métamodèles de la collaboration sont conçus pour des usages précis et **n'ont** pas la même granularité. Vis-à-vis de notre problématique, le couple UsiXML et AMENITIES nous paraît proposer les représentations des concepts de haut niveau les plus appropriées.

Ces métamodèles **d'ensemble** négligent la représentation des tâches collaboratives et, à ce titre, des métamodèles dédiés tels que COMM [66, 65] ou CTML [126, 127, 39] permettent **d'exprimer** des tâches conformément au modèle 3C [34, 40] et à la matrice de Johansen [63]. CTML est par ailleurs adapté à **l'IDM**.

2.3 Supporter la coordination à bas niveau

Après avoir couvert la modélisation des concepts de haut niveau, notre attention se porte sur ceux à bas niveau, **c'est-à-dire** le niveau de **l'interaction**. Bien que nous plaçons ces concepts au niveau le plus concret, nous verrons **qu'il** est nécessaire de les abstraire dans une certaine mesure afin **qu'ils** puissent **s'insérer** dans une approche générative. En effet, que ce soit la conscience de groupe ou les flux de travaux, leur mise en **œuvre** dépend **d'information** de plus haut niveau qui nécessitent alors un pont **d'abstraction**.

Dans cette section, nous passons en revue les métamodèles permettant de décrire la conscience de groupe et surtout **d'identifier** quand celle-ci est nécessaire vis-à-vis de modèles de plus haut niveau (section 2.3.1). Nous aurons une démarche similaire quant aux flux de travaux (section 2.3.2) qui, nous le verrons, représentent une notion transverse des différents niveaux **d'abstraction**.

2.3.1 Métamodèles et interactions pour la conscience de groupe

Lorsqu'il s'agit de la mettre en **œuvre**, la conscience de groupe est souvent considérée comme étant une vue sur une situation ou un contexte. Comme nous l'avons vu en section 1.1, le contexte **d'interaction** regroupe toutes les informations pertinentes vis-à-vis de **l'instant** de **l'interaction**. On y retrouve les informations caractérisant les utilisateurs, les appareils abritant les IHM, les particularités de **l'environnement** physique ainsi que, dans notre approche, tout élément décrivant **l'activité** (e.g. les tâches, **l'organisation**, les rôles, etc.). Procurer une conscience de groupe revient alors à permettre **d'accéder** à ces informations de manière transparente (i.e. en favorisant **l'absorption d'information** par gestaltisme, voir section 1.2.4.1 page 27), soit à fournir une vue sur des données particulières du contexte.

En termes de modélisation et de génération **d'IHM** par **l'IDM**, produire ces vues sur le contexte soulève plusieurs questions qui définissent nos critères **d'analyse**. « Comment modéliser le besoin en information ? » « Comment savoir quelles informations sont pertinentes à un instant donné ? » « Et comment, à partir **d'abstractions**, tendre vers une IHM fournissant ces informations ? » À cet effet, nous nous intéressons ici aux métamodèles permettant de décrire et mettre en pratique la conscience de groupe.

2.3.1.1 Abstraction de la conscience de groupe

Vis-à-vis de **l'IDM**, les approches pour la conscience de groupe visent à abstraire les besoins en informations et à générer les vues nécessaires (ou widgets de **boîtes** à outils). Ces besoins en conscience de groupe sont souvent exprimés à partir de la classification de Gutwin & Greenberg [53] (i.e. le détail du **who, what, where, when & how** ; voir section 1.2.4).

Penichet et al. [88, 87], dans leurs travaux sur la méthodologie TOUCHE, introduisent des extensions pour la conscience de groupe aux interfaces abstraites (AUI) et concrètes (CUI) du langage UsiXML [70]. Au niveau abstrait, TOUCHE permet de spécifier si un conteneur devra apporter de la conscience de groupe via un nouveau type de conteneur, les **Abstract Workspace Awareness Container** (ou AWAC, voir figure 2.12, page 73).

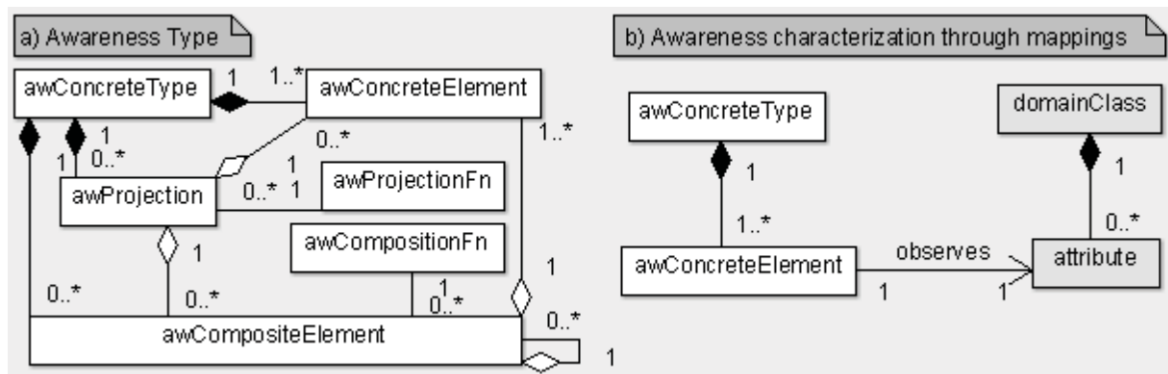


Figure 2.26 – Métamodèle d’awareness de Figueroa-Martinez et al. [37] dans leur approche d’UsiXML [70]. Représenté sous la forme d’un diagramme de classes UML.

Les extensions d’AUI permettent aussi de spécifier la nature de l’information de groupe (i.e. *embodiement*, *feedthrough* sur un artefact ou la visibilité des actions d’autrui). Puis, lorsque les AUI sont transformées en CUI, tous les composants issus d’un AWAC sont mis sous l’égide d’un autre conteneur concret (voir figure 2.13, page 73) qui spécifie alors que ces composants fournissent de la conscience de groupe. Pour ce faire, TOUCHE indique que ces composants peuvent se baser sur des boîtes à outils telles que MAUI [58] cité précédemment. Via ces conteneurs concrets, TOUCHE permet en fait de définir des widgets pour la conscience de groupe.

Figueroa-Martinez et al. [37] proposent également des extensions à UsiXML [70], mais à l’instar de TOUCHE, les auteurs partent sur l’abstraction même de la classification de Gutwin & Greenberg [53]. Pour cela, ils introduisent un nouveau métamodèle à UsiXML, celui d’awareness, qui permet de décrire la classification de Gutwin & Greenberg sous forme de catégories (e.g. *who, what, where*, etc.) et d’éléments (e.g. *location, action, embodiement*, etc.) d’une manière proche du métamodèle du domaine. Pour cause, les éléments de ce métamodèle sont alors associés aux éléments du domaine auxquels ils correspondent (voir figure 2.26). Les auteurs proposent ensuite des métamodèles permettant d’identifier des situations où une conscience de groupe serait requise. Ces situations sont définies via des contraintes sur le modèle de contextes d’UsiXML ainsi que sur le modèle du domaine qui détermine la source des informations à afficher. Ces deux types de contraintes sont alors associées à un modèle de tâches, ce qui liera la tâche à une situation et un besoin en conscience de groupe (voir figure 2.27). Enfin, lors de la transformation du modèle de tâches en AUI puis CUI, si les contraintes sont satisfaites, il est possible d’ajouter une visualisation sur les éléments du domaine correspondants au besoin en conscience de groupe. La démarche de Figueroa-Martinez et al. [37] est intéressante mais limitée à un lien avec le modèle de domaine en ce qui concerne la source des informations de groupe et aussi limitée au modèle de contexte (qui dans UsiXML se limite à la vision de Dey & Abowd [31]) pour ce qui est de l’identification de situations.

Gallardo et al. [41] proposent une ontologie de la conscience de groupe. Illustré en figure 2.28, ce métamodèle a pour vocation d’être une base d’analyse des besoins afin de concevoir des éditeurs graphiques collaboratifs. Cette ontologie s’articule autour de la notion de mécanisme de conscience de groupe qui consiste à informer de l’état d’une

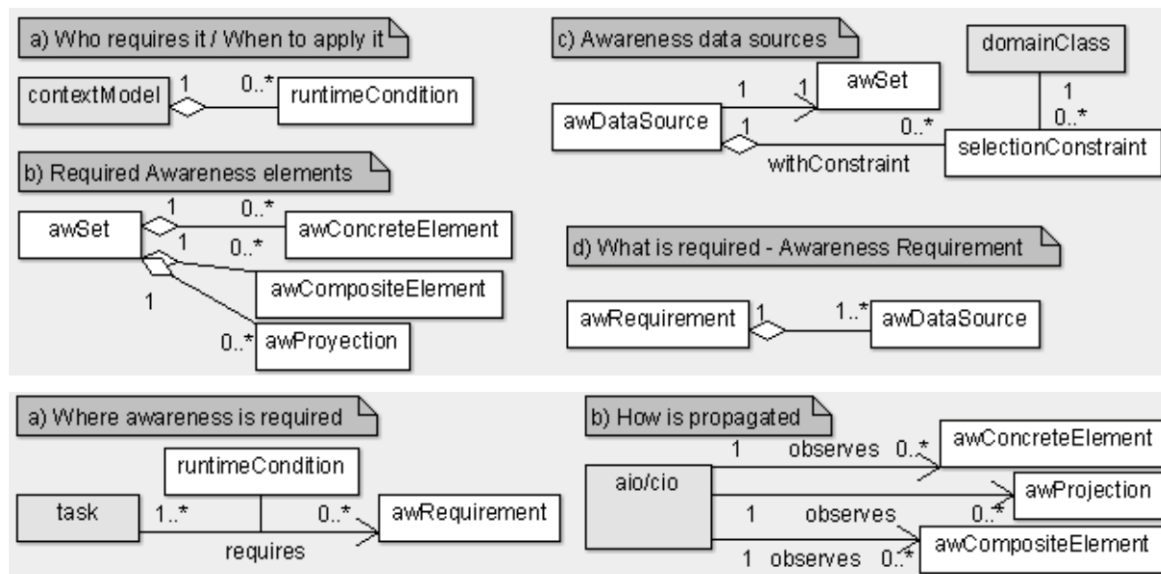


Figure 2.27 – Métamodèle de besoins en conscience de groupe de Figueroa-Martinez et al. [37] dans leur approche d’UsiXML [70]. Représenté sous la forme d’un diagramme de classes UML.

action d’un membre du groupe de travail, à travers un élément du *workspace* donné. Concrètement, l’ontologie met en avant les utilisateurs et leurs actions qui seront visibles à travers des widgets de conscience de groupe. Gallardo et al. [41] ont notamment construit leur ontologie autour de quatre widgets : le panneau de session (i.e. liste des utilisateurs connectés), la vue radar partagée, le télépointeur et l’historique des actions. Ensuite, suivant leur méthodologie, cette ontologie permet d’initialiser des modèles productifs au sein du cadre IDM *Eclipse Modeling Framework* (EMF)²⁶. Ces modèles sont alors transformés par le générateur d’éditeurs *Graphical Modeling Framework* (GMF, un projet Eclipse)²⁷ en intégrant les widgets décrits lors des phases amont de modélisation. Cette approche est évidemment limitée à la conception d’éditeurs graphiques qui n’ont pas nécessairement d’autres besoins en conscience de groupe que les quatre widgets précédents. Néanmoins, leur approche est plutôt concrète dans la mesure où les auteurs aboutissent à une implémentation technique basée sur leur modèle ontologique.

Belkadi et al. [8] abordent la conscience de groupe sous l’angle de la situation et en proposent un métamodèle dans le but de servir de source d’informations aux IHM. Pour cela, ils passent en revue de nombreux travaux, dont ceux de De Farias et al. [29] et Garrido et al. [44] (entre autres), afin de déterminer sur quelles informations la conscience de groupe repose. Ils en tirent les informations suivantes :

Les entités basiques qui sont les utilisateurs et leur état (e.g. identité, rôle, tâche courante, etc.) ainsi que celui des ressources ;

Les entités opérationnelles qui représentent l’activité (e.g. buts, flux de travaux) et les tâches ;

26. <http://www.eclipse.org/modeling/emf/>

27. <http://www.eclipse.org/modeling/gmf/>

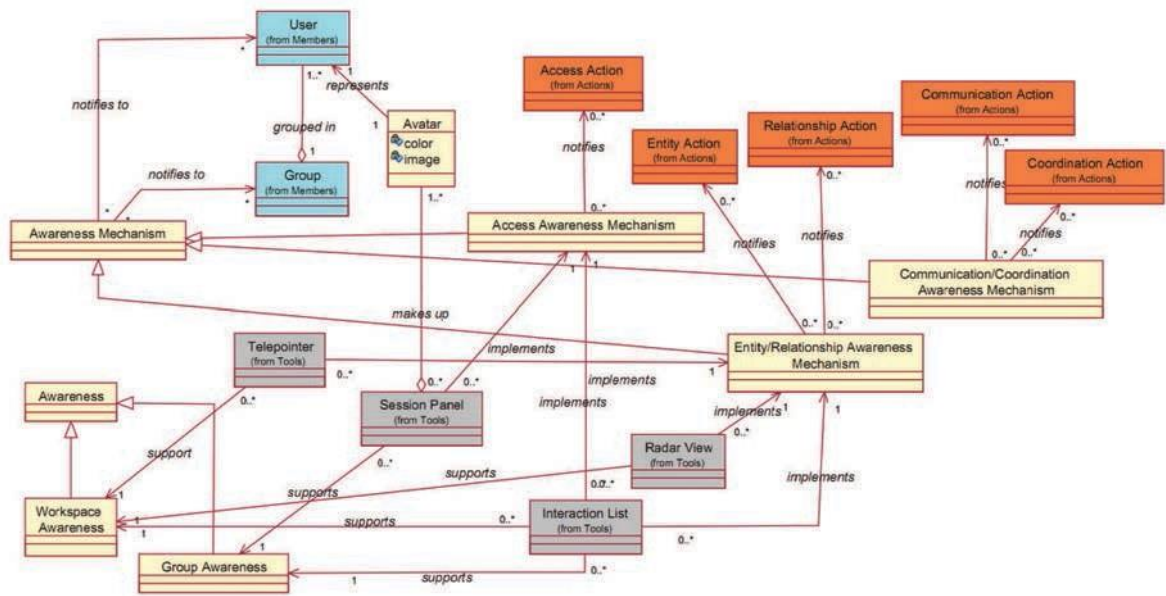


Figure 2.28 – Ontologie de la conscience de groupe de Gallardo et al. [41].

Les **entités communautaires** qui sont la description des organisations et des règles qui les régissent ;

Les **entités collaboratives** qui déterminent le fonctionnement de la collaboration, sur base du modèle 3C [34, 40] ;

Les **contraintes** qui sont les règles délimitant l'étendue des entités (e.g. contraintes organisationnelles, géographiques, etc.).

L'état de ces entités représente alors une situation. Belkadi et al. [8] résumant alors cette schématisation dans un métamodèle, illustré en figure 2.29. Ce métamodèle est alors proche d'une définition de l'ensemble de la collaboration telle que nous l'avons étudié en section 2.2.1 et met en avant le fait que la conscience de groupe repose sur la totalité du contexte d'interaction. Notons que ce métamodèle n'est pas utilisé à des fins d'IDM par Belkadi et al. [8] mais a pour vocation de diriger la phase de conception d'IHM collaboratives en identifiant clairement le besoin en conscience de groupe.

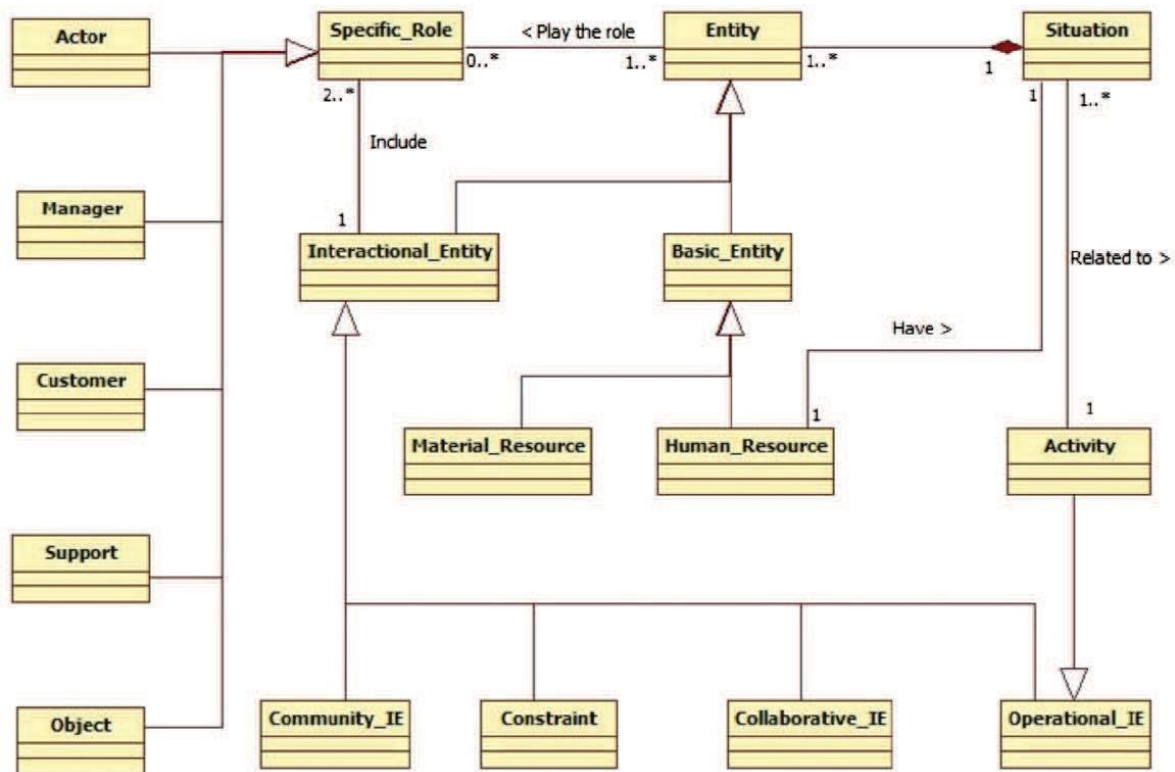


Figure 2.29 – Métamodèle de situation de Belkadi et al. [8]. Représenté sous la forme d'un diagramme de classes UML.

2.3.1.2 Synthèse

L'analyse de ces métamodèles pour la conscience de groupe nous permet **d'isoler** le fait **qu'il s'agisse d'un** mécanisme situationnel. C'est-à-dire que la conscience de groupe est pertinente vis-à-vis **d'une** situation donnée dont il **s'agira** de représenter l'état. Une situation est représentée par une vue sur **l'ensemble** du contexte **d'interaction**, soit le contexte utilisateur, de **l'organisation**, de **l'activité**, et le contexte des autres utilisateurs. En ce sens, la conscience de groupe repose sur **l'intégralité** du modèle de contexte, soit **l'ensemble** de la couche de haut niveau qui peut être exploitée. Elle doit donc être accessible en toute circonstance et ne pas être juste un modèle **d'entrée** pour la génération **d'IHM**.

Vis-à-vis de notre analyse, récapitulée en table 2.4, nous notons en premier lieu que les métamodèles étudiés couvrent des champs différents, mais complémentaires. Penichet et al. [88, 87] apportent via TOUCHE des extensions aux AUI et CUI **d'UsiXML** [70] qui permettent en bout de chaîne **d'initialiser** des composants dédiés à la conscience de groupe. Néanmoins, TOUCHE ne propose pas **d'abstraction** des besoins en conscience de groupe à plus haut niveau, ceux-ci étant déterminés par méthodologie au moment de la spécification des AUI (à travers les AWAC). Figueroa-Martinez et al. [37] complètent cela en étendant UsiXML avec un métamodèle de besoins ainsi **qu'un** métamodèle **d'identification** de situations, se limitant par contre à des contraintes sur le modèle de contexte **d'UsiXML** (qui **n'inclut pas l'activité**). Belkadi et al. [8], quant à eux, apportent la modélisation de situations plus complètes qui inclut **l'activité**.

2.3. Supporter la coordination à bas niveau

Table 2.4 – Récapitulatif des métamodèles de la conscience de groupe étudiés. Légende: **9** très adéquat à notre problématique, **Q** adéquat avec des réserves, **Q** présente des limitations significatives.

Métamodèle	Concepts		
	Modélisation du besoin en conscience de groupe	Identification de situations	Mise en œuvre
Penichet et al. [88, 87]	Q Pas d'expression des besoins à haut niveau, mais intégration de la conscience de groupe à partir des AUI/CUI (et déduit par méthodologie)	N/A	9 Boîte à outils dédiée à la conscience de groupe
Figuroa-Martinez et al. [37]	Q Lié (et limité) au modèle du domaine	Q Contraintes sur le modèle de contexte uniquement et les tâches	Q Points d'observation sur le domaine dans les AUI/CUI
Gallardo et al. [41]	N/A	N/A	Q Limitée à quatre types de widgets
Belkadi et al. [8]	N/A	Q Contexte de l'activité	N/A

2.3.2 Métamodèles et application des flux de travaux

La modélisation et la mise en pratique des flux de travaux est généralement moins problématique que la conscience de groupe du fait de l'**existence** de standards industriels tels que BPMN²⁸, XPDL²⁹, BPEL³⁰ ou, dans une moindre mesure, UML³¹ et ses **flowcharts**. Les flux de travaux permettent de décrire l'**activité** dans son ensemble, soit les interdépendances entre toutes les tâches et les rôles qui constituent l'**activité**. Il s'agit d'une représentation à un niveau macroscopique qui **n'a** pas pour vocation de décrire le niveau d'**interaction** à la manière d'un arbre de tâches. Néanmoins, on associe communément une tâche d'un flux de travaux à une IHM dédiée, ou tout du moins un écran spécifique dans un état spécifique. Le flux de travaux intervient alors en tant que chef d'**orchestre** et détermine quels acteurs ont accès à quelles IHM vis-à-vis des activités actives du flux. Dans le domaine du génie logiciel, la plupart des environnements de développement permettent de concevoir des applications sur base de flux de travaux, comme c'est le cas d'**Eclipse** (via un support de BPEL et BPMN) ou de Microsoft Visual Studio (via Windows Workflow Foundation³²). Certaines solutions de collecticiels peuvent également mettre en **œuvre** des flux de travaux (e.g. EGroupware³³).

2.3.2.1 Flux de travaux et MBUI

Dans le cas des MBUI et des espaces de modélisation pour la conception de collecticiels, on distingue divers tendances dans l'**usage** des flux de travaux. Comme nous l'avons vu en section 2.2.2, certains métamodèles de tâches collaboratives ont une approche similaire aux flux de travaux.

Les formalismes de description de tâches CCTT [78] et COMM [66, 65] ont des approches des flux de travaux similaires. Chaque tâche individuelle dispose de son propre arbre de tâches et un autre arbre de tâches (arbre de synchronisation) se charge de la synchronisation générale de l'**activité**. Bien que les auteurs **n'avancent** pas cet usage comme relevant d'un flux de travaux, la sémantique véhiculée ainsi que la mise en application est très proche, sans toutefois avoir l'**expressivité** des standards dédiés (notamment en termes de dépendance temporelle et de parallélisme des tâches).

Wurdel et al. [126, 127, 39], avec le métamodèle de tâches CTML, abordent le sujet à la manière de flux de travaux. Contrairement aux approches susnommées, CTML ne se base pas sur un modèle chapeautant la dépendance entre les autres modèles. CTML découpe l'**activité** en un modèle de tâches par rôles, les interdépendances entre tâches sont alors gérées via des pré et post-conditions sur les sous-tâches grâce à un langage de contrainte sur les modèles. Ces contraintes peuvent se substituer aux messages/événements que l'on retrouve dans les flux de travaux. CTML se revendique d'**ailleurs** équivalent à une approche par flux de travaux.

28. <http://www.bpmn.org/>

29. <http://www.wfmc.org/xpdl.html>

30. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>

31. <http://www.uml.org/>

32. <http://msdn.microsoft.com/en-us/vstudio/jj684582>

33. <http://www.egroupware.org/index.php>

Les méthodologies de conception de collecticiels CIAM [76], AMENITIES [44, 45] et TOUCHE [88, 87] ont une approche similaire des flux de travaux. Sans passer par des standards spécifiques aux flux de travaux, ces trois méthodologies ont une étape **d'identification de l'activité** qui consiste à déterminer les rôles, leurs tâches et les interdépendances entre toutes ces notions. CIAM par exemple, propose un modèle **d'inter/action** (voir figure 2.4) dérivant du diagramme d'états-transitions **d'UML**. AMENITIES dispose de sa propre extension à UML, COMO-UML, pour décrire **l'activité** sous un principe similaire (voir figure 2.8). TOUCHE propose un métamodèle **d'activité** indépendant **d'UML** (voir figure 2.10). Nous avons vu tous ces modèles en section 2.1.1.

FlowiXML de Garcia et al. [51, 42] proposent un métamodèle **d'organisation** et de flux de travaux pour UsiXML [70] que nous avons abordé en section 2.2.1. **L'approche** de FlowiXML est de modéliser en premier lieu les rôles et **l'organisation** pour ensuite décrire les flux de travaux associés. Chaque tâche identifiée dans les flux de travaux est alors modélisée avec ConcurTaskTrees [84] ce qui, à travers la chaîne de génération de CAMELEON [21, 113, 23], donnera lieu à des IHM pour chaque tâche. FlowiXML est avant tout un outil de modélisation et ne spécifie pas la manière dont les IHM interagissent entre elles après génération, **c'est-à-dire ce qu'il** advient des flux de travaux en dehors de leur modélisation.

Les travaux de Boukhebouze et al. [15, 14] se portent également sur UsiXML, mais du point de vue du standard BPEL avec UsiWSC (pour UsiXML Web Service Consumption). BPEL est un standard originellement prévu pour décrire la manière dont sont consommés des services web. UsiWSC est fait pour cela, avec une emphase sur la génération et la composition des IHM relatives aux services décrits. Ici, les flux de travaux BPEL se substituent aux modèles de tâches car ils englobent la description des services web qui permet **d'en** dériver des IHM. La particularité **d'UsiWSC** est de garder les flux de travaux à **l'exécution** des IHM pour en interpréter l'état et maintenir la cohérence entre les IHM des services actifs et la méthode de consommation décrite par les flux de travaux (voir figure 2.30).

2.3.2.2 Synthèse

En guise de synthèse, nous prenons en point de référence les standards de modélisation qui proposent la meilleure expressivité et représentation tout en étant largement admis **d'un** point de vue industriel. En les comparant à **d'autres** approches, nous tenons compte de leur paradigme de représentation, de leur compatibilité avec les standards ainsi que de leur méthode de mise en **œuvre**, ce que nous résumons en table 2.5.

Bien que les approches dérivées de CTT [84] ne se présentent pas comme des alternatives à la modélisation de flux de travaux, nous pouvons souligner que leur approche est similaire. En effet, elle consiste à modéliser **l'interdépendance** entre des tâches individuelles déterminées par rôle. Ces formalismes sont toutefois limités en termes **d'expressivité** de flux de travaux, ce qui est notamment dû aux limites de CTT concernant les transitions entre tâches. CTML [126, 127, 39] fait toutefois cas particulier en proposant de gérer les interdépendances avec un langage bien plus expressif et intégré à **l'IDM**.

Concernant leur mise en **œuvre**, la règle générale semble être de considérer une activité **d'un** flux de travaux comme étant réalisé à travers une tâche. Dans le cas des approches

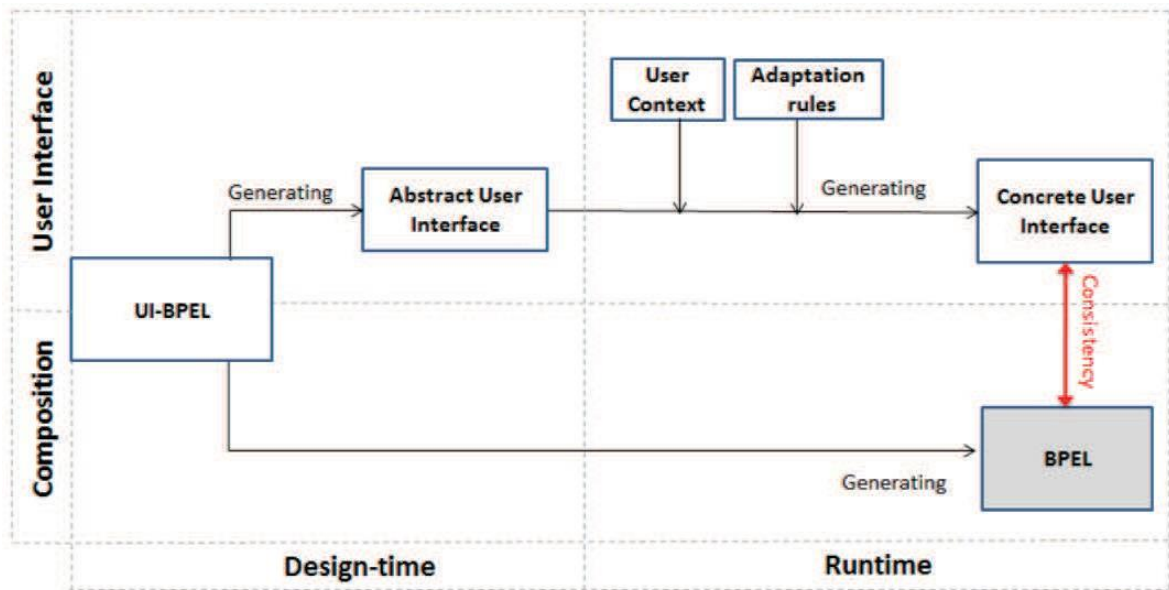


Figure 2.30 – L'architecture d'UsiWSC de Boukhebouze et al. [15, 14].

IDM que sont UsiWSC [15, 14] et CTML, les flux de travaux sont par ailleurs gardés à l'exécution et interprétés afin d'être mis en application. L'état du flux de travaux détermine alors quelles tâches sont accessibles et leur réalisation agit sur cet état. Dans ces recherches, les flux de travaux sont alors une notion transverse **puisque** ils servent autant à décrire l'**activité** à haut niveau, qu'à orchestrer l'**IHM** à bas niveau.

Table 2.5 – Récapitulatif des métamodèles de flux de travaux étudiés. Légende : **9** très adéquat à notre problématique, **Q** adéquat avec des réserves, **Q** présente des limitations significatives.

Métamodèles	Concepts		
	Approches de flux de travaux	Expressivité	Mise en œuvre
CCTT [78] / COMM [66, 65]	Q Arbre de synchronisation	Q Limitée	Q Un modèle de tâches par activité du flux
CTML [126, 127, 39]	Q Langage de contraintes (OCL)	9 Comparable aux standards	9 Un modèle de tâches par activité du flux et interprétation à l'exécution
CIAM [76]	Q États-transition UML	9 Comparable aux standards	Q Un modèle de tâches par activité du flux
AMENITIES / COMO-UML [43, 45]	Q États-transition UML	9 Comparable aux standards	Q Un modèle de tâches par activité du flux
TOUCHE [88, 87]	Q Métamodèle d'organisation (simili États-transition UML)	9 Comparable aux standards	Q Un modèle de tâches par activité du flux
FlowiXML [51, 42]	9 Dérivé des standards BPMN et BPEL	9 Standards	Q Un modèle de tâches par activité du flux
UsiWSC [15, 14]	9 Standard BPEL	9 Standards	Q Un service web par activité du flux et interprétation à l'exécution

Résumé de section

La conscience de groupe est situationnelle. C'est-à-dire **qu'**elle répond à des situations particulières en proposant des informations appropriées. De plus, ces informations consistent en une vue puisant dans les données de **l'**ensemble du contexte de **l'**activité collective (i.e. les modèles de haut niveau).

Proposer de la conscience de groupe est déterminé par les points suivants :

Identifier des situations ;

Définir le besoins en information dans ces situations ;

Répondre à ces besoins en fournissant ces informations dans **l'**IHM.

Les flux de travaux se traduisent par une IHM par activité, soit la définition **d'un** arbre de tâches. Leur description est identique à la définition **d'un** modèle **d'**activité (i.e. **l'**interdépendance entre les tâches). Il **s'agit** donc **d'une** notion transverse qui se modélise à haut niveau et **s'applique** à bas niveau.

2.4 Conclusion de chapitre

Dans ce second chapitre, nous avons établi un état de **l'art** sur les différentes facettes qui composent notre problématique. Pour rappel, nous nous positionnons sur la conception et la génération **d'IHM** adaptées au contexte de **l'activité** collective. À cet effet, nous avons considéré cette problématique sous **l'angle** des MBUI qui représentent une solution potentielle pertinente.

Pour évaluer cette solution, nous avons scindé notre état de **l'art** suivant les axes de recherche établis en chapitre 1 (voir section 1.4). Dans un premier temps, nous avons considéré les approches généralistes couvrant le plus large spectre de la problématique. À ce titre, nous avons passé en revue un ensemble de méthodologies de conception de collecticiels basées sur les MBUI (section 2.1.1), notamment CIAM [76], AMENITIES [44, 45] et TOUCHE [88, 87]. En guise de synthèse, nous avons soulevé le fait que ces méthodologies arborent des démarches par étapes similaires et définies comme suit :

1. La modélisation des rôles et de leur hiérarchie (i.e. le sociogramme de **l'organisation**) ;
2. La modélisation des tâches et de leur attribution aux rôles/acteurs ;
3. La modélisation des interdépendances entre les tâches et **l'accès** aux ressources (i.e. **l'activité** dans son ensemble) ;
4. La modélisation des tâches utilisateurs.

En dehors de TOUCHE, ces méthodologies **n'ont** pas pour vocation **d'aboutir** à une génération **d'IHM** par IDM. Nous avons donc porté notre attention sur des travaux allant en ce sens en sortie de ces méthodologies, dont les travaux autour de TOUCHE mais aussi les travaux sur le langage WSL [121] pour la génération **d'IHM** adaptatives par widgets (section 2.1.2). Eu égard de cette revue, nous concluons sur les limites suivantes dans ces approches :

- La représentation conceptuelle de la collaboration **n'est** pas toujours propice à **l'IDM** alors **qu'il s'agit d'une** notion importante du contexte, et donc essentielle dans **l'adaptation** des IHM ;
- Le formalisme de description des tâches CTT ne suffit pas à décrire convenablement des tâches collaboratives ;
- La conscience de groupe **n'est** pas traitée à tous les niveaux **d'abstraction** et ne tire pas partie des informations modélisées en amont de la génération ;
- Les flux de travaux, bien que modélisés en amont, ne sont pas mis en application de manière opérationnelle.

Nous avons alors orienté notre revue de la littérature sur ces derniers points en deux temps. Tout **d'abord** en nous concentrant sur les aspects haut niveau de la problématique, **c'est-à-dire** la modélisation théorique de la collaboration (section 2.2.1) et la modélisation des tâches collaboratives (section 2.2.2). En termes de métamodèle de la collaboration, UsiXML **s'approche** le plus de notre problématique mais **n'est** pas complet sur tous les concepts tandis que AMENITIES dispose **d'une** bonne modélisation mais **n'est** pas orienté vers **l'IDM**. Une association des deux approches nous semble propice. Du côté des métamodèles de tâches collaboratives, les approches dérivées de CTT [84] que sont CCTT [78], COMM [66, 65] et CTML [126, 127, 39] nous semblent en accord avec le modèle 3C [34, 40]

et la matrice de Johansen [63], bien que CTML nous parait être la seule approche particulièrement pensée pour **l'IDM**.

En second temps, nous nous sommes intéressés au support de la coordination à bas niveau à travers une revue portée sur la conscience de groupe (section 2.3.1) puis sur les flux de travaux (section 2.3.2). Dans le cas de la conscience de groupe, nous avons souligné que celle-ci dépend de situations qui ne sont pas toujours prévisibles, particulièrement si **l'on** considère **l'adaptation** des IHM. Ainsi, pour supporter convenablement la conscience de groupe, il **s'agit d'exploiter l'ensemble** des modèles de haut niveau qui permettent **d'identifier** ces situations et de servir de sources **d'informations**. À travers la littérature, nous avons vu la mise en **œuvre** de la conscience de groupe consiste en les étapes suivantes :

1. Identifier des situations ;
2. Définir les besoins en information de conscience de groupe dans ces situations ;
3. Répondre à ces besoins en fournissant ces informations dans **l'IHM**.

Nous avons vu que TOUCHE couvre notamment ces deux derniers points de manière statique (en spécialisant les AUI indépendamment de situations) alors que les travaux de Figueroa-Martinez et al. [37] proposent une extension pertinente à UsiXML [70] pour la modélisation de situations mais limitée à des contraintes sur le modèle du domaine. Il ne semble ainsi pas y avoir **d'approche** couvrant tous ces aspects en adéquation avec notre problématique. Enfin, dans le cas des flux de travaux, nous avons noté que ceux-ci véhiculent une sémantique identique à la façon dont est modélisée la synchronisation des tâches dans les métamodèles de tâches collaboratives. Il **s'agit ainsi d'une** notion qui nous parait transverse entre les concepts de haut et bas niveau et **qu'il** convient de concilier. Vis-à-vis de leur application, un consensus semble être de considérer la modélisation **d'un** arbre de tâches par activité de flux de travaux, le flux pilotant alors **l'accès** et la synchronisation des IHM générées.

Synthèse de la littérature

Dans l'optique de proposer une approche de conception et de génération d'IHM adaptatives pourvues d'un support à la coordination, il nous semble judicieux de construire une réponse en considérant les éléments de contribution suivants qui synthétisent ce chapitre et qui viennent en réponse aux hypothèses formulées (voir page 50) :

1. **Comblent la modélisation des concepts de haut niveau** : c'est-à-dire compléter un ensemble de métamodèles décrivant toutes les facettes du contexte de l'activité collective telles que nous les avons défini. Ici, le métamodèle le plus pertinent nous semble être en provenance d'AMENITIES [43, 45] alors que celui le plus propice à l'IDM est celui d'UsiXML [70]. Un rapprochement nous paraît opportun. Vis-à-vis du cas de la modélisation de tâches collaboratives, le métamodèle de CTML [126, 127, 39] apporte la meilleure réponse IDM pour compléter UsiXML. Ce point de contribution est lié à l'hypothèse H1 ;
2. **Établir un support de la coordination** : nous savons ici que la conscience de groupe s'opère en trois étapes (i.e. identifier des situations, y associer des besoins et répondre à ces besoins) dont TOUCHE [88, 87] et les travaux de Figueroa-Martinez et al. [37] apportent des solutions à divers niveaux d'abstraction, mais aucune approche d'ensemble ne semble se présenter. Il convient alors d'établir des métamodèles et mécanismes de génération mettant en œuvre ces trois étapes. Ce point de contribution est lié à l'hypothèse H2 ;
3. **Élargir le champ des méthodologies de conception à la génération** : les méthodologies étudiées représentent une bonne base via des étapes de modélisation bien segmentées et pertinentes. Il serait toutefois intéressant d'aller plus loin, à l'image de TOUCHE [88, 87] qui propose de modéliser les IHM. Cependant, aucune approche ne semble aller jusqu'à la génération effective des IHM, tout comme le manque de considération de l'adaptation. Pour compléter cette démarche, des travaux tels que ceux autour d'UsiXML [70] dont WSL [121] et les travaux de Figueroa-Martinez et al. [37] nous paraissent intéressants à considérer. Ce point de contribution est lié à l'hypothèse H3.

De manière générale, les travaux sus-nommés les plus pertinents sont bâtis autour de TOUCHE [88, 87] pour ce qui est des aspects méthodologiques et sur UsiXML [70] pour ce qui est de la modélisation des concepts et du support de la génération d'IHM. Ces deux approches représentent un socle intéressant à étendre vis-à-vis des autres travaux passés en revue qui sont pertinents dans leurs domaines respectifs.

Chapitre 3

Un cadre de conception d'IHM adaptatives pour la coordination

Sommaire

3.1	Modélisation du contexte de l'activité collective	113
3.1.1	Contribution au métamodèle de la collaboration.....	114
3.1.2	Choix d'un métamodèle de flux de travaux.....	118
3.1.3	Métamodèles de tâches et du domaine.....	121
3.1.4	Adaptation au contexte de l' activité collective.....	123
3.2	Prise en compte de la conscience de groupe par widgets adaptatifs.....	130
3.2.1	Modélisation des besoins en conscience de groupe.....	130
3.2.2	Réponse sous forme de widgets adaptatifs.....	133
3.2.3	Composition d'interfaces et dépôt d'arbres de tâches de widgets	136
3.3	Prise en compte des flux de travaux.....	142
3.3.1	Application des flux de travaux.....	142
3.3.2	Génération d'un tableau de bord	143
3.4	Intégration dans une méthodologie de conception.....	147
3.4.1	Étapes de modélisation.....	147
3.4.2	Itération pour l' identification des besoins en conscience de groupe	149
3.5	Conclusion de chapitre.....	154

Ce troisième chapitre constitue la mise en place de notre proposition. Eu égard aux limitations de l'état de **l'art** que nous avons identifié, notre démarche est de proposer un cadre de modélisation et de génération d'IHM adaptatives pourvues **d'un** support à la coordination. À cet effet, **l'orientation** de notre proposition **s'articule** autour des axes suivants :

Une contribution à la modélisation du contexte de l'activité collective : c'est-à-dire contribuer aux métamodèles de haut niveau qui constituent le contexte et plus particulièrement vis-à-vis des métamodèles de la **collaboration**, **de l'activité** ainsi que des tâches collaboratives;

Une contribution à la modélisation et au support de la coordination : c'est-à-dire mettre en **œuvre** les concepts de bas niveau que sont la conscience de groupe et les flux de travaux, notamment en définissant des métamodèles **d'abstraction** qui permettront de générer des IHM intégrant ces concepts ;

Une contribution méthodologique : afin **d'intégrer** les éléments précédents dans une démarche de conception IDM uniformisée.

Dans la construction de cette proposition, il nous semble pertinent de considérer TOUCHE [88, 87, 89] comme base méthodologique ainsi que UsiXML [70] comme base de modélisation et de cadre de génération **d'IHM**. En effet, même si ce couple ne répond pas complètement à notre problématique, il nous semble le plus adéquat dans une démarche **d'IDM**, **d'autant** que de potentielles solutions se trouvent dans des travaux qui gravitent autour de TOUCHE [88, 87, 89] et **d'UsiXML** [70, 37].

En sus de cette orientation, notre volonté est de proposer un cadre de modélisation pensé pour **l'IDM** avec un minimum de disparités entre la phase de conception (i.e. qui nécessite des modèles conceptuellement adaptés aux méthodologies) et la phase de production des IHM (i.e. qui nécessite des modèles productifs). Nos choix seront donc dirigés avec **l'intention d'offrir** des métamodèles autant conceptuels que productifs et adaptés à une démarche **d'IDM**.

Enfin, nous nous positionnons sur la problématique de la conception, c'est-à-dire que notre proposition consiste avant tout à présenter un cadre de conception **d'IHM** adaptatives pour la coordination. En ce sens, nous ne prenons pas position vis-à-vis des problématiques liées à **l'exécution** de telles IHM, comme la qualité de **l'adaptation**, **l'utilisabilité** des IHM générées ou **l'évaluation** de **l'apport** de la conscience de groupe qui sont des domaines de recherche en soi.

Structuration du chapitre et de la proposition

Nous présentons dans ce chapitre un cadre de conception IDM pour la génération **d'IHM** adaptatives bénéficiant **d'un** support à la coordination. Ce chapitre **s'articule** autour des éléments de cette proposition qui font **l'objet d'une** contribution. Aussi, les éléments propres à CAMELEON [21, 113, 23] ou UsiXML [70] qui ne font pas **l'objet** de contributions ne seront pas discutés ici (tels que les heuristiques de transformation).

Le cadre que nous proposons, dont la structuration est illustrée en figure 3.1, est établi sur le principe de CAMELEON et se décompose comme suit :

Des métamodèles d'abstraction de l'activité collective (section 3.1) représentant les modèles de haut niveau du cadre CAMELEON, c'est-à-dire ceux qui définissent **l'ensemble** du contexte (i.e. modèles de tâches, du domaine, **d'organisations** et **d'acteurs**). Il **s'agit** des modèles qui servent de point **d'entrée** au processus de génération et qui conduisent aux transformations vers modèles **d'AUI** puis CUI ;

La prise en compte de la coordination qui passe par des métamodèles **d'abstraction** qui **s'insèrent** dans le processus de génération afin de supporter les éléments suivants :
– **Le support de la conscience de groupe** (section 3.2), où nous proposons **d'adopter** une architecture par composition **d'arbres** de tâches de widgets dédiés à la conscience de groupe. Cette composition est déterminée par des modèles qui

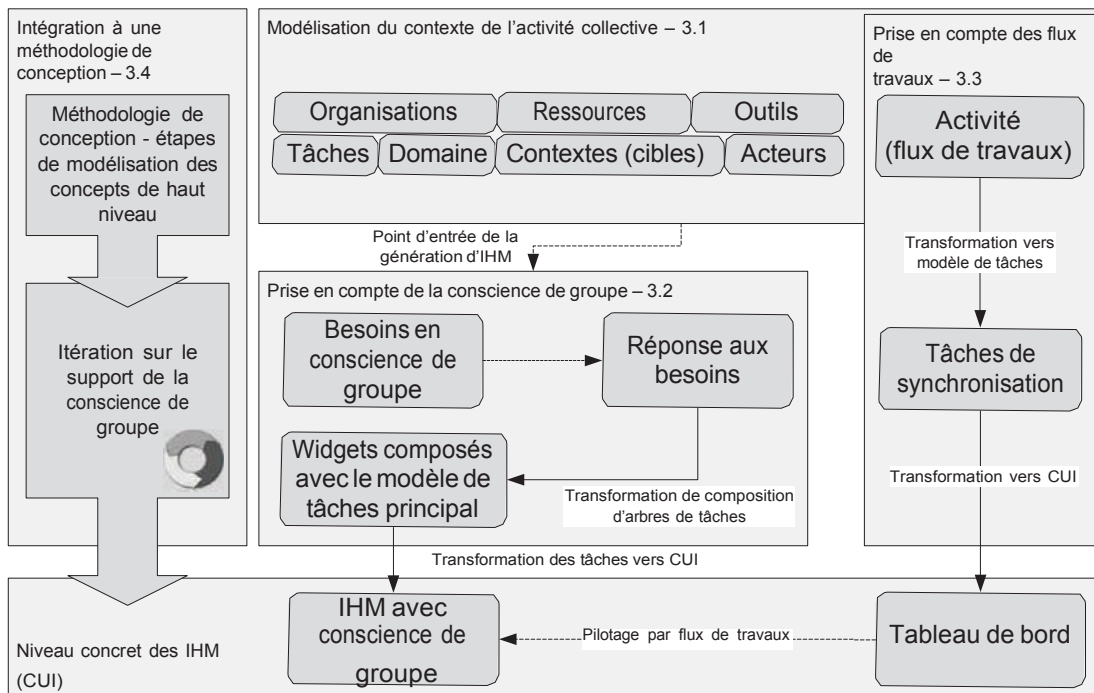


Figure 3.1 – Structuration du cadre de conception et de génération proposé.

suivent les étapes de la conscience de groupe que nous avons mis en avant (i.e. l'identification de besoins, le calcul d'une réponse adaptée et la présentation des informations résultantes) ;

- **Le support des flux de travaux** (section 3.3), que nous voyons comme un mécanisme de pilotage des IHM générées individuellement. Pour superviser ce pilotage et gérer l'accès aux IHM du système, nous proposons la génération d'une IHM de tableau de bord.

Une **méthodologie itérative de conception** (section 3.4) qui détermine la manière dont sont construits les modèles par étapes sur la base de la synthèse des méthodologies que nous avons passé en revue. Nous y ajoutons un jeu d'étapes supplémentaires propres à la prise en compte de nos propositions vis-à-vis de la conscience de groupe. Ces étapes dédiées se présentent sous la forme d'un cycle itératif impliquant les utilisateurs afin d'identifier au mieux les situations et besoins liés à la conscience de groupe.

3.1 Modélisation du contexte de l'activité collective

Dans cette section, nous présentons notre positionnement vis-à-vis de l'état de l'art et les contributions que nous proposons pour répondre à notre problématique en matière de modélisation du contexte de l'activité collective. Pour rappel, nous considérons le contexte de l'activité collective comme étant composé des facettes suivantes : l'activité (i.e. l'inter-

dépendance entre les tâches), les acteurs, les organisations, les ressources, les outils (i.e. les IHM), le domaine (i.e. les concepts manipulés dans les IHM), les tâches et les cibles (i.e. les triplets **d'**environnement utilisateur, physique & matériel).

