

Une Formalisation du Contexte dans les Environnements Coopératifs Nomades

Manuele Kirsch-
Pinheiro
*Laboratoire LSR – IMAG,
Grenoble, France*
+33 4 76 82 72 11
kirsch@imag.fr

Marlène Villanova-
Oliver
*Laboratoire LSR – IMAG,
Grenoble, France*
+33 4 76 82 72 80
villanov@imag.fr

Jérôme Gensel
*Laboratoire LSR – IMAG,
Grenoble, France*
+33 4 76 82 72 80
gensel@imag.fr

Hervé Martin
*Laboratoire LSR – IMAG,
Grenoble, France*
+33 4 76 82 72 80
martin@imag.fr

RESUME

Dans cet article, nous présentons un modèle de représentation objet du contexte d'un utilisateur nomade dans un collecticiel sur le Web. En raison de l'évolution de l'informatique nomade, ces systèmes tendent de plus en plus à devenir « sensibles au contexte » et ainsi à adapter leur comportement au contexte de l'utilisateur afin de mieux le satisfaire. Le modèle de contexte proposé ici inclut les aspects physiques (la localisation ou le dispositif utilisé) mais également les aspects liés au processus coopératif dans lequel l'utilisateur est impliqué (groupe, rôle, activités, etc.). Cette formalisation de la notion de contexte permet la séparation entre les informations relatives au contexte et celles propres au système, facilitant ainsi la conception des systèmes collaboratifs, traditionnellement complexes. Le modèle de contexte proposé permet au système d'adapter l'information de conscience de groupe destinée à chaque membre participant à une tâche collective par une reconnaissance des circonstances dans lesquelles il accède au système.

ABSTRACT

In this paper, we propose an object oriented model for describing a mobile user's context inside a Web-based groupware system. These systems are confronted to the growing use of mobile devices, such as laptops, PDAs or cellular phones, which requires special context-aware capabilities. Our model formalizes the context notion, decoupling context and system information, which consequently reduces the system development complexity. We consider the context as a set of basic elements that identify the circumstances in which the user is accessing the system. These elements include elements related to physical aspects (the user's location and device), as well as those related to the cooperative process in which the user is involved (group, role, activities, etc.).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UBIMOB 05, May 31 – June 3, 2005, Grenoble, France.

Copyright 2005 ACM X-XXXXX-XXX-X/XX/XXXX ...\$5.00.

Categories and Subject Descriptors

H.4.1 [Information System Application]: Office Automation – *groupware* H.5.3 [Information Interface and Presentation]: Group and Organization Interface – *computer-supported cooperative work, collaborative computing*.

General Terms

Algorithms, Management, Design

Mot-clés

Informatique nomade, informatique sensible au contexte, travail coopératif assisté par ordinateur.

Keywords

Mobile computing, context-aware systems, cooperative work (CSCW).

1. INTRODUCTION

Le domaine des ordinateurs portables et de poche est en constante évolution. Les ordinateurs portables sont de plus en plus légers et désormais aussi puissants que les ordinateurs de bureau, les ordinateurs de poche, quant à eux, d'une taille sans cesse réduite, permettent de les avoir sur soi en permanence. De plus, ces dispositifs peuvent aujourd'hui être connectés facilement aux réseaux sans fils [13]. Ainsi, grâce à ces évolutions, un utilisateur, équipé de dispositifs divers (ordinateur de bureau, portable ou de poche, téléphone portable, etc.), est en mesure d'accéder au Web, et, par conséquent, aux systèmes disponibles, depuis des lieux divers et nombreux (son bureau, chez lui, l'aéroport, etc.).

Ce nouveau type d'utilisateur, que nous nommons ici utilisateur *nomade* car il peut se déplacer à tout moment, est la cible privilégiée des technologies et des systèmes dits *sensibles au contexte* (en référence à l'anglais "*context-aware computing*" [5][7]). Un système sensible au contexte a pour but d'adapter les informations ou services qu'il fournit à l'utilisateur aux circonstances courantes d'utilisation. En d'autres termes, la sensibilité au contexte est la capacité d'un système à percevoir la situation dans laquelle se trouve l'utilisateur et d'adapter en conséquence le comportement du système (les services, les

données et l'interface) [6]. Avec l'émergence et la fréquence d'un type d'utilisation qualifiée de nomade, les systèmes accessibles sur le Web se doivent nécessairement d'être, eux aussi, sensibles au contexte.