Les enjeux de ces modèles sont importants car ils servent de source au processus de génération **d'IHM**, mais aussi de données à ces IHM, notamment en termes de conscience de groupe (i.e. identification de situations puis observations des valeurs). Aussi, notre proposition en termes de modélisation est orientée suivant ces deux axes :

L'applicabilité des métamodèles afin de couvrir au mieux le spectre des concepts compris dans le contexte de **l'activité collective**. Ceci entre notamment en compte dans la pertinence des modèles dont dépendra le champ des possibles en termes de génération **d'IHM**. La représentation de ces modèles est également à considérer afin de faciliter la communication entre les intervenants du processus de conception ;

Être proche de la pratique de l'IDM et ainsi produire des métamodèles **s'intégrant** convenablement dans des **chaînes** de transformations. Cet aspect permet de simplifier les transformations et donc **d'augmenter l'acceptation** du paradigme de **l'IDM** vis-à-vis **d'une** ingénierie logicielle classique.

En somme, il **s'agit** de modéliser toutes les facettes du contexte de **l'activité collective** telles que requises par notre problématique. Le but de la modélisation du contexte est **d'inscrire** ces éléments dans le pilotage des transformations qui conduiront **à l'IHM finale**, et donc de servir de facteurs **d'adaptation**.

La figure 3.2 reprend le plan de cette section par rapport aux points sujets à contributions. Sur base **d'UsiXML** [70], nous proposons dans cette section une contribution au métamodèle de la collaboration (section 3.1.1), puis aux éléments plus spécifiques que sont les flux de travaux (section 3.1.2) et le couple de métamodèles de tâches et du domaine (section 3.1.3). Enfin, nous verrons comment ces contributions **s'insèrent** dans une **chaîne** de transformations de modèles afin de proposer des IHM adaptées (section 3.1.4). Vis-à-vis **d'UsiXML**, nous considérons le métamodèle de cibles (i.e. le triplet de contraintes sur les environnements physique, matériel et utilisateur), ainsi que celui de CUI comme adéquats pour notre problématique. Nous **n'aborderons** par conséquent pas ces éléments.

3.1.1 Contribution au métamodèle de la collaboration

En section 1.2.3, nous avons tiré une synthèse des concepts remarquables de la collaboration. Nous avons ensuite étudié le support de ces concepts dans les efforts de modélisation présents dans la littérature. Nous attendons **d'un** métamodèle de la collaboration les éléments suivants :

Les activités qui représentent les objectifs et la décomposition du travail en tâches ainsi que leur ordonnancement pour **l'obtention d'un** résultat. Nous avons vu que cet aspect se schématise par flux de travaux et est traité dans une section dédiée en 3.1.2 ;

Les acteurs qui sont les individus qui se coordonnent ou coopèrent pour la réalisation **d'une** tâche. Les acteurs doivent être identifiables individuellement (e.g. nom, prénom, coordonnées, etc.), rattachés à des rôles qui déterminent leur accès aux tâches ;

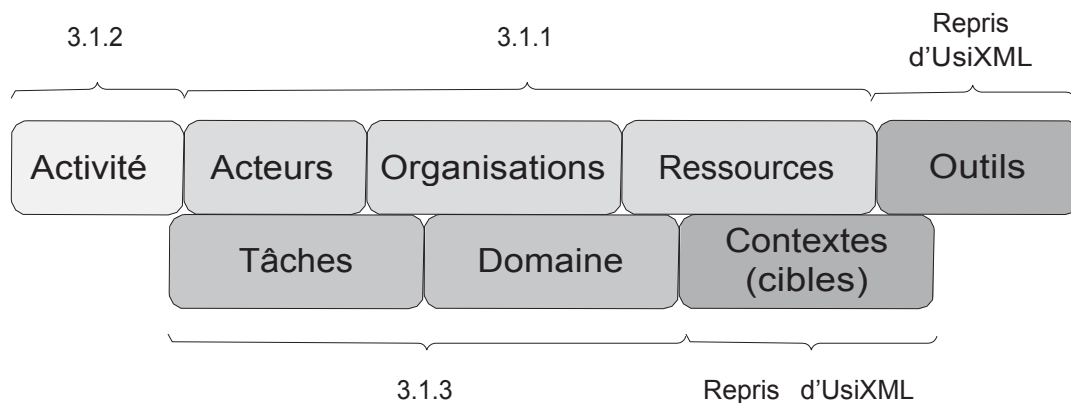


Figure 3.2 – Structuration des contributions de la proposition à la modélisation du contexte de l'activité collective.

Les organisations qui représentent les relations entre les acteurs, leur hiérarchie et les règles qui régissent la communication et la coordination des acteurs. Les organisations doivent être décomposables en groupes, sous-organisations et permettre de formaliser les relations inter-organisationnelles ;

Les ressources qui représentent les éléments nécessaires à la réalisation **d'une** tâche et qui peuvent être le résultat produit par **d'autres** tâches en amont. Les ressources peuvent être composées ou liées à **d'autres** ;

Les outils qui médiatisent l'exécution des tâches et/ou la coordination. Il **s'agit** généralement du système interactif en lui-même. Dans notre cas, il **s'agit** des IHM qui seront générées et sont ici représentées par les arbres de tâches et les modèles plus concrets **d'AUI** et **CUI**.

Les points importants que nous cherchons donc à apporter dans un métamodèle de la collaboration sont les acteurs, les organisations ainsi que les ressources. Dans notre état de **l'art** en section 2.2.1, nous avons souligné UsiXML [70] et AMENITIES [44, 45] comme étant les approches qui proposent la meilleure couverture de ces concepts. UsiXML propose une représentation des ressources, des outils et des activités adéquate tandis que AMENITIES contribue davantage à la représentation fine des organisations et la hiérarchie des acteurs.

En plus de ces éléments, nous avons vu en section 2.1.1 que les méthodologies de conception de collecticiels **s'orientent** autour des étapes suivantes :

1. La modélisation des rôles et de leur hiérarchie (i.e. le sociogramme de **l'organisation**) ;
2. La modélisation des tâches et de leur attribution aux rôles/acteurs ;
3. La modélisation des interdépendances entre les tâches et **l'accès** aux ressources (i.e. **l'activité** dans son ensemble) ;
4. La modélisation des tâches utilisateurs en détail.

Dès lors, il nous paraît judicieux d'également considérer ces étapes dans la définition d'un métamodèle de la collaboration afin de pouvoir les supporter. Ceci implique d'avoir des éléments de modélisation suffisamment indépendants pour être construits par étapes.

Le métamodèle de la collaboration que nous proposons (voir figure 3.3) tient compte des avantages de ces deux approches, la démarche étant de contribuer à la représentation d'UsiXML en y apportant la granularité d'AMENITIES concernant la modélisation d'organisations tout en anticipant les besoins des étapes de la méthodologie de conception. Il se compose des éléments suivants :

Les **OrganizationalUnits** (ou **organisations**) qui décrivent les entités organisationnelles (e.g. entreprise, département, unité, filiale, sous-division, groupe d'utilisateurs, etc.) et peuvent être liées par **OrganizationRelationships** afin de déterminer les liens existants entre ces entités (e.g. maison mère, partenaire, client, etc.). Les organisations sont composées d'utilisateurs ;

Les **Users** (ou **utilisateurs**) qui sont les individus identifiés individuellement, car nécessaires à la constitution d'organisations et de groupes, tout en gardant la notion d'**UserStereotype** d'utilisateurs d'UsiXML. Ces stéréotypes sont notamment utiles à des fins d'adaptation. Les utilisateurs peuvent être associés à des rôles et des tâches ;

Les **Roles** (ou **rôles**) qui déterminent les droits des utilisateurs mais également leur hiérarchie à travers les associations par **RoleRelationship** (e.g. dirige, supervise, etc.). La composition des éléments d'organisations, utilisateurs et rôles permet de définir le sociogramme de l'activité collective tel qu'exigé par les méthodologies de conception ;

Les **Tasks** (ou **tâches**) qui sont ici identifiées uniquement par leur nom, importance et complexité. Une tâche peut être liée à un rôle et un acteur, ainsi qu'à des ressources qui pourraient être nécessaires à la réalisation de la tâche. Ces associations permettent de convenir à l'étape d'attribution des rôles et tâches dans une méthodologie de conception ;

Les **Activities** (ou **activités**) qui décrivent la décomposition du travail sous forme de flux de travaux avec les tâches. Notons que la portion du métamodèle concernant l'activité (i.e. les flux de travaux associés aux tâches) est en partie masquée, seules les associations directes étant visibles (i.e. entités **Workflow** et **Activity**). La modélisation des flux de travaux fait l'objet de la section suivante. Plusieurs tâches peuvent être associées à une activité (et vice versa) afin de permettre la réutilisation de tâches existantes (e.g. il est parfois nécessaire de composer avec des applications métiers pré-existantes dans une organisation).

Enfin, l'étape de modélisation des tâches utilisateurs en détail est pris en charge par un métamodèle de tâches dédié avec lequel un lien est opéré dans le modèle de collaboration (attribut **TDAUri** de la classe **ComputableTask** qui représente une tâche informatizable). Les arbres de tâches représentent ici la notion d'outils médiatisant la réalisation des tâches. Nous verrons ce métamodèle en section 3.1.3.

3.1. Modélisation du contexte de l'activité collective

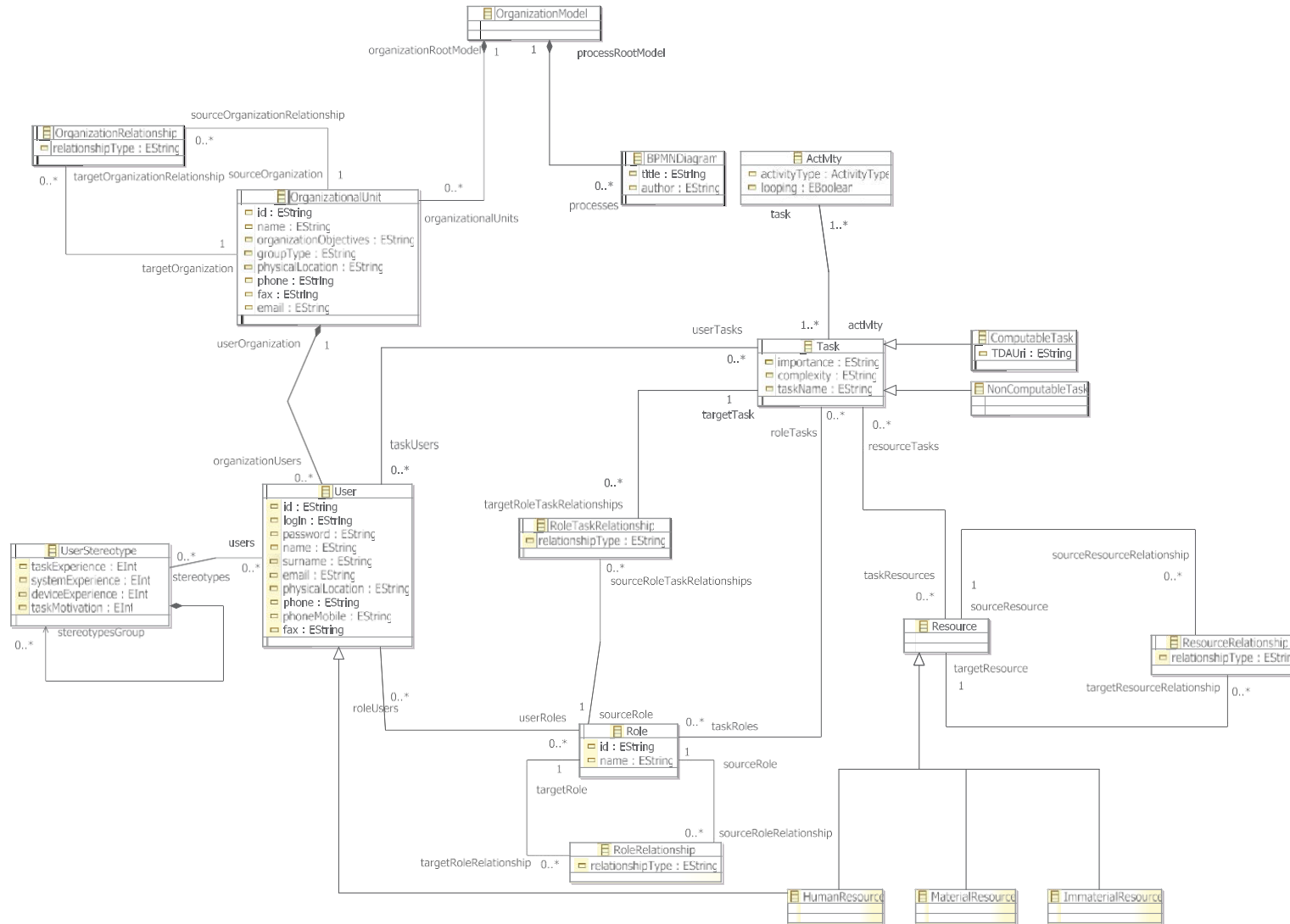


Figure 3.3 – Métamodèle de la collaboration. Représenté sous la forme d'un diagramme de classes UML.

3.1.2 Choix d'un métamodèle de flux de travaux

Nous avons vu que la description de l'**activité** (i.e. la définition des tâches et leur interdépendance temporelle) se fait par flux de travaux. Dans l'état de l'**art** (voir section 2.3.2), nous avons considéré les standards industriels que sont BPMN³⁴ et BPEL³⁵, ainsi que des métamodèles issus de la recherche en IDM comme UsiXML [70] ou des modèles cadrant des méthodologies de conception comme ceux d'**AMENITIES** [44, 45]. Pour rappel (voir section 1.2.4.2), la modélisation de flux de travaux dépend des éléments suivants :

- Les **pools** (ou piscines) qui représentent une activité de manière globale et qui sont rattachées à une unité organisationnelle. Elles contiennent des **swimlanes** ;
- Les **swimlanes** (ou couloirs) qui représentent la séquence de tâches propre à un acteur, ou type d'acteur (généralement par rôle). Elles contiennent des **tasks**, **events** et **gateways** reliés entre eux par **flows** afin de déterminer la séquence ;
- Les **tasks** (ou tâches) qui représentent une tâche atomique au sein de l'**activité** et associée à une **swimlane**. Il s'agit des tâches utilisateurs qui peuvent être réalisées à travers (ou non) une ou plusieurs IHM ;
- Les **flows** (ou flux) qui relient **tasks**, **events** et **gateways** entre eux de façon unidirectionnelle dans le temps ;
- Les **events** (événements) qui signalent les conditions de début des tâches ou celles activées par la fin d'une autre. Il peut s'agir d'envoi de messages, de conditions temporelles (e.g. date de début), d'événements sans pré/post-conditions ou de règles plus complexes ;
- Les **gateways** (portes) qui contrôlent l'enchaînement logique des tâches (e.g. tâches parallèles, exclusives, etc.).

Vis-à-vis de ces points, les travaux qui nous paraissent les plus proches de ces définitions sont UsiXML [70] (via son extension FlowiXML [51, 42]) et AMENITIES [44, 45]. En effet, FlowiXML est dérivé de BPMN et BPEL qui rend son métamodèle très adéquat, sans pour autant prétendre à une compatibilité avec ces standards. De l'autre côté, AMENITIES (ainsi que ses homologues CIAM [76] et TOUCHE [88, 87]) emploie un métamodèle dérivant des diagrammes d'états-transitions d'**UML**. L'inconvénient de cette approche est de reposer sur des termes et symboles différents des standards de flux de travaux, mais permet tout même de véhiculer la même sémantique. Nous avons aussi noté dans l'état de l'**art** que la plupart des approches de modélisation de tâches arborent un formalisme qui permet de les considérer comme des flux de travaux (sans pour autant employer les mêmes dénominations ni la même granularité).

Cependant, nous avons vu que dans la pratique, les approches qui mettent en œuvre les flux de travaux sont généralement basées sur des standards industriels telles que UsiWSC [15, 14] qui emploie BPEL, ou FlowiXML qui est dérivé de BPMN.

Nos contraintes étant de reposer sur des métamodèles proches d'une réalité pratique et technique, nous avons choisi de considérer un métamodèle de flux de travaux issu d'un

34. <http://www.bpmn.org/>

35. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>

standard industriel, soit BPMN. En ce sens, nous suivons l'approche de FlowiXML, mais avec la prétention d'être compatible avec BPMN.

Ce choix se justifie par le fait que ce standard est largement admis et documenté, ce qui se traduit par un formalisme plus souvent connu des intervenants. De plus, il est supporté par la plupart des outils industriels dédiés aux flux de travaux. **D'un** point de vue technique, certains standards sont représentés dans des outils ou environnements de développement. **C'est** le cas de BPMN qui est intégré à EMF par la présence **d'un** métamodèle Ecore (i.e. le langage de modélisation **d'EMF**) et **d'un** éditeur graphique. Il **s'agit** donc **d'un** métamodèle propice à l'**IDM** de par son intégration à l'**architecture Model-2-Model d'Eclipse**.

Ces raisons nous conduisent à proposer l'**usage** de BPMN tel quel, dont le métamodèle Ecore est présenté en figure 3.4.

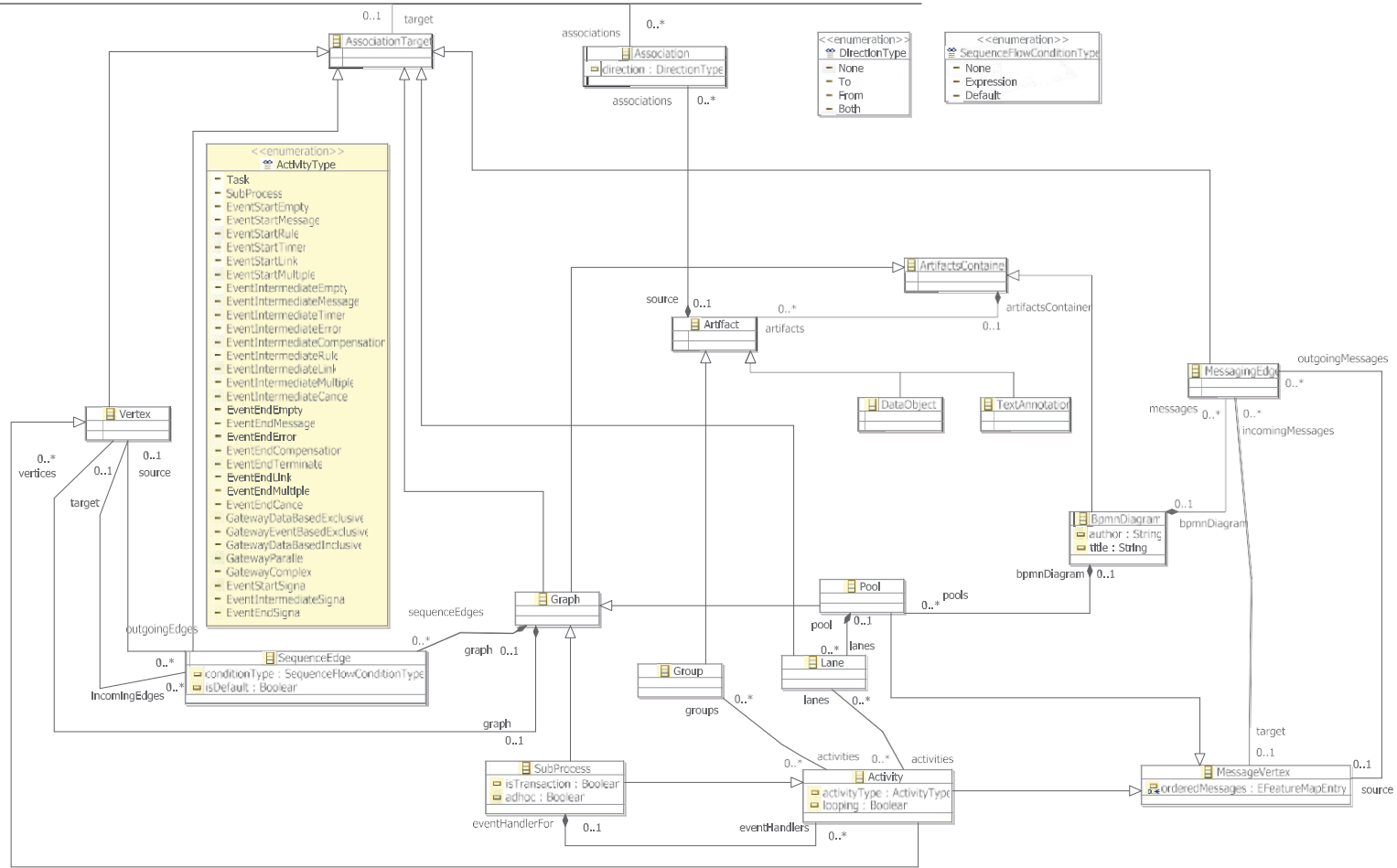


Figure 3.4 – Métamodèle de BPMN 2.0 dans son implémentation Ecore. Des éléments propres à la représentation graphique ont été occultés pour plus de lisibilité. Représenté sous la forme d'un diagramme de classes UML.

3.1.3 Métamodèles de tâches et du domaine

Le couple de métamodèles du domaine et de tâches représente le socle de base de la génération **d'IHM**. Il s'**agit** de la plus haute abstraction dans le cadre CAMELEON [21, 113, 23]. Leur rôle est de décrire **l'IHM** à un niveau élémentaire, c'est-à-dire la description des interactions via un arbre de tâches (e.g. tâche de sélection, consultation, commande, entrée, etc.) manipulant des concepts du domaine (e.g. nom, date, numéro de commande, etc.).

3.1.3.1 Métamodèle du domaine

Le métamodèle du domaine est très certainement le plus trivial de tous. Il consiste à décrire les concepts qui seront manipulés dans les arbres de tâches ainsi que les liens qui peuvent exister entre ces concepts. Ce métamodèle prend communément la forme **d'un** diagramme de classes UML, ou plus simplement un modèle de données relationnel (tel que dans les travaux sur CAMELEON [21, 113, 23] et UsiXML [70]). Le métamodèle du domaine doit ainsi permettre de définir des entités (ou classes), leurs propriétés et éventuelles associations, voire héritages **d'autres** entités.

Il est possible de reposer directement sur le métamodèle **d'UML** (**d'autant qu'il** dispose **d'une** instance Ecore au sein **d'EMF**), mais celui-ci englobe toute la spécification UML, c'est-à-dire **l'ensemble** des treize types de diagrammes supportés par le standard. Cette complexité le rend fastidieux à manipuler dans une **chaîne** de transformations. En ce sens, nous préférons opter pour un métamodèle du domaine minimal et taillé à nos besoins spécifiques en se contentant **d'un** schéma relationnel de données avec néanmoins des relations **d'héritage** et de composition entre entités.

Ce métamodèle, illustré en figure 3.5, est basé sur celui **d'UsiXML** à l'**exception** des instances du schéma de données. En effet, UsiXML permet, dans un même modèle du domaine, de décrire la structure des données et les données en elles-mêmes, suivant ainsi le principe de **l'IDM**. Ceci tend à complexifier le métamodèle et sa maintenance (deux niveaux **d'abstraction** siégeant au sein **d'un** même modèle). Par ailleurs, UsiXML a été pensé pour être modulaire et ses métamodèles sont ainsi indépendants et peuvent être rendu compatibles avec des métamodèles externes. Nous avons pris le parti de dissocier structure et données, pour ces raisons. Dans notre cas, les données pourront être prises en charge par tout autre format ou gestionnaire de données (e.g. base de données, fichiers XML, etc.), bien que ceux-ci peuvent être considérés comme des instances du modèle du domaine.

3.1.3.2 Métamodèles de tâches et d'AUI

Dans notre revue de la littérature, nous nous sommes intéressés aux métamodèles de tâches collaboratives (section 2.2.2). Nous avons conclu que CCTT [78] et COMM [66, 65] sont des représentations graphiques pertinentes mais **l'approche** la plus à même de **s'insérer** dans **l'IDM** est CTML [126, 127, 39] de par son usage du langage OCL³⁶ pour exprimer les contraintes de collaboration. De plus, nous considérons dans notre proposition que la

36. <http://www.omg.org/spec/OCL/>

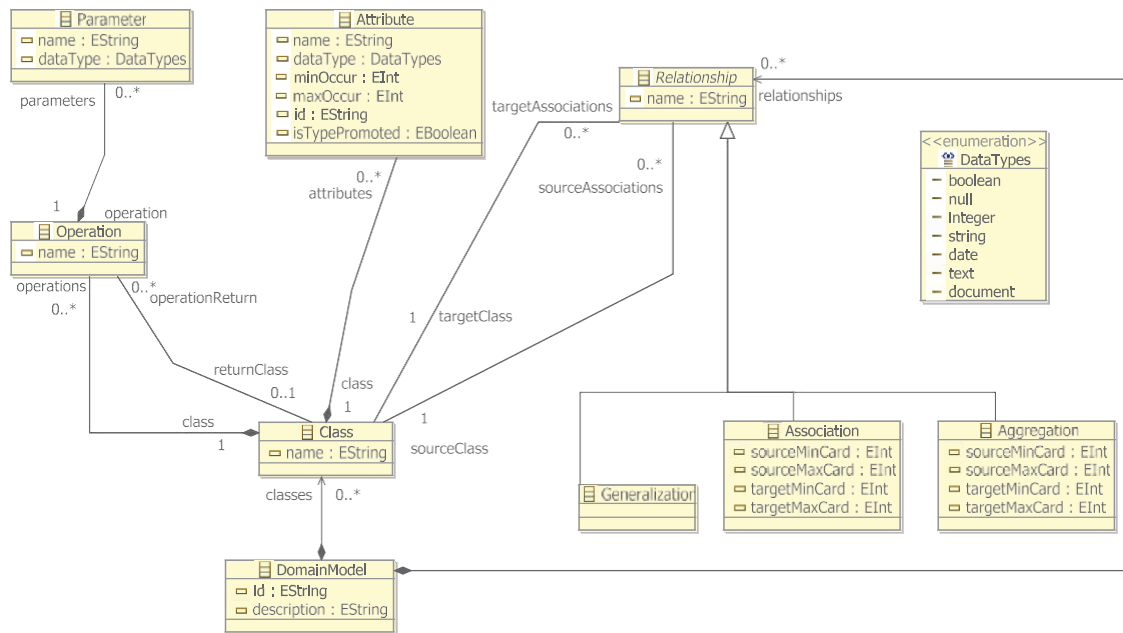


Figure 3.5 – Métamodèle du domaine. Représenté sous la forme d’un diagramme de classes UML.

synchronisation entre les tâches est opérée par flux de travaux et que par conséquent un arbre de synchronisation n’est pas nécessaire. Nous verrons également en section 3.3.1 que l’usage d’OCL peut s’insérer dans un pilotage des flux de travaux.

Nous proposons donc un métamodèle de tâches sur base de CTML, à deux différences notables près :

Des opérateurs temporels n-aires à la MAD [93] à l’inverse des opérateurs binaires de CTT [84]. Dans le métamodèle de CTT et de ses dérivés (dont CCTT, COMM et CTML), une tâche est décomposée en n sous-tâches puis les sous-tâches voisines reliées par un opérateur temporel LOTOS [117] binaire. Dans MAD, les opérateurs sont portés directement par les tâches parentes et affectent les n sous-tâches directes. D’un point de vue sémantique, les deux formalismes sont équivalents mais demandent d’organiser l’arbre différemment. La figure 3.6 illustre cette différence.

Spécification en AUI des arbres de tâches proposant ainsi d’intégrer le métamodèle d’*Abstract User Interface* à celui des tâches. Pour rappel, un arbre de tâches détermine si une tâche est interactive, exécutée par le système ou abstraite (i.e. un conteneur d’autres tâches) et une AUI reprend cette caractérisation en ajoutant le type de l’interaction (i.e. sortie/entrée d’informations, commande, sélection, navigation ou conteneur). Lors de la génération d’une IHM, le cadre de référence CAMELEON [21, 113, 23] indique une transformation de modèles de tâches vers AUI. Or, comme soulevé par Sottet [103], cette transformation ne peut être automatique car la sémantique d’un arbre de tâches n’est pas suffisante pour déterminer tous les types d’une AUI (e.g. une tâche interactive peut être une tâche de sélection comme elle peut être une tâche d’entrée ou de navigation) et donc d’opérer des transfor-

mations automatiques. Celles-ci étant alors semi-automatiques et la sémantique est apportée par les concepteurs. Nous suivons alors la démarche de Sottet [103] qui est **d'intégrer** ce typage des AUI (i.e. AUIType) dans le métamodèle de tâches puisque la connaissance sur ce typage est apportée par le concepteur. **L'intérêt est d'alléger** le processus de génération, au détriment **d'une** représentation graphique intermédiaire.

Ces deux particularités se justifient dans **l'interaction** de ces présents travaux de thèse avec un autre projet du Centre de Recherche Public Henri Tudor, à savoir le projet GENIUS [104] qui traite de la prise en compte de **l'utilisabilité** dans la génération **d'IHM**. Comme nous le verrons au chapitre suivant en section 4.1.1, ce projet a servi de cadre général à nos travaux et un effort de mutualisation des éléments relatifs au processus de génération **d'IHM** a été mené. Ainsi, les choix **d'employer** des arbres de tâches n-aires et **d'y** intégrer la spécification **d'AUI** sont hérités du projet GENIUS avec lequel cette thèse partage le socle technique que nous aborderons lors de la présentation du démonstrateur issu de ces travaux (section 4.1).

La figure 3.7 reprend notre proposition de métamodèle de tâches qui représente notre approche du compromis entre un modèle conceptuel et productif. Nous lui associons une représentation graphique illustrée en figure 3.8 que nous reprendrons dans les exemples qui suivront en chapitre 4. Ce métamodèle se constitue des éléments suivants :

Les **Tasks** (ou tâches) qui sont les **nœuds** individuels de **l'arbre** de tâches. Ces **nœuds** peuvent être abstraits, exécutés par le système, **par l'utilisateur** ou interactifs. Ils portent également le type de **l'AUI** auquel ils correspondent parmi entrée/sortie, commande, sélection, navigation ou conteneur abstrait. Les **nœuds** portent également les post/pré-conditions OCL déterminant les conditions **d'activation** et **d'enchaînement** entre les tâches ;

Les **Operators** (ou opérateurs) qui sont les opérateurs n-aires, basés sur LOTOS, déterminant les liens logiques et temporels entre les **nœuds** de **l'arbre** ;

Les **Associations** qui déterminent les concepts du domaine manipulés par chaque **nœud**.

Dans ce métamodèle, **l'accès** aux concepts du domaine dans les arbres de tâches se fait par notation pointée via **l'attribut manipulatedConceptId** de l'élément **Association** (e.g. «NomDeClasse.propriété1»). Nous considérons que, en dehors du domaine, un arbre de tâches doit pouvoir manipuler **l'ensemble** des concepts du contexte de **l'activité collective** (i.e. **l'organisation**, les acteurs, **l'activité**). Il est donc possible de pointer vers **n'importe** quel modèle (e.g. organisation, activité) et de les manipuler dans un arbre de tâches de la même manière que le domaine.

Enfin, notons la présence dans le métamodèle des entités **WidgetProperty** et **Default-Parameter** qui sont propres à notre approche de la conscience de groupe et détaillées en section 3.2. La propriété **d'importance** sur les concepts manipulés est elle traitée en section suivante dans le cadre de **l'adaptation**.

3.1.4 Adaptation au contexte de l'activité collective

Nous venons de voir dans les sections précédentes les détails de modélisation du contexte vis-à-vis de notre proposition. Une fois ces éléments modélisés, ceux-ci servent de

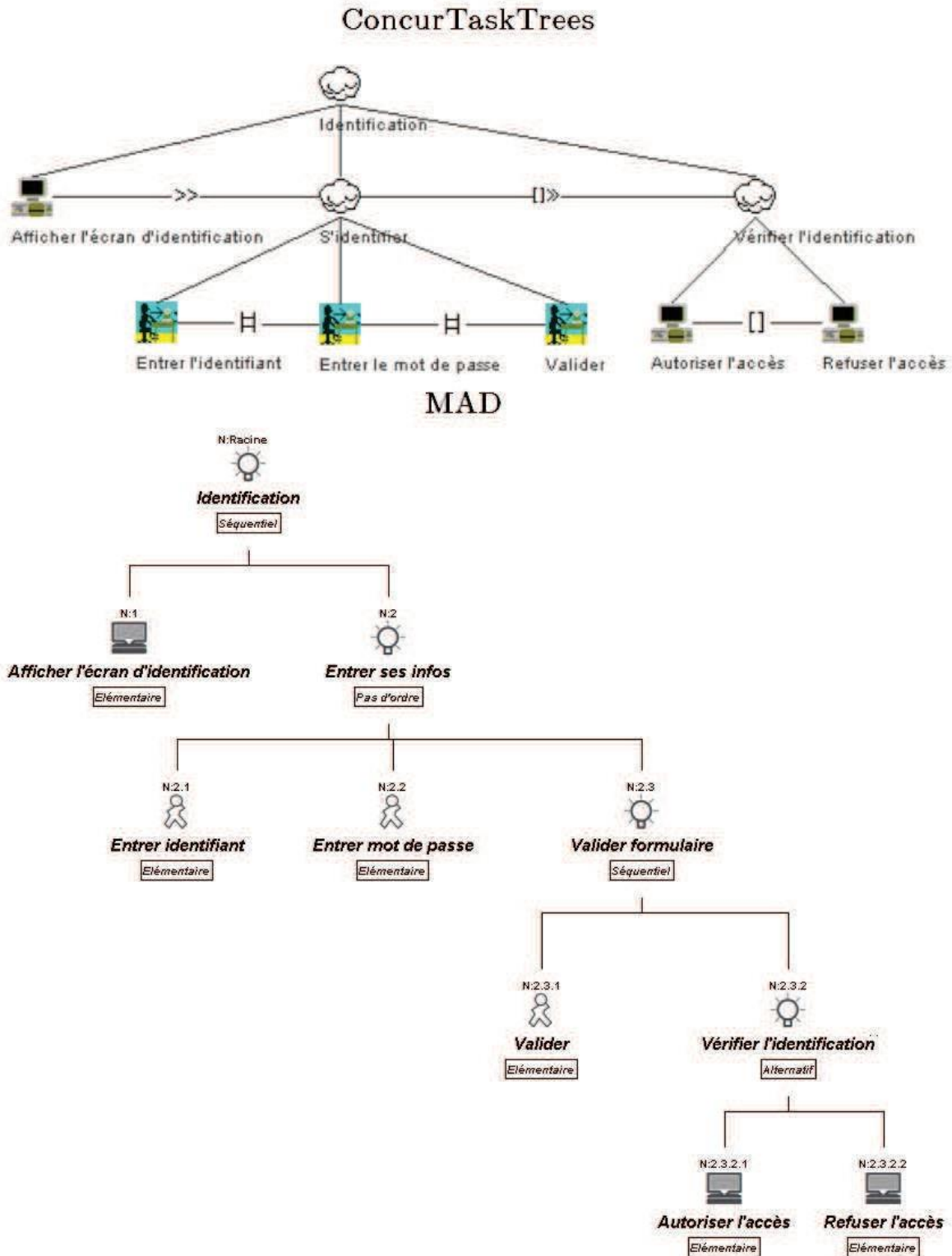


Figure 3.6 – La même tâche d'identification représentée par un arbre ConcurTask-Trees [84] et un arbre MAD [93].

3.1. Modélisation du contexte de l'activité collective

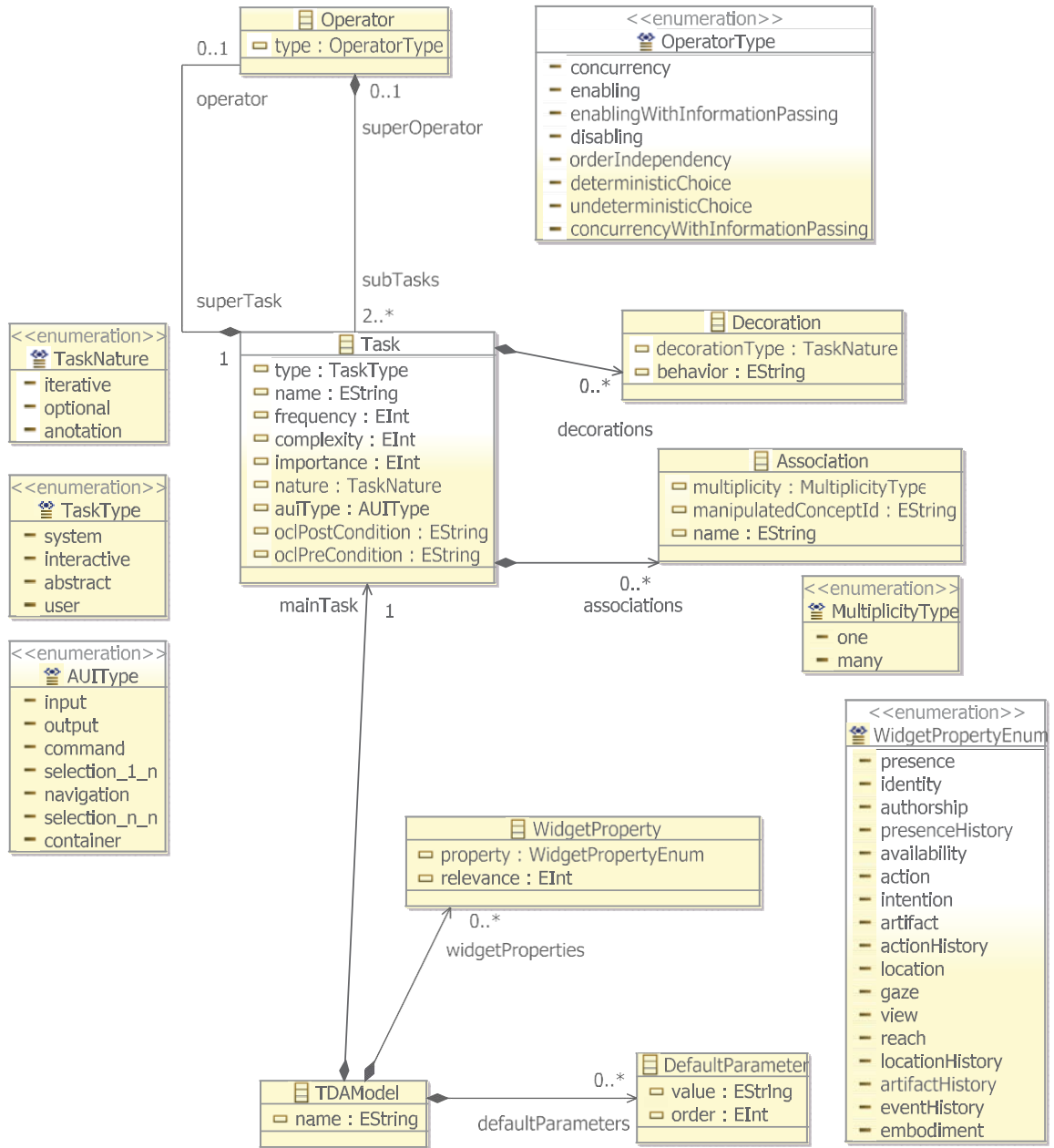


Figure 3.7 – Métamodèle de tâches. Représenté sous la forme d'un diagramme de classes UML.

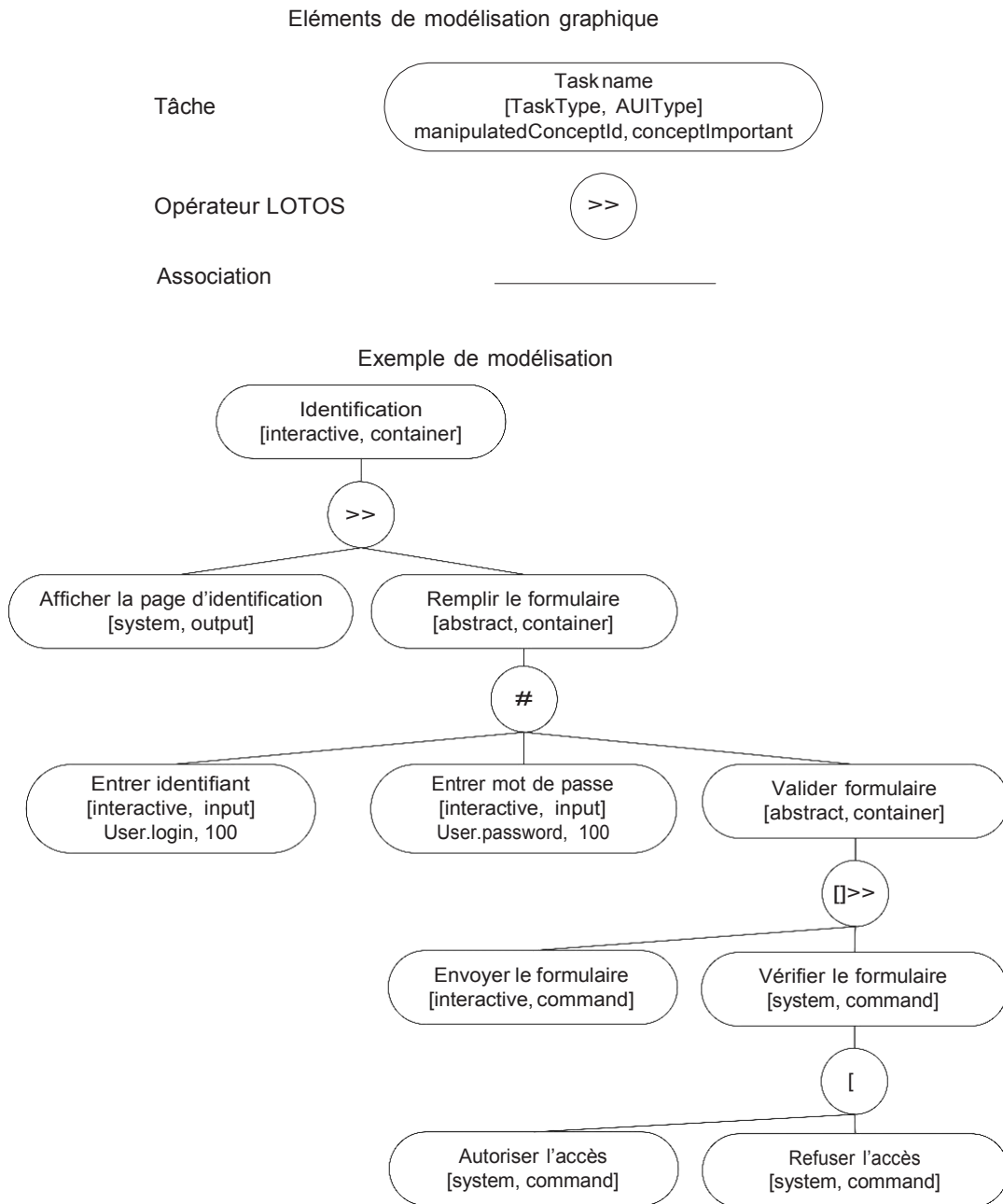


Figure 3.8 – Représentation graphique du métamodèle de tâches.

point d'entrée du processus de génération des IHM. En termes de génération, nous n'émettons pas de propositions concernant les heuristiques de transformations et renvoyons aux travaux sur le sujet (voir section 1.3.2.2).

Nous nous positionnons cependant vis-à-vis de l'**adaptation** des IHM qui se doivent de tenir compte des différents modèles (i.e. organisations, activité, cibles et tâches). Sans prétendre à l'**exhaustivité** en matière d'**adaptation**, notre proposition s'**articule** autour des mécanismes suivants :

Adaptation à la cible (i.e. le contexte d'interaction)

- **Adaptation au matériel** : notamment aux contraintes d'**espace d'affichage** et des différences modales introduites par les écrans tactiles. À cet effet, la notion d'**« importance »** des concepts manipulés dans les modèles de tâches entre en ligne de compte, en changeant la présentation des concepts à l'**importance** moindre si l'**espace d'affichage** vient à être exigü. Nous avons exposé ce type d'**adaptation** en section 1.3.2.2 et celle-ci permet de changer la représentation d'**un** concept ou de **l'omettre si** l'espace vient à manquer. Cette notion d'**importance** est héritée de CTT [84] et a notamment été traitée par Paterno et al. [86]. Enfin, pour supporter différents types de matériels, en l'**absence** de clavier physique les composants pourront être adaptés à l'**utilisation** d'écrans tactiles, en proposant par exemple des menus et boutons plus gros et en évitant de recourir à des infobulles ;
- **Adaptation à l'environnement physique** : comme par exemple à la luminosité ambiante pour ajuster le contraste (une fonction bien souvent présente en standard dans la plupart des appareils de nos jours), au bruit ambiant pour passer de feedback sonores à des feedback visuels ou tactiles en cas d'**environnement** bruyant, ou encore à la localisation géographique pour proposer des services différents (une notion qui sera surtout considérée vis-à-vis de la conscience de groupe en section 3.2) ;
- **Adaptation à l'utilisateur** : dans un premier temps à travers les stéréotypes d'**utilisateurs** (et tel que suggéré par CAMELEON [21, 113, 23]) qui permettent de définir des groupes de personnes ayant des particularités communes (e.g. maîtrise d'**une** tâche, déficience visuelle, etc.). Il est dès lors possible, à la génération de l'IHM, de proposer par exemple des menus d'**aides** ou des composants plus imposants dans le cas d'**une** déficience ;

Adaptation à l'activité collective

- **Adaptation au rôle** : en dehors des stéréotypes, le rôle de l'**utilisateur** entre également en ligne de compte en proposant des représentations différentes de certains concepts suivant le rôle. Par exemple, pour une tâche de visualisation de planning, une vue par diagramme de Gantt sera pertinente pour un chef de projet, alors **qu'une** vue par calendrier individuel sera plus pertinente pour les exécutants. Le rôle des utilisateurs est aussi déterminant dans la gestion des droits d'**accès** qui, ici, est gérée par génération en élaguant les arbres de tâches suivant les tâches non attribuées à un rôle spécifique ;
- **Adaptation à l'état de l'activité** : c'est-à-dire une adaptation suivant l'état courant du flux de travaux en proposant un accès aux tâches actives. Cette notion est décrite plus en détail en section 3.3 vis-à-vis de l'**application** des flux de travaux.

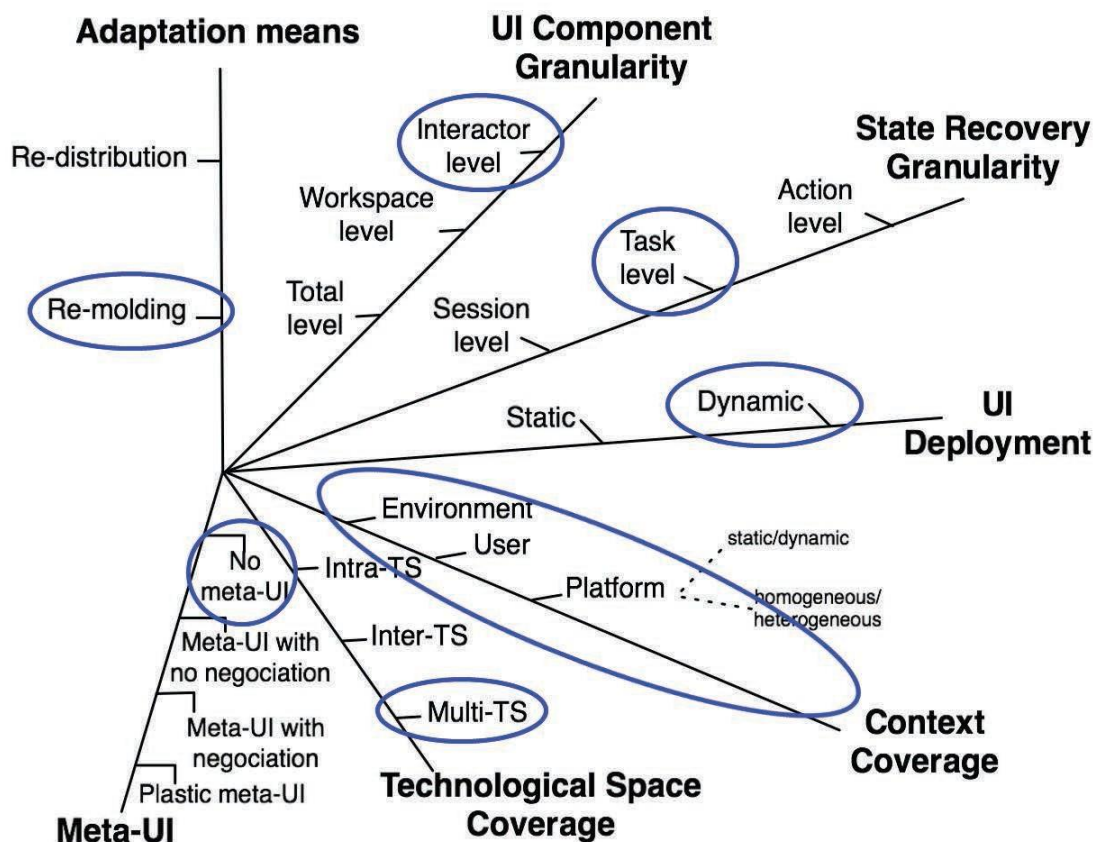


Figure 3.9 – Notre positionnement (entouré) sur l’espace problème de CAMELEON (en anglais), repris de Coutaz et al. [26].