Dans toutes les définitions de systèmes sensibles au contexte, la notion de *contexte* est présente, sans pour autant qu'il en existe une définition claire et unique. Au contraire, la plupart des systèmes sensibles au contexte éludent la question de la définition et exploitent, en général, une notion de contexte qui se limite à la localisation de l'utilisateur et aux caractéristiques de son dispositif (voir, par exemple, [5] [12]). De surcroît, ces systèmes ne séparent pas les informations qui font référence au contexte, de celles qui sont relatives au domaine d'application traité par le système [6]. Par conséquent, ces systèmes deviennent difficiles à concevoir, et les composants liés à la détection et à la manipulation du contexte ne sont pas (ou sont très peu) réutilisables dans d'autres systèmes. Par exemple, l'étude de l'application *Campus Aware* [5] ou de l'architecture *MoCA* [16], nous permet de noter l'inexistence d'une représentation propre et dédiée au contexte. Dès lors, en l'absence d'une formalisation de la notion de contexte, l'introduction et l'exploitation dans ces systèmes d'une information contextuelle, autre que la localisation, comme, par exemple, les activités de l'utilisateur, ne peuvent se faire facilement, ni reposer sur la réutilisation de composants préexistants. Nous pensons qu'une formalisation de la notion de contexte est donc un pré-requis nécessaire à la conception des systèmes sensibles au contexte.

Dans cet article, nous proposons un modèle objet qui formalise la notion de contexte en séparant les données propres au contexte de celles liées au domaine d'application. Notre objectif est de tendre vers une représentation simple, extensible et surtout qui puisse être aisément mise en œuvre pour la conception de systèmes sensibles au contexte. Parmi eux, les *systèmes coopératifs*, appelés "*collecticiels*", sur le Web, comme, par exemple ToxicFarm [17], sont tout particulièrement les cibles de notre proposition. Ces systèmes sont de plus en plus conçus dans l'optique d'une utilisation nomade, répondant ainsi à l'adoption croissante des nouvelles technologies mobiles par les organisations.

Alors que la notion de contexte est souvent envisagée dans le cadre des interactions entre un utilisateur isolé et le système [1], nous adoptons un point de vue différent qui considère les interactions utilisateur/système *dans un environnement coopératif*. Ceci nécessite la prise en compte d'éléments relatifs aux processus de coopération impliquant cet utilisateur (que nous considérons de plus comme nomade dans notre approche, et donc susceptible d'employer divers dispositifs mobiles).

Le modèle que nous proposons considère l'utilisateur lui-même comme une partie intégrante de la notion de contexte, en tant que membre d'un ou plusieurs groupes de travail et au titre de sa participation à un ou plusieurs processus coopératifs. Nous avons identifié plusieurs éléments qui nous semblent essentiels pour une représentation de contexte qui soit à la fois simple et générique. Ces éléments se divisent en deux catégories regroupant d'une part, les aspects physiques de l'environnement de l'utilisateur (sa localisation, le dispositif qu'il utilise, etc.), et, d'autre part, les aspects collaboratifs qui concernent l'utilisateur (ses activités, ses rôles dans le processus de coopération, etc.).

Après avoir rappelé les principales définitions de la notion de contexte existantes, nous précisons notre propre perception (section 2). Le modèle objet formalisant notre vision du contexte est exposé dans la section 3. La section 4 propose une application possible de ce modèle pour procéder à l'adaptation des informations délivrées à l'utilisateur. Nous comparons notre approche aux travaux voisins dans la section 5 avant de présenter nos conclusions et perspectives.

2. DEFINITION DE CONTEXTE

Toute proposition visant à prendre en compte la notion de contexte dans le fonctionnement d'un système implique naturellement de considérer la façon dont ce contexte peut être représenté et codé [9]. Pour Brézillon [4], une représentation efficace du contexte en machine, tant en termes de modélisation des connaissances que de raisonnement à partir de celles-ci, est un problème non encore résolu, et ce, aussi bien du point de vue de la programmation que de son utilisation.

En amont de la représentation du contexte, il est nécessaire d'en retenir une définition. Il est clair, au regard de la littérature, qu'il n'existe pas une seule et unique définition du contexte. Notamment, divers domaines de recherche se sont attelés à définir la notion de "contexte" [3][14]. Par exemple, une analyse sociale s'intéressera au "contexte dans lequel les interactions entre un individu et un système ont lieu" [8]. Ce contexte, autant organisationnel et culturel que physique, joue ici un rôle critique en fournissant les moyens d'interpréter et de comprendre l'action [8]. Le contexte dans lequel une action se produit ne se limite pas à sa simple localisation ou aux acteurs qu'elle implique, mais il inclut aussi tout élément capable d'aider la bonne identification et compréhension de cette action. Dans ce sens, le contexte donne du sens à l'action.

Cependant, cette définition est trop large pour être facilement (et efficacement) exploitable dans un système sensible au contexte. Ces systèmes utilisent le plus souvent la notion de contexte afin de permettre des requêtes, en association avec d'autres informations (par exemple, l'utilisation de la localisation comme paramètre de la requête), et afin d'adapter le comportement et la réponse du système à des patrons d'utilisation (par exemple un utilisateur sur un dispositif donné) [9].

Une autre définition du contexte, proposée par Dey [7], considère plus particulièrement la conception et le développement des systèmes sensibles au contexte. Pour l'auteur, le contexte est construit à partir de tous les éléments d'information qui peuvent être utilisés pour caractériser la situation d'une entité. Une entité correspond ici à toute personne, tout endroit, ou tout objet (en incluant les utilisateurs et les applications eux-mêmes) considéré(e) comme pertinent(e) pour l'interaction entre l'utilisateur et l'application. Cependant, l'impossibilité d'énumérer tous les éléments d'information pertinents pour toutes les situations est évidente, car ces éléments changent inévitablement d'une situation à l'autre [7]. En outre, nous pouvons voir la notion de contexte comme une notion récursive, un contexte local faisant toujours référence à un contexte plus large, le rendant, de fait, impossible à décrire totalement [4].