Nous nous positionnons ainsi par rapport aux travaux sur CAMELEON [26] et son espace problème (voir figure 3.9). Notre but est de proposer une adaptation aux appareils les plus couramment utilisés de nos jours, soit les ordinateurs (de bureau ou portable), les *smartphones* et les tablettes, et ceci vis-à-vis du contexte de l’activité collective (i.e. *context coverage*). Cette adaptation se fait au niveau des composants (i.e. *UI component*). Nous n’apportons pas de réponse à la redistribution de l’interface (i.e. *adaptation means*) et ne donnons aucun contrôle à l’utilisateur sur le processus d’adaptation (i.e. *meta-UI*).

Le cas de la continuité de l’interaction (i.e. *state recovery*), de la dynamicit  de l’adaptation (i.e. *UI deployment*, c’est-à-dire si l’adaptation est pr -mod lis e ou dynamiquement faite   l’ex cution) et de la couverture technologique (i.e. *technological space*) sont   notre sens des probl matiques d’impl mentation technique et non conceptuelles. Nous les aborderons dans le cas de l’impl mentation de notre proposition, en section 4.1.

Résumé de section

Pour synthétiser cette section, nous proposons dans la modélisation du contexte de l'activité collective les éléments suivants :

Un **métamodèle de la collaboration** pour la modélisation des organisations, des acteurs, des tâches et des ressources ;

L'utilisation de **BPMN** pour la modélisation de l'activité ;

Un **métamodèle de tâches collaboratives** pour la modélisation de tâches en accord avec l'**IDM** ;

Des **mécanismes d'adaptation** qui tiennent compte des éléments précédents pour la génération d'**IHM**.

3.2 Prise en compte de la conscience de groupe par widgets adaptatifs

D'un point de vue pratique, nous avons vu en section 2.3.1 que mettre en **œuvre** la conscience de groupe consiste à fournir une vue adaptée sur une situation ou le contexte. Pour cela, deux grands types **d'approches s'opposent** : reposer sur une **boîte à outils** spécifique (e.g. MAUI [58, 59]) ou fournir des widgets dédiés (e.g. WSL [121]). La première est générique mais ne cible pas des situations précises de coordination, alors que la seconde se veut ciblée mais subit les inconvénients du polymorphisme vis-à-vis du problème de **l'adaptation** (i.e. devoir concevoir a priori toutes les représentations **d'un widget**). Trouver un équilibre entre ces approches nous semble approprié.

Notre approche de la prise en compte de la conscience de groupe consiste à générer des widgets adaptatifs dédiés à la présentation **d'informations** de groupe. En suivant ce qui a été suggéré à travers notre revue de la littérature, nous considérons la conscience de groupe comme étant situationnelle, **c'est-à-dire** que la présence **d'informations** relatives au groupe doit dépendre de situations de collaboration précises pour être pertinente. Notre approche par widgets **l'est donc également** et se résume en deux temps : **l'identification d'une situation**, puis une réponse adaptée sous forme de widgets. Afin de palier à **l'inconvénient** du polymorphisme, nous modélisons les widgets de la même manière que les autres IHM du système, **c'est-à-dire** comme des modèles de tâches à travers une **chaîne** de transformations, et non comme des CUI/FUI définies a priori.

Dans cette section, nous exposons en premier lieu notre approche de la modélisation des besoins de conscience de groupe qui servent à détecter des situations (section 3.2.1). Puis, nous détaillons notre approche des widgets (section 3.2.2) et leur insertion par composition **d'arbres** de tâches dans les IHM (section 3.2.3).

3.2.1 Modélisation des besoins en conscience de groupe

Dans la littérature, nous **n'avons** identifié que les travaux de Figueroa-Martinez et al. [37] comme relevant de la modélisation de besoins en conscience de groupe. Dans ces travaux, **l'identification d'une situation** se limite à la modélisation de contraintes sur le modèle du domaine. La conscience de groupe étant relative à **l'ensemble** du contexte de **l'activité** collective, nous estimons que les situations doivent être exprimées par rapport à **l'ensemble** des modèles. Une situation est ensuite associée à un besoin en informations sur le groupe qui sont communément caractérisées suivant la nomenclature de Gutwin et al. [53, 55] (e.g. le qui, quoi, où, comment, etc.). **D'autres** travaux, tels que ceux autour de TOUCHE [88, 87], emploient cette nomenclature pour abstraire la conscience de groupe (i.e. au niveau de **l'AUI** dans le cas de TOUCHE). La réponse aux besoins exprimés consiste à fournir une vue sur ces informations lorsque la situation associée est identifiée. Ainsi, identifier un besoin en conscience de groupe passe par les composantes suivantes :

Caractériser une situation, **c'est-à-dire** définir les conditions contextuelles qui délimitent un besoin particulier (i.e. l'état de **l'ensemble** du contexte). Ces conditions peuvent par exemple être attachées à une tâche précise qui soit en plus collaborative (condition qui pourrait se résumer à **l'affirmation** « toute tâche impliquant plus de

deux acteurs ») et uniquement lorsque **l'utilisateur** est en déplacement (« en tout lieu différent du bureau ») ;

Caractériser le besoin, c'est-à-dire définir les informations ou types **d'informations** sur le groupe qui sont pertinents dans le cadre de la situation définie. Par exemple, dans la situation précédemment décrite, il peut être utile de savoir quels utilisateurs sont disponibles et leur tâche courante, besoin qualifié par les critères de **presence**, **identity**, **availability** et **action** suivant la caractérisation de Gutwin et al. [53, 55].

Notre proposition est **d'apporter** une meilleure vue sur le contexte (et donc **l'identification** des situations) et **d'adapter** la formulation de besoins à notre vue par widgets. Nous présentons le métamodèle proposé en figure 3.10 qui arbore une structuration similaire à la décomposition susnommée :

Une **collection de Conditions** qui définit la situation qui nécessite une conscience de groupe, une condition étant une contrainte sur un modèle du contexte. Les contraintes sont exprimées à travers le langage OCL³⁷ et peuvent **s'appliquer** à **l'ensemble** des modèles de l'écosystème (i.e. modèles **d'organisation**/de la collaboration, de tâches, du domaine, de cibles ou de **l'activité**) permettant ainsi **d'identifier** précisément une situation au delà du simple modèle du domaine ;

Une **collection de Required Properties** qui exprime le besoin en conscience de groupe. Dans une logique **d'abstraction**, le type **d'informations** se base sur la caractérisation de Gutwin et al. [53, 55], dont les valeurs utilisées sont décrites en table 3.1. La propriété **d'importance** (qui s'échelonne de 0 à 100, 100 relevant de **l'indispensable** et 0 de **l'inutilité**) représente, comme son nom **l'indique**, **l'importance** de la propriété vis-à-vis de la tâche et de la situation données. Ceci permet notamment de proposer des mécanismes **d'adaptation**, à **l'image** de la génération par arbre de tâches, en occultant par exemple des propriétés si **l'espace** ne **s'y** prête pas. Cette propriété entre également en ligne de compte dans **l'insertion** de widgets dans la tâche principale, spécifiée en section 3.2.3 ;

Un élément **Widget Requirement** qui cible la tâche principale ou un **nœud** dans son arbre de tâches (**taskContainerName**) où un besoin en conscience de groupe est identifié. **C'est** à cet endroit que le ou les widgets répondant au besoin devront être insérés. La propriété **alwaysVisible** détermine si le besoin est propre uniquement au **nœud** ciblé (**false**) ou à **l'ensemble** des sous-tâches dans **l'arbre** (**true**). Ceci permet de définir si un besoin affecte **l'ensemble** de la tâche ou une portion particulière.

Un tel métamodèle permet **d'exprimer** une situation précise via des contraintes sur **l'ensemble** du contexte de **l'activité** collective. Un exemple **d'instance** est donné en figure 3.11. Les conditions exprimées en OCL décrivent une situation où une tâche est réalisée par plus **d'un** acteur (condition1, « self.users->size() >= 2 ») et lorsque les acteurs concernés ne sont pas géo-localisés au même endroit (condition2, « self.users->collect(self.location)->asSet()->size() >= 2 »). Le besoin en termes **d'informations** est alors relatif aux propriétés **presence**, **identity**, **authorship**, **action**, **availability** et **location**.

Les éléments **Widget Parameter** et la propriété **widgetName** sont discutés en section (3.2.3).

37. <http://www.omg.org/spec/OCL/>

3.2. Prise en compte de la conscience de groupe par widgets adaptatifs

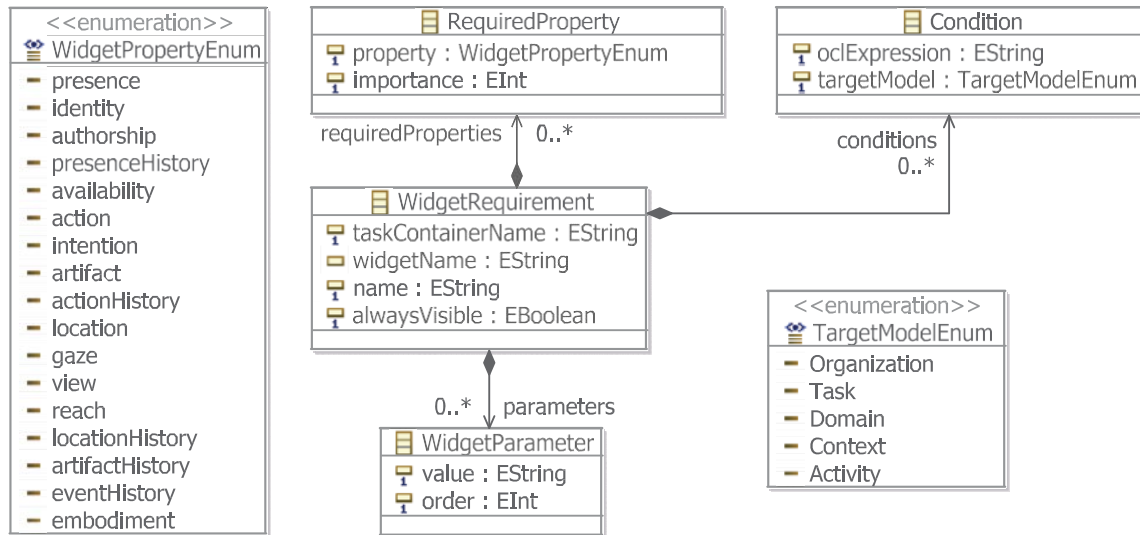


Figure 3.10 – Métamodèle de besoins en conscience de groupe. Représenté sous la forme d'un diagramme de classes UML.

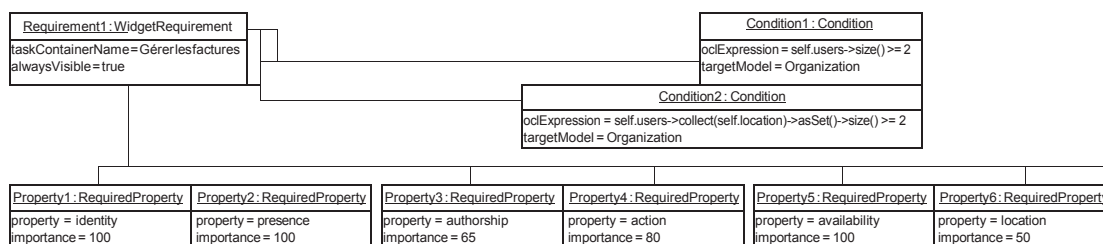


Figure 3.11 – Exemple de modèle de besoins en conscience de groupe. Représenté sous la forme d'un diagramme d'objets UML.

Table 3.1 – Propriétés d'abstraction de la conscience de groupe utilisées dans le méta-modèle de besoins. Basées sur la caractérisation de Gutwin et al. [53, 55].

Propriétés (en anglais)	Besoins associés
<i>presence</i>	Savoir si d'autres acteurs utilisent l'IHM
<i>presence history</i>	Savoir quand d'autres acteurs ont utilisé l'IHM
<i>identity</i>	Savoir qui sont les acteurs qui utilisent l'IHM
<i>embodiment</i>	Savoir à quoi ressemblent les acteurs
<i>availability</i>	Savoir si les acteurs sont disponibles
<i>action</i>	Savoir ce que font les acteurs
<i>action history</i>	Savoir ce qu'ont fait les acteurs
<i>authorship</i>	Savoir la paternité d'une action
<i>intention</i>	Savoir le but derrière l'action d'un acteur
<i>artifact</i>	Savoir si des ressources sont manipulées
<i>artifact history</i>	Savoir quand des ressources ont été manipulées
<i>location</i>	Savoir où sont les acteurs
<i>location history</i>	Savoir où ont été les acteurs
<i>gaze</i>	Savoir vers où se porte le regard des acteurs
<i>view</i>	Savoir ce que regardent les acteurs
<i>reach</i>	Savoir ce que peuvent voir les acteurs
<i>event history</i>	Savoir quels événements se sont produits

3.2.2 Réponse sous forme de widgets adaptatifs

Une fois le besoin en conscience de groupe défini, il convient **d'y** apporter une réponse adéquate sous la forme **d'une** vue sur les informations requises. Nous avons vu en sections 2.1.2 et 2.3.1 deux types **d'approches** : **l'usage** de boîtes à outils dédiées à la conscience de groupe avec TOUCHE [88, 87, 89], et **l'usage** de widgets pour encapsuler les vues sur les informations avec WSL [121].

Les boîtes à outils pour la conscience de groupe sont dédiées au partage de **l'espace** de travail, et ne permettent pas de répondre à tous types de besoins. Par exemple, dans le cas de TOUCHE, représenter un historique des actions utilisateurs **n'est** pas automatiquement généré par boîtes à outils, il est nécessaire de modéliser cette interaction en amont via **l'arbre** de tâches pour **qu'elle** soit présente.

Les widgets présentent le même problème mais ont **l'intérêt** de permettre une factorisation de portions **d'IHM** fréquemment utilisées et donc **d'augmenter** la réutilisation de modèles. Mettre en **œuvre la conscience** de groupe fait souvent appel à des IHM telles que des listes **d'acteurs** connectés, des historiques de modification, des vues radar. Leur inconvénient, notamment dans WSL [121], est **d'arborer** un format polymorphe pour ce qui est du problème de **l'adaptation**. Or, ceci revient à modéliser **a priori** toutes les représentations possibles **d'un** widget, limitant **l'adaptation** et augmentant la complexité de la conception.

Pour palier à ce défaut, tout en reposant sur leur concept, nous proposons de considérer les widgets comme étant des IHM comme les autres dans le processus de génération. **C'est-à-dire** de les abstraire par des arbres de tâches et **d'en** générer des IHM adaptées au

3.2. *Prise en compte de la conscience de groupe par widgets adaptatifs*

contexte. La figure 3.12 représente **l'arbre** de tâches **d'un** tel widget qui affiche une liste des acteurs connectés à **l'IHM**. **L'importance** donnée aux concepts permet de générer des IHM adaptées suivant des contraintes (comme **l'espace** disponible) sans avoir à modéliser toutes ces représentations a priori.

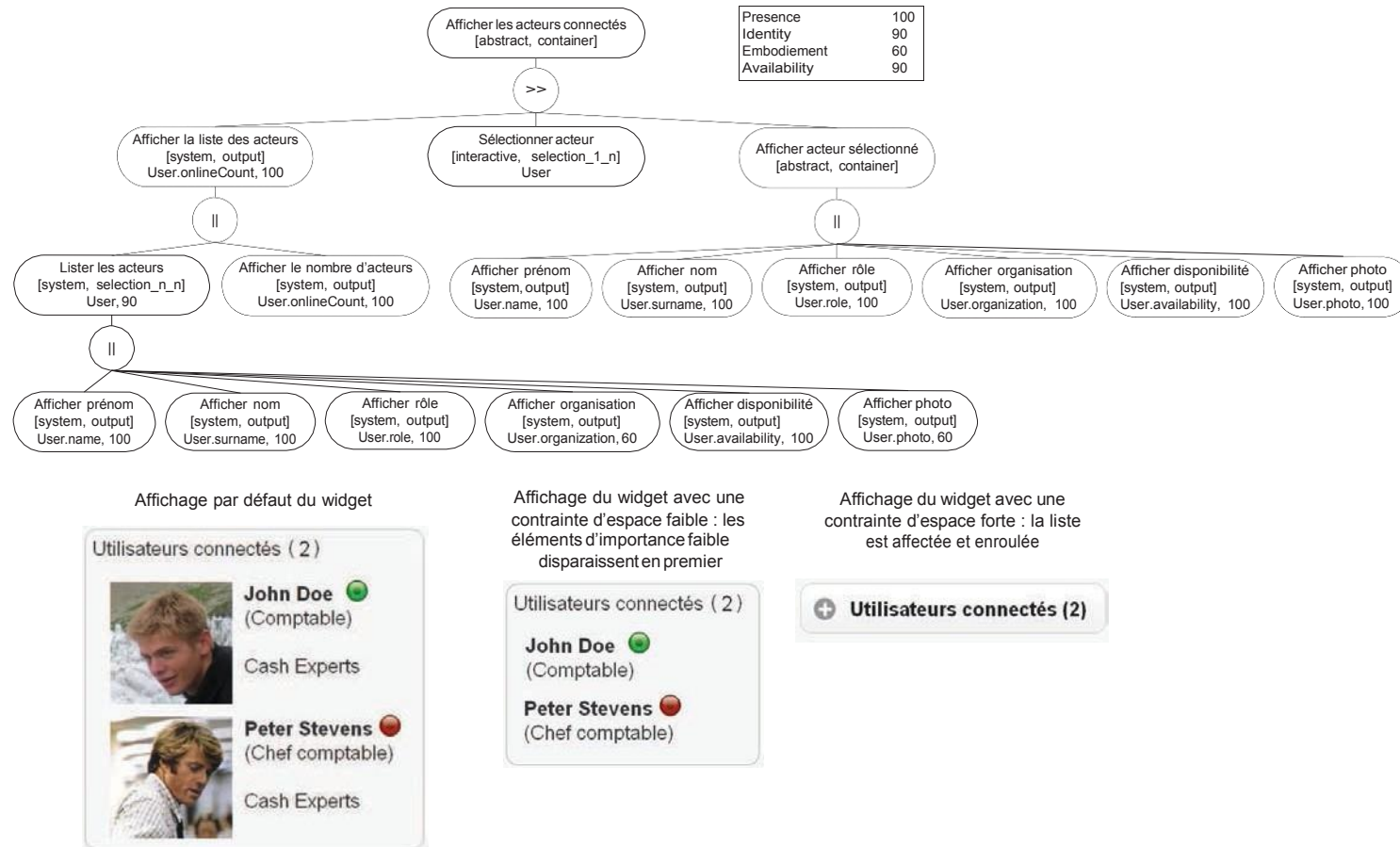


Figure 3.12 – Exemple d'un modèle de tâches d'un widget listant les acteurs connectés et différentes représentations adaptées en fonction de l'importance des concepts.

Afin d'identifier à quel(s) besoin(s) un widget peut répondre, le métamodèle de tâches dispose d'un élément **WidgetProperty** qui détermine quel(s) critère(s) de conscience de groupe y sont associés (voir figure 3.7, page 125). La propriété de **relevance** représente avec quel degré de pertinence (de 0 à 100) un widget répond à une propriété de conscience de groupe. Cette pertinence est donnée vis-à-vis de la qualité de la représentation de la propriété et de son accessibilité (i.e. l'effort requis pour l'utilisateur à trouver l'information). Sa définition est laissée à l'appréciation du concepteur du widget ou pourrait notamment être déterminée par test utilisateur. Par exemple, dans un widget qui affiche le nom des acteurs avec une importance forte (e.g. 100), il peut considérer que le widget répond à la propriété d'**identity** avec une pertinence de 100 car l'information sera toujours visible. À l'inverse, si l'affichage d'une information nécessite le survol d'un élément pour qu'une infobulle apparaisse, son accessibilité est alors indirecte et le concepteur peut choisir d'opter pour une pertinence moindre. Nous couvrirons un cas concret de modélisation de widget et de définition de ses propriétés en section 4.3.2.1.

3.2.3 Composition d'interfaces et dépôt d'arbres de tâches de widgets

Nous venons de discuter de la définition des besoins en conscience de groupe puis de la modélisation de widgets répondant à ces besoins. Il convient désormais d'intégrer ces widgets à l'arbre de tâches ciblé par le modèle de besoins. Cette opération se réalise en deux temps, tout d'abord en déterminant quel(s) widget(s) répondent au mieux aux besoins, puis en intégrant les arbres de tâches des widgets concernés dans l'arbre de tâches principal.

Vis-à-vis de l'état de l'art, les travaux autour de WSL [121] n'apportent pas de réponse à la méthode de composition de l'interface avec les widgets décrits. En effet, WSL est avant tout un langage de spécification de CUI polymorphes et qui peuvent s'insérer dans d'autres CUI. Il n'est de fait pas conçu pour un niveau d'abstraction comme celui des arbres de tâches.

3.2.3.1 Sélection des widgets répondant aux besoins

Dans un premier temps, il est nécessaire d'identifier quel(s) widget(s) peuvent répondre aux besoins modélisés. Nous avons vu que, d'un côté, les modèles de besoins expriment l'importance des types d'informations de conscience de groupe requis. Ces types correspondent à la caractérisation de Gutwin et al. [53, 55]. Du côté des widgets, leur arbre de tâches est associé à des types d'informations qu'ils fournissent suivant la même grille de caractérisation et avec un taux de pertinence.

Sélectionner des widgets consiste alors à trouver un ensemble de widgets couvrant les besoins définis avec une pertinence au moins égale à l'importance du besoin. L'algorithme de sélection de widgets a donc deux objectifs : maximiser la couverture des besoins et minimiser le nombre de widgets (afin d'éviter une surcharge d'informations moins importantes). Dès lors, cet algorithme tombe dans le champ de l'optimisation multiobjectifs, un domaine de recherche en soi. Notre but ici n'est pas de proposer un algorithme de sélection optimal, mais simplement une base conceptuelle. Il serait également possible d'ajouter des

3.2. Prise en compte de la conscience de groupe par widgets adaptatifs

objectifs en lien avec **l'adaptation**. La mesure de **l'adaptation** est également un domaine complexe que nous ne cherchons pas à couvrir de manière pertinente, bien conscients de cette limitation dans notre proposition. Pour ces raisons et à défaut de validation de cet aspect, nous laissons le choix de **l'algorithme** de sélection ouvert.

Afin de capitaliser **l'usage** de widgets, nous suggérons **l'emploi d'un dépôt d'arbres** de tâches de widgets, à **l'image** de la proposition de WSL [121]. Le rôle **d'un** dépôt est de stocker des modèles de widgets et de **servir d'interface** de requêtes, **c'est-à-dire** que **c'est** au dépôt **qu'incombe** de répondre aux besoins en conscience de groupe et de renvoyer **l'ensemble** de widgets adéquat en réponse. **L'intérêt d'un** dépôt est de favoriser la réutilisation des widgets, que ce soit au sein **d'un** même système ou partagé avec **d'autres** systèmes, mais aussi de favoriser **l'enrichissement d'une** base commune de widgets, à la manière **d'une** approche par services.

3.2.3.2 Modèles de widgets paramétrables

Une fois les widgets sélectionnés, ceux-ci doivent être intégrés au niveau de la tâche ciblée par le modèle de besoin. Nous avons vu **qu'à** ce sujet WSL [121] **n'a** pas pour vocation de répondre à la problématique de composition. En dehors de notre problématique de support de la coordination, certains travaux sur les MBI nous apportent des éléments de réponse.

C'est notamment le cas des travaux de Sinnig et al. [102] qui consistent à identifier des motifs **d'interactions** récurrents (e.g. se connecter à une application, utiliser un champ de recherche, naviguer une liste, etc.) et à stocker ces fragments génériques **d'arbres** de tâches indépendamment pour les réutiliser dans un cadre spécifique. Nous pouvons également citer les travaux de Blouin et al. [10] qui abordent le sujet sous **l'angle** de la modélisation par aspects. Ici, une tâche est considérée comme pouvant être réalisée de différentes manières. Les portions **d'une** tâche qui peuvent être réalisées de différentes manières sont alors modélisées sous forme **d'aspects**, une notion assez proche des motifs **d'interactions**, qui eux-mêmes renvoient à un paradigme similaire aux widgets.

En ce sens, nous avons pensé les widgets comme des éléments **d'IHM** génériques. En effet, en dehors du cadre de la conscience de groupe, **l'utilisation** de widgets permet de factoriser des portions **d'IHM** fréquemment utilisées ou de mettre en **œuvre** des bonnes pratiques. Par exemple, afficher une liste et permettre de consulter le détail de ses éléments est une tâche courante, seuls les concepts du domaine manipulés changent **d'une** instance à **l'autre**. Pour cela, nous permettons de détourner **l'usage** premier des modèles de besoin en conscience de groupe via l'élément *WidgetParameter* et la propriété *widgetName*. Un widget générique peut alors être transformé en un arbre de tâches de widget paramétré (voir figure 3.13).

3.2.3.3 Composition d'arbres de tâches

Les widgets étant censés être accessibles à tout instant au niveau de la tâche ciblée, nous considérons leur insertion dans la tâche principale comme relevant **d'une** tâche parallèle (déterminée par un opérateur LOTOS [117] **d'«** indépendance de **l'ordre** **»**), tel **qu'illustré** par la figure 3.14. Cette insertion se fait via transformation de **l'arbre** de tâches

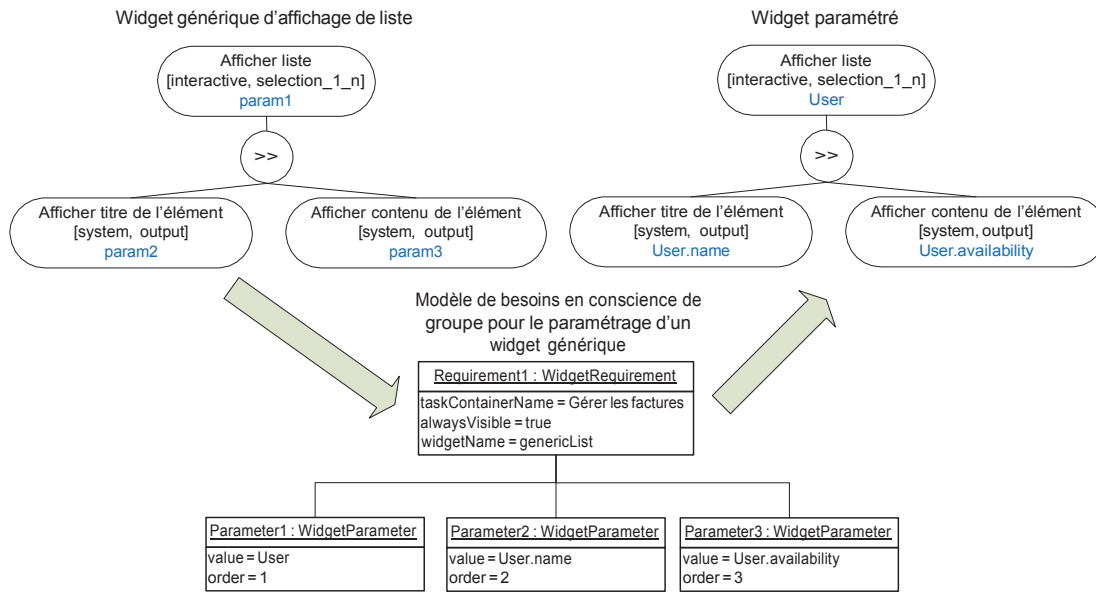


Figure 3.13 – Schématisation du paramétrage d'un widget générique via un modèle de besoin.

principal. La démarche est alors d'insérer un opérateur de parallélisme entre la tâche ciblée par le modèle de besoins et ses branches. Les branches décalées se retrouvant sans parent, une tâche intermédiaire de conteneur abstrait est créée. Un second conteneur est créé de l'autre côté de l'arbre afin d'héberger les widgets à insérer, sous l'égide d'un opérateur parallèle.

3.2. Prise en compte de la conscience de groupe par widgets adaptatifs

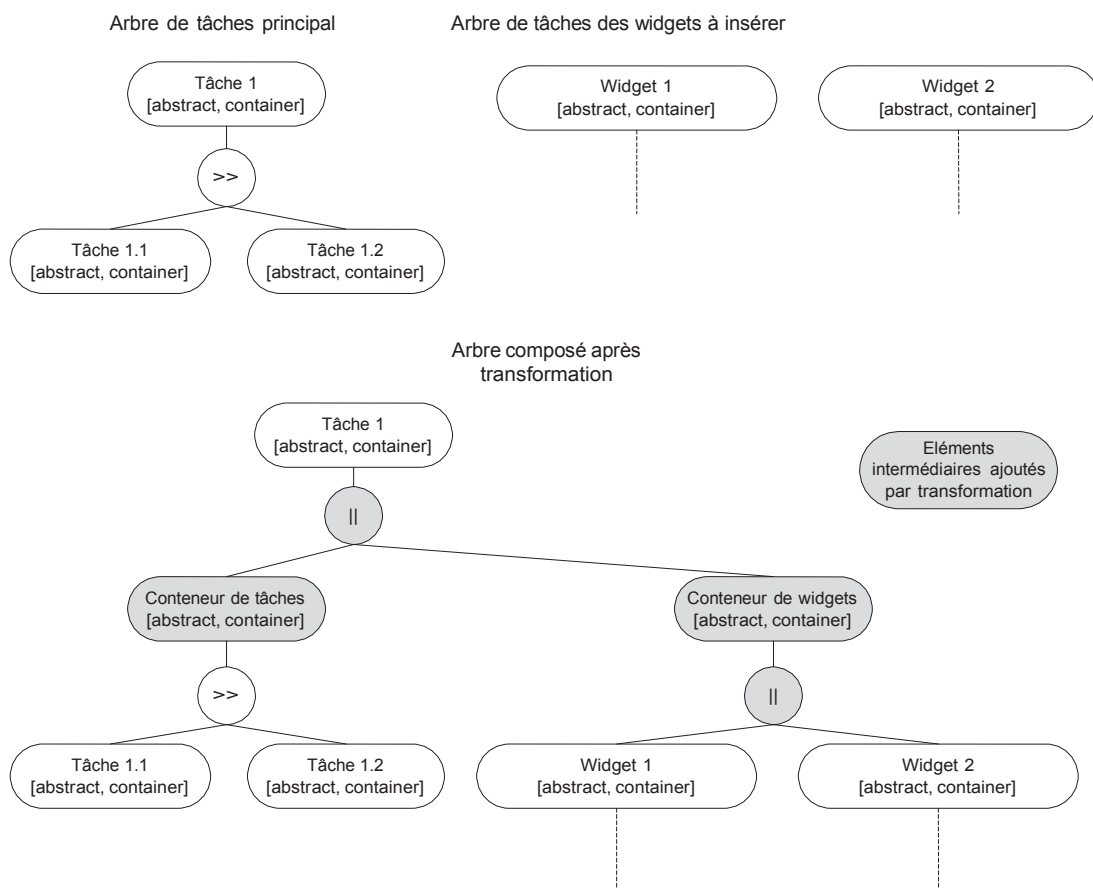


Figure 3.14 – Schéma d'une transformation d'un arbre de tâches pour y insérer des widgets. La tâche 1 est la tâche ciblée par le modèle de besoins.

Résumé de section

Pour résumer cette section, nous avons proposé des métamodèles et des mécanismes pour la génération de widgets adaptatifs dédiés à la conscience de groupe. Notre proposition **s'articule** sur trois plans :

La modélisation des besoins et situations via la proposition **d'un** métamodèle de besoins en conscience de groupe ;

Un dépôt de modèles de widgets adaptatifs qui détermine une réponse adéquate aux besoins modélisés en fournissant un ensemble de widgets, eux-mêmes consistant en des arbres de tâches permettant de générer des IHM adaptées ;

La présentation de widgets par composition en insérant les arbres de tâches des widgets suggérés par le dépôt dans **l'arbre** de tâches ciblé par le modèle de besoin.

La figure 3.15 reprend ces éléments et illustre le processus de sélection et de composition des widgets.

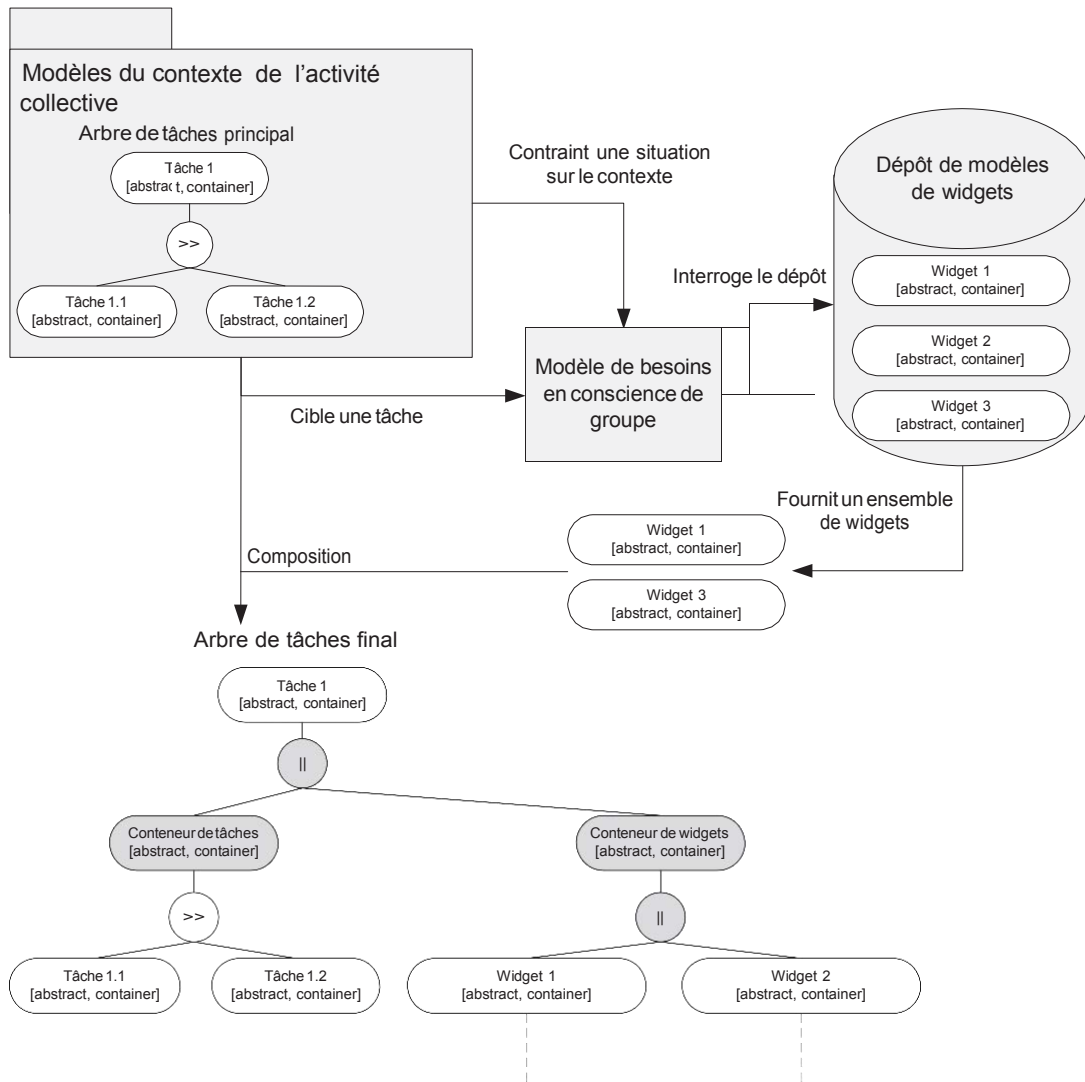


Figure 3.15 – Synthèse du mécanisme d'insertion de widgets.

3.3 Prise en compte des flux de travaux

Nous avons conclu en section 2.3.2 que les flux de travaux ont un emploi double vis-à-vis de notre approche. Du point de vue de la modélisation, ils permettent de formaliser **l'activité** (i.e. **l'interdépendance** entre les tâches en vue **d'atteindre** un objectif commun) et du point de vue de **l'interaction**, ils mettent en application les procédés, soutenant ainsi la coordination.

En section 3.1.2 nous avons traité les aspects de modélisation du contexte de **l'activité** collective pour porter notre choix sur BPMN pour ce qui est de la modélisation des flux de travaux. Dans cette section, nous présentons la facette pratique de ce que nous proposons, **c'est-à-dire** la mise en **œuvre** de la coordination par flux de travaux.

Pour cela, nous abordons le sujet en deux temps. Tout **d'abord** nous traitons de **l'application** des flux de travaux (section 3.3.1) en positionnant notre proposition par rapport aux travaux en MBUI. Puis, nous abordons le problème de la présentation des IHM associées à un flux de travaux (section 3.3.2) en proposant la génération **d'un tableau** de bord.

3.3.1 Application des flux de travaux

Vis-à-vis de la mise en **œuvre** des flux de travaux, nous avons discuté du sujet dans le cas des MBUI en section 2.3.2. Nous avons vu que la tendance générale est de considérer **qu'une** activité de flux de travaux est réalisée à travers une IHM spécifique, et donc associée à un arbre de tâches. Ceci, que ce soit dans les approches de conception de collectifs basées sur UML (CIAM [76], AMENITIES [44] et TOUCHE [88, 87, 89]), les métamodèles de tâches collaboratives (CCTT [78], COMM [66, 65] et CTML [126, 127, 39]) ou les extensions **d'UsiXML** [70] (FlowiXML [51] et UsiWSC [15, 14]). Il convient donc **d'associer** un métamodèle de flux de travaux à un métamodèles de tâches collaboratives. Dans les sections 3.1.2 et 3.1.3, nous avons opté **pour l'usage de BPMN** pour la modélisation de flux de travaux ainsi que de dériver de CTML pour ce qui est de modéliser les tâches.

Ces choix se justifient davantage ici. Tout **d'abord**, les approches par modèles de tâches (CCTT et COMM) **n'ont pas l'expressivité** nécessaire pour se substituer entièrement aux formalismes dédiés aux flux de travaux ou même des dérivés **d'UML**. Ces derniers, dont CIAM et AMENITIES, ont plus pour vocation de servir **d'aide** à la conception que de modèles productifs dans une **chaîne d'IDM**. FlowiXML propose une modélisation pertinente en associant ConcurTaskTree (CTT) [84] à des flux de travaux dérivés de BPMN, mais ne tire pas parti de cette définition pour synchroniser les IHM générées, seuls les arbres de tâches entrant en compte dans le processus de génération. UsiWSC, quant à lui, fait cas particulier en étant spécifique à la composition de services web (décrits en BPEL) mais a cependant un intérêt dans son interprétation à **l'exécution** du flux de travaux pour la synchronisation des services.

Notre proposition de mise en **œuvre** des flux de travaux se positionne entre FlowiXML (pour ce qui est de la modélisation) et UsiWSC (pour ce qui est de **l'exécution**). Sur base de BPMN et de notre métamodèle de tâches dérivé de CTML, nous établissons des liens entre activités de flux de travaux et tâches à travers les annotations de BPMN, tel **qu'illustré** en figure 3.16 et de manière semblable à FlowiXML. Ce lien est établi dans

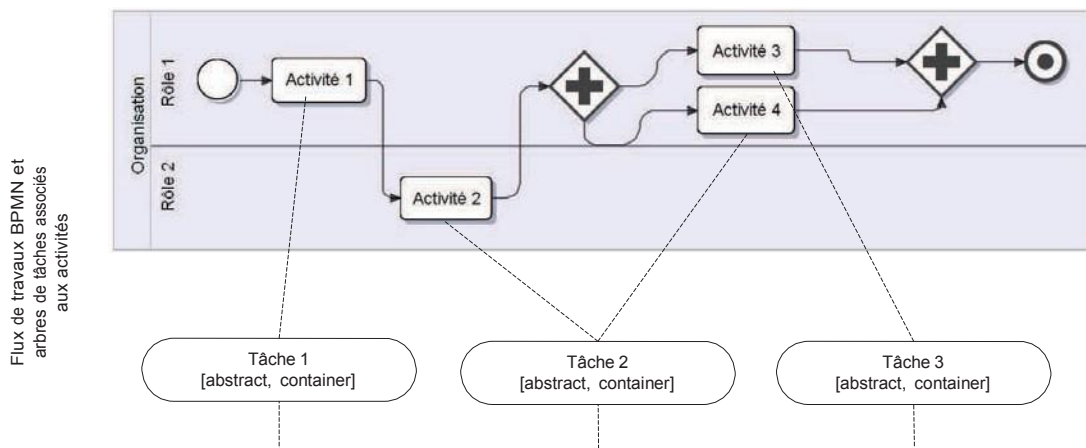


Figure 3.16 – Schéma de l'association entre modèle de flux de travaux (formalisme BPMN) et modèles de tâches pour la mise en œuvre des flux de travaux.

notre métamodèle de la collaboration (voir figure 3.3) en associant une tâche à une activité. Lors de la génération d'IHM, chaque arbre de tâches donne lieu à une IHM et, afin de combler les manques de FlowiXML, nous suggérons une interprétation des flux de travaux à l'exécution afin d'orchestrer les différentes IHM. De plus, les pré/post-conditions OCL³⁸ sur les tâches permettent d'exprimer les transitions entre les activités d'un flux de travaux lorsqu'une tâche est complétée, à la manière de CTML (qui n'exige pas de flux de travaux en lui-même).

3.3.2 Génération d'un tableau de bord

Si dans la littérature MBUI la modélisation et l'application des flux de travaux sont traités (notamment via FlowiXML [51, 42] et UsiWSC [15, 14]), la question de l'accès aux IHM générées ne semble pas abordée. En effet, appliquer un flux de travaux implique qu'une IHM soit accessible ou non suivant l'état du flux et du rôle de l'utilisateur. Que présenter à l'utilisateur dans l'un ou l'autre cas? FlowiXML, qui se positionne sur la modélisation, n'aborde pas le sujet de la génération basée sur un flux de travaux. UsiWSC, quant à lui, apporte une réponse mais vis-à-vis de la composition de services web et non basée sur des IHM générées depuis des arbres de tâches.

Nous considérons qu'un élément de réponse à cette question est apporté par les métamodèles de tâches collaboratives que sont CCTT [78] et COMM [66, 65]. Ces approches, qui sont avant tout des formalismes graphiques et non productifs, abordent (comme nous l'avons vu en section 2.2.2) la synchronisation entre les tâches en positionnant un arbre par dessus les tâches individuelles. Il n'y a alors qu'un pas à considérer un arbre de synchronisation CCTT ou COMM pour générer une IHM de synchronisation des tâches. Nous qualifierons ici une telle IHM d'extra-IHM, qui est définie chez Sottet [103] comme étant

38. <http://www.omg.org/spec/OCL/>

une IHM de contrôle sur les modèles, en l'occurrence sur le modèle de flux de travaux dans notre cas. Proposer une extra-IHM a un intérêt double : gérer l'accès aux IHM par rôle suivant l'état du flux de travaux et avoir une vue sur l'état du contexte de l'activité collective pour renforcer la conscience de groupe. Or, dans notre proposition, nous avons mis en avant la pertinence de l'usage d'un standard de modélisation de flux de travaux (BPMN) pour remplir ce rôle.

Nous suggérons alors la génération d'une extra-IHM, ou tableau de bord, à partir des flux de travaux BPMN, qui aura pour but de gérer l'accès aux IHM associées aux activités des flux de travaux. Afin de tirer parti du processus de génération d'IHM à partir d'arbres de tâches et d'uniformiser notre proposition, notre approche consiste à transformer les flux de travaux en un arbre de tâches servant de support à la génération du tableau de bord. L'objectif est alors que l'arbre de tâches du tableau de bord décrive l'accès aux IHM liées aux activités BPMN, sous condition de l'état du flux de travaux (en rendant des branches accessibles ou non), tel que illustré en figure 3.17.

Un autre avantage de générer un tableau de bord sur base d'un arbre de tâches est de lui permettre d'être ciblé par des besoins en conscience de groupe et par conséquent d'être composé avec des widgets. De plus, dans la section 1.2.2, nous avons vu que la coordination ne consiste pas toujours en la standardisation des procédés par flux de travaux et qu'une application doit être flexible à ce sujet. Pour cela, nous considérons les tâches ne faisant pas partie d'un flux de travaux comme étant liées à d'autres formes de coordination (i.e. ajustement mutuel, standardisation des qualifications et standardisation des résultats) et son IHM doit par conséquent être accessible à tout instant dans le tableau de bord (et donc non soumise à l'état courant du flux de travaux). Cela peut être le cas d'IHM dédiées à la communication (e.g. messagerie) ou aux tâches individuelles (e.g. prise de notes, agenda).

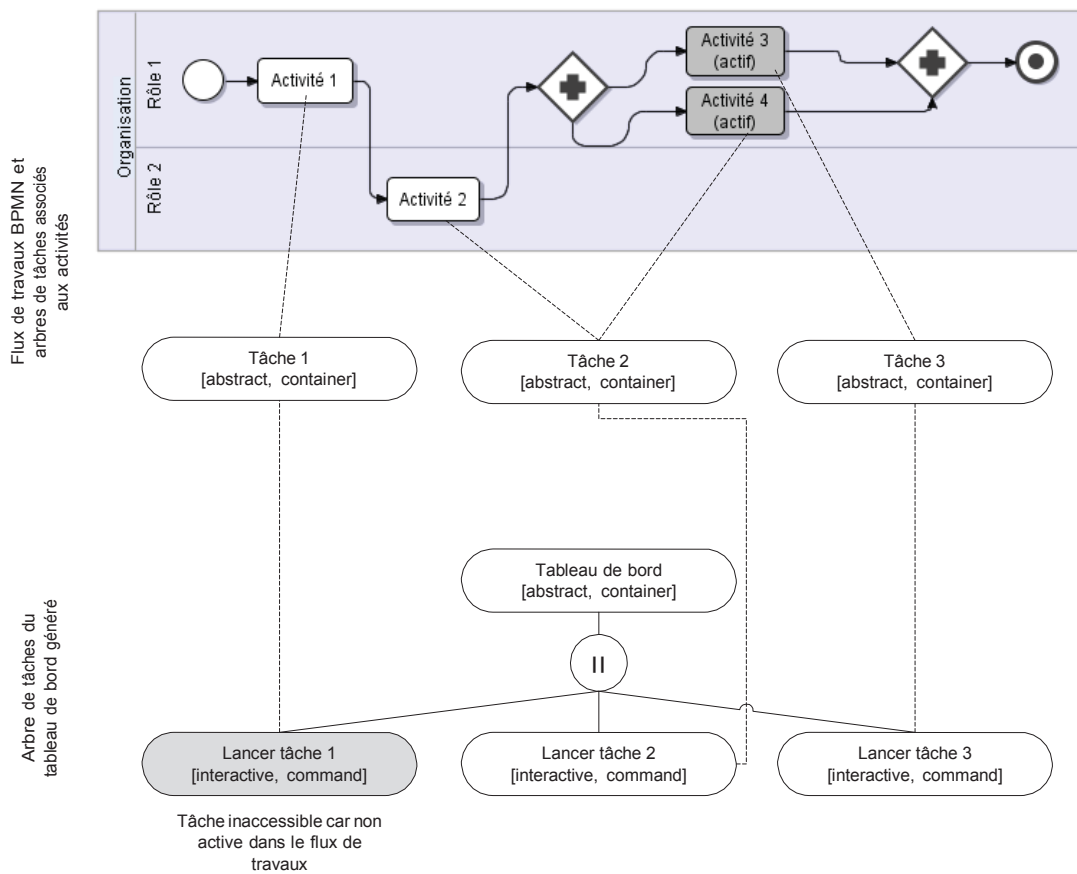


Figure 3.17 – Schéma de la génération d'un tableau de bord à partir de flux de travaux pour la gestion de l'accès aux tâches.

Résumé de section

En guise de synthèse de cette section, nous pouvons résumer le positionnement de notre proposition sur les flux de travaux via ces deux axes :

L'association de tâches aux activités BPMN afin de générer des IHM destinées à supporter la réalisation des activités. Le flux de travaux, interprété à **l'exécution**, détermine la synchronisation des tâches et celles-ci pilotent **l'avancement** du flux de travaux à travers les pré/post-conditions OCL des modèles de tâches ;

La génération d'un tableau de bord depuis BPMN afin de proposer une méthode **d'accès** aux IHM générées. Ce tableau de bord est transformé depuis un modèle BPMN en modèle de tâches qui détermine **l'accès** aux IHM suivant l'état du flux de travaux et le rôle de **l'utilisateur**.

La figure 3.17 reprend la prise en compte des flux de travaux et la génération **d'un** tableau de bord.

3.4 Intégration dans une méthodologie de conception

Dans les sections précédentes, nous avons exposé les aspects conceptuels de notre proposition, notamment nos contributions de modélisation à haut niveau en section 3.1 (i.e. la définition du contexte de l'**activité** collective) puis nos contributions à bas niveau avec la mise en **œuvre** de la conscience de groupe en section 3.2 et des flux de travaux en section 3.3. Il convient désormais **d'intégrer** ces éléments dans une méthodologie de conception qui détermine la manière dont se construisent et **s'imbriquent** les différents modèles.

Au chapitre précédent (voir section 2.1.1), nous avons passé en revue différentes méthodologies de conception de collectifs, dont CIAM [76], AMENITIES [44] et TOUCHE [88, 87, 89]. Ces méthodologies **s'organisent** en étapes de modélisation que nous avons synthétisé comme suit :

1. La modélisation des rôles et de leur hiérarchie (i.e. le sociogramme, **c'est-à-dire l'organisation**) ;
2. La modélisation des tâches et de leur attribution aux rôles/acteurs ;
3. La modélisation des interdépendances entre les tâches et **l'accès** aux ressources (i.e. **l'activité** dans son ensemble) ;
4. La modélisation des tâches utilisateurs (i.e. arbre de tâches).

Mais vis-à-vis de notre problématique, ces étapes ne couvrent pas tous les aspects de notre proposition, notamment en termes **d'adaptation** aux cibles et de prise en compte de la conscience de groupe. Nous introduisons alors deux nouvelles étapes spécifiques à ces aspects :

5. La modélisation des cibles (i.e. contexte **d'interaction**) ;
6. La modélisation des besoins en conscience de groupe.

Si l'étape de modélisation des cibles est déjà décrite dans les travaux autour de CAME-LEON [21, 113, 23], l'étape relative à la conscience de groupe nécessite des précisions. En effet, la conscience de groupe étant relative à **l'identification** de situations précises, leur modélisation implique une bonne connaissance des pratiques métiers. À cet effet, nous verrons en section 3.4.2 que nous orientons cette étape vers un cycle itératif où les utilisateurs sont impliqués. La figure 3.18 expose ces étapes, le cycle itératif et les métamodèles associés à chacun **d'entre eux** dans le support de la méthodologie.

3.4.1 Étapes de modélisation

D'un point de vue méthodologique, nous reprenons donc les grandes étapes de modélisation des autres méthodologies de conception que nous avons passé en revue. Nous détaillons ci-après comment les éléments de modélisation que nous avons proposé dans les sections précédentes **s'imbriquent** dans ces étapes.

1. **Modélisation du sociogramme** qui consiste à identifier quels sont les acteurs et leurs liens sous forme de groupes et **d'organisations** ainsi que le rôle **qu'ils** jouent dans ces éléments. Vis-à-vis de notre métamodèle de la collaboration (voir figure 3.3,

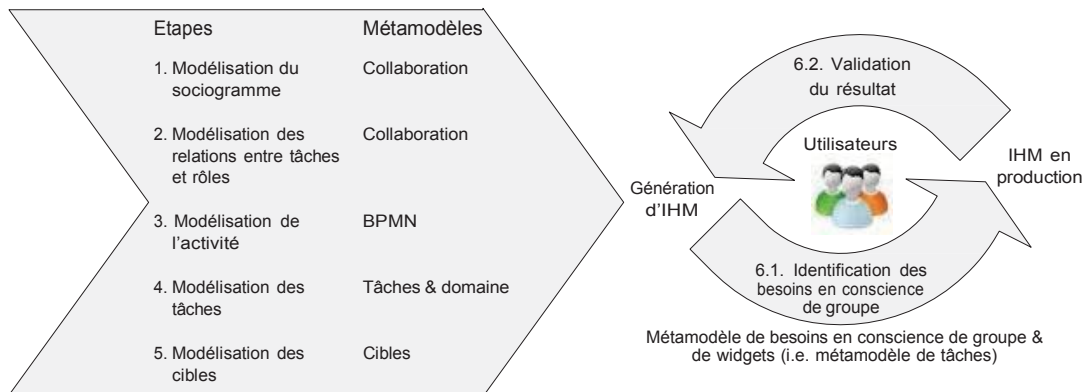


Figure 3.18 – Résumé des étapes méthodologiques et des métamodèles utilisés.

page 117), il s'agit de spécifier les éléments *user* et *user stereotype* qui définissent les acteurs, les éléments *organizational unit* et *organization relationship* qui définissent à la fois les organisations, les groupes à l'intérieur de celles-ci ainsi que les règles qui les unissent (e.g. sous-division, département, sous-traitant, partenaire, filiale, etc.). Enfin, les éléments *role* et *role relationship* définissent les rôles que les acteurs jouent et les relations hiérarchiques entre les rôles (e.g. supervision, délégation, etc.). La figure 3.19 illustre cette étape à laquelle nous associons une représentation graphique décrite en figure 3.20 ;

2. **Modélisation des relations entre tâches et rôles** qui consiste à compléter le modèle de la collaboration avec les tâches et leurs ressources nécessaires. Les tâches, uniquement identifiées par leur nom, sont associées aux acteurs qui les réalisent ainsi qu'aux rôles qui conditionnent l'attribution des tâches aux acteurs. Ces associations permettent de définir l'équivalent des tables d'adjacence présentes dans les autres méthodologies comme CIAM [76] et AMENITIES [44] ou le diagramme d'organisation de TOUCHE [88, 87, 89]. Ici, il est question d'identifier les tâches à haut niveau, sans aller dans la description précise des interactions (i.e. arbres de tâches). La figure 3.19 illustre cette étape superposée à l'étape précédente alors que sa représentation graphique est affichée en figure 3.20 ;
3. **Modélisation de l'activité** qui consiste à décrire l'interdépendance temporelle entre les tâches. Il s'agit ici d'établir les modèles de flux de travaux via BPMN qui font appel aux tâches et aux rôles pour déterminer l'accessibilité des tâches et leur synchronisation ;
4. **Modélisation des tâches** qui consiste à spécifier en détails les interactions des tâches, c'est-à-dire de définir les arbres de tâches en fonction de l'activité modélisée. Nous considérons également à cet instant la modélisation du domaine afin de pouvoir être manipulé dans les tâches ;
5. **Modélisation des cibles** qui consiste à définir les triplets d'environnement matériel, physique et utilisateur en vue de supporter une adaptation sur ces cibles. Il s'agit

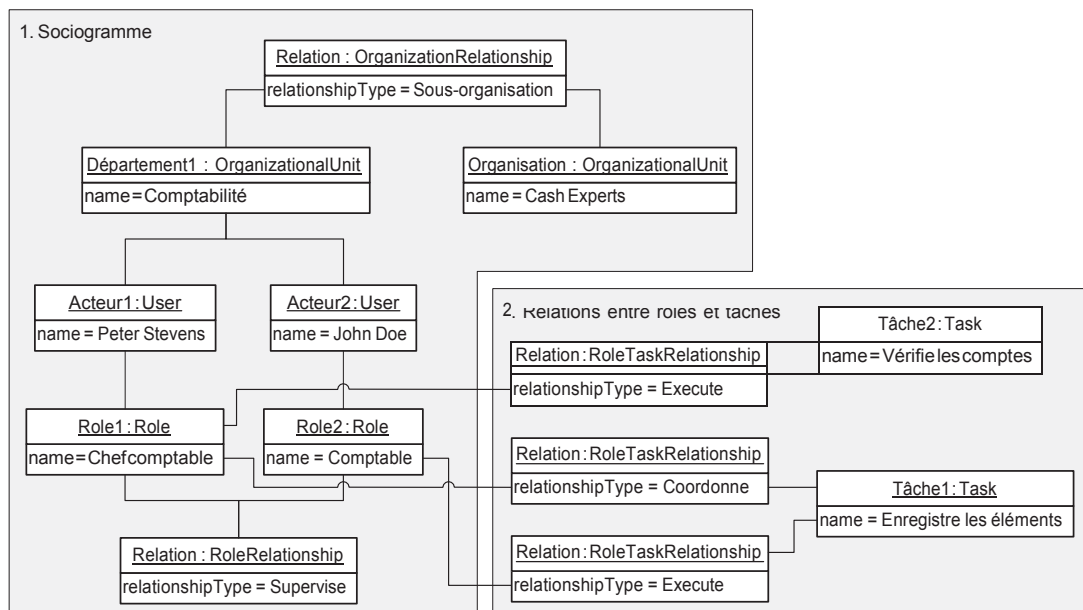


Figure 3.19 – Exemple de construction du modèle de la collaboration illustrant les étapes 1 et 2 de la méthodologie, représenté par un diagramme d’objets UML.

de modéliser les environnements particuliers que nous souhaitons supporter dans la génération des IHM. Ces triplets pilotent la génération en mettant en œuvre certains mécanismes d’adaptation, tels que ceux associés aux contraintes d’espace d’affichage, matériel ou aux particularités des utilisateurs. Concernant cette étape, la modélisation des cibles n’est pas nouvelle et nous nous référons aux travaux sur CAMELEON [21, 113, 23] pour cela.

En sortie de ces étapes, nous couvrons l’ensemble des modèles du système à l’exception de ceux définissant les besoins en conscience de groupe, qui font l’objet de la dernière étape décrite en section suivante.

3.4.2 Itération pour l’identification des besoins en conscience de groupe

Par rapport aux méthodologies de conception que nous avons vu et des grandes étapes précédemment décrites, il reste à aborder le sujet de l’identification des besoins en conscience de groupe et donc l’insertion de widgets de support à la conscience de groupe dans l’interface. Nous avons aussi vu que la conscience de groupe dépend de situations précises et que prévoir ces différentes situations est difficile en phase amont de conception, car elles impliquent une forte connaissance métier.

Pour répondre à cela, nous proposons d’orienter la méthodologie vers les utilisateurs et de les impliquer dans un cycle itératif de définition des besoins en conscience de groupe.

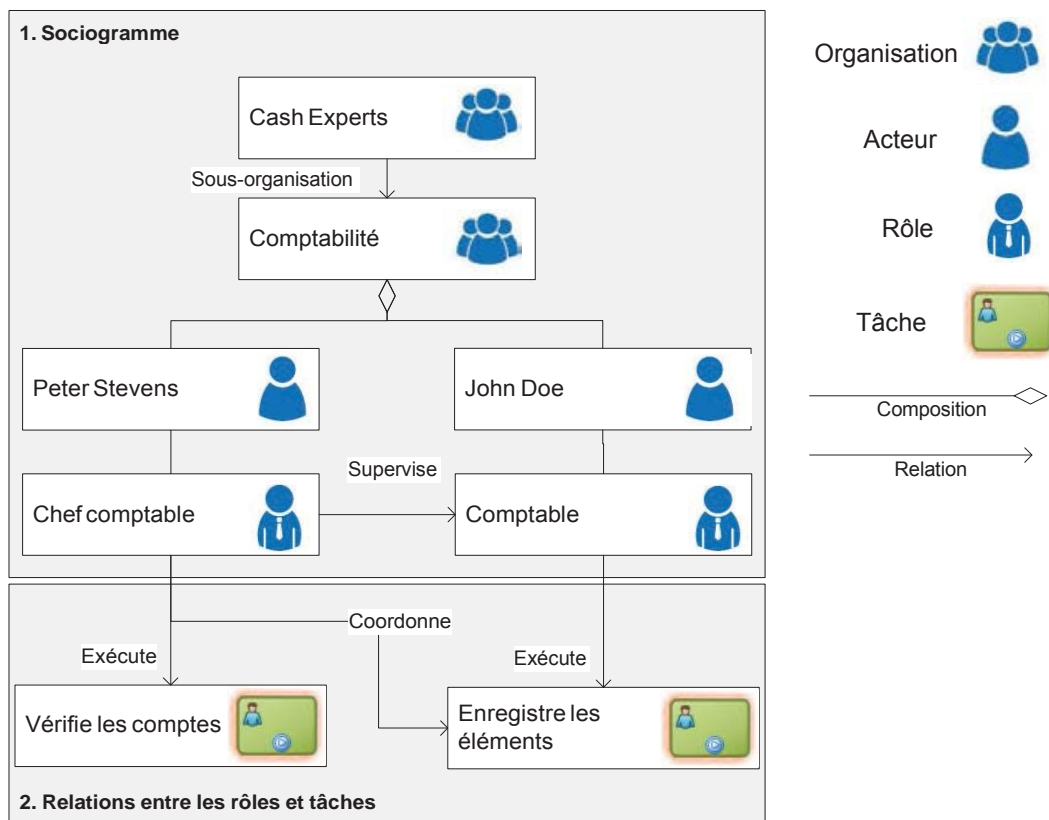


Figure 3.20 – Représentation graphique associée au modèle de la collaboration illustrant et reprenant l'exemple de la figure 3.19.

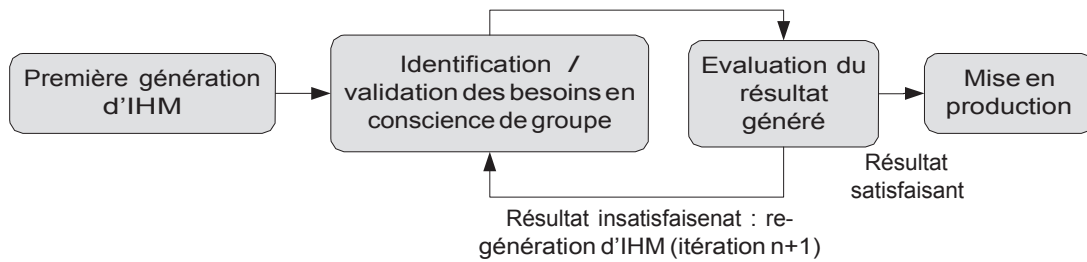


Figure 3.21 – Schéma du processus itératif de définition des besoins en conscience de groupe.

Ainsi, en sortie de l'étape 5 de la méthodologie, une première génération **d'IHM** est produite qui consiste en la base du processus itératif. L'objectif est alors **d'impliquer l'utilisateur** dans chaque tâche pour identifier le besoin en conscience de groupe. Cette implication se fait en deux étapes et sous forme itérative : identifier les besoins et valider les résultats (voir figure 3.21).

3.4.2.1 Etape 6.1 - Identifier les besoins

Nous avons défini nos métamodèles de besoins et de widgets autour de la caractérisation de Gutwin et al. [53, 55] qui nous permet **d'abstraire** des règles **d'insertion** de widgets suivant les conditions **d'une** situation. L'étape **d'identification** des besoins consiste pour **l'utilisateur** à spécifier les critères de Gutwin et al. [53, 55] dans une situation et une tâche données. Cette spécification des besoins passe par un questionnaire qui a été construit sur base de la classification de Gutwin et al. [53, 55] et dont le but est **d'identifier l'importance** des types **d'information** de conscience de groupe (voir annexe A.1). Ces éléments permettent ensuite au concepteur de définir des modèles de besoins.

Dans ce questionnaire, nous faisons également une distinction entre un besoin non-fonctionnel (i.e. propre à la conscience de groupe) et un besoin fonctionnel (i.e. indispensable à la tâche). En effet, comme nous l'avons vu, la conscience de groupe est un mécanisme non-fonctionnel, **c'est-à-dire** que la présence **d'information** sur le groupe favorise la coordination par gestaltisme mais **n'est pas censée être** indispensable à la réalisation **d'une** tâche. Si tel est le cas, nous quittons la frontière de la conscience de groupe, ce qui relève **d'un** besoin fonctionnel qui doit être remonté au niveau de la modélisation de la tâche plutôt que de relever **d'un** modèle de besoins en conscience de groupe.

À ce stade de la méthodologie, une nouvelle génération **d'IHM** peut être produite avec une réponse éventuelle aux besoins en conscience de groupe. Cependant, cette réponse dépend du dépôt de widgets existant, qui en début de conception peut être déjà fourni tout comme il peut être vide. Il incombe donc au concepteur de vérifier que les besoins trouvent une réponse et le cas échéant de définir le ou les widgets adéquats dans le dépôt. Ainsi, le dépôt de widgets est construit au fur et à mesure des déroulements de la méthodologie. De plus, certains modèles de besoins peuvent **s'avérer être** génériques (e.g. comme le besoin

d'une liste d'acteurs connectés au système) et à terme, il est possible d'imaginer que la première génération d'IHM tire partie de ces règles génériques, sous la forme de bonnes pratiques identifiées.

3.4.2.2 Etape 6.2 - Valider les résultats

Une fois la nouvelle génération d'IHM produite, un nouveau cycle débute. Dans un premier temps, il s'agit d'évaluer la réponse apportée par les widgets générés. Pour cela, un nouveau questionnaire (voir annexe A.2) est présenté aux utilisateurs, tirant à nouveau parti des critères de Gutwin et al. [53, 55]. Ici, ce n'est plus l'importance des besoins qui est évaluée, mais le taux de pertinence avec lequel la solution répond aux critères selon les utilisateurs (voir annexe A.2). Ainsi, le concepteur est en mesure de savoir si les widgets sélectionnés par le dépôt sont bel et bien pertinents et éventuellement d'ajuster la caractérisation ou la modélisation des widgets pour un nouveau cycle d'évaluations.

L'algorithme de sélection de widgets du dépôt n'étant pas optimal et afin d'aider le concepteur dans sa tâche, une partie de cette évaluation pourrait se faire automatiquement à l'aide de métriques sur la réponse donnée par le dépôt de widgets. Par exemple, une fois les widgets sélectionnés par le dépôt, il est possible de savoir avec quelle pertinence un besoin de conscience de groupe est couvert sur base des modèles. Un taux de satisfaction des besoins peut alors être calculé et donner au concepteur une première estimation de ce que pourront être les retours des utilisateurs.

En sus de la pertinence de la réponse aux besoins, il est également demandé à l'utilisateur de se prononcer sur la qualité des représentations choisies pour mettre en avant les informations présentes mais mal représentées (voir annexe A.3). En effet, la pertinence de certaines représentations dépend du rôle de l'utilisateur et de la situation, d'où la nécessité de valider la modélisation des widgets par situation.

Enfin, l'étape 6.1 est reproduite afin de déterminer si les besoins convergent. En effet, il se peut qu'une fois le résultat produit, l'utilisateur perçoive sa situation différemment et ajuste ses besoins face au résultat présenté. Si les besoins convergent et que l'évaluation du résultat est satisfaisante, le cycle itératif est terminé et les IHM peuvent entrer en phase d'exploitation.

Résumé de section

Pour résumer notre positionnement par rapport aux méthodologies de conception de collecticiels, nous avons montré comment nos propositions s'insèrent dans celles-ci via les étapes suivantes :

Les grandes étapes des autres méthodologies qui consistent à modéliser :

1. Lesociogramme de**l'activité**;
2. Les relations entre tâches et rôles;
3. **L'activité** (par flux de travaux);
4. Les tâches;
5. Les cibles;

Nous avons notamment vu comment nos propositions de métamodèles s'insèrent dans ces étapes.

Un cycle itératif dédié à la conscience de groupe car son support est lié à des situations difficilement identifiables en conception. En ce sens, ce cycle consiste à impliquer **l'utilisateur** dans la spécification de modèles de besoins en conscience de groupe, sur base **d'IHM** générées en sortie des étapes précédentes. Ce cycle se compose ainsi :

1. **L'identification** des besoins, sur base **d'un** questionnaire issu de la caractérisation de Gutwin et al. [53, 55];
2. La validation des IHM générées, sur base **d'un** questionnaire similaire afin d'évaluer la réponse donnée en termes de widgets.

3.5 Conclusion de chapitre

Dans ce troisième chapitre, nous avons proposé un cadre de conception et de génération d'IHM adaptées à l'activité collective. Pour ce faire, nous avons pris position par rapport aux manques que nous avons identifiés dans la littérature au chapitre 2. Le cadre que nous proposons, présenté de manière abstraite en figure 3.1 (page 113), a structuré notre proposition suivant les points faisant l'objet de contributions.

Dans un premier temps, nous nous sommes concentrés sur la modélisation du contexte de l'activité collective (section) 3.1. Nous avons abordé le sujet par quatre aspects particuliers :

- **La proposition d'un métamodèle de la collaboration** (section 3.1.1), consistant en un compromis entre les métamodèles d'UsiXML [70] et d'AMENITIES [44] quant à la modélisation des organisations, des acteurs, des tâches et des ressources ;
- **Le choix d'un métamodèle de flux de travaux** (section 3.1.2), qui décrit l'activité (i.e. l'interdépendance entre les tâches) et qui s'est porté sur un standard industriel, BPMN ;
- **La proposition de métamodèles de tâches et du domaine** (section 3.1.3), en partant sur la base de CTML [126, 127, 39] en y apportant des adaptations plus propices à l'IDM ;
- **Le positionnement de ces métamodèles dans la génération d'IHM adaptatives** (section 3.1.4), notamment sur les méthodes d'adaptation fonctionnelle suivant l'activité et de la présentation suivant les cibles.

Ces métamodèles constituent le contexte de l'activité collective et servent de point d'entrée au processus de génération des IHM.

Nous avons ensuite traité de ce processus de génération, et plus particulièrement vis-à-vis du support de la coordination que nous avons abordé en deux temps.

En premier temps, nous avons exposé notre proposition quant à la prise en compte de la conscience de groupe en section 3.2. Cette notion dépendant de la réponse à des situations précises, notre approche se présente en trois étapes :

- **Modéliser les besoins en conscience de groupe** (section 3.2.1), à travers un métamodèle définissant une situation (issue de contraintes sur le contexte de l'activité collective) associée à un besoin décrit suivant les critères de Gutwin et al. [53, 55] de la conscience de groupe ;
- **Permettre une réponse aux besoins sous la forme de modèles de widgets adaptatifs** (section 3.2.2), en considérant une approche par widgets sous forme de modèles de tâches, palliant les contraintes des widgets polymorphes. Ces widgets sont dédiés à présenter des informations de conscience de groupe et sont adaptatifs ;
- **Mettre en œuvre cette réponse par composition de modèles de tâches** (section 3.2.3), c'est-à-dire en proposant un mécanisme de composition entre les widgets qui répondent au modèle de besoins et la tâche principale. Ainsi, l'ensemble peut être généré par transformation d'un seul tenant.

En second temps, nous avons abordé notre proposition du support de la coordination par flux de travaux en section 3.3. Nous nous sommes notamment positionnés sur les aspects suivants :

- **L'application des flux de travaux sur les tâches** (section 3.3.1), en considérant l'as-

sociation des modèles de tâches aux activités BPMN. Les pré/post-conditions des modèles de tâches déterminent l'avancement dans le flux de travaux qui synchronise l'accès aux tâches suivant les rôles;

- **La génération d'un tableau de bord** (section 3.3.2), en transformant les modèles BPMN en arbre de tâches pour être insérés dans le processus de génération d'IHM. Ce tableau de bord régit la présentation des différentes IHM suivant l'état du flux de travaux et le rôle de l'utilisateur.

Enfin, dans le but d'unifier ces briques de conception, nous avons exposé la manière dont ces éléments s'inscrivent dans une méthodologie de conception de collecticiels en section 3.4. Cette méthodologie reprend les étapes de celles vues dans la littérature en les adaptant à nos propositions du support de la coordination (section 3.4.1) :

1. Le sociogramme de l'activité;
2. Les relations entre tâches et rôles;
3. L'activité (par flux de travaux);
4. Les tâches;
5. Les cibles;

En sortie de ces étapes, une première génération d'IHM a lieu et mène à des étapes propres au support de la conscience de groupe (section 3.4.2) :

6. Un cycle itératif impliquant l'utilisateur
 - 6.1. L'identification des besoins en conscience de groupe ;
 - 6.2. La validation des IHM générées pour répondre à ces besoins.

Nous proposons ce cycle itératif afin d'identifier des besoins difficilement exprimables sans une connaissance du domaine métier visé. À travers une série de questionnaires, les utilisateurs finaux sont impliqués dans le processus qui permettra d'établir des modèles de besoins en conscience de groupe, et donc de présenter des widgets en support suivant les situations identifiées.

Ces différents éléments de proposition trouvent une instance dans un cas d'étude au chapitre suivant. Cette instanciation des concepts a ainsi pour but pour valider nos propositions.

Synthèse de la proposition

En synthèse de ce chapitre, nous proposons un cadre de conception et de génération d'IHM adaptatives uniformisé avec un support à la coordination. Nous reprenons sa structure détaillée en figure 3.22.

Ce cadre **s'articule** ainsi autour des propositions suivantes en vue de répondre à nos hypothèses de recherche (voir page 50) :

1. **Des métamodèles adaptés à la modélisation de l'activité collective** qui permettent de définir les organisations, les acteurs, les tâches, l'activité, les ressources, les outils et le domaine. Ces éléments constituent la base du processus de génération des IHM en fournissant des paramètres **d'adaptation** adéquats au contexte de l'activité collective ;
2. **La prise en compte de la coordination** qui passe par des modèles **d'abstraction** qui **s'insère** dans le processus de génération afin de supporter les éléments suivants :
 - **Le support de la conscience de groupe**, via une approche par widgets et l'abstraction des besoins en information de conscience de groupe. En définissant un modèle de besoins qui cible une situation et une tâche principale, nous proposons **l'usage d'un** dépôt de widgets qui a pour charge de fournir une réponse aux besoins exprimés. Cette réponse est matérialisée par la composition des arbres de tâches des widgets et celui de la tâche principale pour ensuite être transformés en une IHM unique ;
 - **Le support des flux de travaux**, de par une association entre les modèles de tâches et les activités **d'un** modèle BPMN pour la synchronisation et le pilotage des flux de travaux. Le modèle BPMN est par ailleurs transformé en modèle de tâches décrivant ainsi un tableau de bord qui régit **l'accès** aux IHM générées en accord avec l'état du flux de travaux à **l'exécution**.
3. **L'intégration de ces éléments dans une méthodologie de conception** qui détermine la manière dont sont construits les modèles précédents. Une attention particulière est portée sur le support de la conscience de groupe en introduisant un cycle itératif de modélisation des besoins en informations sur le groupe en impliquant les utilisateurs.

Le chapitre suivant **s'affaire** ainsi à la validation de cette proposition, notamment centrée sur les hypothèses émises.

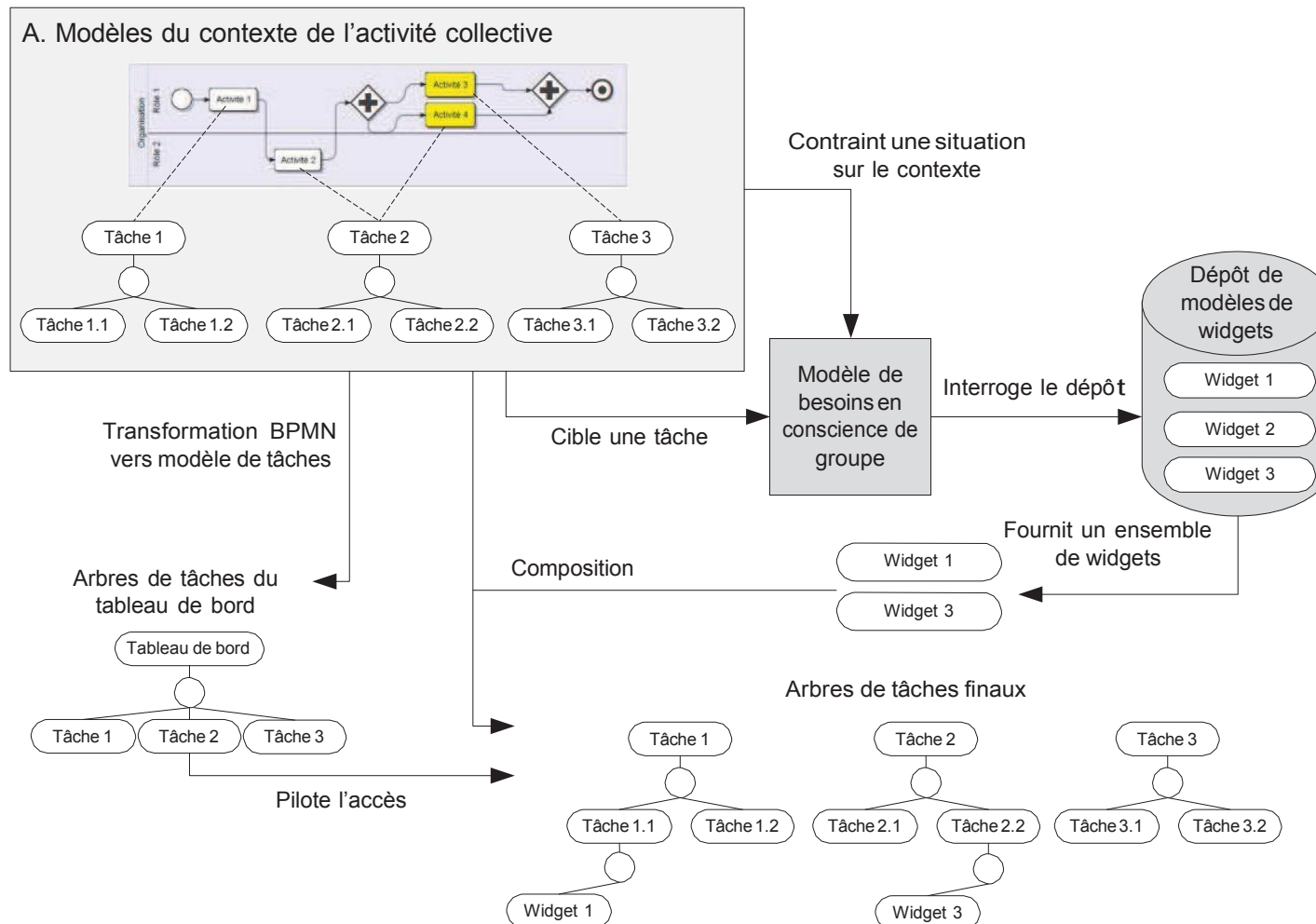


Figure 3.22 – Résumé de la structure du cadre de modélisation et de génération d'IHM proposé.

Chapitre 4

Instanciation des concepts via la suite d'outils ADAGIOS sur un cas d'étude

Sommaire

4.1	La suite d'outils ADAGIOS	161
4.1.1	Le générateur et interpréteur d'IHM de GENIUS	162
4.1.2	Extensions d'ADAGIOS à GENIUS	165
4.2	Cas d'étude : le secteur AIC	173
4.2.1	Détail du secteur	173
4.2.2	Le scénario de la réunion de chantier	176
4.2.3	Application de la méthodologie	178
4.3	Expérimentation sur les besoins en conscience de groupe . .	189
4.3.1	Protocole de test	189
4.3.2	Synthèse des résultats de l'expérimentation	197
4.4	Conclusion de chapitre	211

Au chapitre précédent, nous avons exposé nos propositions vis-à-vis de notre problématique. Pour rappel, leur but est de tendre vers une approche par MBUI pour la génération d'IHM adaptatives venant en support à la coordination.

Pour cela, nous avons proposé une architecture MBUI appuyée d'une méthodologie de conception sur base d'UsiXML [70] et TOUCHE [88, 87, 89], en y apportant notre vue sur le contexte de l'activité collective, la prise en compte de la conscience de groupe et des flux de travaux pour venir en support à la coordination. Dans nos travaux, nous avons notamment émis des propositions vis-à-vis des points suivants, relatifs aux hypothèses formulées (voir page 50) :

- des métamodèles pour la description du contexte de l'activité collective ;
- un processus de transformation pour la génération d'IHM adaptatives considérant la conscience de groupe (par composition de widgets) et les flux de travaux (par génération d'un tableau de bord) ;

- une méthodologie de conception héritée de TOUCHE avec une emphase sur l'abstraction de la conscience de groupe et une implication des utilisateurs.

Il est dès lors nécessaire d'évaluer ces aspects de proposition afin d'en déterminer leur validité et leur applicabilité par rapport à la problématique initiale.

Concernant la démarche de validation (synthétisée en table 4.1), les aspects de modélisation requièrent une validation en deux temps, tout d'abord conceptuelle (i.e. la capacité des métamodèles à décrire les concepts étudiés), puis technique (i.e. la capacité de générer des IHM tenant compte des éléments modélisés). Pour cela, les aspects techniques sont instanciés par un démonstrateur qui sert alors de socle à l'application des métamodèles sur un cas d'étude pour une validation conceptuelle. Enfin, concernant la méthodologie de conception, celle-ci tire parti du cas d'étude pour être validée. Nos propositions relatives à la conscience de groupe (i.e. le volet conceptuel de l'hypothèse H2 et l'étape 6 de la méthodologie) qui ne peuvent être validées par seul le cas d'étude, feront l'objet d'une expérimentation impliquant des utilisateurs afin de dérouler l'étape 6.

Table 4.1 – Résumé de la démarche de validation des hypothèses de recherche (voir page 50).

Hypothèses	Aspects / couverture	Validation
H1	Technique	Démonstrateur, section 4.1
	Conceptuel	Cas d'étude, section 4.2
H2	Technique	Démonstrateur, section 4.1
	Conceptuel	Expérimentation, section 4.3
H3	Étape 1 à 5	Cas d'étude, section 4.2
	Étape 6	Expérimentation, section 4.3

Structuration du chapitre

Dans ce chapitre, nous nous attelons donc à instancier ces différents concepts pour validation. Pour ce faire, nous présentons tout d'abord une suite d'outils qui est conforme aux métamodèles et à l'architecture de notre proposition. Cette suite d'outils se nomme ADAGIOS (*frAmework for moDeling And Generation of adaptlve ui featuring cOordination Support*) et permet la modélisation et la génération d'IHM adaptatives selon nos propositions afin de valider les aspects techniques par démonstration. En section 4.1, nous abordons ADAGIOS en présentant en premier lieu un générateur d'IHM que nous avons utilisé et issu d'un autre projet, à savoir GENIUS [104].

Nous évaluerons l'applicabilité de cette suite d'outils dans la méthodologie de conception proposée en la déroulant à travers un cas d'étude. Ce cas d'étude, issu du domaine métier de l'Architecture, Ingénierie et Construction (AIC), permet à la fois de valider la capacité des métamodèles à supporter un cas concret et le bon déroulement de la méthodologie. En section 4.2, nous présentons ce cas d'étude et un scénario plus précis pour dérouler les étapes 1 à 5 de la méthodologie (présentée en section 3.4).

L'étape 6 de cette méthodologie, celle consistant à identifier les besoins des utilisateurs en conscience de groupe, est celle faisant l'objet des plus fortes propositions par rapport

aux autres étapes. En effet, nous proposons ici une étape itérative impliquant les utilisateurs sur base de nos métamodèles de besoins en conscience de groupe. Sa validation fera l'objet de la section 4.3 à travers une expérimentation avec des utilisateurs finaux. Le but de cette expérimentation est d'évaluer la pertinence de nos choix technologiques et méthodologiques dans la considération de la conscience de groupe.

4.1 La suite d'outils ADAGIOS

Dans cette section, nous nous concentrons sur la validation du processus de génération des IHM, c'est-à-dire la partie technique de nos travaux à travers un démonstrateur. Pour cela, nous avons instancié les concepts de notre proposition à travers la suite d'outils ADAGIOS. Ces outils ont été conçus dans le but de supporter la modélisation du contexte de l'activité collective suivant les métamodèles définis au chapitre précédent et de supporter le déroulement d'une méthodologie de conception. ADAGIOS permet ensuite de générer les modèles de CUI des IHM et propose un environnement d'exécution interprétant ces modèles. Ainsi, cette suite d'outils repose sur les moyens suivants :

Un ensemble d'éditeurs graphiques :

- de modèles de la collaboration (pour supporter les étapes 1 et 2 de la méthodologie) ;
- de modèles de flux de travaux BPMN (pour supporter l'étape 3) ;
- de modèles de tâches (pour supporter l'étape 4), et donc aussi de widgets (pour supporter l'étape 6) ;
- de modèles de cibles (pour supporter l'étape 5) ;
- de modèles de besoins en conscience de groupe (pour supporter l'étape 6) ;

Un environnement technique, basé sur un projet existant (GENIUS [104]) :

- pour la génération des CUI à travers une chaîne de transformations de modèles ;
- pour l'exécution des IHM à travers une interprétation des modèles de CUI dans un environnement Web.

Pour donner corps à ces moyens, cette suite d'outils a été réalisée au sein de *Eclipse Modeling Framework*³⁹ (EMF) qui est un environnement pour l'IDM. Il permet notamment la définition de métamodèles et la génération d'outils tels que des éditeurs de modèles conformes. Le tout est supporté par le moteur de transformation *ATL Transformation Language*⁴⁰ (ATL). Ces choix ont été fait dans le cadre d'un projet du Centre de Recherche Public Henri Tudor, à savoir le projet GENIUS [104] que nous abordons en section suivante (4.1.1). Pour cause, nous reprenons de GENIUS sa partie technique (i.e. son générateur et interpréteur d'IHM) qui est le fruit de travaux mutualisés avec l'équipe du projet. En section 4.1.2, nous mettons en avant comment nos propositions trouvent leur place sur ce socle technique afin de proposer un ensemble cohérent d'outils.

39. <http://www.eclipse.org/modeling/emf/>

40. <http://www.eclipse.org/atl/>

4.1.1 Le générateur et interpréteur d'IHM de GENIUS

Pour la génération et l'exécution des IHM, ADAGIOS se base sur le fruit d'un autre projet qui est le projet GENIUS⁴¹ [104] (2010-2012). Il s'agit d'un travail de recherche mené au sein du Centre de Recherche Public Henri Tudor (financé par le Fonds National de la Recherche⁴²) et qui a servi de cadre global à ces présents travaux de thèse. L'objectif du projet était de traiter du sujet de l'utilisabilité des IHM dans le cadre de la génération de celles-ci par l'IDM.

La proposition de GENIUS est de capitaliser le savoir des ergonomes dans les transformations de modèles [104]. Pour ce faire, les transformations se voient associer un modèle d'utilisabilité spécifiant l'impact (positif ou négatif) d'une transformation sur des critères ergonomiques tels que ceux établis par Bastien & Scapin [94]. Le but est alors de proposer un ensemble de transformations ayant divers impacts sur la génération d'une IHM et de pouvoir choisir celles qui génèrent une IHM de qualité optimale suivant les critères ergonomiques que l'on souhaite privilégier (sans pour autant impacter trop négativement les autres critères). Ces choix de règles de transformations s'opèrent via un cycle itératif de conception impliquant des utilisateurs.

Ces présents travaux de thèse ont été menés au sein de l'équipe du projet GENIUS ayant pour ligne de recherche l'ingénierie des IHM. Ceci notamment dans le but de mutualiser le socle technique des propositions respectives qui sont en lien avec la génération d'IHM suivant le cadre de référence CAMELEON [21, 113, 23].

Nous nous intéressons donc ici uniquement à la base technologique du projet qui trouve une instance dans notre proposition, c'est-à-dire au générateur d'IHM et à l'environnement d'exécution associé.

4.1.1.1 Chaîne de transformation de GENIUS

Afin de générer des IHM, GENIUS repose sur le cadre CAMELEON [21, 113, 23] en proposant de transformer des modèles de tâches en modèles de CUI. Le métamodèle de tâches de GENIUS fait écho à celui que nous avons avancé au chapitre précédent concernant notre choix d'un métamodèle de tâches (voir section 3.1.3, page 121). Ainsi, GENIUS repose sur un métamodèle de tâches intégrant les aspects d'AUI. Par ailleurs, le métamodèle du domaine est lui aussi intégré au métamodèle de tâches, donnant lieu à un métamodèle unique d'entrée nommé *Task & Domain Abstraction* (TDA).

Un modèle de TDA est ainsi transformé en trois modèles distincts visant à fournir une sémantique d'exécution de l'IHM (schématisé en figure 4.1) :

Un modèle de CUI qui décrit la présentation, les composants de l'IHM et la navigation ;

Un modèle de données qui est ici une instance du modèle du domaine dans un formalisme plus aisément exploitable à l'exécution ;

Un modèle d'automate fini (i.e diagramme d'états-transitions UML) qui vient en complément du modèle de CUI pour résoudre des problématiques liées à l'exécution. En

41. <http://www.tudor.lu/fr/projets/genius/>

42. <http://www.fnr.lu/>

effet, comme nous l'avons souligné en section 2.2.2 (page 87), un modèle de tâches induit des transitions non explicites (e.g. retours, annulations, répétitions, etc.) **qu'il** est nécessaire de **connaître** à l'exécution. La génération **d'un** automate fini raffine notamment les liens de transitions qui peuvent exister entre les tâches, tels que les retours arrières ou interruptions. Il **s'agit** en fait **d'une** représentation enrichie **d'un** arbre de tâches plus appropriée à l'exécution de l'IHM. Un modèle **d'automate** fini permet également **d'aisément** préserver ou **connaître** l'état **d'une** IHM vis-à-vis de sa tâche parente, permettant interruption, reprise ou annulation sur base **d'un** modèle. Cet automate pilote l'état de la CUI et ses transitions possibles. Il permet également **d'avoir** une traçabilité des éléments **d'un** modèle de tâches transformés en CUI en faisant office de *mapping* entre ces deux modèles. Il est ainsi possible de remonter au modèle de tâches à partir de l'exécution de l'IHM à des fins **d'analyses**. La figure 4.2 donne un exemple d'automate fini généré depuis un arbre de tâches. Cette démarche fait écho aux travaux sur CTML [126, 127, 39], un métamodèle de tâches décrivant également des automates finis pour leur exécution.

Vis-à-vis de nos travaux, GENIUS a été adapté aux métamodèles **d'ADAGIOS** tout en préservant les mêmes heuristiques de transformations afin de bénéficier des avancées effectuées sur le premier projet.

Ainsi, ADAGIOS repose sur ses propres métamodèles tels que présentés en chapitre précédent. La différence avec GENIUS se note au niveau du métamodèle de tâches qui dans notre cas spécifie en sus des pré/post-conditions sur les tâches qui délimitent les conditions de transitions entre celles-ci. Cette notion nous est utile dans la gestion des flux de travaux et de l'état des tâches que nous abordons en section suivante. De plus, ADAGIOS dispose **d'un** modèle du domaine dédié, lui aussi transformé en modèle de données suivant les mêmes heuristiques. Les données générés sont toutefois enrichies par le métamodèle de la collaboration qui apporte au domaine la spécification des utilisateurs et des organisations qui dans le cas **d'ADAGIOS** sont des concepts manipulables par les tâches.

En sortie de cette génération, nous disposons du même ensemble de modèles (i.e. CUI, modèle de données et automate fini) qui spécifie la sémantique **d'exécution** de l'IHM. Cette séparation peut être comparée à une approche par Modèle-Vue-Contrôleur (MVC) où le modèle serait les données, la vue serait la CUI et le contrôleur pris en charge par l'**automate** fini. La mise en application de ces modèles de sortie est quant à elle traitée en section suivante.

4.1.1.2 Interprétation des modèles de CUI à l'exécution

Lors de notre introduction à la génération **d'IHM** suivant le cadre CAMELEON [21, 113, 23] (voir section 1.3.2.2, page 42), nous avons vu que plusieurs approches se distinguent quant à l'exécution des IHM. **D'un** côté, il est possible de transformer les modèles de CUI en modèles de FUI, ou plutôt **d'engénérer** le code de l'**application** directement dans l'**espace** technique des cibles choisies (e.g. générer du code Java pour mobile correspondant à la CUI). De **l'autre** côté, l'**approche** adoptée par GENIUS, et donc ADAGIOS, est **d'interpréter** à l'exécution les modèles de CUI, accompagnés dans notre cas des modèles **d'automates** finis et des modèles de données (issus du domaine et de l'**organisation**).

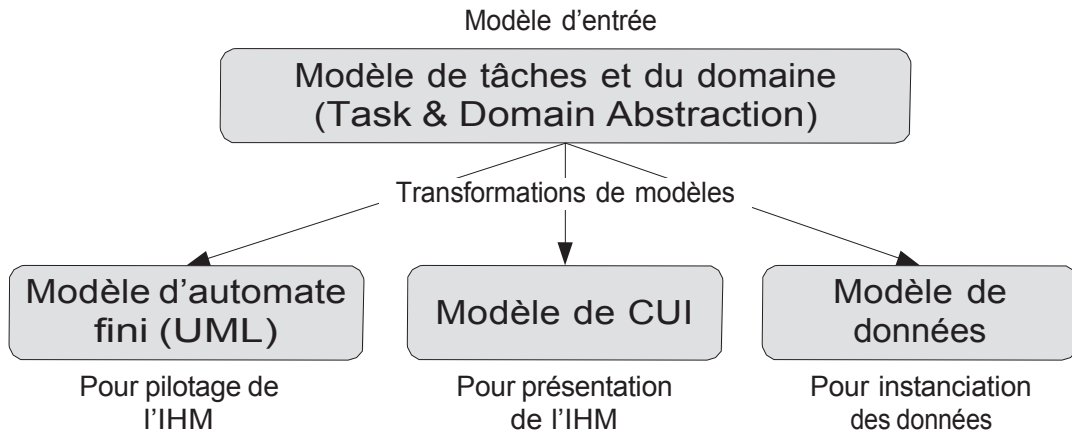


Figure 4.1 – Schéma des transformations issues de GENIUS.

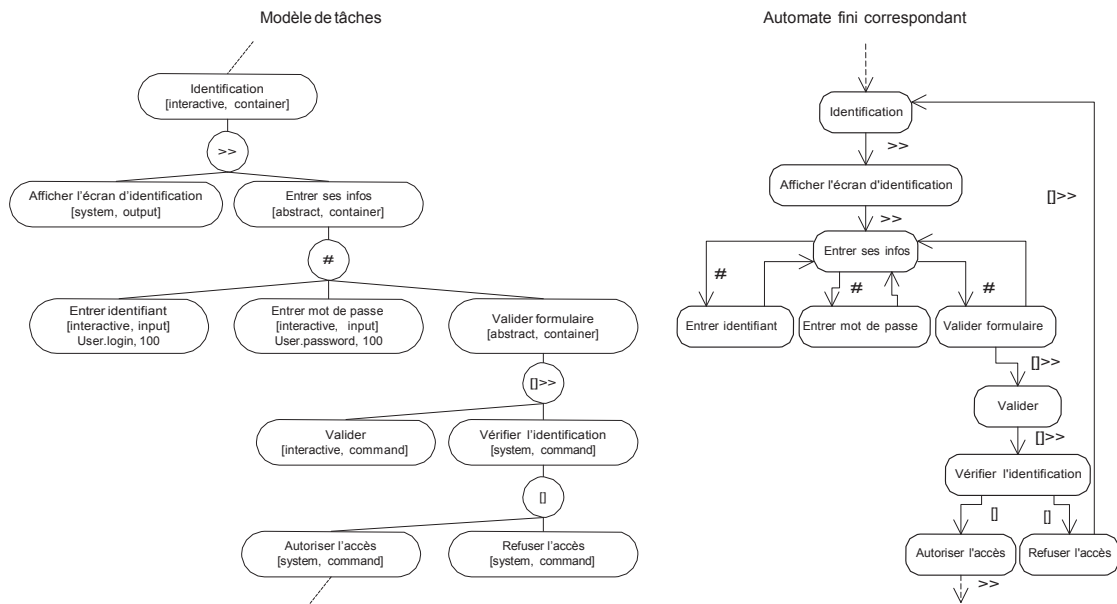


Figure 4.2 – Exemple d'un automate fini UML issu d'une transformation d'un arbre de tâches dans GENIUS.

Reposer sur l'interprétation des modèles à l'exécution permet aux IHM d'être indépendantes de l'espace technique de l'environnement cible, sous réserve que l'interpréteur en question abstraie ces espaces techniques. Ce qui nous intéresse davantage dans l'interprétation des modèles est la possibilité de maintenir ces modèles « à chaud » lors de l'exécution, c'est-à-dire de pouvoir les modifier et de permettre à ces changements d'être observés directement par l'interpréteur. Ceci permet notamment de disposer d'un socle technologique intéressant pour la mise en place d'une adaptation fonctionnelle en temps-réel.

Techniquement parlant, l'exécution des IHM est faite suivant une architecture MVC avec comme point d'entrée les modèles de données (les « modèles » du schéma MVC), les modèles de CUI (les vues) et les automates finis UML (les contrôleurs). Cette fonction est remplie par un *interpréteur* qui cible un espace technologique web. Son rôle est alors de donner forme aux CUI dans un environnement HTML, Javascript et CSS, c'est-à-dire un moteur de rendu web tel que WebKit⁴³. Le choix d'un environnement web permet de cibler une grande partie des appareils existants via un seul espace technique dans la mesure où les moteurs de rendu web sont largement répandus (du PC au smartphone, en passant par les télévisions modernes, boîtiers internet et montres intelligentes).

Pour interpréter ce modèle MVC, GENIUS fait appel à un mécanisme de *templating*⁴⁴ (qui peut se voir comme une forme de transformation de CUI vers FUI) issu de la bibliothèque Javascript JsRender⁴⁵. L'application d'un *template*, et donc la présentation d'une IHM finale à l'utilisateur, est pilotée par l'état de l'automate fini interprété en Javascript. Celui-ci détermine quelle fenêtre de la CUI doit être présentée et ainsi transformée en HTML par templating. Lors de cette opération, un lien est établi avec les données (sérialisées en JSON) représentant la manipulation du domaine. Enfin, l'IHM finale en HTML renvoie son état à l'automate fini pour mettre à jour le système. Ces mécanismes sont schématisés en figure 4.3, alors que la figure 4.4 illustre le résultat à l'interprétation.

4.1.2 Extensions d'ADAGIOS à GENIUS

Nous venons de voir en section précédente dans quelle mesure la base technique du projet GENIUS nous permet de générer et exécuter des IHM suivant notre proposition. Néanmoins, vis-à-vis de la prise en compte des autres facettes de notre proposition, GENIUS ne couvre pas les notions d'adaptation des IHM et de composition de widgets. Outre l'usage d'un modèle de tâches dédié à ADAGIOS que nous avons décrit en section précédente, il convient à ce cadre technique de supporter nos propositions en termes d'adaptation des IHM et du support de la coordination (i.e. widgets pour la conscience de groupe et tableau de bord pour les flux de travaux).

43. <https://www.webkit.org/>

44. Un *web template system* ou *templating* est un mécanisme qui permet de séparer la présentation d'une page web de la logique et des données. L'intérêt du templating est de factoriser la présentation de certains éléments d'IHM et de faciliter leur maintenance. Dans notre cas, l'usage est détourné pour donner une représentation web aux composants des CUI.

45. <http://www.jsviews.com/#jsrender>



Figure 4.3 – Schéma de l'environnement d'exécution des modèles de CUI via ADAGIOS (sur base de GENIUS).

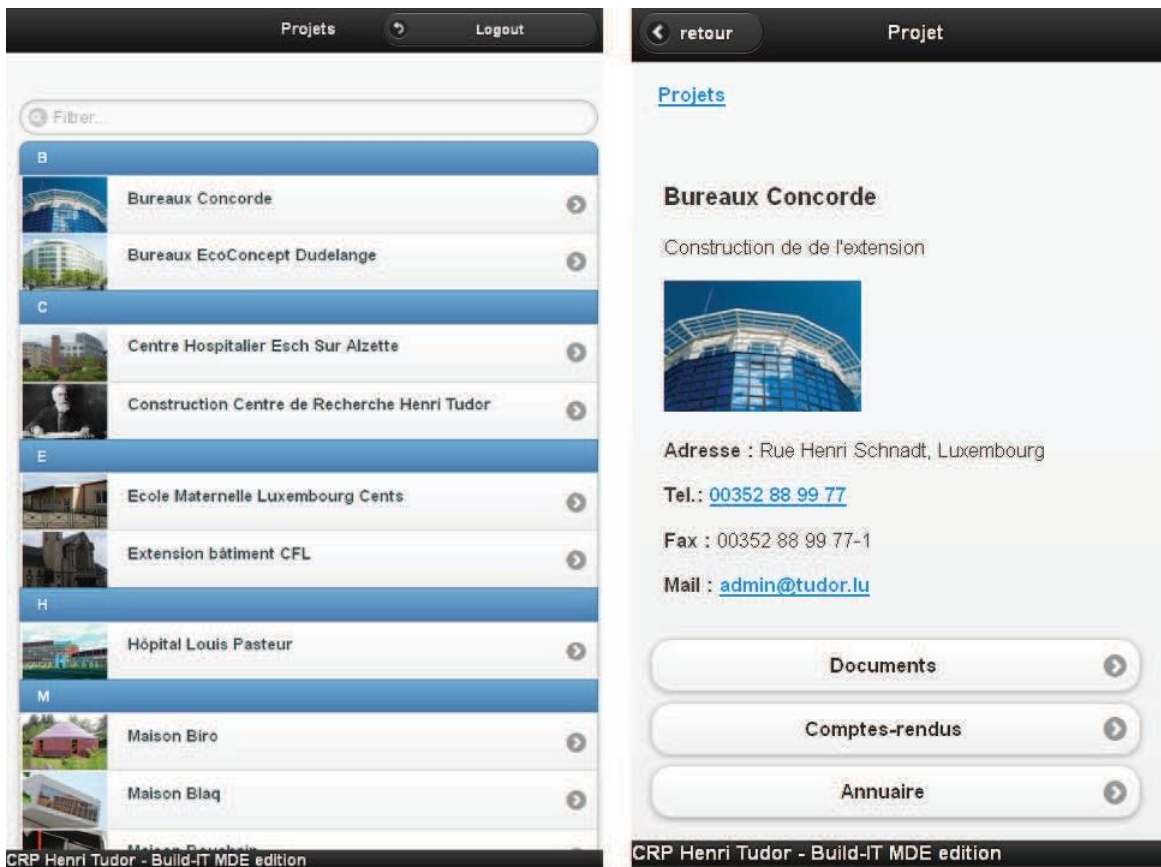


Figure 4.4 – Exemple d'IHM générées et interprétées via GENIUS.

4.1.2.1 Mécanismes d'adaptation

Les travaux originaux de GENIUS ne portant pas sur **l'adaptation** des IHM, son cadre technique a été étendu pour les besoins **d'ADAGIOS**. Pour rappel, nous avons exposé en section 3.1.4 (page 123) notre positionnement dans **l'espace problème de l'adaptation** (i.e. plasticité des IHM). Nous nous positionnons dans une adaptation à la cible (i.e. les environnements matériel, physique et utilisateur) et dans une adaptation à **l'activité** (i.e. rôles et état du flux de travaux).

L'adaptation à l'environnement matériel est en partie prise en compte par **l'interprétation** des modèles faite à **l'exécution** par la base de GENIUS. Comme souligné précédemment, ces modèles permettent une indépendance de **l'espace technologique cible** et les IHM sont alors adaptables à tout environnement logiciel. ADAGIOS apporte une distinction plus spécifique sur le matériel cible en appliquant des templates spécifiques à certains types de matériels. Nous distinguons notamment trois grands types : les PC, les smartphones et les tablettes. Ces types sont discriminés par la surface **d'affichage** disponible et le paradigme de pointage (i.e. tactile ou souris). Par exemple, une interface tactile donne lieu à des composants adaptés à ce paradigme (e.g. plus gros, pas **d'interaction** au survol telles que des infobulles, etc.) et un usage sur smartphone réduit **l'affichage** à une colonne de contenu.

Cette adaptation de la présentation est opérée directement à **l'interprétation** de la CUI à **l'aide d'heuristiques** mises en place grâce aux *CSS Media Queries*⁴⁶, une fonctionnalité du langage CSS permettant **d'appliquer** des attributs de style différents suivant des attributs matériels tels que la taille d'écran, **l'orientation** ou le support des couleurs du périphérique **d'affichage**.

Ce principe est également utilisé pour mettre en application **l'adaptation** de la présentation en fonction du degré **d'importance** donné aux concepts du domaine dans les modèles de tâches. Nous avons vu en section 3.1.4 (page 123) que les interactions manipulant des concepts du domaine peuvent être annotées par un attribut **d'importance** qui détermine à quel degré un concept est critique pour la réalisation de la tâche. Les concepts à **l'importance** moindre sont alors candidats à une adaptation de leur présentation, jusqu'à leur occultation complète. La figure 4.5 illustre le résultat **d'une** IHM adaptée via ADAGIOS.

L'adaptation à l'environnement physique est effectuée à travers **l'API** de géolocalisation introduite par HTML5. Il **s'agit** du seul paramètre de **l'environnement physique** qui a été mis en **œuvre** au sein **d'ADAGIOS**, à défaut de moyens adéquats pour la captation **d'autres** paramètres physiques dans un espace technologique web (e.g. luminosité, bruit ambiant, etc.). La géolocalisation permet toutefois de différencier des cibles du point de vue de **l'environnement physique**.

L'interprétation des modèles à **l'exécution** permet également de détecter tout changement dans les modèles du contexte de **l'activité collective**, notamment dans le flux de travaux, les utilisateurs et **l'organisation**. Ceci permet une adaptation à **l'environnement de l'utilisateur** et à **l'activité**. Ainsi, si un changement a lieu dans **l'un** de ces modèles, il est dès lors possible **d'adapter** fonctionnellement **l'IHM** (e.g. en fonction des changements de rôles, de **l'organisation** ou de l'évolution du flux de travaux). Ce mécanisme **n'est pas** directement pris en compte au niveau de **l'interpréteur** de la CUI, mais passe par un

46. <http://www.w3.org/TR/css3-mediaqueries/>

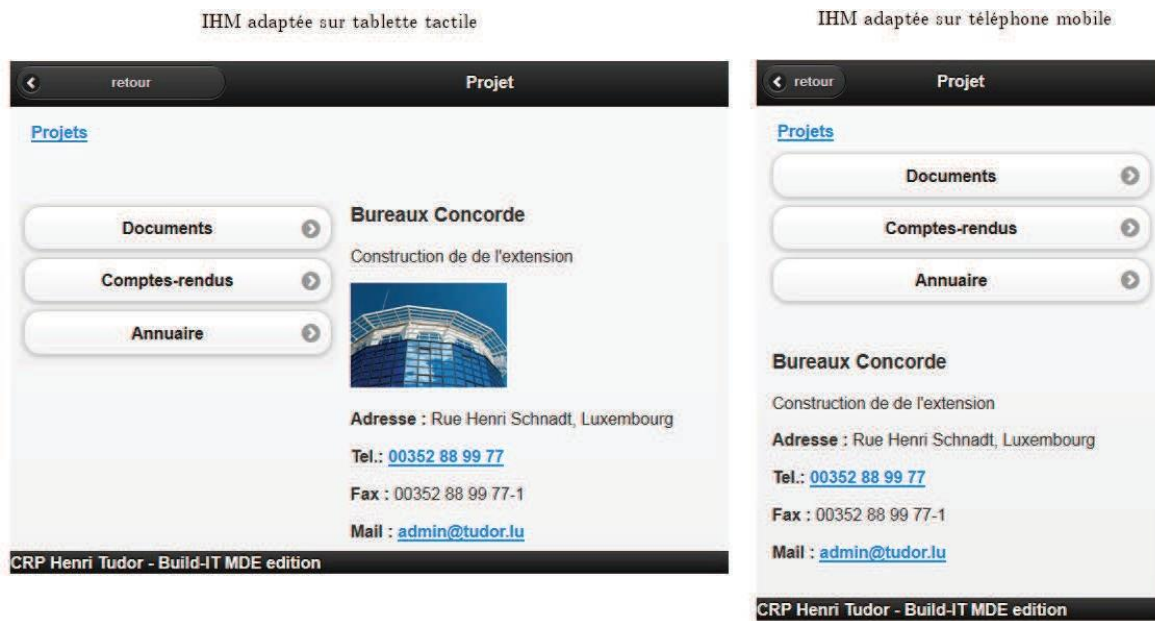


Figure 4.5 – Exemple de l'adaptation d'une IHM générée. Lors de l'adaptation vers téléphone mobile, l'IHM passe dans un sens de lecture vertical et l'image du chantier disparaît en raison de son importance moindre et de l'espace disponible.

appel à la chaîne de génération qui transforme à nouveau les modèles d'entrée en une CUI adaptée et ré-interprétée à la volée. À ce titre, ADAGIOS introduit une architecture client-serveur en séparant les parties génération et interprétation de CUI. La partie serveur est alors dédiée à la génération des CUI sous l'initiative de clients qui se contentent d'interpréter ces CUI et de signaler tout changement de contexte au serveur. Le principe est de permettre de générer des CUI adaptées pour chaque client (et donc utilisateur) sur la base d'un écosystème de modèles d'entrée unique et partagé.

4.1.2.2 Support de la coordination

Dans notre proposition, nous avons pris position dans le support de la coordination suivant deux axes : la conscience de groupe (via une composition de widgets dédiés) et les flux de travaux (intégrés dans le contexte de l'activité collective et appuyés par la génération d'un tableau de bord). Il convient donc pour ADAGIOS de couvrir ces aspects et dans ce but nous abordons le sujet en trois temps :

- A. Supporter la méthodologie de conception en fournissant des éditeurs de modèles conformes à nos métamodèles, dont des éditeurs pour la modélisation de la collaboration (i.e. l'organisation, les utilisateurs et leur lien avec les tâches), des flux de travaux, des arbres de tâches, du domaine, de cibles et de besoins en conscience de groupe ;
- B. Intégrer l'usage des modèles de besoin en conscience de groupe dans la chaîne de génération des IHM, c'est-à-dire instancier le concept de dépôt de widgets et leur composition avec l'IHM finale ;

C. Intégrer l'**usage** des modèles de flux de travaux, c'est-à-dire les mettre en application à l'**exécution** de l'**IHM** et de générer la CUI **d'un** tableau de bord.

A. Éditeurs de modèles du contexte de l'activité collective

Du point de vue de la méthode de conception, il est important de pouvoir disposer **d'outils** de modélisation adéquats. Dans notre cas, il est nécessaire de supporter les métamodèles que nous proposons, c'est-à-dire la modélisation de la collaboration (suivant la représentation graphique associée), de l'**activité** et des besoins en conscience de groupe. Ces éditeurs **s'appuient** également sur l'**environnement** EMF qui permet de couvrir toute la **chaîne** de l'**IDM**, du métamodèle à l'éditeur de modèles associé pour ensuite être transformé. Un des intérêts dans l'**usage d'EMF** est de pouvoir générer des éditeurs graphiques sur base de métamodèles Ecore⁴⁷ assez aisément. Notons que l'éditeur de flux de travaux BPMN (i.e. modèle de l'**activité**) est un éditeur déjà présent dans l'**environnement** EMF et nous l'**avons** repris tel quel. L'éditeur **d'arbres** de tâches et du domaine, quant à lui, est issu de GENIUS. Nous l'**avons** toutefois enrichi avec les notions qui nous sont propres, c'est-à-dire la propriété **d'importance** sur les tâches et les pré/post-conditions OCL, en produisant en sortie deux modèles distincts de tâches et du domaine (GENIUS intégrant les deux au sein du même métamodèle).

B. Dépôt et composition de modèles de widgets

Les modèles en sortie de ces éditeurs doivent ensuite être pris en compte dans la **chaîne** de transformations qui conduit au trio de modèles de CUI, données et automate fini UML. Pour ce qui est du support de la conscience de groupe, il convient **d'appliquer** les besoins modélisés. Pour rappel, les modèles de besoins disposent de deux composantes : l'**identification** du besoin lui-même (défini par des propriétés de conscience de groupe) et du contexte dans lequel ce besoin existe (exprimé par un ensemble de règles OCL délimitant le contexte). Ces modèles sont pris en charge par un dépôt de widgets conçu en Java. Son rôle est de vérifier les contraintes OCL sur les modèles du contexte (i.e. identification de la situation) et d'établir un ensemble de widgets répondant au besoin. Cette première étape de vérification du contexte est opérée à travers EMF-Query, un outil de requêtage permettant de vérifier la validité de contraintes OCL. Si toutes les contraintes sont satisfaites, le dépôt de widgets renvoie un ensemble de widgets en réponse.

Vis-à-vis de l'**algorithme** de sélection de modèles de widgets, nous avons mis en place un algorithme glouton faisant un choix local optimal. C'est-à-dire que pour un besoin en conscience de groupe donné (e.g. un besoin **d'identification** a une importance de 90), le premier widget répondant de manière minimale à ce besoin (e.g. un widget **d'une** pertinence **d'au** moins 90) est sélectionné si aucun autre widget déjà sélectionné ne couvre ce besoin. Il ne **s'agit** pas **d'un** algorithme donnant une solution globale optimale. Notre but est ici, dans un premier temps, de démontrer une faisabilité technique. De plus, nous **n'avons** pas modélisé suffisamment de widgets pour mettre en évidence des cas limites dans la sélection de widgets. Il **s'agit** bien évidemment **d'une** perspective de recherche afin de fournir un ensemble de widgets couvrant optimal.

47. Ecore est le langage de modélisation **d'EMF**.

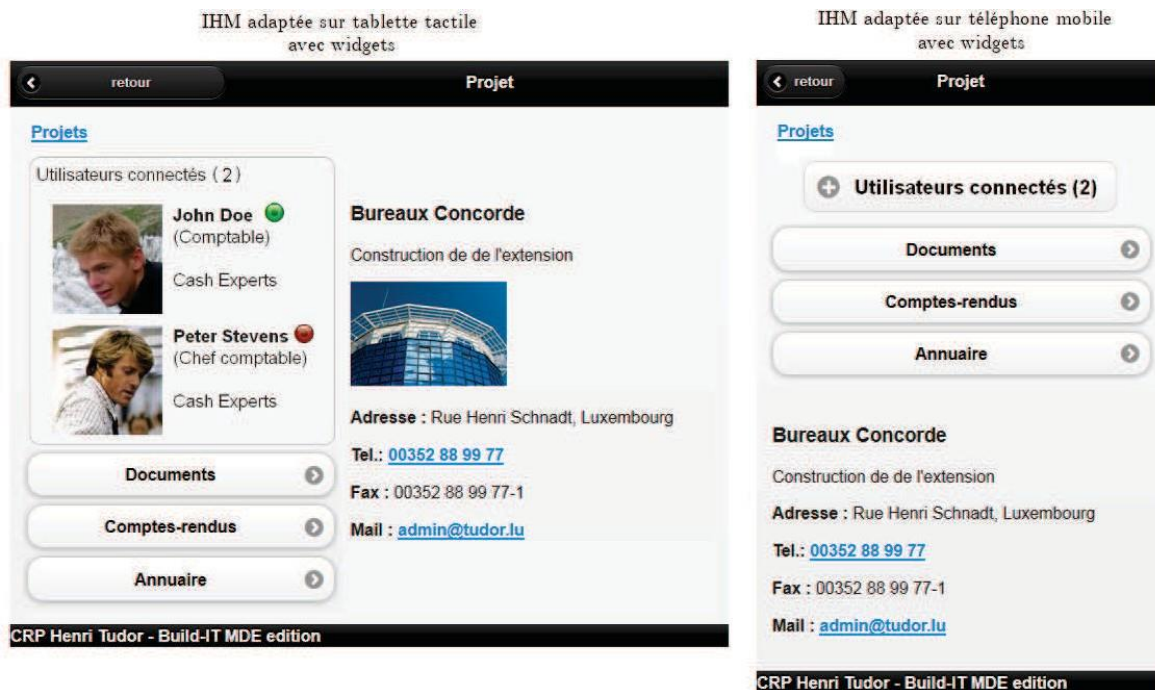


Figure 4.6 – Exemple de l'adaptation de widgets générés. Dans la déclinaison mobile, le widget est replié par défaut et les photos disparaissent.

Une fois cet ensemble de widgets établi, il est nécessaire de l'intégrer par composition dans le modèle de tâches principal. Cette composition est effectuée par transformations de raffinement ATL en deux étapes. Tout d'abord, en paramétrant les widgets avec les concepts du domaine qu'ils doivent manipuler (voir section 3.2.3.3, page 137), puis en raffinant le modèle de tâches principal en y insérant les modèles des widgets suivant l'heuristique d'insertion décrite en section 3.2.3 (page 136). Une fois intégrés au modèle de tâches et après transformation en CUI (suivant les heuristiques de GENIUS), les widgets disposent des mêmes propriétés d'adaptation que le reste de l'IHM. La figure 4.6 illustre l'adaptation d'un widget dans une IHM générée par ADAGIOS.

C. Application les flux de travaux et génération d'un tableau de bord

Enfin, ADAGIOS intègre le support des flux de travaux tel que décrit dans la proposition. Pour ce faire, le modèle de l'activité BPMN est interprété côté serveur où son état est mis à jour suivant la complétion des tâches réalisées côté client. Cette complétion est déterminée par les pré/post-conditions OCL définies sur les tâches et sont vérifiées à nouveau par EMF-Query. Lorsqu'un changement dans l'état du flux de travaux a lieu, un changement de contexte peut être détecté et engendre une re-génération de la CUI qui sera mise à jour côté client.

La notion de tableau de bord est également prise en compte. Pour cela, ADAGIOS introduit dans la chaîne de transformation un passage du modèle BPMN vers un modèle de tâches décrivant un tableau de bord. Le rôle de ce tableau, comme décrit en section 3.3.2 (page 143), est de fournir un accès aux autres tâches sous condition de l'état du flux de

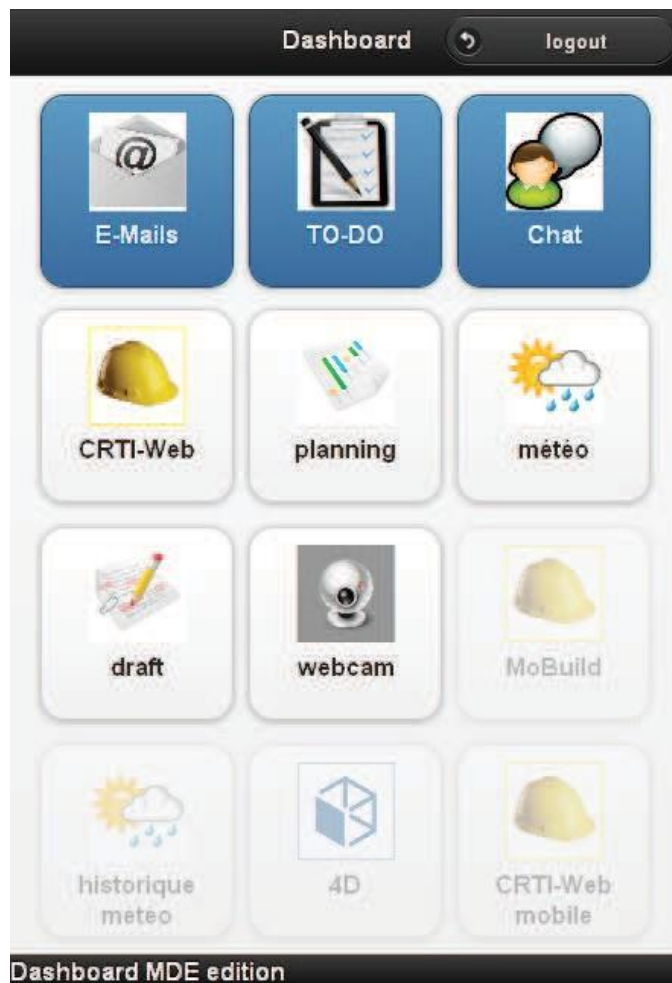


Figure 4.7 – Exemple d'un tableau de bord généré et interprété par ADAGIOS. En bleu, les tâches fixes ne relevant pas du flux de travaux, alors que les tâches grisées ne sont pas accessibles dans l'état courant de l'activité.

travaux. En termes de CUI générée pour ce tableau, nous générons des boutons amenant la navigation vers les CUI des autres tâches. Ces boutons sont actifs ou non suivant l'état du flux de travaux. Nous prenons également en compte les tâches ne relevant d'aucun flux de travaux. Dans une activité, il arrive que des tâches soient ponctuelles et non inscrites de manière formelle dans le flux de l'activité. Ces tâches doivent alors être accessibles à tout instant indépendamment de l'état de l'activité. La figure 4.7 expose un exemple d'IHM de tableau de bord généré par ADAGIOS.

Résumé de section

Pour résumer cette section, nous avons ici instancié techniquement nos propositions via une suite **d'outils** : ADAGIOS. Cette suite **d'outils** propose des éditeurs pour la modélisation du contexte de **l'activité** collective, mais aussi pour la modélisation de besoins en conscience de groupe. Un générateur et interpréteur **d'IHM**, basé sur celui du projet GENIUS [104], permet de produire et exécuter des interfaces agrémentées de widgets pour le support de la conscience de groupe et **d'un** tableau de bord pour la gestion de flux de travaux.

Cette suite **d'outils** fait office de démonstrateur et valide ainsi les aspects techniques de notre proposition ainsi que **l'applicabilité** technique des métamodèles proposés. Nous avons plus particulièrement démontré faisabilité de la génération de widgets sur base de modèles de besoins en conscience de groupe et support des modèles de flux de travaux via la génération **d'un** tableau de bord et de leur interprétation à **l'exécution** pour piloter les IHM.

Concernant les hypothèses émises (voir page 50), cette section nous permet de démontrer la validité technique des hypothèses H1 (i.e. générer des IHM adaptatives sur base de la modélisation du contexte de **l'activité** collective) et H2 (i.e. générer des IHM supportant la conscience de groupe et les flux de travaux), de par la possibilité de générer des IHM y répondant.

4.2 Cas d'étude : le secteur AIC

Nous venons, en section précédente, de traiter de la mise en **œuvre** technique de nos propositions à travers **l'outil** ADAGIOS. Si ce démonstrateur prouve la faisabilité technique de nos propositions et **l'utilisation** des métamodèles dans une **chaîne** de transformation, il convient de valider leur applicabilité dans la méthodologie de conception proposée. En effet, un cadre technique **n'a** de sens que **s'il s'insère** dans une réalité de conception.

Afin **d'attester** de **l'applicabilité** de nos concepts dans une méthodologie de conception, nous présentons en section 4.2.1 un cas d'étude concret dans le domaine métier de **l'Architecture**, de **l'Ingénierie** et de la Construction (AIC). Nous entrerons dans les détails **d'un** scénario particulier en section 4.2.2 pour ensuite dérouler la méthodologie avec ADAGIOS en section 4.2.3 où nous couvrons les étapes 1 à 5, l'étape itérative relative à la conscience de groupe étant traitée plus en détails en section 4.3.

4.2.1 Détail du secteur

Dans leur validation, nos travaux trouvent un cas d'étude concret dans le secteur de **l'Architecture**, de **l'Ingénierie** et de la Construction (AIC)⁴⁸. Il **s'agit** en effet **d'un** secteur complexe qui fait appel à une grande variété **d'acteurs** et **d'activités** qui tiennent place dans des situations hétérogènes. Cette industrie implique la collaboration **d'un** grand nombre de personnes et le maintien **d'un** nombre de documents important de divers types.

Lorsque **l'on** évoque un projet de construction **d'un** bâtiment, on découpe généralement celui-ci en un certain nombre de phases :

1. Les études préalables et la programmation du projet ;
2. Les études principales qui passent par un avant-projet et **l'obtention** des différents permis de construire ;
3. La consultation des entreprises via les marchés publics ;
4. La préparation et la planification du chantier ;
5. Le chantier ;
6. La livraison ;
7. Le cycle de vie de **l'ouvrage**.

Durant ces différentes phases, plusieurs organisations collaborent et celles-ci peuvent revêtir des structures différentes. Par exemple, les relations inter-organisationnelles relèvent de la bureaucratie (voir section 1.2.2, page 20), les organisations en elles-mêmes sont de type hiérarchique ou simple, alors que les chantiers prennent la forme **d'une** structure missionnaire bien que **l'activité** au jour le jour relève de **l'adhocratie**. Il **s'agit** donc **d'un** contexte multi-organisationnel, multi-structurel, certains acteurs appartenant à plusieurs organisations et leur rôle pouvant évoluer au fil du projet (e.g. tel que le rôle de **l'architecte** qui durant la phase d'études participe à l'élaboration technique alors **qu'il** orchestre la réalisation durant le chantier). La figure 4.8 présente les liens entre les parties prenantes **d'un** projet de construction, indépendamment de toute structure sous-jacente.

48. Qui inclut le secteur du **Bâtiment** et Travaux Publics (BTP) mais ne **s'y** limite pas

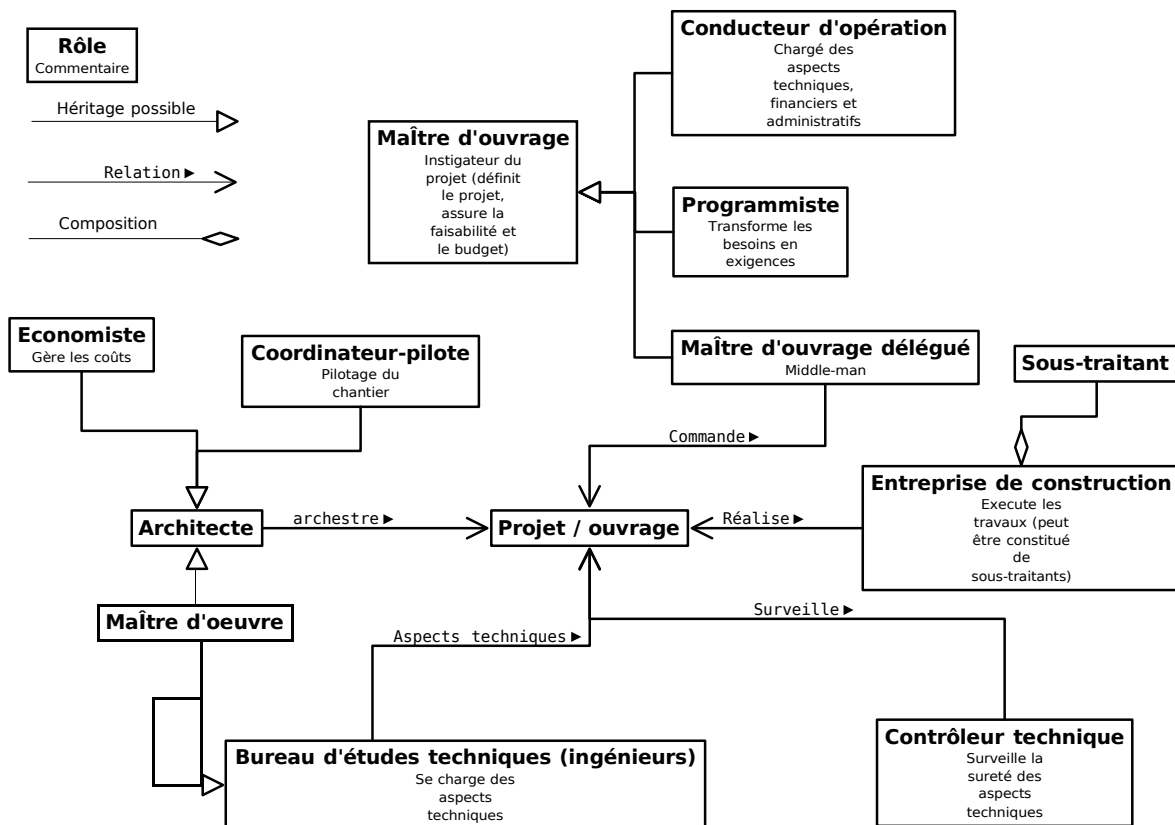


Figure 4.8 – Liens entre les parties prenantes d'un projet de construction.

Pour assurer le bon déroulement de toutes les phases **d'un** projet, la coordination **s'opère** notamment via une standardisation forte des procédés, ceux-ci pouvant être règlementés par des lois, comme la loi de « **Maîtrise d'Ouvrage** Publique⁴⁹ » (propre aux marchés publics) et la standardisation des résultats qui est notamment soutenue par des normes de qualité. La standardisation des procédés permet de mettre en place une coordination stricte entre les acteurs, mais aussi de délimiter leurs missions, un point important dans une industrie où les litiges sont fréquents et peuvent être coûteux (en termes de temps et **d'argent**). Il est donc essentiel de délimiter les responsabilités de chacun et **d'avoir** une traçabilité des décisions.

Cependant, lors de la phase de chantier, la coordination des acteurs ne relève pas nécessairement de procédés. En effet, **l'activité** de construction est parsemée de décisions ou **d'incidents** qui peuvent être réglés rapidement et indépendamment de tous procédés, et donc non-inscrits dans un cycle formel (e.g. corriger une malfaçon, résoudre un retard de livraison, etc.). Il **s'agit** dès lors **d'une** coordination informelle, par ajustement mutuel et ne relevant **d'aucune** structuration particulière (i.e. adhocratie).

Afin de supporter cette activité, les IHM viennent notamment supporter la gestion documentaire, la planification, la modélisation ou la centralisation des données. Cependant, le secteur AIC est un domaine assez conservateur vis-à-vis de **l'adoption** de nouvelles technologies. Les différents acteurs ont des affinités avec les technologies très variées et les environnements (e.g. chantier bruyant et poussiéreux, cabane exigüe, etc.) ne se prêtent pas toujours à **l'usage d'IHM**. Néanmoins, les appareils mobiles tels que les tablettes et smartphones sont de plus en plus adoptés dans **l'activité** de construction [17].

Ainsi, le secteur AIC implique plusieurs degrés de complexité dans son contexte :

Une complexité organisationnelle. Dans le secteur de **l'AIC**, un projet de construction implique une multitude **d'organisations** qui peuvent revêtir des structures variées, notamment suivant les différentes étapes du projet. Elles peuvent être de type hiérarchique ou transverse, voire même adhocratique (voir section 1.2.2). En effet, il est commun dans ce domaine de faire face à des problèmes ponctuels ou de devoir prendre des décisions rapides qui induisent une structure organisationnelle alors peu prédictible en amont. Ce type de structure ne dure généralement que le temps de la résolution du problème durant laquelle les acteurs **s'auto-**organisent ;

Une coordination de tout instant. Ce secteur regroupe une grande variété **d'acteurs** qui doivent collaborer dans ce contexte organisationnel. Certaines activités sont coordonnées par des procédés figés, voire fixés par des lois, mais aussi ponctuellement indépendamment **d'une** structure. Les outils qui viennent en support à la coordination doivent supporter des organisations et procédés précis, tout comme des structures adhocratiques. Ceux-ci, pour autant, proposent généralement des IHM conçues pour un service et une organisation donnés, manquant de flexibilité ;

Des acteurs mobiles. La majeure partie des tâches sous-entend une forte mobilité des acteurs. Les situations sont variées et demandent aux acteurs **d'interagir** sur différents lieux (e.g. chantier, bureau d'études, dans les transports, etc.) qui ont notamment des particularités différentes (e.g. espace disponible, matériel à disposition,

49. Loi n°85-704 du 12 juillet 1985, sur Légifrance : <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000693683&categorieLien=cid>

etc.). **L'utilisation d'appareils** mobiles tels que les smartphones ou tablettes est de plus en plus adoptée. Ceci représente un écosystème **d'appareils** avec lequel les IHM doivent composer ;

Un contexte d'interaction varié. Les triplets **d'environnement** matériel, physique et utilisateur impactent également **l'activité** AIC. Nous avons déjà soulevé **l'hétérogénéité** du matériel et les utilisateurs sont tous très différents, notamment de par leurs rôles et leur affinité avec la technologie. Quant à **l'environnement** physique, son incidence sur **l'activité** porte jusqu'à un facteur de décision. En effet, les conditions ambiantes telles que la météo peuvent être déterminantes dans la compréhension **d'un** événement sur chantier tout comme dans la planification du projet.

Comme toute industrie complexe, le secteur de **l'AIC n'est pas exempt d'incertitudes** et de dysfonctionnements, les projets étant en plus tous différents et les visions à long terme limitées. **C'est** notamment ce type de facteurs qui ont conduit à la standardisation **d'un** maximum de procédés dans **l'activité** (dont la définition de la loi MOP) et un certain nombre de recherches sont encore centrées sur **l'identification** et la limitation des sources de dysfonctionnements, comme le soulignent Shen et al. [101].

Par conséquent, le secteur AIC représente un contexte qui relève bien **d'un** besoin en flexibilité, tant macroscopique (i.e. organisationnelle) que microscopique (i.e. coordination des acteurs). Au long de la section suivante, nous nous attardons plus précisément sur un scénario propre à ce secteur afin **d'instancier** un cas concret pour validation.

4.2.2 Le scénario de la réunion de chantier

Pour concrétiser notre approche, nous nous focalisons sur le scénario particulier du cycle de la réunion de chantier. Lors de la phase de construction **d'un** projet, des suivis administratifs, techniques et financiers ont lieu périodiquement. La réunion de chantier entre dans le cadre de la gestion administrative et technique. Celle-ci est généralement hebdomadaire et se tient sur le chantier dans un lieu temporaire aux commodités réduites (e.g. cabane de chantier pré-fabriquée). Son but consiste à veiller à la bonne réalisation des travaux et à prévenir tout problème. Plus particulièrement, elle permet de vérifier **l'avancement** des travaux, de coordonner **l'ensemble** des corps de métier, de gérer le planning (suivant **l'avancement** et la livraison des matériaux à prévoir, ou les retards en vue), de définir ou valider des choix de matériaux et de gérer les problèmes (détecter/anticiper les malfaçons qui engendrent des surcoûts).

La réunion de chantier **s'organise** autour **d'un** cycle qui comprend les étapes suivantes :

1. **La préparation**, qui consiste à établir **l'ordre** du jour de la réunion en passant en revue les points non clôturés du compte-rendu de la réunion précédente. Ces points sont complétés par un suivi des plannings, des budgets et des demandes des entreprises. Durant cette étape, le coordinateur doit pouvoir accéder aux comptes-rendus précédents et pré-rédiger celui à venir. Toutes autres informations relatives au chantier peuvent être utiles durant cette préparation : planning, documents, plans, historique et prévision météo, webcam du chantier, etc. ;
2. **La visite du chantier**, qui est conduite et animée par le coordinateur (e.g. généralement **l'architecte** dans le cas **d'un** petit chantier, ou un « coordinateur pilote » le cas

échéant) en présence des autres acteurs (e.g. entreprises, ingénieurs, etc.). Elle a lieu juste avant la réunion et consiste à vérifier si les anomalies présentes à l'ordre du jour ont été résolues et si de nouvelles anomalies sont apparues, ainsi qu'à constater l'avancement des travaux. Ici, le coordinateur doit pouvoir accéder à l'ordre du jour, y ajouter des points ou les mettre à jour et documenter les anomalies repérées (e.g. photo, notes, etc.);

3. **La réunion de chantier**, animée par le coordinateur et en présence de tous les acteurs. Elle consiste à passer en revue les points à l'ordre du jour. Durant la réunion, le coordinateur doit pouvoir traiter les points à l'ordre du jour et donc pouvoir le consulter et pré-rédiger les éléments du compte-rendu. Pour animer convenablement la réunion, il doit pouvoir accéder aux documents du chantier, aux plannings ou à toute information utile pour interpréter un événement (e.g. historique météo);
4. **La rédaction du compte-rendu**, qui entérine les points traités, les décisions prises et les actions à entreprendre. Il définit notamment la liste des acteurs présents à la réunion, un historique météorologique pour pouvoir replacer les circonstances du chantier et pré-établit une liste de points à traiter à la prochaine réunion;
5. **La validation des autres acteurs**, qui après lecture du compte-rendu peuvent réagir sur les points de l'ordre du jour en y apportant des commentaires, puis en validant la bonne tenue du rapport. Suite à quoi, les entreprises coordonnent leurs équipes pour la mise en exécution.

Les acteurs qui doivent être présents à ces réunions sont les suivants :

- **Le coordinateur-pilote**, qui est affilié à la maîtrise d'œuvre et est bien souvent l'architecte du projet. C'est lui qui convoque les autres acteurs à la réunion, qui l'anime et qui conduit la visite de chantier. En amont de la réunion, il prépare celle-ci sur base des points non clôturés du compte-rendu de la réunion précédente. Enfin, il rédige le compte-rendu de la réunion qu'il transmet aux autres acteurs pour approbation;
- **Le maître d'ouvrage**, ou plus généralement une personne déléguée qui s'assure de la bonne tenue des travaux et du budget. Il doit lire et approuver le compte-rendu;
- **Les entreprises exécutantes**, c'est-à-dire un représentant privilégié de chaque corps de métier qui coordonne à son tour ses équipes après avoir approuvé le contenu du compte-rendu de la réunion.

Ce scénario met en exergue les rôles variables de certains acteurs, tel que celui de l'architecte qui en phases de pré-chantier contribue à la conception du projet au sein d'une agence d'architecture alors qu'une fois le chantier débuté il endosse le rôle de coordinateur. De plus, plusieurs types de structures organisationnelles entrent en ligne de compte dans ce cycle de réunion :

- **La hiérarchie**, propre à la réunion en elle-même, où le coordinateur supervise les autres acteurs. Elle définit aussi le rapport entre maîtrise d'œuvre (i.e. chargé du projet) et maîtrise d'ouvrage (i.e. propriétaire du projet);
- **La structure missionnaire**, propre à la manière indépendante dont s'organisent les entreprises;
- **L'adhocratie**, durant la vie du chantier et les observations au jour le jour;

Le cycle de la réunion de chantier implique également divers environnements, les phases préparatoires et de rédaction pouvant s'opérer soit en agence d'architecture, soit au bureau

ou au domicile de l'architecte (suivant sa situation) alors que les réunions ont lieu sur le chantier dans un cabanon pré-fabriqu .

D s lors, ce sc nario est un bon exemple de contexte complexe que nous souhaitons prendre en compte   travers notre proposition.

4.2.3 Application de la m thodologie

Dans cette section, nous reprenons les  tapes 1   5 de la m thodologie de conception (voir section 3.4, page 147) en les appliquant au cas d' tude. En la d roulant sur un cas d' tude, nous cherchons   mettre en avant l'expressivit  et les limitations de la proposition en termes de mod lisation. Notre attention se porte donc sur la pertinence des m tamod les et leur insertion dans la m thodologie.

Dans l'illustration de ce cas d' tude, par souci de lisibilit , nous limitons les exemples   un seul projet de construction et aux acteurs en lien avec la t che de gestion de comptes-rendus de r union de chantier. La mod lisation du cas d' tude telle qu'elle a  t  men e dans ces travaux comprenait 21 projets autour de 7 organisations et 27 acteurs.

1. Mod lisation du sociogramme

Cette premi re  tape consiste    tablir le **sociogramme de l'activit **, c'est- -dire les relations entre les organisations ainsi que leur composition. Une organisation est compos e d'**acteurs** (qui peuvent appartenir   diff rentes organisations) affili s   un ou plusieurs r les. Les organisations peuvent  tre associ es les unes aux autres afin de d finir des groupes, des affiliations, des liens particuliers, etc.

Dans notre cas d' tude, nous avons vu que nous consid rions les parties prenantes suivantes : la **ma trise d'ouvrage**, la **ma trise d' uvre** et les entreprises de construction. Ces parties prenantes **s'organisent** autour d'un projet. Afin d'**illustrer** la mod lisation du cas d' tude, nous consid rions ici les organisations et acteurs relatifs   un seul projet (et   la t che de gestion de comptes-rendus), r partis comme suit :

- **Ma trise d'ouvrage**, qui est repr sent e par le **CRP Henri Tudor** et qui comprend l'acteur suivant :
 - **Eva Aufgehen**, qui est **ma tre d'ouvrage d l gu ** au sein de l'organisation ;
- **Ma trise d' uvre**, qui est prise en charge par le **Bureau d' tudes Leborgne** et comprenant l'acteur suivant :
 - **Emmanuel Leborgne**, qui est **architecte** du projet et **coordinateur** du chantier. En tant que coordinateur, il supervise les entreprises de construction par l'interm diaire de leur repr sentant ;
- Les **entreprises de construction**, qui sont ici **Toitures Lamotte** (couvreur) et **Roberto Plumbing Works** (plombier) et qui comprennent :
 - **Alain Lamotte**, qui est le **repr sentant** de Toitures Lamotte ;
 - **Roberto Guy**, qui est le **repr sentant** de Roberto Plumbing Works.

Ces acteurs et organisations sont impliqu s dans un projet de **Nouveau B timent** pour le compte du CRP Henri Tudor. La figure 4.9 met en avant le sociogramme de ce cas d' tude tel que mod lis  au sein d'ADAGIOS. Les organisations et leurs liens sont bien repr sent s et l'**affiliation** des acteurs est telle que explicit e ci-dessus.

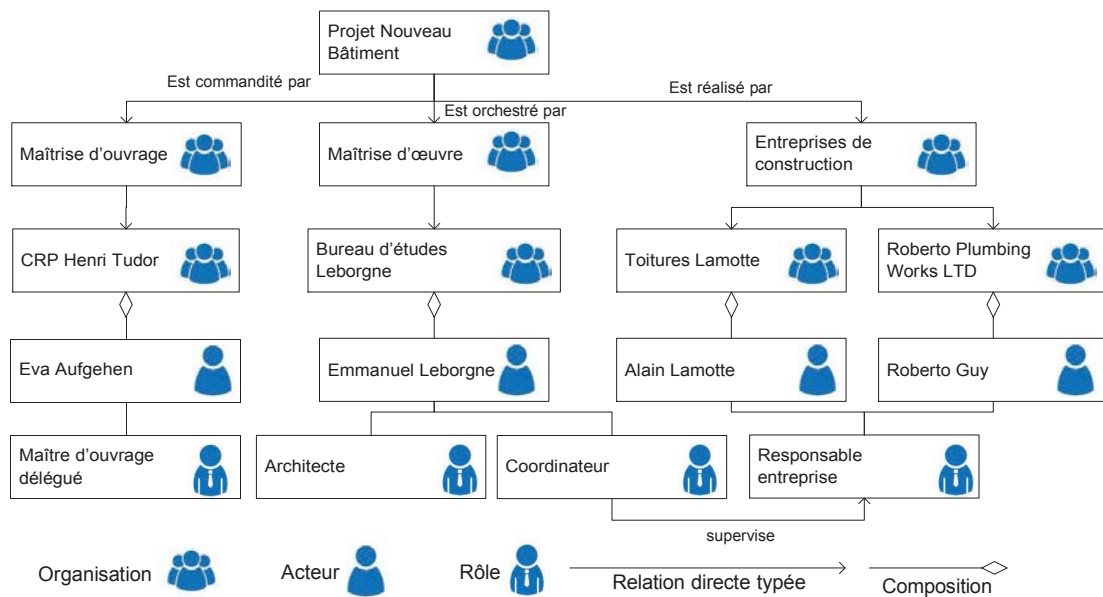


Figure 4.9 – Représentation graphique du modèle de la collaboration pour la définition du sociogramme du cas d'étude.

2. Modélisation des relations rôles-activités

Dans cette seconde étape de modélisation, nous repartons du sociogramme en complétant le modèle de la collaboration avec les liens entre les rôles et les activités à identifier.

Dans le cycle de la réunion de chantier, nous distinguons plusieurs phases : une préparation, une visite du chantier, la réunion en elle-même, la rédaction du compte-rendu puis la validation de celui-ci par les acteurs. Durant ces phases, les différents acteurs doivent réaliser divers activités. Il incombe notamment au coordinateur du chantier de :

- Préparer la réunion de chantier ;
- Conduire la visite du chantier ;
- Rapporter les anomalies pendant la visite ;
- Identifier les points clôturés du compte-rendu précédent durant la visite ;
- Animer la réunion de chantier ;
- Rédiger le compte-rendu ;
- Prendre connaissance des commentaires des autres acteurs sur le compte-rendu.

Le maître d'ouvrage ainsi que les représentants des entreprises de construction doivent quant à eux :

- Lire le compte-rendu de la réunion de chantier ;
- Ajouter des commentaires au compte-rendu, si besoin.

Enfin, les représentants des entreprises de construction doivent *coordonner leurs équipes* une fois le compte-rendu validé.

La figure 4.10 complète le sociogramme en rattachant les différents rôles aux activités nécessaires. À noter que le rôle d'architecte n'apparaît pas ici car les tâches sont attribuées au rôle de coordinateur qui peut éventuellement être l'architecte (voir figure 4.9). Cette

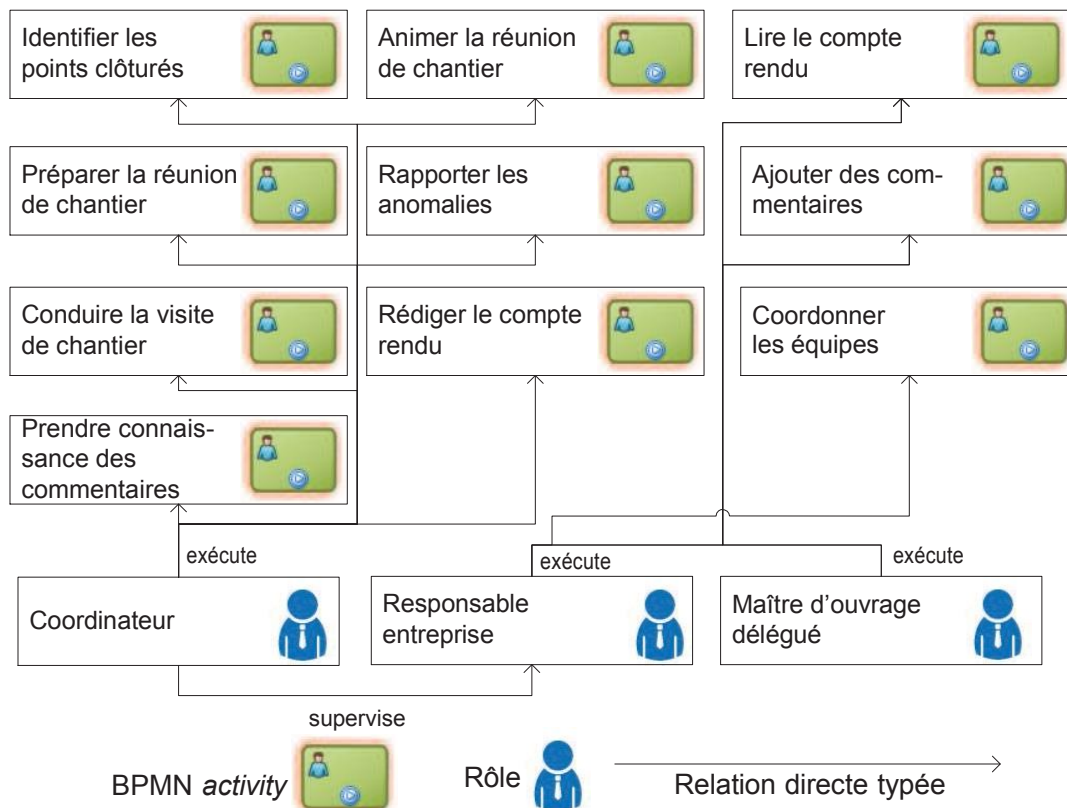


Figure 4.10 – Représentation graphique du modèle de la collaboration pour la définition des relations rôles-activités du cas d'étude. Le sociogramme est éludé par souci de lisibilité.

étape ne présente pas de particularité et permet de mettre en place la base des flux de travaux définis en étape suivante.

3. Modélisation de l'activité (i.e. flux de travaux)

Sur base des activités et des rôles attachés qui ont été identifiés en étape précédente, il convient désormais de construire le flux de travaux qui représente l'activité globale. C'est-à-dire de définir le flux BPMN correspondant au cas d'étude. Les rôles sont ainsi associés à des *pools* BPMN et les activités sont reprises du modèle de collaboration défini. Cette étape consiste alors à détailler le flux avec les dépendances de synchronisation entre toutes les activités. La figure 4.11 expose le flux de travaux résultant de la description du cas d'étude (voir section 4.2.2).

Cette étape consiste également à préparer la suivante en définissant en amont les tâches utilisateurs nécessaires à la réalisation d'une activité et qui donneront lieu à des IHM générées. Pour cela, chaque activité du flux BPMN est associée à une ou plusieurs tâches par annotation. Par exemple, l'activité de « préparation de la réunion de chantier » requiert pour le coordinateur d'avoir accès à certains documents et aux comptes-rendus

précédents. Ainsi, les tâches suivantes sont identifiées :

- Consulter les comptes-rendus ;
- Consulter **l'historique** météo ;
- Consulter le planning du chantier ;
- Consulter les plans du chantier.

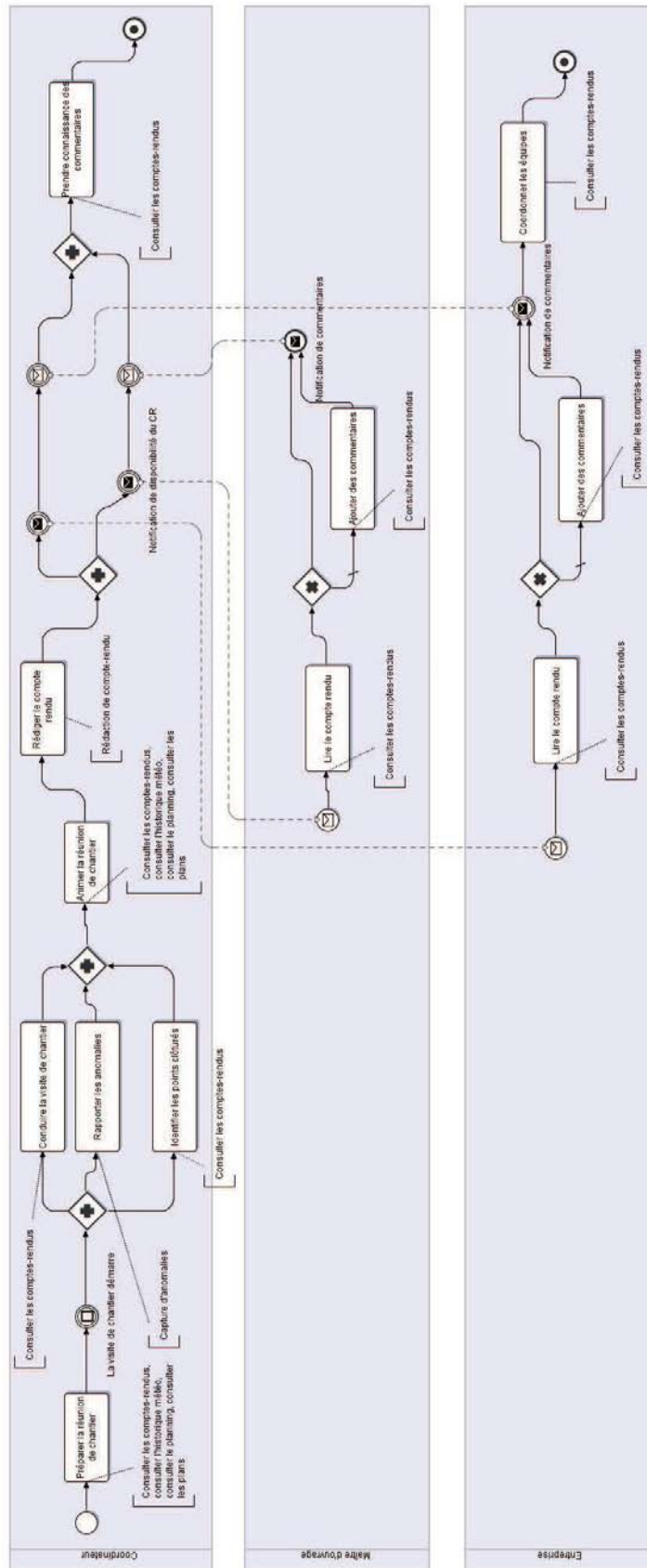


Figure 4.11 – Modèle BPMN de l'activité du cycle de la réunion de chantier.

4. Modélisation des tâches et du domaine

Cette étape consiste à reprendre les tâches identifiées pour chaque activité en étape précédente et à détailler les interactions et manipulation du domaine. Pour cela, il convient de modéliser le domaine et les arbres de tâches qui le manipule.

Vis-à-vis de l'illustration du cas d'étude, nous nous focalisons ici sur la tâche la plus importante, c'est-à-dire la gestion des comptes-rendus qui est reprise par tous les acteurs et qui intervient à plusieurs phases.

Pour ce faire, nous sommes partis **d'une** spécification fonctionnelle existante **d'une** application de gestion de compte-rendu commerciale se nommant CRTI-weB. CRTI-weB est un système de gestion de comptes-rendus de réunion de chantier et de documents divers. Cet outil a été conçu dans le cadre **d'un** projet du CRP Henri Tudor (i.e. le projet Build-IT, mené de 2004 à 2009) et a abouti à un produit dorénavant exploité par **l'industrie**. Il **s'agit** donc **d'un** point de référence fonctionnel éprouvé par les pratiques du secteur. En partant des spécifications fonctionnelles de CRTI-weB, notre but **n'est** pas de comparer une application existante avec un résultat généré par ADAGIOS, mais de servir de base concrète quant aux spécifications **d'une** telle application dans le cas de notre cas d'étude.

Dès lors, nous avons construit le modèle de la tâche « consulter les comptes-rendus » sur base des spécifications fonctionnelles de CRTI-weB. Cette tâche consiste à pouvoir effectuer les actions suivantes :

- **Sélectionner un projet**, parmi la liste des projets auxquels **l'acteur** a accès (suivant ses rôles et affiliations) ;
- **Consulter les documents d'un projet**, qui sont des fichiers téléchargés dans le système et ayant des méta-données (i.e. **detail**) ;
- **Commenter les documents**, ainsi que pouvoir consulter les commentaires des autres acteurs ;
- **Consulter l'annuaire d'un projet**, qui est une liste des acteurs impliqués dans celui-ci ;
- **Consulter les comptes-rendus d'un projet**, qui sont composés **d'observations** ;
- **Consulter la liste des présents à une réunion** ;
- **Consulter les observations d'un compte-rendu** ;
- **Commenter les observations**.

Ainsi, la figure 4.12 expose le modèle du domaine relatif à cette tâche tandis que la figure 4.13 illustre le modèle de tâches.

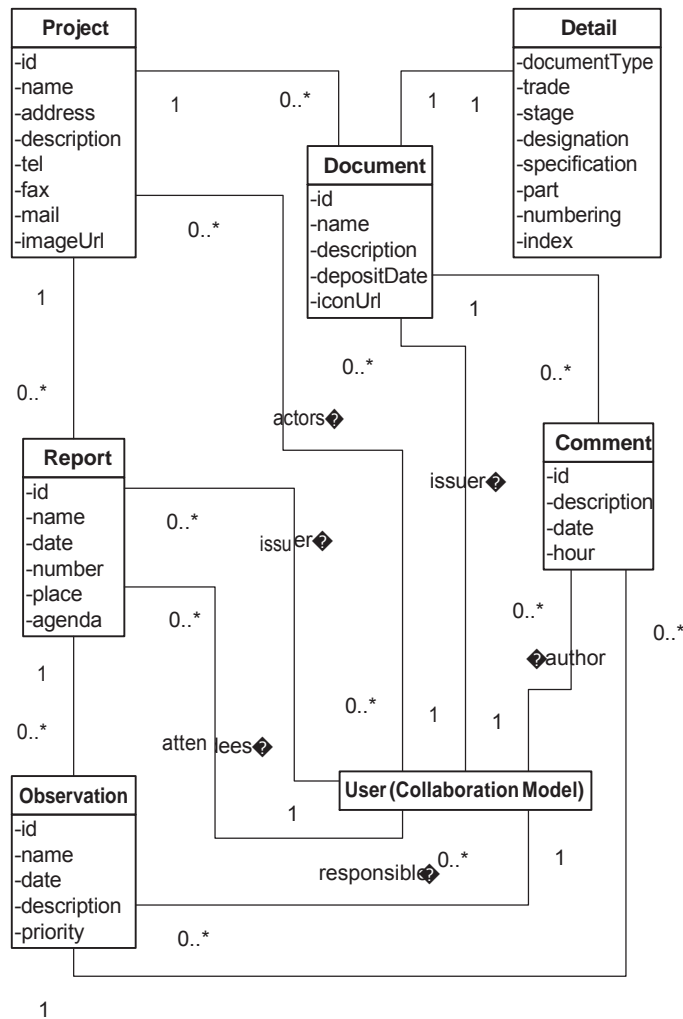


Figure 4.12 – Partie du modèle du domaine relatif à la tâche « consulter les comptes-rendus », sous la forme d'un diagramme de classes UML.

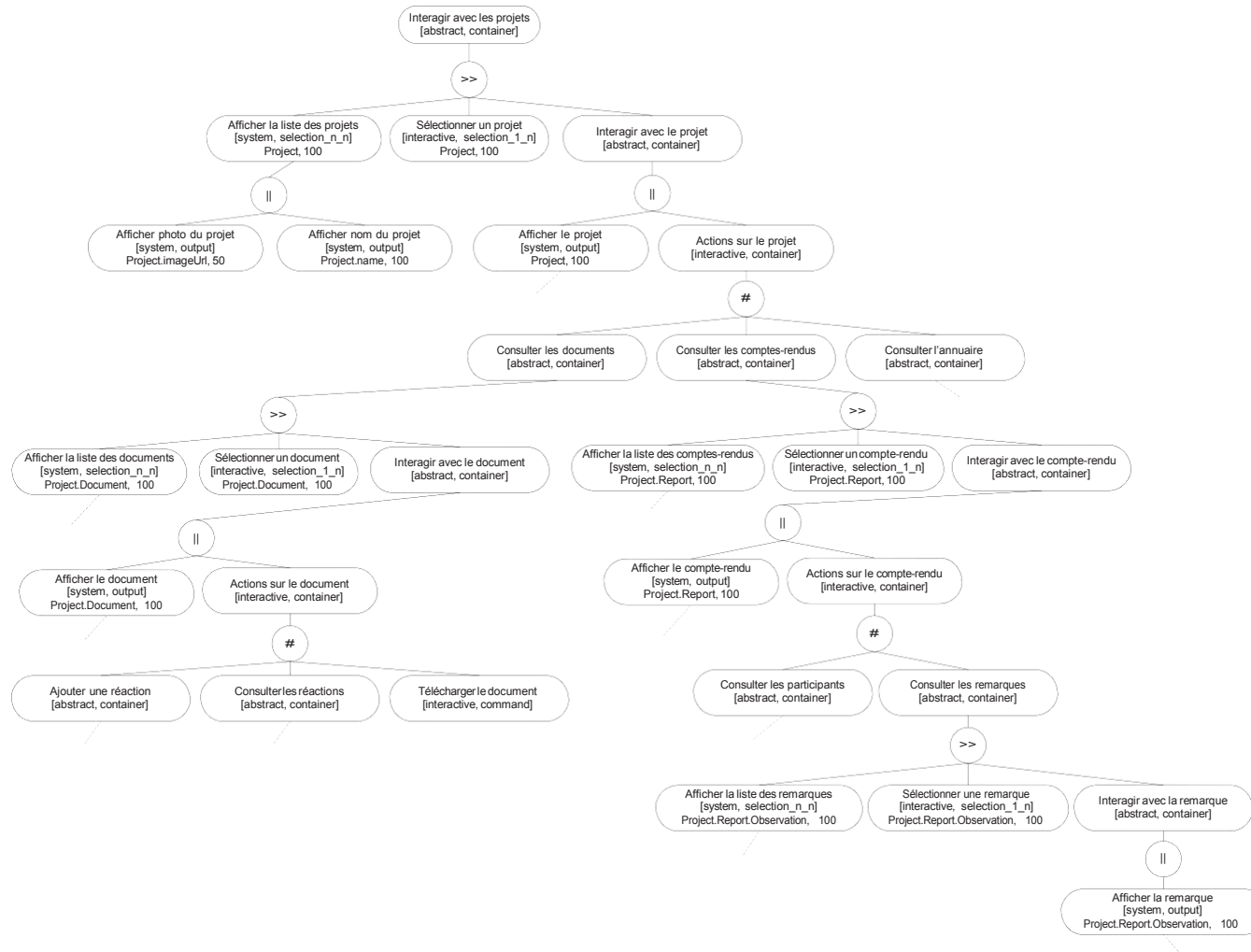


Figure 4.13 – Modèle de la tâche « consulter les comptes-rendus ». Les branches relatives aux affichages sont occultées pour plus de lisibilité.

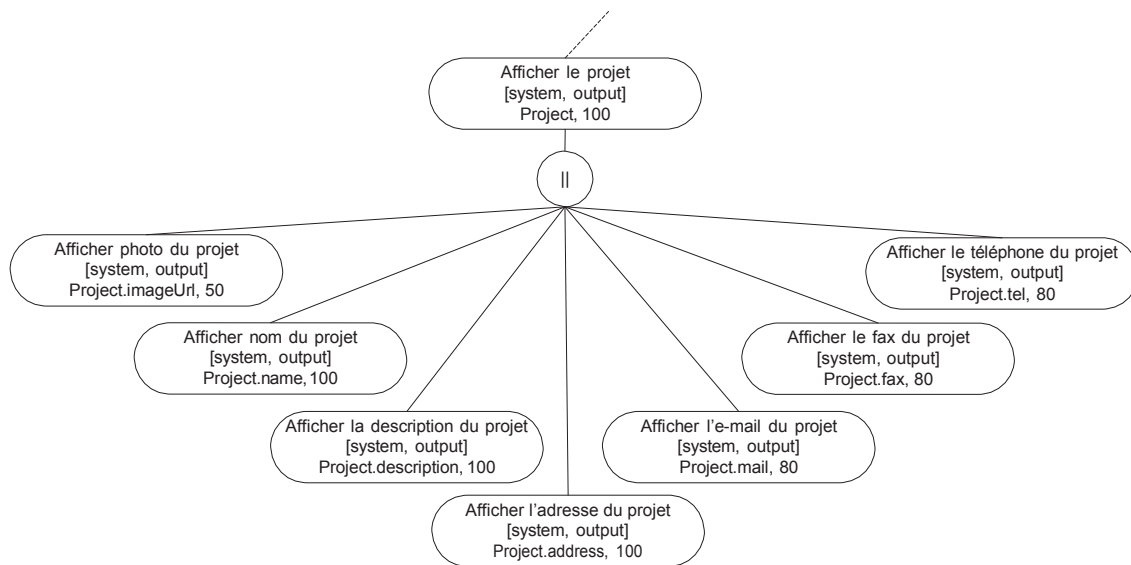


Figure 4.14 – Partie du modèle du modèle de la tâche « consulter les comptes-rendus » relatif à l’affichage des informations d’un projet en tenant compte de l’importance des concepts du domaine.

Vis-à-vis de l’adaptation des IHM, il convient également lors de cette étape de prêter attention à l’importance des tâches. Pour rappel, à chaque concept du domaine manipulé par une sous-tâche, un attribut d’importance peut être associé. Plus l’importance est basse, plus cette sous-tâche sera candidate à une adaptation de sa présentation. C’est ici au concepteur de déterminer si un concept du domaine est critique ou non pour la tâche donnée. La figure 4.14 reprend une portion de l’arbre de tâches avec le détail d’importance pour l’affichage des informations d’un projet. Nous pouvons noter une importance faible pour l’image du projet, qui disparaîtra si l’espace d’affichage vient à manquer. Quant aux informations de contact, celles-ci sont critiques pour la tâche mais peuvent être adaptées dans une certaine mesure (i.e. placées dans un menu déroulant plus compact).

5. Modélisation des cibles

En dehors de l’attribut d’importance associé aux manipulations du domaine dans les arbres de tâches, l’adaptation des IHM dépend également d’autres facteurs à modéliser. Il s’agit des cibles (i.e. les triplets d’environnements utilisateur, physique et matériel) qu’il est nécessaire de modéliser pour cadrer l’adaptation. Concrètement, cela correspond à modéliser les particularités du matériel prévu (e.g. smartphone, tablette et ordinateur), des propriétés physiques (e.g. chantier bruyant et lumineux) ainsi que les stéréotypes d’utilisateur (e.g. expérience, déficience, etc.).

Ainsi, la table 4.2 reprend les environnements modélisés pour le cas d’étude qui dans le cas d’ADAGIOS ont été modélisés à travers les métamodèles d’UsiXML (que nous utilisons tels quels).

Table 4.2 – Points remarquables des cibles du cas d'étude vis-à-vis des attributs au sein d'UsiXML.

Environnements	Déclinaisons		
Matériel	Smartphone keyboard : none pointingDevice : touchscreen screenWidth : 640 screenHeight : 960 isSoftKeyCapable : true hasTouchScreen : true	Tablette keyboard : none pointingDevice : touchscreen screenWidth : 800 screenHeight : 1280 isSoftKeyCapable : true hasTouchScreen : true	PC keyboard : yes pointingDevice : mouse screenWidth : 1920 screenHeight : 1080 isSoftKeyCapable : true hasTouchScreen : no
Physique	Chantier isNoisy : true isStressing : true lightingLevel : high	Bureau d'étude isNoisy : true isStressing : true lightingLevel : moderate	Domicile isNoisy : false isStressing : false lightingLevel : moderate
Utilisateur	Architecte taskExperience : high systemExperience : high taskMotivation : high	Architecte taskExperience : high systemExperience : moderate taskMotivation : high	Architecte taskExperience : high systemExperience : high taskMotivation : high

En sortie de la 5^{ième} étape, la suite **d'outils** nous a permis de générer une première version de **l'IHM** décrite avec la méthodologie (un exemple étant donné en figure 4.5, page 168), et ce, conformément au cas d'étude. Ceci nous permet de valider la pertinence des métamodèles choisis pour la description du contexte de **l'activité** collective. Cette première génération **d'IHM** est néanmoins dépourvue de support à la conscience de groupe qui sera défini en étape 6 de la méthodologie.

Résumé de section

Pour résumer cette section, nous validons ici **l'applicabilité** de nos métamodèles dans le déroulement de la méthodologie de conception proposée, notamment dans ses étapes 1 à 5. Pour cela, nous avons mis en **œuvre** celles-ci via un cas d'étude concret dans le domaine AIC avec une emphase sur le scénario de la réunion de chantier.

Nous avons déroulé la méthodologie sur ce cas d'étude, ce qui démontre de la pertinence des métamodèles et de la démarche. Cependant, pour être pertinent il faudrait dérouler la méthodologie dans plusieurs cas d'étude. Un modèle étant une représentation **d'un** système dans un contexte particulier, il est difficile de prétendre à **l'exhaustivité**. Néanmoins, aucun problème particulier **n'a** été décelé dans la modélisation de ce cas d'étude.

Concernant les hypothèses émises (voir page 50), cette section nous permet de démontrer la validité conceptuelle de **l'hypothèse** H1 (de par **l'applicabilité** des métamodèles à un cas d'étude) ainsi que de **l'hypothèse** H3 (en couvrant une modélisation jusqu'à l'étape 5 de la méthodologie de conception).

4.3 Expérimentation sur les besoins en conscience de groupe

Dans les sections précédentes, nous avons présenté la base **d'outils ADAGIOS** pour ensuite la confronter à un cas d'étude concret dans le secteur AIC. Nous avons notamment déroulé la méthodologie de l'étape 1 à 5 pour nous arrêter à la première génération **d'IHM**. Ici, nous nous concentrons sur l'étape 6 de la méthodologie qui consiste à impliquer les utilisateurs dans un cycle itératif afin **d'identifier** leurs besoins en conscience de groupe. Cette étape fait **l'objet d'une** évaluation dédiée car elle représente un pan particulier de notre proposition. En effet, cette étape itérative dépend de notre métamodèle de besoins en conscience de groupe et du mécanisme de composition de widgets que nous cherchons donc à valider **d'un** point de vue méthodologique.

L'expérimentation est donc davantage centrée sur la boucle **d'itération** de définition des besoins en coordination, notamment via des questionnaires. **L'enjeu** est d'évaluer la pertinence des questionnaires (qui découlent des modèles et de la caractérisation de Gutwin et al. [53, 55]), leur complétude et leur translation en modèles de besoins, puis d'évaluer la prise en compte des besoins fonctionnels qui en découlent et la capacité de la méthodologie à converger vers une IHM de qualité satisfaisante.

Étant donné le faible échantillon de résultats obtenu (5 utilisateurs, 5 scénarios sur 2 itérations chacun), cette expérimentation **n'a** pas pour ambition de répondre de manière forte et quantitative à nos propositions, mais plutôt **d'en** dégager les tendances via une évaluation qualitative.

En section 4.3.1, nous détaillons le protocole de test encadrant cette expérimentation, notamment à travers la définition de scénarios nous permettant de détecter les cas limites de nos propositions. Enfin en section 4.3.2, nous présentons les résultats de **l'expérimentation** en nous focalisant sur les points marquants.

4.3.1 Protocole de test

Le but de **l'expérimentation** étant de valider nos modèles (et leur utilisation) **d'abstraction** des besoins en conscience de groupe, il est nécessaire de construire un protocole de test autour des points à valider. Ainsi, dans l'étape itérative, nous cherchons plus particulièrement à évaluer les aspects suivants :

Du côté utilisateur : la pertinence des questionnaires basés sur la caractérisation de Gutwin et al. [53, 55] de la conscience de groupe et **l'implication** de **l'utilisateur** dans la méthodologie ;

Du côté concepteur : la capacité **d'ADAGIOS** à prendre en compte les besoins identifiés, **c'est-à-dire** la charge de travail pour le concepteur dans la définition de règles précises (i.e. modèles de besoins en conscience de groupe) et éventuellement dans la conception de nouveaux widgets ou la modification de widgets existants.

Pour explorer ces deux facettes, nous avons établi un ensemble de scénarios de test visant à explorer les possibilités de la méthodologie. Le protocole **s'articule** autour de ces scénarios et consiste à impliquer des utilisateurs venant du domaine AIC.

4.3.1.1 Définition de scénarios de test

Afin de construire les scénarios de test, il convient de déterminer les points de variabilité qui permettent de mettre en avant les axes précédemment définis (i.e. les points de vue utilisateur et concepteur dans la méthodologie).

Dans la méthodologie de conception du point de vue de **l'utilisateur**, le but est de définir les besoins en conscience de groupe par **l'intermédiaire** de questionnaires. À **l'issue** de cette étape, plusieurs cas de figure peuvent se présenter suivant **l'itération** en cours :

- **En itération 0**, en sortie directe de l'étape 5 :
 - **L'utilisateur n'identifie pas** de besoins en conscience de groupe, **l'itération s'arrête** ;
 - **L'utilisateur identifie** des besoins en conscience de groupe, le concepteur les traduit en modèles ;
- **En itération 0+n**, où des widgets sont présents :
 - **L'utilisateur valide** le résultat, **l'itération s'arrête** ;
 - **L'utilisateur n'est pas satisfait** et raffine éventuellement ses besoins, le concepteur le prend en compte ;
 - **L'utilisateur n'exprime plus** de besoins, le concepteur retire le modèle de besoins correspondant.

En fonction de ces cas de figure, la tâche du concepteur varie selon deux paramètres : la modélisation de besoins en conscience de groupe et la modélisation de widgets. Les implications sont alors les suivantes :

- Des besoins doivent être modélisés ou des besoins existants doivent être modifiés/raffinés. Ceci a un éventuel impact sur la modélisation des widgets :
 - Le dépôt de widgets couvre les besoins, il **n'y a pas** de travail supplémentaire pour le concepteur ;
 - Le dépôt de widgets ne couvre pas tous les besoins, le concepteur doit modéliser un ou des widgets ;
 - Le dépôt de widgets ne couvre pas tous les besoins, mais un widget peut être étendu.
- Des besoins existants doivent être retirés, le modèle de besoins est simplement supprimé.

Ces différents cas de figure représentent les points que le protocole de tests doit amener à explorer. Ces derniers sont résumés en figure 4.15 et attribués à un numéro afin de les répartir dans des scénarios de tests.

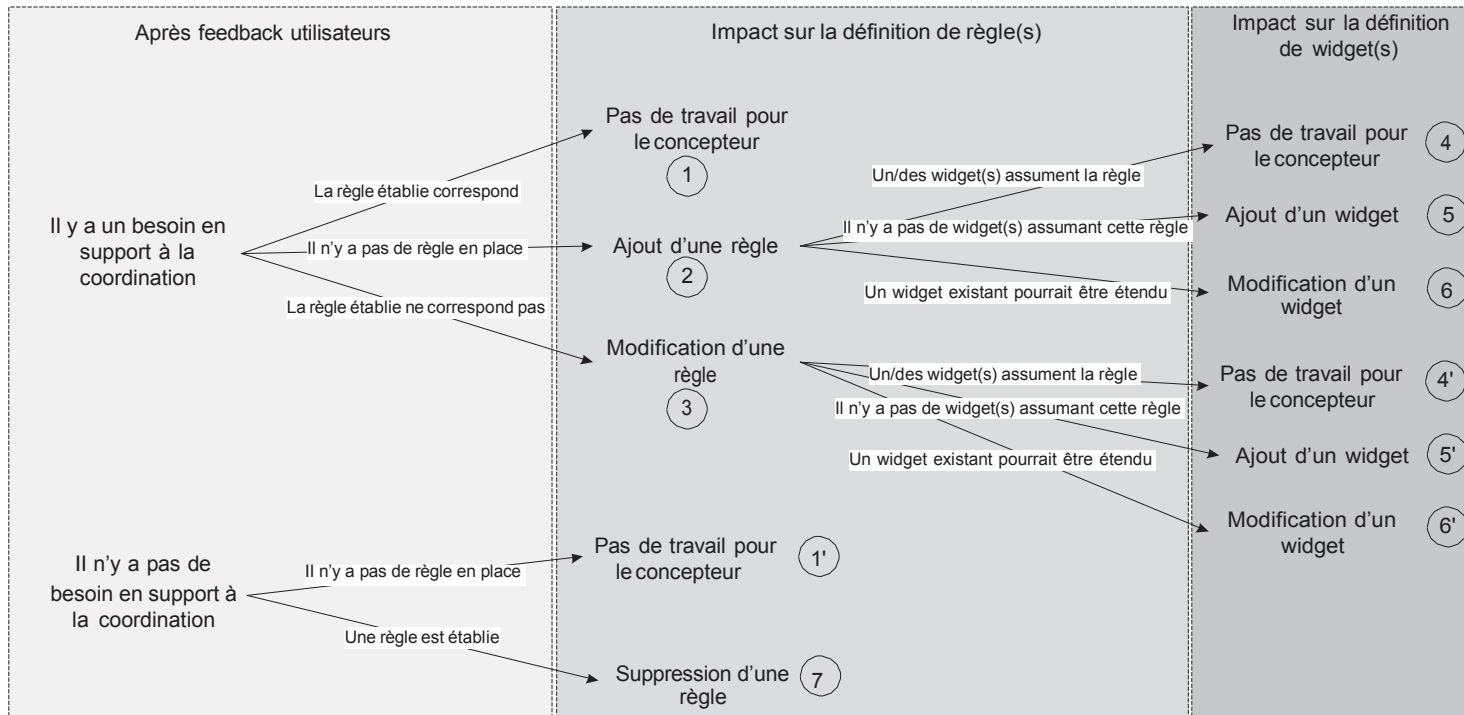


Figure 4.15 – Résumé des cas de figure à couvrir dans l'expérimentation.



Figure 4.16 – Widgets modélisés pour les besoins de l'expérimentation.

Sur cette base, nous avons défini cinq scénarios visant à mettre en évidence ces cas. Ces scénarios consistent à mettre l'utilisateur dans une situation identifiée par une tâche et un contexte particuliers au sein du scénario global de la réunion de chantier. Ces situations ont été élaborées avec l'aide d'un expert du domaine AIC qui est proche du projet CRTI-Web (sur lequel les tâches se basent) afin de s'inspirer de cas réels dans la méthodologie de conception. Le but de ces situations est de couvrir les cas de figure définis en ciblant une itération particulière (e.g. en sortie de l'étape 5, c'est-à-dire l'itération 0, les itérations suivantes étant dénommées par « n+1 ») et en mettant en place des modèles de besoins en conscience de groupe adéquats ou non afin de déceler les limites de notre approche. La table 4.4 reprend les particularités de chaque scénario vis-à-vis des cas couverts.

Pour les besoins de l'expérimentation, nous avons peuplé le dépôt de widgets a priori tout en omettant un widget que nous savions essentiel aux scénarios afin de simuler le cas de la modélisation d'un nouveau widget. Ainsi, nous avons modélisé un widget affichant la caméra du chantier sélectionné, un widget affichant l'historique des dernières actions ainsi qu'un widget affichant la liste des utilisateurs connectés (qui est le widget défini lors de l'expérimentation). Ces widgets sont illustrés en figure 4.16 sous leur forme interprétée sans aucune adaptation appliquée et la table 4.3 détaille les besoins en conscience de groupe couverts par ces widgets. Nous verrons plus en détail l'élaboration d'un widget dans le cadre d'un scénario en section 4.3.2.1.

Pour le suivi de l'expérimentation, une fiche descriptive à l'attention du concepteur est associée à chaque scénario. Cette fiche décrit les particularités du scénario et le résultat hypothétique attendu. Ci-après, un exemple de fiche de scénario, l'ensemble d'entre elles étant disponible en annexe B.

4.3. Expérimentation sur les besoins en conscience de groupe

Table 4.3 – Besoins de conscience de groupe couverts par les widgets de l'expérimentation, basés sur la caractérisation de Gutwin et al. [53, 55].

Widgets	Besoins couverts	Justification du taux de pertinence
Liste d'utilisateurs connectés	presence 100, identity 100, embodiment 60, availability 100	Le concept du domaine lié à l'embodiment (i.e. la photo) a un attribut d'importance de 60 impliquant une disparition de celui-ci sur appareils mobiles.
Webcam	presence 20, action 40	Une webcam ne permet pas d'identifier clairement une personne, mais permet d'estimer la population du chantier (i.e. presence) et d'observer son avancement plus précisément (i.e. action).
Historique des actions	presence 60, presence history 100, identity 60, action 60, action history 100, event history 100, authorship 100, artifact 60, artifact history 100	La presence , l'identity , l'artifact et l'action peuvent être déterminées indirectement en inférant l'information sur l'ensemble de l'historique . Par exemple, si une personne s'est connectée (i.e. presence history) mais non déconnectée, il est possible de déterminer indirectement qu'elle est présente actuellement. Dans le cas d'une information indirecte, nous considérons une pertinence moindre.

Table 4.4 – Description des scénarios.

Scénarios	1	2	3	4	5
Itération	0+n	0+n	0	0+n	0
Support initial des besoins	Mauvais	Bon	Aucun	Mauvais	Aucun
Action anticipée	Modification du modèle de besoins	Aucune	Ajout d'une règle et ajout ou modification de widgets	Suppression d'une règle	Aucune
Cas couverts	3, 4' (4)	1 (1')	2, 5 (5'), 6 (6')	7	1' (1)

Scénario ③

Particularité(s) :

Il n'y a pas de modèle de besoins prédéfini et il y a un besoin de support à la coordination.

Mise en situation :

« Au bureau, vous souhaitez consulter les remarques relatives à une entreprise afin de préparer la prochaine réunion de chantier. »

Tâche demandée : dans le projet « Construction Centre de Recherche Henri Tudor », consultez la remarque « problème de livraison » du compte-rendu n°2.

Contexte :

Activité = Lit le compte rendu
Localisation = Bureau d'étude
Matériel = Ordinateur de bureau
Rôle = Maître d'œuvre

Écran(s)/Tâche(s) concerné(s) :

« Consulter une remarque. »

Widgets présents :

Aucun (pas de règle en place).

Feedback / résultats anticipés :

Un modèle est créé pour répondre au besoin. La base de données ne contient pas l'intégralité des widgets nécessaires (i.e. utilisateurs connectés, webcam). Il est nécessaire d'en ajouter ou d'en modifier.

Cas couverts :

② ⑤ ⑥

4.3.1.2 Le déroulement du protocole

Le protocole en lui-même consiste à dérouler la méthodologie de conception de l'étape 6 avec des utilisateurs experts du domaine AIC. Le but étant de reproduire une situation réelle de conception, le protocole reprend donc les mêmes étapes auquel il ajoute une étape préliminaire visant à présenter le contexte aux utilisateurs et à leur permettre de **s'approprier l'outil** généré.

Durant ces étapes, les utilisateurs sont accompagnés par un concepteur dont le rôle est assuré par **l'auteur** de cette thèse, impliquant un caractère subjectif dans l'évaluation côté concepteur. Le but de cet accompagnement **n'est** pas de remplir les questionnaires mais de clarifier les consignes (ou problèmes méthodologiques) si nécessaire, mais aussi **d'inciter** les utilisateurs à expliciter oralement leurs réflexions (sous la forme d'un entretien semi-directif) en vue de **l'analyse** du résultat. Une attention particulière a été portée sur ces directions afin de **n'introduire** aucun biais dans les réponses des utilisateurs. Tous les entretiens ont également fait **l'objet d'un** enregistrement audio.

Le protocole se déroule comme suit (et schématisé en figure 4.17) :

- A **Présentation de l'IHM générée à l'utilisateur**, afin que **l'utilisateur** puisse **s'approprier l'application** de gestion de comptes-rendus. Le scénario global de la réunion de chantier lui est également présenté afin **d'exposer** les tenants et aboutissants de **l'IHM** générée. Lors de cette étape, **l'utilisateur** est libre de manipuler **l'application** ;
- B **Mise en situation sur un scénario de test**, tel que délimité précédemment, afin de placer **l'utilisateur** dans le contexte associé au scénario ;
- C **Évaluation des besoins en conscience de groupe**, où, après présentation du questionnaire (voir annexe A.1), **l'utilisateur** définit ses besoins en se positionnant sur la grille d'évaluation par rapport au scénario et à l'écran associé. **L'utilisateur n'est** pas guidé dans son évaluation mais le conducteur des tests reste à disposition pour éventuellement expliciter les questions en cas de consigne mal comprise. Dans le cas **d'une** itération non initiale (i.e. 0+n), le questionnaire **d'identification** des besoins est pré-rempli avec les données issues de **l'itération** précédente ;
- D **Définition de modèles et de widgets par le concepteur**, en accord avec les résultats du questionnaire précédent. Il **s'agit** de définir un modèle de besoins en conscience de groupe qui cible le contexte donné par le scénario et de vérifier que le dépôt dispose **d'une** réponse aux besoins. Le cas échéant, le concepteur crée ou adapte un, ou des, widgets adéquats en modélisant leurs modèles de tâches ;
- E **Évaluation de la pertinence des widgets générés**, où **l'IHM** après re-génération (et prise en compte du modèle de besoins) est présentée à **l'utilisateur** pour validation de la réponse générée. À cet effet, le second questionnaire (voir annexe A.2) est exposé à **l'utilisateur** qui évalue alors la pertinence des widgets présents dans le scénario donné. Durant cette étape, **l'utilisateur** est libre de manipuler **l'application**.

Pour chacun des scénarios, deux itérations de la méthodologie sont effectuées avec chaque utilisateur, ceci afin **d'avoir** un aperçu de la convergence des besoins et de la qualité de la réponse apportée par la re-génération de **l'IHM**.

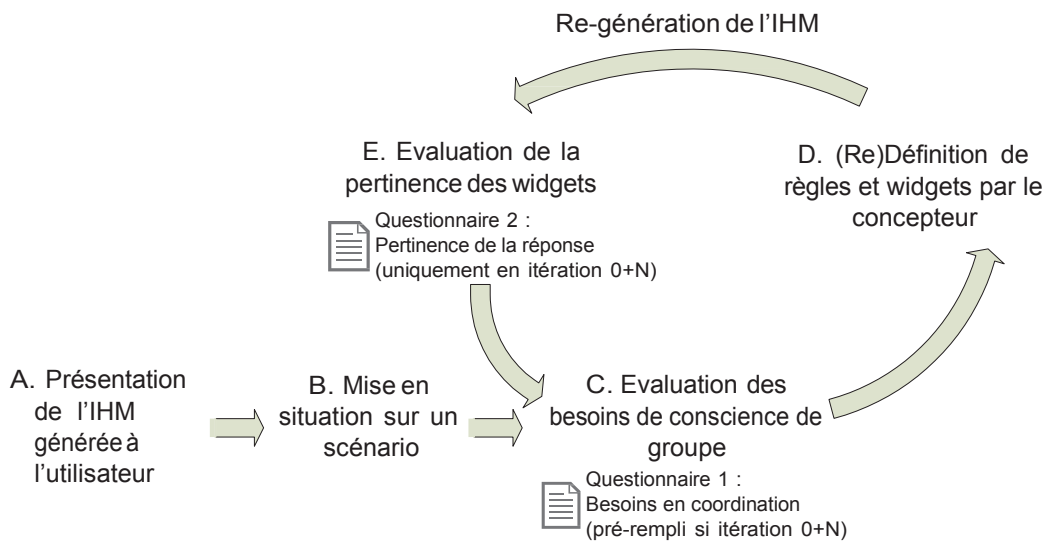


Figure 4.17 – Protocole de test.

Lors du déroulement de ce protocole, plusieurs points ont fait l'objet d'une attention particulière :

- **Vis-à-vis de la méthodologie** : la clarté des consignes, la convergence de la méthodologie et l'usage des questionnaires ;
- **Vis-à-vis de la place de l'utilisateur** : la difficulté de se projeter dans une situation, de positionner son besoin dans les questionnaires, la pertinence de la caractérisation de Gutwin et al. [53, 55] pour exprimer ses besoins et la complexité de la méthodologie à travers plusieurs scénarios ;
- **Vis-à-vis du rôle du concepteur** : la difficulté de transcrire les besoins en modèles et en règles OCL et la difficulté de concevoir de nouveaux widgets.

4.3.1.3 Les utilisateurs

Dans le cadre de cette expérimentation, des utilisateurs du domaine AIC ont été sollicités. Ainsi, cinq personnes ont pris part aux tests, dont quatre ayant une formation d'architecte et une d'ingénieur civil. Parmi elles, quatre exercent en tant que chercheurs en science de l'architecture, une en tant qu'architecte indépendant et toutes ont une bonne affinité avec les outils informatiques. Une de ces personnes était familière avec les travaux de Gutwin et al. [53, 55] sur la conscience de groupe. Enfin, chaque utilisateur a été sollicité sur deux scénarios différents dans le but d'avoir au moins deux retours d'expérience sur chacun des scénarios.

4.3.2 Synthèse des résultats de l'expérimentation

Nous présentons ici le déroulement **d'un** scénario de test ainsi que la synthèse des points remarquables que nous cherchions à traiter à travers l'expérimentation. Rappelons que les faits notés vis-à-vis du concepteur ont potentiellement un caractère subjectif, ce rôle ayant été assumé par **l'auteur** de cette thèse. De manière générale, les scénarios ont bien fonctionné et ont permis de mettre en évidence les cas limites que nous escomptions dans notre analyse.

4.3.2.1 Exemple d'un scénario testé

Afin **d'illustrer** le déroulement de l'expérimentation, nous détaillons ici le cheminement du protocole sur un scénario via un seul utilisateur. En l'occurrence, nous détaillons le scénario 3 dont la fiche descriptive est visible en page 194. La particularité de ce scénario est de ne couvrir aucun besoin en conscience de groupe initialement et le résultat attendu est alors la création **d'un** modèle de besoins sur base des réponses de **l'utilisateur** au questionnaire concerné. La conséquence attendue après re-génération de **l'IHM** est **l'insertion d'un** (ou plusieurs) widget(s), avec l'éventualité de devoir en concevoir un si le dépôt ne dispose pas de réponse adéquate aux besoins.

La situation de ce scénario correspond à la phase de préparation **d'une** réunion de chantier. À cet effet, le **maître d'œuvre** (i.e. le rôle de **l'utilisateur** dans ce scénario) doit consulter les remarques du compte-rendu de la réunion précédente. Ces remarques permettent d'établir **l'ordre** du jour de la réunion à venir, voire de prendre des mesures en amont si applicable (e.g. contacter un fournisseur pour palier un retard sur une livraison).

Nous reprenons ici le déroulement du protocole :

- A **Présentation de l'IHM générée à l'utilisateur** : dans ce scénario, aucun besoin en conscience de groupe **n'a** été défini au préalable, **l'IHM** présentée à **l'utilisateur** est donc vierge de tout widget. **L'utilisateur s'approprie l'application** et ses fonctionnalités en la manipulant librement dans son intégralité;
- B **Mise en situation sur un scénario de test** : dans un premier temps, **l'utilisateur** est mis dans une situation générale : « Au bureau, vous souhaitez consulter les remarques relatives à une entreprise afin de préparer la prochaine réunion de chantier. » Cette situation est alors précisée en demandant à **l'utilisateur** de se focaliser sur une tâche particulière et de naviguer jusqu'à celle-ci dans **l'application** : « dans le projet **Construction Centre de Recherche Henri Tudor**, consultez la remarque **problème de livraison** du compte-rendu n°2. » L'écran correspondant à cette tâche est illustré en figure 4.18;
- C **Évaluation des besoins en conscience de groupe** : une fois mis en situation, le questionnaire **d'identification** des besoins est présenté à **l'utilisateur** (voir annexe A.1 page 226). **L'utilisateur** est guidé dans la compréhension des différentes questions mais **n'est** pas influencé dans ses réponses. Les besoins exprimés par **l'utilisateur** dans cet exemple sont illustrés en table 4.5. Dans **l'ensemble**, deux types de besoins sont exprimés par **l'utilisateur** : des informations sur les autres acteurs (essentiellement dans le but de « pouvoir les contacter et de **connaître** leur disponibilité en cas

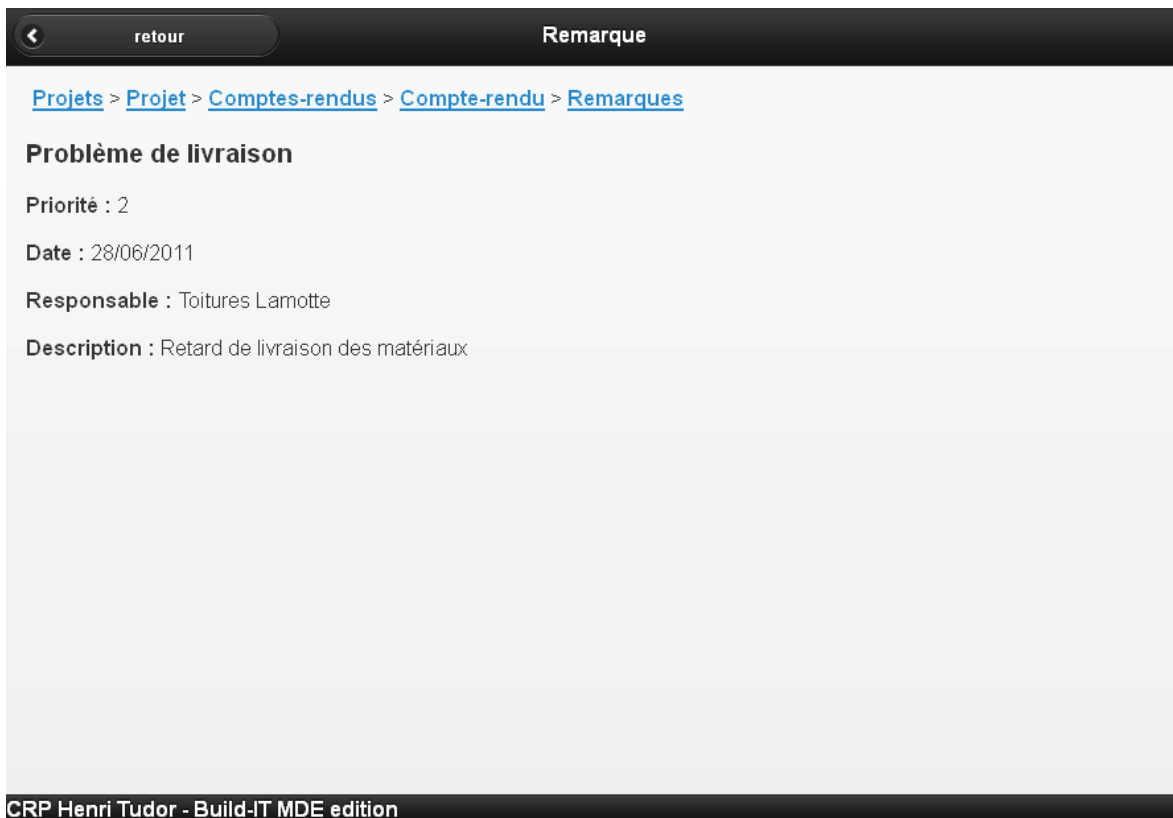


Figure 4.18 – IHM générée sans aucun modèle de besoins en conscience de groupe au préalable. L'écran correspond à la tâche « consulter une remarque. »

4.3. Expérimentation sur les besoins en conscience de groupe

Table 4.5 – Réponses de l'utilisateur en termes de propriétés sur la caractérisation de Gutwin et al. [53, 55] et leur correspondance en taux d'importance.

Propriétés (en anglais)	Réponses	Taux d'importance
<i>presence</i>	Très utile	100
<i>presence history</i>	Plutôt utile	60
<i>identity</i>	Très utile	100
<i>embodiment</i>	Plutôt utile	60
<i>availability</i>	Très utile	100
<i>action</i>	Pas vraiment utile	30
<i>action history</i>	Très utile	100
<i>authorship</i>	Très utile	100
<i>intention</i>	Pas vraiment utile	30
<i>artifact</i>	Plutôt utile	60
<i>artifact history</i>	Très utile	100
<i>location</i>	Pas du tout utile	0
<i>location history</i>	Pas du tout utile	0
<i>gaze</i>	Pas du tout utile	0
<i>view</i>	Pas du tout utile	0
<i>reach</i>	Pas du tout utile	0
<i>event history</i>	Très utile	100

de problème ») et des informations sur les ressources (afin de « savoir si un document relatif à un problème a bien été pris en compte ou mis à jour par les acteurs concernés »);

- D **Définition de modèles et de widgets par le concepteur** : les réponses du questionnaire sont reprises par le concepteur qui associe à chaque besoin (**requirement**) un taux d'importance. Lors de l'expérimentation, ce taux a été fixé via l'échelle suivante sur les réponses : **pas du tout utile** à 0, **pas vraiment utile** à 30, **plutôt utile** à 60, et **très utile** à 100. L'option **indispensable** correspond également à un taux de 100 et est présente afin d'identifier des besoins qui relèveraient de spécificités fonctionnelles omises à la conception, un besoin en conscience de groupe étant par nature non essentiel à la réalisation d'une tâche (et donc non fonctionnel). Une fois les taux fixés, le concepteur définit un modèle de besoins qui décrit la situation (i.e. le contexte concerné par le scénario) et les besoins. Ce modèle est visible en figure 4.19. L'IHM peut alors être re-générée afin d'observer si le dépôt de widgets dispose des éléments adéquats pour répondre aux besoins. Or, à ce stade de l'expérimentation, le dépôt ne dispose que des widgets de **webcam** et d'**historique des actions**, et ne peut donc pas répondre complètement aux besoins de **presence**, **identity**, **embodiment** et **availability**. Il est donc nécessaire de concevoir un nouveau widget, en l'occurrence celui de **liste des utilisateurs connectés**. Le concepteur définit le modèle de tâche du widget de manière similaire à celui que nous avons déjà illustré en figure 3.12 (page 135). Les concepts affichés par ce widget et leur importance (qui détermine la manière dont le widget est adapté) sont définis en fonction des besoins en octroyant un fac-

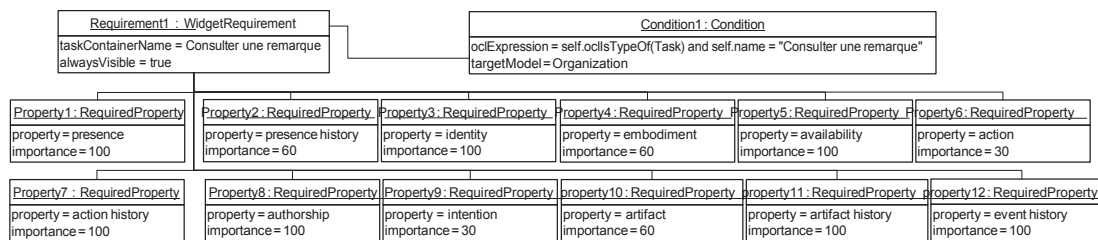


Figure 4.19 – Modèle de besoins en conscience de groupe défini en fonction des résultats du questionnaire de l'utilisateur. Représenté sous la forme d'un diagramme d'objets UML.

leur **d'importance** aux concepts du domaine en accord avec les besoins utilisateurs modélisés. Par exemple, les besoins de **presence**, **identity** et **availability** étant forts, le concepteur définit les concepts de **nom**, **prénom**, **disponibilité** avec une importance de 100 afin qu'ils ne soient pas adaptés en cas de manque d'espace d'affichage. À l'inverse, le besoin **d'embodiment** étant moindre, le concept de **photo** est défini avec une importance réduite à 60, ce qui engendrera une adaptation, notamment sur appareils mobiles. Ce widget se voit alors attribuer des métadonnées décrivant avec quelle pertinence (attribut de **relevance** dans le metamodelle de widget) ce widget répond à des besoins de conscience de groupe, ici la **presence** (100), **l'identity** (90), **l'embodiment** (60) et **l'availability** (90). Dès lors, le dépôt de widgets peut répondre de manière plus complète aux besoins exprimés par **l'utilisateur** et une IHM peut lui être reproposée ;

E **Évaluation de la pertinence des widgets générés** : la nouvelle IHM pourvue de widgets est présentée à **l'utilisateur** (voir figure 4.21) qui est libre de manipuler **l'application**. Un nouveau questionnaire (voir annexe A.2, page 227) est rempli par **l'utilisateur** afin de déterminer si les concepts présents sont bel et bien perçus par **l'utilisateur** comme les informations de conscience de groupe demandées. En sus, de ce questionnaire, un avis plus précis sur chacun des widgets présents est demandé à **l'utilisateur** (voir questionnaire en annexe A.3, page 228) afin de vérifier si la granularité de **l'information** est adéquate à la tâche en cours. Dans le présent cas, les informations ont été perçues correctement par **l'utilisateur**, c'est-à-dire que les besoins en conscience de groupe ont bien été couverts par les widgets sélectionnés. Cependant, le second questionnaire (voir figure 4.20) a montré une satisfaction médiocre sur le widget **d'historique des actions**, **l'utilisateur** ayant jugé les informations comme étant bien présentes, mais en trop grande quantité. En effet, ce widget affiche un historique sur toutes les actions effectuées sur toute **l'application** et par quiconque. Le retour utilisateur suggère qu'il serait plus pertinent et utile d'avoir un historique en lien avec l'écran ou la ressource actuellement consultée. Une itération supplémentaire du protocole serait alors nécessaire pour que le concepteur revienne à la définition de ce widget et présente à nouveau **l'IHM** pour validation à **l'utilisateur**.

Cet exemple de mise en **œuvre** du protocole de test et de la méthodologie nous permet notamment de mettre en avant la conception **d'un** widget et le lien entre les attributs de

4 nouvelle(s) action(s)

- * 24/04 12:00 Jérôme C. s'est déconnecté
- * 24/04 11:55 Jérôme C. a modifié la remarque "Remarque sur la toiture" → clic
- * 24/04 11:50 Jérôme C. a ajouté la remarque "Remarque sur la toiture" ↓
- * 24/04 11:45 Jérôme C. s'est connecté
- * 23/04 16:14 Thomas A. s'est déconnecté
- * 23/04 14:32 Thomas A. a créé le compte-rendu "Compte-rendu n1" T. A. B.
- * 23/04 14:30 Thomas A. s'est connecté

Pas même niveau d'info.

Pas du tout satisfait	Pas vraiment satisfait	Plutôt satisfait	Très satisfait
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si vous n'êtes pas complètement satisfait, avez-vous des suggestions pour améliorer ce widget ?

- * Si personne connectée → icône ds colonne.
- * Pouvoir filtrer actions par personne.
Par exemple ds le détail d'une personne =
- * Tri par calendrier / par jour
- * Filtrer action suivant rubrique sélectionnée.
! droit accès.

Figure 4.20 – Résultats du questionnaire de satisfaction sur le widget d'historique des actions.

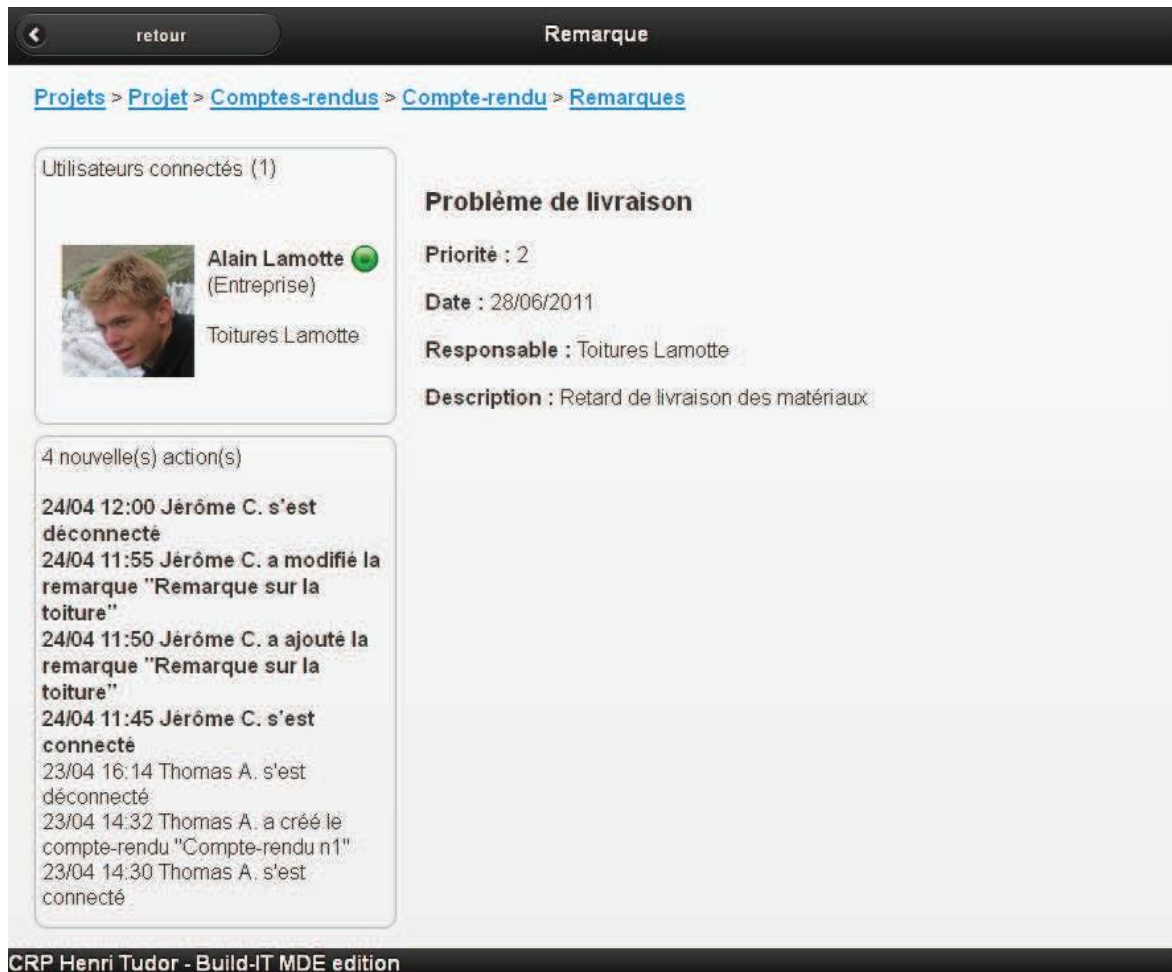


Figure 4.21 – IHM générée une fois le modèle de besoins en conscience de groupe défini.

4.3. Expérimentation sur les besoins en conscience de groupe

besoin de conscience de groupe (**requirement**), la pertinence des widgets vis-à-vis de ces besoins (**relevance**) et **l'importance** des concepts du domaine qui représentent ces besoins au sein des widgets.

Dans notre proposition, nous avons laissé à **l'appréciation** du concepteur la manière dont ces valeurs sont fixées. Dans la pratique de **l'expérimentation**, nous avons toutefois mis en place des valeurs homogènes. Par exemple, dans le cas **d'un** besoin en **embodiment** exprimé à 60 (plutôt utile), nous avons créé un widget affichant une photo avec une importance équivalente de 60. Le widget affichant cette photo **s'est** aussi vu attribuer un taux de pertinence de 60 dans sa réponse au besoin.

Si nous avons uniformisé ces valeurs, il est pour autant difficile de déterminer une possible corrélation entre ces trois paramètres, tant leur définition est subjective au concepteur et portant sur des éléments différents (**l'importance d'un** concept est lié à **l'adaptation**, alors que les **requirements** et la **relevance** sont propres à la conscience de groupe). Il serait cependant intéressant de continuer des recherches en ce sens afin d'établir un éventuel lien logique entre ces propriétés, voire **d'automatiser** la chaîne de définition des valeurs associées. Nous pouvons ici nous référer aux travaux de Figueroa-Martinez et al. [37] (voir section 3.2.1, page 130) qui dans leur modélisation des besoins en conscience de groupe (via une extension à UsiXML [70]), établissent un lien entre un besoin particulier et les concepts du domaine qui y répondent. Si ces travaux ne traitent pas **d'adaptation**, ils représentent néanmoins un point intéressant pour poursuivre la réflexion sur le lien entre besoin et domaine.

4.3.2.2 Synthèse à travers le rôle du concepteur

Perspective de modèles de besoins génériques

Dans la majorité des cas, la méthodologie et ADAGIOS ont montré de bons résultats (que nous détaillons davantage dans cette synthèse). Les questionnaires permettent **d'identifier** convenablement les besoins en conscience de groupe, notamment ceux concernant les autres acteurs (i.e. la catégorie de critères associée à **qui?**) et de produire des widgets adéquats. **D'autres** critères sont plus difficilement identifiables pour des raisons que nous évoquons ci-après.

Les informations liées aux acteurs sont celles qui ont été le mieux perçues par les utilisateurs mais aussi les plus récurrentes à travers les scénarios. Les modèles de besoins qui en résultent sont par ailleurs très similaires **d'un** scénario à **l'autre**, seules quelques contraintes sur le contexte diffèrent.

Vis-à-vis du concepteur, dans ces cas simples, les résultats des questionnaires ont pu être repris à **l'identique** étant donné que ceux-ci ont été construits sur la base du métamodèle de besoins. Le concepteur se concentre alors sur la seconde partie du modèle de besoins, **c'est-à-dire** la modélisation de la situation à travers les contraintes OCL sur le contexte.

La similitude entre les modèles de besoins de certains scénarios nous laissent penser que des modèles génériques peuvent être établis. Une perspective de ces travaux serait alors de déterminer si des situations génériques et récurrentes existent dans les activités collectives afin de déterminer si des modèles de besoins peuvent être généralisés et mis en

commun avec différents systèmes. Ceci à la manière de bonnes pratiques de conception mais dont le savoir serait capitalisé dans des modèles de besoins en conscience de groupe.

De plus, dans ces cas génériques, les utilisateurs ont été largement indépendants dans l'identification de leurs besoins. Cette indépendance des utilisateurs et la généralité de certains modèles de besoins nous laisse entrevoir la possibilité **d'automatiser** une partie des questionnaires. Par exemple, il pourrait être intéressant de donner la possibilité aux utilisateurs **d'annoter** les IHM générées avec des besoins en conscience de groupe. Ces annotations pourraient alors être transcrites automatiquement en modèles de besoins pour générer une nouvelle version de l'IHM intégrant des widgets.

Temps de conception amoindris au fil des itérations

Dans notre proposition **d'un** dépôt de widgets (voir section 3.2.2, page 133), nous avons mis en avant la possibilité de tendre vers un dépôt réutilisable afin de capitaliser ces fragments **d'IHM** et réduire les temps **d'itération**. Pour les besoins de **l'expérimentation**, nous **n'avons** modélisé que trois widgets différents mais avons pu observer **l'intérêt** de leur réutilisation sur plusieurs scénarios.

Ainsi, les cas de figure qui ne nécessitaient pas la modélisation ou la modification de widgets existants dans le dépôt ont pu être traités très rapidement du point de vue du concepteur. La tâche du concepteur est alors de définir un modèle de besoins en conscience de groupe suivant les réponses des utilisateurs, ce qui peut être réalisé en quasi-simultané du questionnaire pour régénérer immédiatement une IHM. Lors de **l'expérimentation**, la modélisation des besoins **n'a** pas nécessité plus de dix minutes de travail au pire des cas, permettant **d'enchaîner** les itérations avec les utilisateurs sans interruption.

La perspective de pouvoir disposer **d'un** dépôt de widgets déjà bien fourni et couvrant **l'ensemble** du spectre des besoins en conscience de groupe semble intéressante. En ce sens, il serait également intéressant de se pencher sur un état de **l'art** des interactions pour la conscience de groupe en les classant suivant des critères **d'abstraction** (tels que ceux de Gutwin et al. [53, 55]).

Problèmes liés à l'abstraction de la conscience de groupe

Le frein le plus important que nous avons rencontré concerne **l'abstraction** de la conscience de groupe via la caractérisation de Gutwin et al. [53, 55]. Pour rappel, nous avons défini notre métamodèle de besoins en fonction de cette caractérisation (voir section 3.2, page 130) en **l'associant** à un facteur **d'importance**.

D'un point de vue conceptuel et technique, nous avons montré la possibilité de générer une réponse à ces besoins à travers des widgets couvrant ces facteurs **d'importance**. Les questionnaires **d'identification** et de validation des besoins en conscience de groupe ont été établis par rapport à cette caractérisation qui trouve leur limite **d'un** point de vue méthodologique et par rapport à **l'utilisateur**. Nous mettons en avant trois types de problèmes qui y sont liés :

Des problèmes liés à la granularité des critères. Suivant Gutwin et al. [53, 55], les informations qui peuvent être véhiculées par conscience de groupe sont caractérisées par des catégories **d'informations** relatant du l'état **d'un** acteur ou **d'une** entité (e.g.

4.3. Expérimentation sur les besoins en conscience de groupe

presence, identity, action, etc.). Si cette caractérisation convient pour abstraire les informations et servir de facteur pour la composition de widgets, la granularité de ces informations **n'est** pas remontée par les questionnaires. En effet, si **l'on** prend par exemple un besoin lié au critère de *présence*, aucune information **n'indique** si **l'utilisateur** a besoin **d'un** simple indicateur (e.g. « des utilisateurs sont connectés ») ou **d'une** granularité plus fine (e.g. « il y a 4 utilisateurs connectés »). Nous avons anticipé cette limitation en introduisant un second questionnaire d'évaluation des représentations choisies dans les widgets (voir annexe A.3) en espérant clarifier la granularité en itération n+1. Ceci dévie de notre vue initiale consistant à abstraire la conscience de groupe, car ici nous nous attardons au final sur des aspects de présentation concrets ;

Des **problèmes liés à la compréhension des critères**. De nombreux critères représentent des notions proches mais **d'un** point de vue différent. Ceci peut perturber des utilisateurs peu sensibilisés à la conscience de groupe, ce qui est censé être le cas dans une situation réelle et ne peut être exigé des utilisateurs. Par exemple, des informations telles que la *présence* et *l'identité* présentent une ambiguïté auprès de la plupart des utilisateurs, une présence étant selon eux une notion inséparable de *l'identité* de la personne associée. **D'autres** couples **d'informations** font débat, tels que les *actions* et les *événements*, la *disponibilité* et *l'action*, ou encore les *artefacts* et les *ressources*. Il convient de bien délimiter ces informations pour un public **d'utilisateurs**, quitte à s'écarter du formalisme de la caractérisation de Gutwin et al. [53, 55] et des termes propres au domaine du TCAO ;

Des **problèmes liés à la perception de la conscience de groupe**. Les informations de conscience de groupe sont par essence perçues par gestaltisme (voir section 1.2.4.1, page 27) par les utilisateurs, un mécanisme cognitif inconscient. Lors de la phase de validation des besoins, nous demandions aux utilisateurs de confirmer si une information de conscience de groupe était bien présente dans **l'IHM**. Or, certaines informations peuvent être perçues de manière indirecte, comme cela peut être le cas avec un historique des actions horodatées qui permet implicitement de savoir **qu'une** personne était connectée (critère de *présence*) à **l'instant de l'action**. Ces informations indirectes **n'ont** pas été toujours bien perçues par les utilisateurs qui les jugeaient absentes de **l'IHM**, ce qui peut éventuellement remettre en cause la pertinence **d'un** widget qui proposerait **l'information** uniquement de manière indirecte. Ces informations indirectes sont tout de même pertinentes pour la conscience de groupe mais notre méthode de validation **n'est** a posteriori pas adaptée à ce cas précis.

Plusieurs perspectives s'ouvrent vis-à-vis de ces points. **D'une** part, améliorer la méthodologie et notamment les questionnaires pour identifier plus précisément les besoins en conscience de groupe. **D'autre** part, travailler sur une classification des informations de conscience de groupe plus adaptée que celle de Gutwin et al. [53, 55] en allant par exemple dans une granularité plus fine, voire **d'intégrer** des aspects de présentation. Il conviendrait cependant de développer une méthodologie en accord avec ces précisions, avec le risque éventuel **d'ajouter** une complexité non négligeable pour **l'utilisateur** et la définition de modèles de besoins en conscience de groupe.

Il serait également bienvenu de disposer **d'une** classification des besoins ouverte à des informations spécifiques aux domaines métiers. En effet, lors des entretiens, plusieurs utilisateurs nous ont remonté des besoins en coordination propres au secteur AIC, tels que des dates de facturation, de livraison, des causes de retard ou **d'autres** éléments propres à cette activité mais **n'entrant** pas dans les critères de la conscience de groupe. Ces informations ne sont pas essentielles à la réalisation de la tâche mais facilitent grandement la compréhension **d'une** situation.

4.3.2.3 Synthèse à travers le rôle de l'utilisateur L'utilisateur

anticipe le résultat et la surcharge d'informations

Sans avoir présenté au préalable le fonctionnement **d'ADAGIOS** aux utilisateurs et les implications de leurs réponses aux questionnaire, certains **d'entre** eux anticipaient la surcharge **d'informations** à présenter. Par exemple, un utilisateur considère que « les informations sont très utiles uniquement si leur présence **n'est** pas envahissante, » en étayant son propos par rapport à des fréquences de notification importantes ou indésirables.

Ceci représente une réaction très intéressante mais qui **n'est** pas nécessairement souhaitable dans notre cas précis. En effet, **d'un** point de vue méthodologique, nous cherchons à identifier tous les besoins de **l'utilisateur**, indépendamment de l'éventuel résultat. La surcharge **d'informations** est alors traitée via les mécanismes **d'adaptation**, **d'importance** des concepts et de **l'utilité** exprimée par les besoins. Nous considérons que la gestion de la charge cognitive incombe aux transformations de modèles qui portent les bonnes pratiques en termes **d'ergonomie**, une propriété venant du projet GENIUS (voir section 4.1.1, page 162). Afin **d'avoir** toutes les informations dans le processus de génération des IHM, il **n'est** alors pas souhaitable que **l'utilisateur** censure ses besoins vis-à-vis de cet aspect. Les utilisateurs sont toutefois rassurés après présentation de **l'IHM** re-générée en considérant le résultat « plutôt satisfaisant parce **qu'il n'y** a pas beaucoup de chose. » Ce phénomène est difficilement identifiable en dehors notre protocole de tests où nous demandions aux utilisateurs **d'expliquer** oralement leur raisonnement. Ces informations sont toutefois très utiles afin **d'améliorer** les heuristiques **d'adaptation** des IHM.

Retour lié à la vie privée, au droit d'accès ou la responsabilité des utilisateurs

Lors de **l'expérimentation**, les utilisateurs ont exprimé leurs préoccupations au sujet du respect de leur vie privée **ou d'informations** qui pourraient engager leur responsabilité. **L'acceptation** de la conscience de groupe est également en cause, un utilisateur note que « **c'est** intéressant de le savoir, mais je ne souhaite pas partager moi-même cette information et je doute donc que les autres en fassent autant » en évoquant notamment la géo-localisation et la disponibilité. Certaines remarques sont mêmes plus strictes, avec des informations de conscience de groupe qui ne doivent pas être publiques, au risque de problèmes légaux : « si je passe devant un chantier alors que je ne fais que me rendre au bureau, cela peut engager ma responsabilité si mon passage est prouvé. »

Les questionnaires permettent de noter ce refus strict (en considérant une information comme inutile), mais nous **n'avons** pas pris cela en compte dans les modèles de besoins et

dans la manière dont les widgets sont sélectionnés, résultant en des informations présentées non permises. En effet, les besoins ne sont exprimés **qu'en** fonction des informations utiles qui sont alors privilégiées lors de la sélection des widgets. Mais rien ne garantit que toutes les informations présentées ont une utilité et par conséquent certaines des informations inutiles peuvent être parmi celles qui ne sont pas souhaitées. Il convient alors **d'introduire** dans nos modèles de besoins des exceptions afin **d'exclure** strictement ce type **d'informations**.

Perspectives utilisateurs pour le support à la coordination

Au-delà du protocole et de ce que nous cherchons à identifier à travers **l'expérimentation**, les utilisateurs ont émis des besoins ou des suggestions intéressantes en termes de génération **d'IHM** pour le support à la coordination parmi les suivantes :

Interconnexion des informations : une fonctionnalité souvent demandée par les utilisateurs concerne **l'interconnexion** des informations de conscience de groupe. **L'idée** est de lier tous les concepts du domaine entre eux et de pouvoir explorer ces liens **lorsqu'une** information est présentée. Par exemple, dans un widget affichant la liste des acteurs connectés, les utilisateurs souhaitent pouvoir sélectionner un acteur pour voir les informations qui lui sont liées (e.g. rôles, organisations, tâches, projets, etc.), mais aussi de pouvoir explorer ces nouvelles informations (e.g. personnes dans **l'organisation** affichée), et ainsi de suite ;

Contextualiser les informations : dans le secteur AIC, il est très pratique de pouvoir **connaître** le contexte **d'information**. Par exemple, savoir **qu'un** document a été modifié **n'a** pas fondamentalement de sens en soi, mais pouvoir **connaître** le contexte de cette modification (e.g. la météo, **l'info** trafic, les événements sur le chantier, etc.) permet de replacer la situation et de prendre des décisions plus efficaces et rapides. Retracer le contexte **d'une** information permet également d'écartier la responsabilité de certains acteurs. Idéalement, les utilisateurs souhaiteraient pouvoir explorer le contexte de **l'activité** dans le temps, à la manière des modèles 4D qui permettent de visualiser l'évolution dans le temps **d'un** chantier sur une maquette numérique.

4.3.2.4 Synthèse relative à la méthodologie

Convergence de la méthodologie

Dans ces expérimentations, nous avons pu observer une convergence positive dans **l'identification** des besoins des utilisateurs. À travers les scénarios, nous avons trois cas de figure à mettre en avant vis-à-vis des besoins en conscience de groupe :

- Un cas en itération 0, sans besoins prédéfinis ;
- un cas en itération 0+n, avec des besoins qui conviennent déjà à la situation ;
- Un cas en itération 0+n, avec des besoins partiellement ou complètement inadéquats.

Dans chacun de ces cas, à **l'issue** des deux itérations faites sur chacun des scénarios, aucun des utilisateurs **n'a** réajusté ses besoins au-delà **d'une** case dans les questionnaires, avec une majorité de réponses constantes dès la seconde itération. Cet aperçu tend à nous indiquer une bonne convergence de **l'identification** des besoins en conscience de groupe.

La réponse à ces besoins via les widgets est quant à elle plus discutable en raison du point précédent (**l'abstraction** de la conscience de groupe) dont la représentation et la granularité de **l'information** est souvent remise en cause après une première intégration des widgets. Cependant, dans les scénarios où une itération permettait de prendre en compte ces remarques, les modifications effectuées sur les widgets étaient accueillies avec une plus grande satisfaction, ce qui conforte tout de même une convergence de la méthodologie.

Problèmes d'ordre méthodologique liés à la complexité des situations

La méthodologie de conception que nous mettons en avant implique d'établir des modèles de besoins en conscience de groupe assez précis, notamment en termes **d'identification** des situations. Dans le cadre de **l'expérimentation**, nous avons défini cinq scénarios et donc autant de situations dans lesquelles nous demandions aux utilisateurs de se plonger (deux scénarios par utilisateur).

Dans les faits, certains utilisateurs ont éprouvé quelques difficultés à se projeter dans les situations décrites, ou tout du moins à les respecter. Les utilisateurs font alors des écarts aux scénarios et expriment des besoins qui ne correspondent plus aux situations décrites, mais à des situations plus génériques qui englobent les scénarios. Ceci met en avant le caractère parfois vague des situations à identifier et la difficulté pour un utilisateur à cloisonner sa réflexion.

Dès lors, notre vue par situation dans les modèles de besoins semble être trop détaillée et implique de multiplier les scénarios pour être exhaustif alors que des situations plus génériques peuvent exister. Cette multiplication aurait un coût en termes de lourdeur méthodologique, le temps moyen pour couvrir un scénario avec un utilisateur étant de 20 minutes par itération (comprenant le temps **d'intégration** des besoins par le concepteur). Une éventuelle solution serait de rendre plus générale la définition des situations dans les modèles de besoins afin de couvrir des scénarios plus englobants.

À **l'inverse**, les utilisateurs mettent parfois en avant des situations complexes, une information de conscience de groupe pouvant être inutile dans le cas général, mais très utile dès lors que la situation approche un cas particulier. Par exemple, un utilisateur exprime le fait que lors de la préparation **d'une** réunion de chantier, dans la grande majorité des cas il **n'est** pas nécessaire de pouvoir identifier les utilisateurs connectés mais que rarement, un point peut nécessiter un éclaircissement rapide et donc de contacter la personne.

Il convient alors de trouver un équilibre, entre modèles de besoins génériques qui engloberaient un ensemble de situations, et des modèles de besoins sur des situations très précises qui viendraient en cas particulier ou en exception **d'autres** modèles de besoins.

Résumé de section

Pour résumer cette section, nous avons mené une expérimentation afin de valider les aspects relatifs à l'étape 6 de la méthodologie, c'est-à-dire l'identification des besoins en conscience de groupe et leur ré-intégration sous forme de modèles dans le processus de génération.

Au bout de cette expérimentation qui a impliqué des utilisateurs experts du domaine AIC, nous avons relevé les éléments suivants :

Tendances pour la méthodologie :

- Mettre en place des modèles de besoins en conscience de groupe génériques, qui couvriraient des cas de figure couramment rencontrés et ainsi réduire les itérations ;
- Une convergence de la méthodologie, qui au fil de l'expérimentation semble montrer **qu'une** IHM de qualité peut être atteinte en quelques itérations ;
- Des temps de conception amoindris au fil des itérations, ce qui est en partie lié aux points précédents. Une fois un dépôt de widgets bien alimenté, les savoir-faire peuvent être capitalisés pour des temps de conception réduits ;

Tendances dans l'implication des utilisateurs :

- Les utilisateurs anticipent souvent le résultat des questionnaires et l'éventuelle **surcharge d'informations**. Un élément intéressant mais que la méthodologie doit considérer afin d'établir des modèles de besoins complets, la surcharge **d'informations** devant être traitée par adaptation de **l'IHM** ;
- Les utilisateurs ont des préoccupations autour de la gestion de leur vie privée. En effet, dans le domaine AIC, certaines informations de conscience de groupe **n'ont** pas toujours une utilité dans la coordination car celles-ci peuvent engager la responsabilité des utilisateurs. Il convient de pouvoir empêcher ces informations d'être affichées ;

Problèmes décelés :

- La granularité insuffisante de la caractérisation de Gutwin et al. [53, 55], qui dans notre cas ne permet pas via les questionnaires d'élucider explicitement un besoin précis. En effet, le type exact de **l'information** et sa représentation ne sont pas exprimés. De plus, **l'utilisation** brute de cette caractérisation pose des problèmes de compréhension chez les utilisateurs, certains critères étant ambigus. Il convient d'élaborer davantage une abstraction de la conscience de groupe afin de tenir compte de ces éléments ;
- La complexité des situations à modéliser pourrait entraîner une lourdeur méthodologique. Celle-ci requiert que les modèles de besoins soient associés à des situations précises qui sont alors modélisées via le langage de contraintes OCL, un élément qui peut devenir fastidieux **d'autant** que ces situations doivent être identifiées avec les utilisateurs ;

Perspectives suggérées par les utilisateurs :

- Inter-connecter les informations de conscience de groupe, pour que **l'ensemble** de celles-ci soient explorables au besoin, comme par exemple pouvoir afficher les détails **d'un** acteur et les informations qui lui sont associées ;

- Contextualiser les informations, **c'est-à-dire** pouvoir visualiser le contexte **d'un** élément du domaine relatif à son moment de création, comme pouvoir connaître l'état de **l'activité** à l'époque de **l'ajout d'une** remarque. Ceci permet de replacer des éléments dans le contexte pour des prises de décisions plus rapides.

Concernant les hypothèses émises (voir page 50), cette section nous permet de démontrer la validité conceptuelle de **l'hypothèses** H2 (de par la définition de modèles de besoins et de modèles de widgets dans un cas d'étude) ainsi que **l'hypothèse** H3 (pour ce qui est de l'étape 6 de la méthodologie).

4.4 Conclusion de chapitre

Dans ce quatrième chapitre, nous avons instancié les concepts que nous avons mis en avant dans notre proposition en vue **d'obtenir** une validation des éléments de modélisation et de dégager des tendances quant à **l'applicabilité** de la méthodologie de conception. Précédemment, nous avons proposé une architecture MBUI appuyée **d'une** méthodologie de conception pour la génération **d'IHM** venant en support à la coordination. Ces travaux **s'appuient** sur les aspects suivants qui ont fait **l'objet d'une** validation :

- Des métamodèles pour la description du contexte de **l'activité** collective;
- Un processus de transformation pour la génération **d'IHM** adaptatives considérant la conscience de groupe (par insertion de widgets dans **l'arbre** de tâches principal) et les flux de travaux (par génération **d'un** tableau de bord);
- Une méthodologie de conception avec une emphase particulière sur **l'abstraction** de la conscience de groupe et une implication des utilisateurs.

Ainsi, en section 4.1, nous avons présenté le démonstrateur ADAGIOS qui est une suite **d'outils** permettant de modéliser les différentes facettes du contexte de **l'activité** collective. Sur la base **d'un** générateur **d'IHM** existant, GENIUS [104], ADAGIOS permet de produire et interpréter des IHM adaptatives pourvues **d'un** support à la coordination via des widgets de conscience de groupe et un tableau de bord dirigé par un flux de travaux. Ce démonstrateur permet de valider les aspects techniques de nos travaux.

Afin de valider **l'autre** pan de nos propositions, **c'est-à-dire** les aspects méthodologiques et la pertinence des métamodèles dans ceux-ci, nous avons présenté en section 4.2 un cas d'étude concret. Nous avons alors instancié nos concepts en déroulant la méthodologie dans le domaine de **l'Architecture, Ingénierie et Construction (AIC)**, et plus particulièrement à travers un scénario précis autour de **l'activité** de la réunion de chantier. Ce cas d'étude est un bon exemple de contexte mettant en évidence notre problématique initiale. Ainsi, nous avons appliqué la méthodologie dans ses étapes 1 à 5 afin **d'aboutir** à une première génération **d'IHM** dépourvue de support à la conscience de groupe, ce qui atteste de la pertinence de nos propositions jusque là.

Le cas de la conscience de groupe, propre à l'étape 6 de la méthodologie, a fait **l'objet** de la section 4.3 à travers une expérimentation. Cette étape, basée sur le métamodèle de besoins en conscience de groupe, consiste à impliquer les utilisateurs finaux afin **d'identifier** ces besoins via des questionnaires et un cycle itératif. Afin de valider **l'usage** de ces questionnaires et des modèles de besoins en conscience de groupe, cette expérimentation a consisté à impliquer des utilisateurs experts du domaine AIC. Notre intention ici n'était pas d'établir une évaluation quantitative de cette étape de la méthodologie, mais de tirer des tendances via une évaluation qualitative.

Synthèse des résultats

Nous concluons ce chapitre en résumant les résultats des précédentes sections.

Afin de valider les aspects techniques de notre proposition, nous avons présenté la suite **d'outils d'ADAGIOS** faisant office de démonstrateur. Cette suite **d'outils** permet la modélisation des éléments qui constituent le contexte de **l'activité** collective via des éditeurs graphiques de :

- modèles de la collaboration (pour supporter les étapes 1 et 2 de la méthodologie) ;
- modèles de flux de travaux BPMN (pour supporter l'étape 3) ;
- modèles de tâches (pour supporter l'étape 4), et donc aussi de widgets (pour supporter l'étape 6) ;
- modèles de cibles (pour supporter l'étape 5) ;
- modèles de besoins en conscience de groupe (pour supporter l'étape 6) ;

Sur base **d'un** générateur et interpréteur **d'IHM** existant (issu du projet GENIUS [104]), ADAGIOS permet de produire des IHM conformément à nos propositions en composant des widgets (via des modèles de besoins en conscience de groupe) et en générant un tableau de bord (pilote par une interprétation des flux de travaux à **l'exécution**). Cette suite **d'outils** permet de valider les aspects technique de notre proposition, **c'est-à-dire** le processus de transformation supportant la conscience de groupe et les flux de travaux. Ceci permet également de valider techniquement les métamodèles utilisés dans la mesure où ceux-ci permettent la génération **d'IHM** en accord avec nos propositions.

Ces métamodèles et ADAGIOS ont alors été confrontés à une utilisation réelle afin d'évaluer leur pertinence dans un tel cas. Pour cela, nous avons introduit un cas d'étude concret issu du secteur AIC, et plus précisément le scénario **d'une** réunion de chantier. Nous avons alors déroulé la conception **d'IHM** pour ce cas d'étude à travers les étapes 1 à 5 de la méthodologie proposée. En sortie, nous avons pu obtenir des IHM conformes à nos attentes, validant ces aspects méthodologiques ainsi que **l'usage d'ADAGIOS**.

La 6^{ième} étape, qui consiste à identifier les besoins en conscience de groupe en impliquant les utilisateurs dans un cycle itératif, a fait **l'objet d'un** traitement particulier via une expérimentation. Celle-ci a consisté à définir un ensemble de scénarios nous permettant **d'explorer** les cas limites de notre approche. Nous avons cherché à obtenir une validation sur le plan méthodologique, que ce soit du point de vue utilisateur (avec la pertinence des questionnaires) ou concepteur (avec la pertinence de **l'usage** de modèles de besoins). Cette évaluation qualitative nous a permis de mettre en exergue les points suivants dans notre proposition du support de la conscience de groupe :

Des intérêts dans la méthodologie : dans la possibilité de tendre vers des modèles de besoins génériques couvrant des cas de coordination fréquemment rencontrés ; dans la convergence de la méthodologie pour tendre vers une IHM de qualité en quelques itérations ; dans les temps de conception amoindris au fil du peuplement du dépôt de modèles de widgets ;

Des intérêts dans l'implication des utilisateurs : qui comprennent **l'impact** du questionnaire sur l'éventuelle surcharge **d'information** que beaucoup de besoins en conscience de groupe pourraient générer ;

Des problèmes décelés : dans la granularité de la caractérisation de Gutwin et

al. [53, 55], qui dans notre cas ne permet pas via les questionnaires de résoudre explicitement un besoin précis, en plus **d'une** certaine ambiguïté entre certains critères auprès des utilisateurs ; dans la complexité des situations à modéliser (sous forme de contraintes OCL) qui pourrait entraîner une lourdeur méthodologique.

En termes de validation, le démonstrateur ADAGIOS nous a permis de mettre en avant la faisabilité technique de notre proposition et la pertinence de nos métamodèles sur le plan technique, validant ce volet de nos hypothèses H1 et H2. Afin de valider nos métamodèles **d'un** point de vue conceptuel, leur application sur un cas d'étude nous a permis de valider cet aspect sur **l'**hypothèse H1. Ce cas d'étude nous a aussi permis de dégager des observations positives sur la méthodologie de conception **jusqu'en** étape 5, validant **l'**hypothèse H3 jusqu'à cette étape. **L'**expérimentation, quant à elle, avait pour objectif la validation conceptuelle du métamodèle de besoins en conscience de groupe (issu de **l'**hypothèse H2) vis-à-vis de la méthodologie et **l'implication** des utilisateurs dans l'étape 6 (le second volet de validation de **l'**hypothèse H3). Ici, nous avons notamment pointé un manque dans notre abstraction de la conscience de groupe qui présente des limitations pour exprimer la granularité des informations.

Conclusion générale

En conclusion de ces travaux, nous retraçons le cheminement de ce manuscrit de l'établissement de la problématique de recherche aux résultats obtenus. Nous clôturons sur des perspectives de recherche ouvertes par le positionnement et les résultats de nos travaux.

I. Problématique

Pour rappel, ces présents travaux de thèse trouvent leur origine dans l'environnement multi-contextuel dans lequel nous évoluons. Au chapitre 1, nous avons mis en avant la richesse du contexte **d'interaction** dans lequel les utilisateurs interagissent et l'intérêt de sa prise en compte pour les IHM, à la fois dans leur exécution (pour ne pas perdre l'utilisateur) et dans leur conception (pour que les concepteurs puissent en tenir compte). Nous nous sommes ensuite intéressés aux activités collectives qui en termes **d'IHM** ajoutent une complexité supplémentaire. Notre attention **s'est** alors portée sur le support de la coordination dans les IHM en contexte et nous avons à cet effet introduit la notion de contexte de **l'activité** collective (qui englobe les environnements physiques, utilisateurs, matériels et les paramètres de **l'activité**). En nous positionnant sur la conception de telles IHM, nous avons abordé une possible réponse à la conception **d'IHM** en contexte **d'activités** collectives : les *Model-Based User Interfaces* (MBUI). Plus particulièrement, nous avons évoqué les IHM adaptatives qui permettent de tenir compte de la complexité du contexte. Nous avons conclu cette réflexion en posant nos hypothèses de recherche :

- H1. Apporter des métamodèles du contexte de **l'activité** collective en entrée **d'un** processus de génération **d'IHM** permet de produire des IHM adaptatives tenant compte de ces éléments.
- H2. Apporter des métamodèles **d'abstraction** de la pratique de la coordination, notamment de la conscience de groupe et des flux de travaux, permet de générer des IHM mettant en **œuvre** ces principes.
- H3. Une méthodologie de conception tenant compte des hypothèses H1 et H2, **c'est-à-dire** des métamodèles et des mécanismes de génération **d'IHM** associés, permet de couvrir **l'ensemble** de la chaîne de conception **d'IHM** (i.e. de l'élicitation des besoins jusqu'à la production) pour **l'activité** collective.

II. Positionnement

Partant de cette problématique, nous avons scindé notre espace de recherche selon les axes suivants au chapitre 2 :

- La modélisation à haut niveau** qui décrit tout ce qui constitue le contexte de **l'activité collective** (e.g. organisations, flux de travaux, collaboration, tâches, cibles etc.), correspond à une vue théorique sur la coordination ;
- La modélisation à bas niveau** des IHM qui mettent en **œuvre** de manière concrète la coordination (e.g. conscience de groupe, flux de travaux), **c'est-à-dire** les aspects pratiques de la coordination ;
- La méthodologie** de conception générale (e.g. modéliser les besoins métiers, intégration avec les divers intervenants, etc.)

En synthèse de notre état de **l'art** sur ces axes, nous dénotions les perspectives de contributions suivantes répondant à nos hypothèses :

1. **Étendre la modélisation des concepts de haut niveau** : **c'est-à-dire** contribuer à un ensemble de métamodèles décrivant toutes les facettes du contexte de **l'activité collective** telles que nous les avons définies (i.e. **l'activité**, les organisations, les acteurs, les ressources, les outils et le contexte **d'interaction**). Ici, le métamodèle le plus spécifique vis-à-vis de nos besoins nous semble être en provenance **d'AMENITIES** [44] alors que celui le plus propice à **l'IDM** est celui **d'UsiXML** [70]. Un rapprochement nous paraît opportun. Vis-à-vis du cas de la modélisation de tâches collaboratives, le métamodèle de CTML [126, 127, 39] apporte la meilleure réponse IDM pour compléter UsiXML [70], notamment en traitant des problèmes de **l'interprétation** du modèle de tâches à **l'exécution** et de la synchronisation des tâches ;
2. **Établir un support de la coordination** : nous avons relevé ici que la mise en **œuvre** de la conscience de groupe **s'opère** en trois étapes (i.e. identifier des situations, y associer des besoins et répondre à ces besoins) dont TOUCHE [88, 87, 89] et les travaux de Figueroa-Martinez et al [37] apportent des solutions à différents niveaux **d'abstraction**, mais aucune approche **d'ensemble n'a** pu être identifiée. En effet, TOUCHE apporte un abstraction de la conscience de groupe dans les AUI alors que Figueroa-Martinez et al. se penchent sur la modélisation du besoin en information. Il convient alors d'établir des métamodèles et mécanismes de génération mettant en **œuvre** les trois étapes mentionnées ;
3. **Élargir le champ des méthodologies de conception à la génération d'IHM** : les méthodologies étudiées (CIAM [76], AMENITIES [44] et TOUCHE [88, 87, 89]) représentent une solide base pour la conception de collecticiels via des étapes de modélisation bien segmentées et pertinentes dans la considération du contexte de **l'activité collective**. Il serait toutefois intéressant **d'aller** plus loin, à **l'image** de TOUCHE qui propose de modéliser les IHM. Cependant, aucune approche ne semble aller jusqu'à la génération effective des IHM, tout comme le manque de considération de **l'adaptation**. Pour compléter cette démarche, des travaux tels que ceux autour

d’**UsiXML** [70], dont WSL [121] et les travaux de Figueroa-Martinez et al. [37] nous apportent des éléments pour étendre une méthodologie au support de la conscience de groupe, notamment via des modèles d’**abstraction** des besoins en information sur le groupe.

III. Résumé des contributions

Pour répondre aux limitations de l’état de l’**art**, et sur base principale d’**UsiXML** [70] et de TOUCHE [88, 87, 89], nous avons proposé en chapitre 3 un cadre de conception d’**IHM** adaptatives par MBUI. Ce cadre se concentre sur le support de la coordination et s’**insère** dans une méthodologie générale afin de documenter la construction des modèles associés. Nous relevons les contributions suivantes, dont la figure C.1 reprend la synthèse technique :

Des **métamodèles pour la modélisation de l’activité collective** (voir point A en figure C.1) qui permettent de définir les organisations, les acteurs, les tâches, l’**activité**, les ressources, les outils et le domaine. Ces éléments constituent les modèles d’**entrée** du processus de génération des IHM en fournissant des **paramètres d’adaptation** adéquats au contexte de l’activité collective ainsi que le support de la méthodologie de modélisation ;

La **prise en compte de la coordination** qui passe par des modèles intermédiaires qui s’**insèrent** dans le processus de génération afin de supporter les éléments suivants :

- **La conscience de groupe** (voir point B en figure C.1), via une approche par génération de widgets (i.e. fragments d’**IHM** dédiés à une tâche spécifique) et l’**abstraction** des besoins en information de conscience de groupe. Cette approche se base sur le mécanisme de mise en **œuvre** de la conscience de groupe (i.e. identifier des situations, y associer des besoins et répondre à ces besoins). Ainsi, nous avons défini un métamodèle de besoins qui cible une situation (i.e. contraintes sur le contexte exprimées en OCL) et une tâche principale auxquels des besoins sont associés suivant la caractérisation de Gutwin et al. [53, 55]. Pour répondre à ces besoins, notre approche est de reposer sur des widgets fournissant les informations qui répondent aux besoins exprimés (e.g. en affichant la photo des utilisateurs si un besoin d’**embodiment** existe) et décrits par des modèles de tâches afin de pouvoir en générer les IHM à travers le même processus que l’**application** principale (et donc d’être adaptables au même titre que le reste de l’**IHM**). Le lien entre les modèles de besoins et les modèles de widgets est opéré par un dépôt de modèles de widgets qui détermine lesquels répondent aux besoins en conscience de groupe. Ces modèles sélectionnés sont alors composés par transformation avec le modèle de tâche principal afin de former un unique arbre de tâches pour générer une IHM complète ;
- **Les flux de travaux** (voir point C en figure C.1), via des associations entre les modèles de tâches et les activités d’un modèle BPMN pour la synchronisation et le pilotage des flux de travaux. Le modèle BPMN est par ailleurs transformé en modèle de tâches décrivant ainsi un tableau de bord (ou extra-IHM) qui régit

L'accès aux IHM générées en accord avec l'état du flux de travaux interprété à l'exécution.

L'insertion de ces éléments dans une méthodologie de conception qui permet de déterminer la manière dont sont construits les modèles précédents. Une attention particulière est portée sur le support de la conscience de groupe en introduisant un cycle itératif de modélisation des besoins relatifs en impliquant les utilisateurs. Cette méthodologie se base sur les étapes communes à **d'autres** travaux que sont CIAM [76], AMENITIES [44] et TOUCHE [88, 87, 89] en y apportant des étapes propres à **l'adaptation** au contexte **d'interaction** et à la conscience de groupe (i.e les étapes 5 et 6). Elle se décompose en étapes qui consistent à modéliser les éléments suivants :

1. Le sociogramme de **l'activité** (i.e. les organisations, rôles et acteurs) ;
2. Les relations entre tâches et rôles ;
3. **L'activité** en elle-même, par flux de travaux ;
4. Les tâches (i.e. arbres de tâches) ;
5. Les cibles (i.e. le contexte **d'interaction**, pour supporter une adaptation) ;
6. Les besoins en conscience de groupe, de manière itérative avec **l'implication d'utilisateurs** finaux. Ceci passe par **l'identification** de ces besoins sur base **d'un** questionnaire issu de la caractérisation de Gutwin et al. [53, 55], puis par la validation des IHM générées, sur base **d'un** questionnaire construit sur la même caractérisation afin d'évaluer la réponse donnée en termes de widgets.

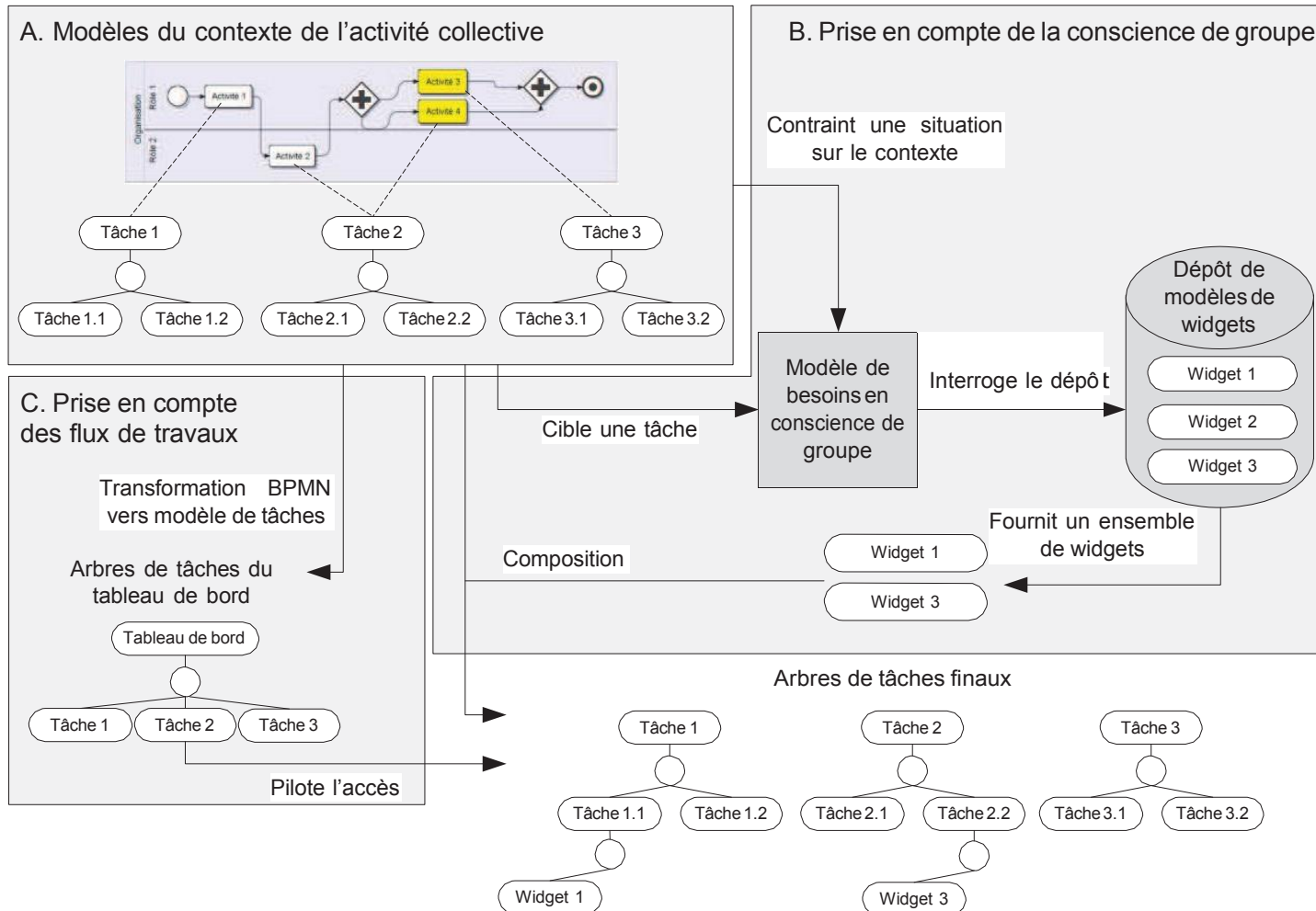


Figure C.1 – Synthèse de la structure du cadre de modélisation et de génération d'IHM proposé.

IV. Résultats

Enfin, au chapitre 4, nous présentons les résultats. **D'un** point de vue technique, nous avons démontré la faisabilité de notre approche via le démonstrateur ADAGIOS développé dans le cadre de cette thèse. Celui-ci fournit des outils pour la modélisation du contexte de **l'activité** collective ainsi que les éléments suivants :

Des éditeurs graphiques pour la modélisation de :

- la collaboration ;
- des flux de travaux BPMN ;
- des tâches et de widgets ;
- des tâches ;
- des cibles ;
- des besoins en conscience de groupe ;

Un environnement technique, basé sur un projet existant (GENIUS [104]) :

- pour la génération des CUI, dont la prise en compte des modèles de besoins en conscience de groupe (appliqués par des widgets) et les flux de travaux (appliqués par un tableau de bord) ;
- et **l'interprétation** de ces CUI dans un environnement web, afin de pouvoir maintenir les modèles à **l'exécution**.

Nous avons ensuite soumis ADAGIOS à la méthodologie de conception proposée à travers un cas d'étude concret dans le secteur de **l'Architecture, Ingénierie et Construction**. Ce secteur est représentatif de la complexité du contexte de **l'activité** collective. En effet, sur les différentes phases **d'un** projet de construction (i.e. études préalables, avant-projet, appel **d'offre**, planification, chantier, vie de **l'ouvrage**), de nombreux acteurs interviennent et les organisations peuvent revêtir différentes formes suivant ces phases. En sus, les acteurs sont mobiles sur des contextes **d'interaction** hétérogènes (e.g. chantier, bureau d'étude, etc.) et se coordonnent de différentes manières en fonction de la tâche (e.g. la coordination est informelle sur chantier, mais le suivi des travaux doit être tracé précisément). Via ce cas d'étude, nous avons instancié les concepts **d'un** scénario particulier, celui de la réunion de chantier et plus particulièrement en nous focalisant sur les tâches de gestion documentaire des comptes-rendus de ces réunions hebdomadaires. Ceci nous a permis de dérouler correctement la méthodologie de l'étape 1 à 5 avec en sortie une première génération **d'IHM** conforme au résultat attendu du cas d'étude.

L'étape 6 de la méthodologie a fait **l'objet d'une** validation dédiée car, contrairement aux étapes précédentes, elle implique les utilisateurs. Cette étape consiste à identifier les besoins en conscience de groupe de manière itérative avec des utilisateurs finaux. Cette partie de la méthodologie est celle qui concerne les plus fortes propositions. Nous avons donc mis en place une expérimentation afin **d'explorer** les cas limites de la méthodologie et la pertinence du métamodèle de besoins employé. Les résultats de **l'expérimentation** ont permis de dégager les tendances et problèmes suivants :

Intérêt de la méthodologie :

- La possibilité de mettre en place des modèles de besoins en conscience de groupe génériques, qui couvrent des cas de figure couramment rencontrés et ainsi réduire les itérations ;

-
- Une convergence de la méthodologie, qui au fil de l'expérimentation semble montrer **qu'une** IHM répondant aux besoins de conscience de groupe peut être atteinte en quelques itérations;
 - Des temps de conception amoindris au fil des itérations, ce qui est en partie lié aux points précédents. Une fois un dépôt de modèles de widgets bien alimenté, les savoir-faire peuvent être capitalisés pour des temps de conception réduits;

Intérêt de l'implication des utilisateurs :

- Les utilisateurs anticipent souvent le résultat des questionnaires **d'identification** des besoins en conscience de groupe et l'éventuelle surcharge **d'informations**. Un élément intéressant que la méthodologie doit considérer afin d'établir des modèles de besoins complets, la surcharge **d'informations** devant être traitée par adaptation de **l'IHM**;
- Les utilisateurs ont des préoccupations autour de la gestion de leur vie privée. En effet, dans le domaine AIC, certaines informations de conscience de groupe **n'ont pas toujours** une utilité dans la coordination car celles-ci peuvent engager la responsabilité des utilisateurs (e.g. la géo-localisation). Il convient de pouvoir empêcher ces informations d'être affichées;

Problèmes décelés :

- La granularité insuffisante de la caractérisation de Gutwin et al. [53, 55], qui dans notre cas ne permet pas via les questionnaires d'éliciter explicitement un besoin précis. En effet, le type exact de **l'information** et sa représentation ne sont pas exprimés. De plus, **l'utilisation** brute de cette caractérisation pose des problèmes de compréhension chez les utilisateurs, certains critères étant ambiguës (e.g. **action** et **event history**). Il convient d'élaborer davantage une abstraction de la conscience de groupe afin de tenir compte de ces éléments;
- La complexité des situations à modéliser pourrait entraîner une lourdeur méthodologique. Celle-ci requiert que les modèles de besoins soient associés à des situations précises qui sont alors modélisées via le langage de contraintes OCL⁵⁰, un élément qui peut devenir fastidieux **d'autant** que ces situations doivent être identifiées avec les utilisateurs;

En termes de validation, le démonstrateur ADAGIOS nous a permis de mettre en avant la faisabilité technique de notre proposition et la pertinence de nos métamodèles sur le plan technique, validant ce volet de nos hypothèses H1 et H2.

Afin de valider nos métamodèles **d'un** point de vue conceptuel, leur application sur un cas d'étude nous a permis de valider cet aspect sur **l'hypothèse H1**. Ce cas d'étude nous a aussi permis de dégager des observations positives sur la méthodologie de conception **jusqu'en** étape 5, validant **l'hypothèse H3 jusqu'à** cette étape.

L'expérimentation, quant à elle, avait pour objectif la validation conceptuelle du métamodèle de besoins en conscience de groupe (issu de **l'hypothèse H2**) vis-à-vis de la méthodologie et **l'implication** des utilisateurs dans l'étape 6 (le second volet de validation de **l'hypothèse H3**). Ici, nous avons notamment pointé un manque dans notre abstraction de la conscience de groupe qui présente des limitations pour exprimer la granularité des

50. <http://www.omg.org/spec/OCL/>

informations. Cette expérimentation a également servi à mettre en évidence les limites et perspectives qui suivent.

V. Perspectives

Sur base des limitations identifiées et des résultats de nos travaux, nous énumérons un certain nombre de perspectives de recherche.

À court terme

Caractériser la conscience de groupe plus finement

La perspective qui nous semble la plus intéressante est celle qui concerne **l'usage** de la caractérisation de Gutwin et al. [53, 55] dans nos propositions relatives à la conscience de groupe. Dans nos travaux, nous avons utilisé cette caractérisation telle quelle, sans prétendre à des propositions de **l'ordre** du TCAO. Nous avons basé nos métamodèles de besoins en conscience de groupe sur celle-ci et **l'avons** utilisé pour générer des widgets adéquats. Dans **l'expérimentation** en section 4.3.2, nous avons relevé des limitations dans notre usage de cette caractérisation. En effet, celle-ci est trop abstraite et ne permet pas de définir le niveau de granularité de **l'information** à afficher. Nous pensons **qu'en** travaillant sur une abstraction de la conscience de groupe plus détaillée et adaptée à notre problème, mais toujours dans la mouvance des travaux de Gutwin et al. [53, 55], il soit possible de palier à cette limitation. Ceci implique également de revoir la méthodologie et **l'implication** de **l'utilisateur**.

Améliorer la prise en compte de l'adaptation

Dans notre approche de **l'adaptation**, nous avons simplement mis en **œuvre** des mécanismes connus, notamment via le traitement de **l'importance d'une** tâche ou **d'un** concept du domaine. Notre but initial était de fournir une adaptation de la présentation en fonction des cibles, et une adaptation fonctionnelle en fonction de l'état de **l'activité**.

Une perspective réside dans **l'amélioration** du support de **l'adaptation**, en couvrant davantage **l'espace** problème de la plasticité et en reposant sur des heuristiques plus spécifiques. Il serait par exemple intéressant d'étendre la notion **d'importance** avec un modèle sémantique du domaine, afin de mettre en place des heuristiques **d'adaptation** plus poussées. Dans le cas des widgets de conscience de groupe, il serait intéressant de pouvoir lier un concept à un autre (e.g. une photo représente un **embodiment**, mais peut sémantiquement permettre une **identification**) et **d'adapter** les concepts en fonction de leur lien sémantique. Ce modèle sémantique pourrait être mis en commun avec une caractérisation de la conscience de groupe adaptée, comme spécifié au point précédent.

Traiter plus finement la composition de widgets

Dans notre proposition de cadre technique, nous avons exposé le principe de composition de widgets avec un modèle de tâches principal. **C'est** au dépôt de modèles de widgets

qu'incombe la fonction de sélectionner les widgets qui répondent au mieux aux besoins exprimés. Nous avons ici laissé ouvert le choix de **l'algorithme** de sélection et dans notre validation technique à travers ADAGIOS, nous avons mis en place un algorithme glouton non optimal. Dans le traitement du cas d'étude, nous **n'**avons pas pu entrevoir de limites à ce choix, étant donné le nombre de widgets que nous avons modélisé.

En prévision de cas plus complexes, il convient de déterminer un algorithme de sélection de widgets efficace, qui fournit des réponses optimales en termes de couverture des besoins et de minimisation de la quantité **d'information**. Des réponses peuvent exister dans le domaine de **l'optimisation**, notamment en regardant du côté des algorithmes multi-objectifs.

À moyen terme

Tendre vers un dépôt de modèles de widgets génériques

Après traitement des perspectives susmentionnées, et en supposant qu'elles trouvent une issue favorable, il serait intéressant d'étudier la pertinence de widgets et de modèles de besoins en conscience de groupe génériques. En effet, lors de notre expérimentation, les résultats nous ont laissé entrevoir la possibilité que bon nombre de widgets pourraient être génériques à travers une multitude de situations. Approvisionner un dépôt de modèles de widgets avec un ensemble générique pourrait permettre de répondre rapidement et efficacement à des situations variées, et donc de réduire les temps de conception et **d'itération**. De la même manière, certaines situations (et les besoins associés) pourraient être génériques et être capitalisées dans des modèles de besoins. La perspective étant de pouvoir proposer des IHM ayant déjà un support de la conscience de groupe à la première génération de celles-ci, **c'est-à-dire** en sortie d'étape 5 de la méthodologie, juste avant d'évaluer les besoins.

Automatiser la prise en compte des besoins en conscience de groupe

À supposer **qu'une** caractérisation précise de la conscience de groupe le permette, nous entrevoyons la perspective **d'intégrer l'identification** des besoins utilisateurs directement à **l'IHM**. Dans l'état actuel de nos travaux, cette identification passe par des questionnaires, eux-mêmes basés sur le métamodèle de besoins. Le travail du concepteur consiste à traduire ces questionnaires quasi à **l'identique** en modèles. Dans le cas où nous arrivons à tendre vers une caractérisation efficace de la conscience de groupe, le travail du concepteur pourrait être automatisé, en intégrant par exemple un panneau de retours utilisateurs aux IHM générées, permettant aux utilisateurs **d'exprimer** leurs besoins au cas par cas. Un modèle de besoins pourrait alors être créé pour ensuite régénérer **l'IHM** et proposer un résultat à **l'utilisateur** de manière instantanée, lui permettant ainsi **d'avoir** un contrôle fin sur la génération des IHM.

Traiter la question de la coopération synchrone et des espaces partagés

Dans nos travaux, nous nous sommes concentrés sur le support de la coordination, en traitant par effet de bord une partie du reste du modèle 3C [34]. Par ailleurs, nous

n'avons pas traité tous les pans de la matrice de Johansen [63] dont notamment le travail en espace partagé.

En termes de perspective, il nous semble essentiel de couvrir ces questions, notamment dans le but **d'apporter** un support de la collaboration synchrone et des espaces partagés dans notre approche générative. Des éléments de réponse se situent notamment dans les travaux de Penichet et al. sur TOUCHE [88, 87, 89] qui propose de caractériser les AUI et CUI avec des propriétés de coopération synchrone. De plus, les méthodologies dont nous nous sommes inspirés (principalement CIAM [76] et TOUCHE [88, 87, 89]) proposent de caractériser les tâches suivant la matrice de Johansen, permettant ainsi **d'avoir** une abstraction du type de collaboration attendu dans la génération de l'IHM. La littérature relative à la conscience de groupe aborde également ces aspects, avec un champ de recherche dédié, la conscience de groupe en espace partagé (*workspace awareness*).

À long terme

Évaluer quantitativement la méthodologie

Valider une méthodologie de conception est un travail complexe. Dans notre évaluation, nous nous sommes restreints à une analyse qualitative, en dégagant des tendances. Pour être pertinent, il serait nécessaire de conduire une étude comparative avec **d'autres** méthodologies (telles que des approches non-IDM), **d'avoir** un protocole et des points de comparaison, et surtout **d'avoir** les ressources humaines suffisantes (concepteurs et utilisateurs) pour avoir un échantillon significatif sur un même cas d'étude. Ceci représente une étude considérable, dont la faisabilité reste **aujourd'hui** en suspens.

Évaluer l'apport des IHM adaptatives dans la coordination

Nous **n'avons** pas abordé le sujet de la mesure de **l'apport** des IHM adaptatives pour la coordination. Dans nos travaux, nous nous sommes concentrés sur la conception de telles IHM, et non sur leur évaluation. Une perspective intéressante serait alors de mesurer **l'impact** de **l'adaptation** des IHM dans une activité collective et de déterminer si elle améliore les pratiques. Cette perspective dépend de celle relative à l'évaluation de **l'utilisabilité** des IHM adaptatives, un problème déjà complexe en soi. Une fois cette base acquise, il convient de mener une étude comparative entre IHM en support à la coordination, avec et sans adaptation.

Fin.

Annexe A

Identification et validation des besoins en conscience de groupe

A.1 Identification des besoins en conscience de groupe

Vis-à-vis du rôle, du lieu et de la tâche qui vous sont donnés, les informations suivantes vous paraissent-elles utiles ?

	Pas du tout utile	Pas vraiment utile	Plutôt utile	Très utile	Indispensable
Savoir si d'autres personnes sont connectées	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Connaître l'identité des personnes connectées	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pouvoir identifier visuellement les personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Connaître la disponibilité des autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que font les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir où se situent les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir où ont été les autres personnes dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quand étaient connectées les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce qu'ont fait dernièrement les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que regardent les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que peuvent voir les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir si une ressource est actuellement manipulée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quand et par qui une ressource a été manipulée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quelles actions ont été réalisées dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quels événements se sont produits dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

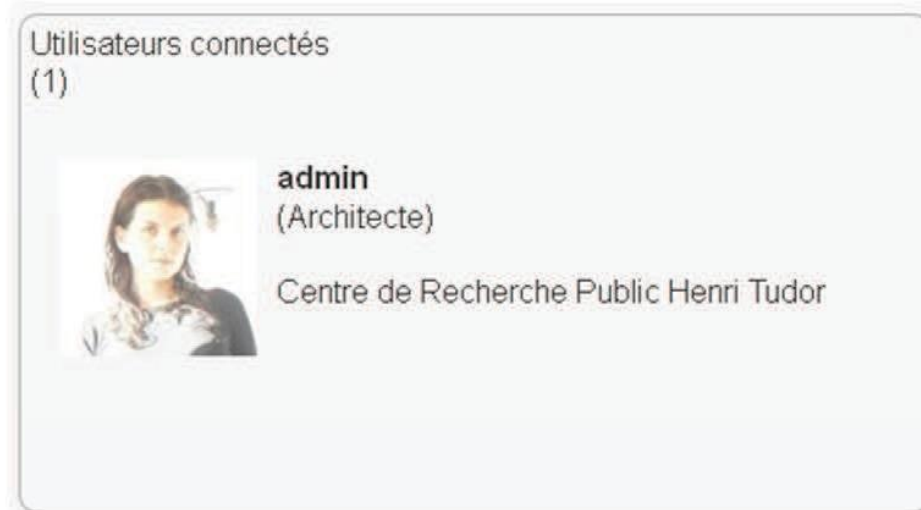
A.2 Validation des besoins en conscience de groupe

Vis-à-vis des informations présentes, trouvez-vous leur représentation adéquate dans le cas qui vous a été présenté ?

	Non applicable	Pas vraiment adéquat	Plutôt adéquat	Très adéquat	Parfait
Savoir si d'autres personnes sont connectées	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Connaître l'identité des personnes connectées	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pouvoir identifier visuellement les personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Connaître la disponibilité des autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que font les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir où se situent les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir où ont été les autres personnes dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quand étaient connectées les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce qu'ont fait dernièrement les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que regardent les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir ce que peuvent voir les autres personnes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir si une ressource est actuellement manipulée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quand et par qui une ressource a été manipulée	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quelles actions ont été réalisées dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Savoir quels événements se sont produits dernièrement	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

A.3 Validation des représentations des widgets

Êtes-vous satisfait des widgets présents dans l'interface ?



Pas du tout satisfait	Pas vraiment satisfait	Plutôt satisfait	Très satisfait
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si vous n'êtes pas complètement satisfait, avez-vous des suggestions pour améliorer ce widget ?

Webcam du chantier
Bureaux Concorde



Pas du tout satisfait	Pas vraiment satisfait	Plutôt satisfait	Très satisfait
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si vous n'êtes pas complètement satisfait, avez-vous des suggestions pour améliorer ce widget ?

4 nouvelle(s) action(s)

24/04 12:00 Jérôme C. s'est déconnecté
24/04 11:55 Jérôme C. a modifié la remarque "Remarque sur la toiture"
24/04 11:50 Jérôme C. a ajouté la remarque "Remarque sur la toiture"
24/04 11:45 Jérôme C. s'est connecté
23/04 16:14 Thomas A. s'est déconnecté
23/04 14:32 Thomas A. a créé le compte-rendu "Compte-rendu n1"
23/04 14:30 Thomas A. s'est connecté

Pas du tout satisfait	Pas vraiment satisfait	Plutôt satisfait	Très satisfait
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Si vous n'êtes pas complètement satisfait, avez-vous des suggestions pour améliorer ce widget ?

Annexe B

Fiches de scénario pour l'expérimentation

Scénario ①

Particularité(s) :

Mauvais support de la conscience de groupe. Une partie des widgets présents n'apporte pas d'informations relatives à la tâche courante.

Mise en situation :

« Vous animez une réunion de chantier et tout les intervenants sont présents autour de la table. »

Tâche demandée : lister les remarques du compte-rendu n°1 du projet « Bureau concorde. »

Contexte :

Activité = Animer la réunion de chantier

Localisation = Chantier

Matériel = Tablette

Rôle = Architecte

Écran(s)/Tâche(s) concerné(s) :

« Consulter la liste des remarques. »

Widgets présents :

Webcam

Feedback / résultats anticipés :

Modification de la règle déjà en place. Disparition du widget webcam.

Cas couverts :

③ ④'

Scénario②

Particularité(s) :

Le support de la conscience de groupe est idéal. Une règle est déjà en place et répond bien au besoin.

Mise en situation :

« Vous préparez une réunion de chantier au bureau et vous devez contacter un intervenant. »

Tâche demandée : consulter l'**annuaire** du projet « bureau concorde. »

Contexte :

Activité = Préparer la réunion de chantier

Localisation = Bureau d'étude

Matériel = Ordinateur de bureau

Rôle = Architecte

Écran(s)/Tâche(s) concerné(s) :

« Consulter l'**annuaire**. »

Widgets présents :

Log des actions

Feedback / résultats anticipés :

Une règle existante correspond et ne nécessite pas de modification.

Cas couverts :

①

Scénario ③

Particularité(s) :

Il n'y a pas de règle pré-définie et il y a un besoin de support à la coordination.

Mise en situation :

« Au bureau, vous souhaitez consulter les remarques relatives à une entreprise afin de préparer la prochaine réunion de chantier. »

Tâche demandée : dans le projet « Construction Centre de Recherche Henri Tudor », consultez la remarque « problème de livraison » du compte-rendu n°2.

Contexte :

Activité = Lit le compte rendu

Localisation = Bureau d'étude

Matériel = Ordinateur de bureau

Rôle = Maître d'œuvre

Écran(s)/Tâche(s) concerné(s) :

« Consulter une remarque. »

Widgets présents :

Aucun (pas de règle en place).

Feedback / résultats anticipés :

Une règle est créée pour répondre au besoin. La base de données ne contient pas l'intégralité des widgets nécessaires (i.e. utilisateurs connectés, webcam). Il est nécessaire d'en ajouter ou d'en modifier.

Cas couverts :

② ⑤ ⑥

Scénario④

Particularité(s) :

Un widget est en place, mais il **n'y** a aucun besoin en conscience de groupe.

Mise en situation :

« Vous êtes sur chantier et vous êtes sur le point de vous connecter à CRTI-Web. »

Tâche demandée : aller sur l'écran de connexion

Contexte :

Activité = Conduit la visite de chantier

Localisation = Sur chantier

Matériel = Smartphone

Rôle = Architecte

Écran(s)/Tâche(s) concerné(s) :

« Identification. »

Widgets présents :

Webcam.

Feedback / résultats anticipés :

Suppression de la règle, disparition des widgets.

Cas couverts :

⑦

Scénario ⑤

Particularité(s) :

Il n'y a pas de règle en place et aucun besoin en conscience de groupe.

Mise en situation :

« Vous êtes sur chantier et vous êtes sur le point de vous connecter à CRTI-Web. »

Tâche demandée : aller sur l'écran de connexion

Contexte :

Activité = Conduit la visite de chantier

Localisation = Sur chantier

Matériel = Smartphone

Rôle = Architecte

Écran(s)/Tâche(s) concerné(s) :

« Identification. »

Widgets présents :

Aucun.

Feedback / résultats anticipés :

Aucune action.

Cas couverts :

①

Bibliographie

- [1] Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26, 11 (1983), 832–843.
- [2] Altenburger, T. Vers une caractérisation des systèmes sensible au contexte : étude de leur impact sur l'adaptation des interfaces. In *Conférence Internationale Francophone sur l'Interaction Homme-Machine* (2010), ACM.
- [3] Altenburger, T. Interfaces adaptatives pour des contextes hautement collaboratif et mobile. In *Conférence Internationale Francophone sur l'Interaction Homme-Machine* (2011), ACM.
- [4] Altenburger, T., Guerriero, A., Vagner, A., and Martin, B. Toward adaptive context-aware user interfaces for better usability and productivity in aec collaborative tasks. In *CIB W78 2010 : 27th International Conference on Applications of IT in the AEC Industry, November 16-19, 2010, Cairo, Egypt* (2010).
- [5] Altenburger, T., Guerriero, A., Vagner, A., and Martin, B. Groupware requirements modelling for adaptive user interface design. In *9th European Conference on Product and Process Modelling (ECPPM 2012), July 25-27th, 2012, Reykjavik, Iceland* (2012).
- [6] Bardram, J., and Hansen, T. The aware architecture : supporting context-mediated social awareness in mobile cooperation. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (2004), ACM, pp. 192–201.
- [7] Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, R., Sheppard, S., and Szezur, M. R. The arch model : seeheim revisited. In *User Interface Developers' Workshop* (1991).
- [8] Belkadi, F., Bonjour, E., Camargo, M., Troussier, N., and Eynard, B. A situation model to support awareness in collaborative design. *International Journal of Human-Computer Studies* 71, 1 (2013), 110 – 129.
- [9] Blouin, A., and Beaudoux, O. Improving modularity and usability of interactive systems with malai. In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems* (New York, NY, USA, 2010), EICS '10, ACM, pp. 115–124.
- [10] Blouin, A., Morin, B., Beaudoux, O., Nain, G., Albers, P., and Jézéquel, J.-M. Combining aspect-oriented modeling with property-based reasoning to improve user interface adaptation. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems* (2011), ACM, pp. 85–94.

- [11] Blumendorf, M., Feuerstack, S., and Albayrak, S. Multimodal user interaction in smart environments : Delivering distributed user interfaces. *Constructing Ambient Intelligence* (2008), 113–120.
- [12] Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., and Tanca, L. A data-oriented survey of context models. *SIGMOD Rec.* **36**, 4 (2007), 19–26.
- [13] Bomsdorf, B. The webtaskmodel approach to web process modelling. In *Task Models and Diagrams for User Interface Design*. Springer, 2007, pp. 240–253.
- [14] Boukhebouze, M., Neto, W., Lim, E., and Thiran, P. UsiWSC : Framework for supporting an interactive web service composition. In *Web Engineering*, M. Brambilla, T. Tokuda, and R. Tolksdorf, Eds., vol. 7387 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 461–464.
- [15] Boukhebouze, M., Neto, W. P. F., and Erbin, L. Yet another bpel extension for user interactions. In *Proceedings of the 30th international conference on Advances in conceptual modeling : recent developments and new directions* (Berlin, Heidelberg, 2011), *ER'11*, Springer-Verlag, pp. 24–33.
- [16] Bourguin, G. *Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le projet DARE*. PhD thesis, Université des Sciences et Technologies de Lille, 2000.
- [17] Bowden, S. L. *Application of Mobile IT in Construction*. PhD thesis, Loughborough University, 2005.
- [18] Brézillon, P. Expliciter le contexte dans les objets communicants. *Les Objets Communicants* **21** (2002), 295–303.
- [19] Browne, D., Totterdell, P., and Norman, M. *Adaptive user interfaces*. Academic Press Ltd., London, UK, UK, 1990.
- [20] Calvary, G. *Plasticité des Interfaces Homme-Machine*. PhD thesis, 2007. Thèse Habilitation à Diriger des Recherches préparée au Laboratoire d'Informatique de Grenoble (LIG), Université Joseph Fourier.
- [21] Calvary, G., Coutaz, J., and Thevenin, D. A unifying reference framework for the development of plastic user interfaces. In *IFIP WG2.7 (13.2) Working Conference, EHCI01, Toronto, Springer Verlag Publ., LNCS 2254, M. Reed Little, L. Nigay Eds* (2001), pp. 173–192.
- [22] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A unifying reference framework for multi-target user interfaces. *Interacting With Computers Vol. 15/3* (2003), 289–308.
- [23] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., and Vanderdonckt, J. Plasticity of user interfaces : A revised reference framework. In *First International Workshop on Task Models and Diagrams for User Interface Design TAMODIA'2002, Bucarest* (2002), pp. 127–134.
- [24] Caminero, J., Rodríguez, M. C., Vanderdonckt, J., Paternò, F., Rett, J., Raggett, D., Comeliau, J.-L., and Marín, I. The serenoa project : Multidimensional context-aware adaptation of service front-ends.

-
- [25] Chesbrough, H. W. *Open innovation : The new imperative for creating and profiting from technology*. Harvard Business Press, 2003.
- [26] Coutaz, J., Balme, L., Calvary, G., Demeure, A., and Sottet, J.-S. An MDE-SOA approach to support plastic user interfaces in ambient spaces. In *Proc. HCI International 2007* (2007), pp. 152–171. Beijing, July 2007.
- [27] De Alwis, B., Gutwin, C., and Greenberg, S. Gt/sd : performance and simplicity in a groupware toolkit. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems* (2009), ACM, pp. 265–274.
- [28] De Bra, P., Brusilovsky, P., and Houben, G. Adaptive hypermedia : from systems to framework. *ACM Computing Surveys (CSUR)* **31**, 4es (1999), 12.
- [29] De Farias, C., Pires, L., and Van Sinderen, M. A conceptual model for the development of CSCW systems. In *Designing cooperative systems : the use of theories and models : proceedings of the 5th International Conference on the Design of Cooperative Systems (COOP'2000)* (2000), Los Pr Inc, p. 189.
- [30] Dey, A., and Abowd, G. The context toolkit : Aiding the development of context-aware applications. In *Workshop on Software Engineering for wearable and pervasive computing* (2000), Citeseer, pp. 431–441.
- [31] Dey, A. K., and Abowd, G. D. Towards a better understanding of context and context-awareness. Tech. rep., Georgia Tech, 1999.
- [32] Dourish, P. Developing a reflective model of collaborative systems. *ACM Transactions on Computer-Human Interaction (TOCHI)* **2**, 1 (1995), 40–63.
- [33] Dourish, P., and Bellotti, V. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work* (1992), ACM, pp. 107–114.
- [34] Ellis, C. A., Gibbs, S. J., and Rein, G. Groupware : some issues and experiences. *Commun. ACM* **34** (January 1991), 39–58.
- [35] Endsley, M. Toward a theory of situation awareness in dynamic systems. *Human Factors : The Journal of the Human Factors and Ergonomics Society* **37**, 1 (1995), 32–64.
- [36] Favre, J.-M., Estublier, J., and Blay-Fornarino, M. *L'ingénierie dirigée par les modèles au-delà du MDA*. Hermès Science Publications Paris, 2006.
- [37] Figueroa-Martinez, J., López-Jaquero, V., Vela, F. L. G., and González, P. Enriching UsiXML language to support awareness requirements. *Science of Computer Programming* (2012).
- [38] Fitzpatrick, G., Tolone, W. J., and Kaplan, S. M. Work, locales and distributed social worlds. In *Proceedings of the 1995 European Conference on Computer-Supported Cooperative Work (ECSCW 95)* (1995), pp. 1–16.
- [39] Forbrig, P., Dittmar, A., Brüning, J., and Wurdel, M. Making task modeling suitable for stakeholder-driven workflow specifications. In *Universal Access in Human-Computer Interaction. Design for All and eInclusion*. Springer, 2011, pp. 51–60.

- [40] Fuks, H., Raposo, A., Gerosa, M. A., Pimental, and Mariano. *Encyclopedia of E-Collaboration*. 2008, ch. The 3C Collaboration Model.
- [41] Gallardo, J., Molina, A. I., Bravo, C., Redondo, M. A., and Collazos, C. A. An ontological conceptualization approach for awareness in domain-independent collaborative modeling systems : Application to a model-driven development method. *Expert Systems with Applications* **38**, 2 (2011), 1099–1118. Intelligent Collaboration and Design.
- [42] Garcia, J. G., and Vanderdonckt, J. Designing workflow user interfaces with UsiXML. *UsiXML 2010* (2010).
- [43] Garrido, J. Especificación de la notación como-uml. Tech. rep., Tech. Rep. No. LSI-2003-2, Granada, Spain, University of Granada, Departamento de Lenguajes y Sistemas Informáticos, 2003.
- [44] Garrido, J., Gea, M., and Rodríguez, M. Requirements engineering in cooperative systems. *Requirements Engineering for Sociotechnical Systems* (2005), 226–244.
- [45] Garrido, J., Noguera, M., González, M., Hurtado, M., and Rodríguez, M. Definition and use of computation independent models in an mda-based groupware development process. *Science of Computer Programming* **66**, 1 (2006), 25–43.
- [46] Garrido, J. E., Penichet, V. M. R., and Lozano, M. D. Improving ubiquitous environments through collaborative features. In *Distributed User Interfaces*, J. A. Gallud, R. Tesoriero, and V. M. Penichet, Eds., Human-Computer Interaction Series. Springer London, 2011, pp. 59–66.
- [47] Green, T., and Hoc, J.-M. What is cognitive ergonomics? *Le Travail Humain* (1991), 291–304.
- [48] Greenberg, S., Roseman, M., Webster, D., and Bohnet, R. Human and technical factors of distributed group drawing tools. *Interacting with Computers* **4**, 3 (1992), 364–392.
- [49] Greenberg, S., and Witten, I. H. Adaptive personalized interfaces-a question of viability. *Behaviour & Information Technology* **4**, 1 (1985), 31–45.
- [50] Griffiths, T., McKirdy, J., Paton, N., Kennedy, J., Cooper, R., Barclay, P., Goble, C., Gray, P., Smyth, M., West, A., et al. An open model-based interface development system : The teallach approach. In *Proceedings of DSV-IS* (1998), vol. 98, Citeseer, pp. 32–49.
- [51] Guerrero Garcia, J., Vanderdonckt, J., and Gonzalez Calleros, J. M. FlowiXML : a step towards designing workflow management systems. *International Journal of Web Engineering and Technology* **4**, 2 (2008), 163–182.
- [52] Gutwin, C. *Workspace awareness in real-time distributed groupware*. PhD thesis, University of Calgary, Calgary, AB, 1997.
- [53] Gutwin, C., and Greenberg, S. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)* **11**, 3 (2002), 411–446.

-
- [54] Gutwin, C., and Greenberg, S. The importance of awareness for team cognition in distributed collaboration. *Team cognition : Understanding the factors that drive process and performance 201* (2004).
- [55] Gutwin, C., Penner, R., and Schneider, K. Group awareness in distributed software development. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (2004), ACM, pp. 72–81.
- [56] Heath, C., Svensson, M., Hindmarsh, J., Luff, P., and Vom Lehn, D. Configuring awareness. *Computer Supported Cooperative Work (CSCW) 11*, 3 (2002), 317–347.
- [57] Hevner, A. R. The three cycle view of design science research. *Scandinavian journal of information systems 19*, 2 (2007), 87.
- [58] Hill, J., and Gutwin, C. Awareness support in a groupware widget toolkit. In *Proceedings of the ACM Conference on Supporting Group Work* (2003).
- [59] Hill, J., and Gutwin, C. The MAUI toolkit : Groupware widgets for group awareness. *Computer Supported Cooperative Work (CSCW) 13*, 5-6 (2004), 539–571.
- [60] Hong, J.-Y., Suh, E.-H., and Kim, S.-J. Context-aware systems : A literature review and classification. *Expert Systems with Applications 36*, 4 (2009), 8509–8522.
- [61] Hutchins, E., and Lintern, G. *Cognition in the Wild*, vol. 262082314. MIT press Cambridge, MA, 1995.
- [62] Jeners, N., and Prinz, W. From groupware to social media-a comparison of conceptual models. In *Collaborative Networks in the Internet of Services*. Springer, 2012, pp. 416–423.
- [63] Johansen, R. *Groupware : Computer support for business teams*. The Free Press, 1988.
- [64] Johnson, P., Wilson, S., Markopoulos, P., and Pycock, J. Adept : Advanced design environment for prototyping with task models. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (1993), ACM, p. 56.
- [65] Jourde, F., Laurillau, Y., and Nigay, L. COMM notation for specifying collaborative and MultiModal interactive systems. In *Proceedings of the second ACM SIGCHI Symposium on Engineering Interactive Computing Systems (Berlin, Germany - June 19-23, 2010) (EICS 2010)* (2010), ACM New York, NY, USA, pp. 125–134.
- [66] Jourde, F., Laurillau, Y., and Nigay, L. e-comm, un éditeur pour spécifier l'interaction multimodale et multiutilisateur. In *Conference Internationale Franco-phone sur l'Interaction Homme-Machine* (New York, NY, USA, 2010), IHM '10, ACM, pp. 225–228.
- [67] Klug, T., and Kangasharju, J. Executable task models. In *Proceedings of the 4th international workshop on Task models and diagrams* (2005), ACM, pp. 119–122.

- [68] Kuutti, K. Activity theory as a potential framework for human-computer interaction research. *Context and consciousness : Activity theory and human-computer interaction* (1996), 17–44.
- [69] Leontiev, A. Activity, consciousness, and personality, 1978.
- [70] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., and López-Jaquero, V. UsiXML : A language supporting multi-path development of user interfaces. *Engineering Human Computer Interaction and Interactive Systems 3425/2005* (2005), 200–220.
- [71] Lonchamp, J. *Le travail coopératif et ses technologies*. 2003.
- [72] Luyten, K., Clerckx, T., Coninx, K., and Vanderdonckt, J. Derivation of a dialog model from a task model by activity chain extraction. In *Interactive Systems. Design, Specification, and Verification*. Springer, 2003, pp. 203–217.
- [73] Malone, T. W., and Crowston, K. What is coordination theory and how can it help design cooperative work systems? In *Proceedings of the 1990 ACM conference on Computer-supported cooperative work* (1990), ACM, pp. 357–370.
- [74] Meixner, G., Seissler, M., and Breiner, K. Model-driven useware engineering. In *Model-Driven Development of Advanced User Interfaces*, H. Hussmann, G. Meixner, and D. Zuehlke, Eds., vol. 340 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, 2011, pp. 1–26.
- [75] Mintzberg, H., and Romelaer, P. *Structure et dynamique des organisations*. Editions d'organisation, 1986.
- [76] Molina, A., Redondo, M., Ortega, M., and Hoppe, U. CIAM : A methodology for the development of groupware user interfaces. *Journal of Universal Computer Science* 14, 9 (2008), 1435–1446.
- [77] Montero, F., and López-Jaquero, V. Idealxml : an interaction design tool. In *Computer-Aided Design of User Interfaces V*. Springer, 2007, pp. 245–252.
- [78] Mori, G., Paternò, F., and Santoro, C. CTTE : support for developing and analyzing task models for interactive system design. *Software Engineering, IEEE Transactions on* 28, 8 (2002), 797–813.
- [79] Nardi, B. *Context and consciousness : activity theory and human-computer interaction*. 1996.
- [80] Nielsen, J. Usability inspection methods. In *Conference companion on Human factors in computing systems* (1994), ACM, pp. 413–414.
- [81] Noguera, M., González, M., Garrido, J. L., Hurtado, M. V., and Rodríguez, M. L. System modeling for systematic development of groupware applications. In *International Conference On Software Engineering Research And Practice* (2006), pp. 750–756.
- [82] Ouahab, K., and Adel, A. Une boîte à outils qui support la conscience de groupe dans les collecticiels.
- [83] Pascoe, J., Ryan, N. S., and Morse, D. R. Human Computer Giraffe Interaction : HCI in the Field. In *Workshop on Human Computer Interaction with Mobile*

-
- Devices* (May 1998), C. Johnson, Ed., GIST Technical Report G98-1, University of Glasgow.
- [84] Paternò, F., Mancini, C., and Meniconi, S. ConcurTaskTrees : A diagrammatic notation for specifying task models. In *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction* (London, UK, UK, 1997), INTERACT '97, Chapman & Hall, Ltd., pp. 362–369.
- [85] Paterno, F., Santoro, C., Mantyjarvi, J., and Mori, G. Authoring pervasive multimodal user interfaces. *International Journal of Web Engineering and Technology* **4**, 2 (2008), 235–261.
- [86] Paterno, F., Santoro, C., and Spano, L. D. Maria : A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Transactions on Computer-Human Interaction (TOCHI)* **16**, 4 (2009), 19.
- [87] Penichet, V. M. R., Lozano, M. D., Gallud, J. A., and Tesoriero, R. User interface analysis for groupware applications in the TOUCHE process model. *Adv. Eng. Softw.* **40** (December 2009), 1212–1222.
- [88] Penichet, V. M. R., Lozano, M. D., Gallud, J. A., and Tesoriero, R. Requirement-based approach for groupware environments design. *J. Syst. Softw.* **83** (August 2010), 1478–1488.
- [89] Penichet, V. M. R., Lozano, M. D., Gallud, J. A., Tesoriero, R., Rodríguez, M., Garrido, J., Noguera, M., and Hurtado, M. Extending and supporting featured user interface models for the development of groupware applications. *Journal of Universal Computer Science* **14**, 19 (2008), 3053–3070.
- [90] Redondo, M. Á., and Bravo, C. Domosim-tpc : Collaborative problem solving to support the learning of domotical design. *Computer Applications in Engineering Education* **14**, 1 (2006), 9–19.
- [91] Rodríguez, M. L., Garrido, J. L., Hurtado, M. V., Noguera, M., and Hornos, M. J. Designing user interfaces for collaborative applications : A model-based approach. In *New Trends on Human-Computer Interaction*, J. A. Macías, A. Granollers Saltiveri, and P. M. Latorre, Eds. Springer London, 2009, pp. 1–11.
- [92] Roseman, M., and Greenberg, S. Building real-time groupware with groupkit, a groupware toolkit. *ACM Transactions on Computer-Human Interaction (TOCHI)* **3**, 1 (1996), 66–106.
- [93] Scapin, D., and Pierret-Golbreich, C. Towards a method for task description : MAD. *Work with display units* **89** (1989), 371–380.
- [94] Scapin, D. L., and Bastien, J. C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & information technology* **16**, 4-5 (1997), 220–231.
- [95] Schilit, B., Adams, N., and Want, R. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications* (1994), IEEE Computer Society, pp. 85–90.

- [96] Schilit, B., and Theimer, M. Disseminating active map information to mobile hosts. *Network, IEEE* 8, 5 (Sep/Oct 1994), 22–32.
- [97] Schlunbaum, E., and Elwert, T. Automatic user interface generation from declarative models. *Computer-Aided Design of User Interfaces. Namur : Presses Universitaires de Namur* (1996), 3–18.
- [98] Schmidt, A., Beigl, M., and Gellersen, H.-w. There is more to context than location. *Computers and Graphics* 23 (1999), 893–901.
- [99] Schmidt, K. Some notes on mutual awareness. *COTCOS, CTI, DTU, Copenhagen, Denmark* (1998).
- [100] Schwab, I., and Kobsa, A. Adaptivity through unobstrusive learning. *KI* 16, 3 (2002), 5–9.
- [101] Shen, W., Hao, Q., Mak, H., Neelamkavil, J., Xie, H., and Dickinson, J. Systems integration and collaboration in construction : a review. In *12th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2008)* (2010), pp. 11–22.
- [102] Sinnig, D., Gaffar, A., Reichart, D., Forbrig, P., and Seffah, A. Patterns in model-based engineering. In *Computer-Aided Design of User Interfaces IV*. Springer, 2005, pp. 197–210.
- [103] Sottet, J.-S. *Méga-IHM : malléabilité des Interfaces Homme-Machine dirigées par les modèles*. PhD thesis, 2008. Thèse de doctorat Informatique préparée au Laboratoire d'Informatique de Grenoble (LIG), Université Joseph Fourier.
- [104] Sottet, J.-S., and Vagner, A. GENIUS : Generating usable user interfaces. Tech. rep., Public Research Center Henri Tudor - Technical Report of the FNR project GENIUS, 2013.
- [105] Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., and Tatar, D. WYSIWIS revised : early experiences with multiuser interfaces. *ACM Transactions on Information Systems (TOIS)* 5, 2 (1987), 147–167.
- [106] Stephanidis, C., Paramythis, A., Sfyraakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G., and Karagiannidis, C. Adaptable and adaptive user interfaces for disabled users in the avanti project. *Intelligence in Services and Networks : Technology for Ubiquitous Telecom Services Volume 1430/1998* (1998), 153.
- [107] Suchman, L. *Plans and situated actions : the problem of human-machine communication*. Cambridge university press, 1987.
- [108] Tarby, J.-C., and Barthet, M.-F. the diane+ method. In *Computer-Aided Design of User Interfaces* (1996), pp. 95–119.
- [109] Tarpin-Bernard, F. La flexibilité dans les collecticiels. *Le temps, l'espace et l'évolutif* (2000), 449–458.
- [110] Thevenin, D. *L'adaptation en Interction Homme-Machine : le cas de la plasticité*. PhD thesis, Université Joseph Fourier, 2001. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (IMAG), Université Joseph Fourier 238 pages.

-
- [111] Thevenin, D., and Coutaz, J. Plasticity of user interfaces : Framework and research agenda. In *Proc. Interact99, Edinburgh, , A. Sasse & C. Johnson Eds, IFIP IOS Press Publ* (1999), pp. 110–117.
- [112] Thevenin, D., and Coutaz, J. Adaptation des IHM : taxonomies et archi. logicielle. In *Proceedings of the 14th French-speaking conference on Human-computer interaction (Conférence Francophone sur l'Interaction Homme-Machine)* (2002), ACM, pp. 207–210.
- [113] Thevenin, D., Coutaz, J., and Calvary, G. *A Reference Framework for the Development of Plastic User Interfaces*. 2003.
- [114] Thevenin, D., Haue, J. B., and Calvary, G. Cadre de conception pour la plasticité des interfaces : Démarche ingénierie, démarche centrée utilisateur. *Revue d'Interaction Homme-Machine* 4, 1 (2003), 119–148.
- [115] Totterdell, P., and Rautenbach, P. Adaptation as a problem of design. *Adaptive user interfaces* (1990), 61–84.
- [116] Trevellyan, R., and Browne, D. P. A self-regulating adaptive system. *SIG-CHI Bull.* 17, SI (1987), 103–107.
- [117] Turner, K. J. The formal specification language LOTOS : A course for users.
- [118] Van Der Aalst, W., and Van Hee, K. *Workflow management : models, methods, and systems*. MIT press, 2004.
- [119] van der Aalst, W. M. The application of petri nets to workflow management. *Journal of circuits, systems, and computers* 8, 01 (1998), 21–66.
- [120] Vanderdonckt, J. M., and Bodart, F. Encapsulating knowledge for intelligent automatic interaction objects selection. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (1993), ACM, pp. 424–429.
- [121] Vellis, G., Kotsalis, D., Akoumianakis, D., and Vanderdonckt, J. Model-based engineering of multi-platform, synchronous and collaborative UIs-extending UsiXML for polymorphic user interface specification. In *Informatics (PCI), 2012 16th Panhellenic Conference on* (2012), IEEE, pp. 339–344.
- [122] Weiser, M. The computer for the 21st-century. *Scientific American* 265 (3) (1991), 94.
- [123] Wieringa, R. Design science as nested problem solving. In *Proceedings of the 4th international conference on design science research in information systems and technology* (2009), ACM, p. 8.
- [124] Wilson, P. *Computer Supported Cooperative Work : An Introduction*. Springer, 1991.
- [125] Winograd, T. Architectures for context. *Human-Computer Interaction* 16, 2 (2001), 401–419.
- [126] Wurdel, M., Sinnig, D., and Forbrig, P. CTML : Domain and task modeling for collaborative environments. *Journal of Universal Computer Science* 14, 19 (2008), 3188–3201.

- [127] Wurdel, M., Sinnig, D., and Forbrig, P. Task-based development methodology for collaborative environments. In *Engineering Interactive Systems*. Springer, 2008, pp. 118–125.

Résumé

De nos jours, nous vivons dans un monde **d'interactions**. Nous sommes entourés **d'appareils** électroniques susceptibles de compliquer ces interactions. De plus, les utilisateurs sont dorénavant mobiles et évoluent dans des environnements changeant. Vis-à-vis de la collaboration, ces conditions peuvent inhiber la productivité. Ce projet de thèse vise à proposer des méthodes pour la conception **d'interfaces** utilisateur capables de tenir compte du contexte à travers **l'utilisation d'interfaces** adaptatives.

La contribution principale de cette thèse consiste en un cadre de référence dirigé par les modèles pour la conception et **l'exécution d'interfaces** utilisateur adaptatives supportant des mécanismes de coordination (i.e. flux de travaux, conscience de groupe). La proposition se présente sous deux aspects :

- Un cadre méthodologique pour **l'aide** à la conception **d'interfaces** supportant la coordination. Il consiste essentiellement en **l'emploi** de méthodes de modélisation **d'exigences** métier via un processus itératif;
- Un cadre technologique qui repose sur **l'ingénierie** basée sur les modèles pour permettre la génération et **l'exécution d'interfaces** utilisateur adaptatives. Il se base sur une architecture orientée **widgets** pour le support de la conscience de groupe afin de promouvoir la coordination.

Mots-clés : interfaces homme-machine, contexte **d'interaction**, adaptation, coordination, ingénierie dirigée par les modèles.

Abstract

Nowadays, we live in a world of interactions. We are surrounded by electronic devices which tend to complexify user interactions. Moreover, users are now mobile and evolve in everchanging environments. Regarding collaboration, these conditions may inhibit productivity. This PhD aims to propose design methods for user interfaces able to consider the context through the use of adaptive user interfaces.

The main contribution of this thesis consists in a model-driven reference framework for the design and the execution of adaptive user interfaces supporting coordination mechanisms (i.e. workflow, group awareness). The proposition is composed of these two facets:

- A methodologic framework to assist in the design of user interfaces supporting coordination. It consists in the use of iterative modelisation methods for requirements engineering;
- A technological framework which relies on model-based engineering to allow the generation and execution of adaptive user interfaces. It makes use of widget-based architecture to support group awareness in order to promote coordination.

Keywords: human-computer interaction, interaction context, adaptation, coordination, model-driven engineering.