En conséquence, nous préférons limiter le travail présenté dans cet article à un cas de figure plus précis. Ce cas concerne l'utilisation d'un système coopératif sur le Web, tel que

ToxicFarm [17]. Les collecticiels sur le Web doivent de plus en plus prendre en compte une utilisation nomade par des utilisateurs disposant d'un ou plusieurs dispositifs mobiles (ordinateur portable ou de poche, téléphone portable, etc.), et qui peuvent se déplacer tout en accédant au système. Les utilisateurs nomades utilisent de plus en plus les dispositifs mobiles pour coopérer avec leurs collègues, comme le constatent Alarcón *et al.* [1] pour qui les solutions mobiles constituent une opportunité intéressante pour le travail en groupe. Par exemple, un utilisateur en déplacement peut, grâce aux nouvelles technologies mobiles, garder à tout moment le contact avec ses collègues de bureau (et vice-versa), ce qui facilite grandement le travail en groupe. Un collecticiel qui se veut sensible au contexte doit, au contraire d'autres systèmes sensibles au contexte, considérer l'utilisateur nomade en tant que participant d'un ou plusieurs processus coopératifs, pour lesquels il collabore avec d'autres utilisateurs, que nous appelons "membres" (d'un groupe). En tant que membre d'un groupe, cet utilisateur a besoin de garder le contact avec ses collègues et d'être conscient des activités réalisées dans le groupe (et qui concernent son propre travail). Par conséquent, outre les éléments qui décrivent la situation physique de l'utilisateur en tant qu'individu (tels que sa localisation et son dispositif actuel), plusieurs éléments relatifs au groupe et au processus coopératif doivent aussi faire partie du contexte de cet utilisateur nomade. Par exemple, dans le cas d'un groupe qui coopère afin de rédiger un rapport en commun, les informations liées à ce rapport et aux activités réalisées par les différents membres sont hautement significatives et participent à la définition du contexte de chacun.

Les éléments mis en évidence ci-dessus sont à la base de notre proposition d'un modèle du contexte d'un utilisateur nomade dans un environnement coopératif. La section suivante introduit ce modèle.

3. UN MODELE OBJET DU CONTEXTE

Bien que nous ayons opté pour la modélisation du contexte dans un cadre précis (celui de l'utilisation d'un collecticiel sur le Web par un utilisateur nomade), il nous apparaît peu réaliste de décrire de façon exhaustive les éléments qui composent le contexte d'un utilisateur. Par conséquent, nous nous orientons vers une représentation générique et facilement extensible. Nous proposons donc un modèle objet qu'il est possible d'étendre grâce aux relations de généralisation et de spécialisation.

En nous basant sur les principaux travaux du domaine du Travail Coopératif Assisté par Ordinateur (TCAO), nous avons procédé à l'analyse des informations jugées pertinentes pour l'identification du contexte d'un utilisateur nomade dans un environnement coopératif. Leiva-Lobos et Covarrubias [11] proposent que le contexte dans lequel des acteurs coopèrent contienne trois axes principaux : spatial, temporel et culturel. A travers ces axes, les auteurs mettent en avant les artefacts qui peuplent l'environnement physique et l'environnement virtuel (à travers axe spatial), l'historique de la coopération ainsi que les prévisions pour le futur (axe temporel), et les pratiques communes au groupe (axe culturel). De façon similaire, Alarcón *et al.* [2] suggèrent la décomposition de la notion de contexte en "sous-contextes". Un premier sous-contexte est associé au travail mené de façon coopérative, un deuxième est relatif exclusivement à l'utilisateur, et enfin un troisième est lié aux documents partagés par le groupe (voir [1]). Dans le premier sous-contexte (appelé

contexte de travail), les auteurs exploitent un contenu constitué par les applications et par le processus coopératif, avec les activités qui le composent et le calendrier qu'il doit respecter. Dans le deuxième sous-contexte (*contexte de l'utilisateur*), ils considèrent les utilisateurs eux-mêmes, en portant une attention particulière à sa localisation "électronique" (c'est-à-dire, les moyens pour le contacter : SMS, messagerie électronique, etc.). Dans le dernier sous-contexte, les objets partagés, avec leurs structures, sont pris en compte.

Ainsi, sur la base de ces travaux, nous avons dégagé cinq points de vue distincts : *espace*, *outil*, *temps*, *communauté*, et *processus*. Chacun d'eux fait référence aux éléments suivants (voir Figure 1) : le point de vue "espace" nous fournit les concepts de *localisation* de l'utilisateur, tandis que le point de vue "outil" fournit les concepts de *dispositif* et d'*application* employés par cet utilisateur; le troisième point de vue ("temps") inclut le concept de *calendrier* partagé ; la notion de "communauté" nous donne les concepts de *groupe*, de *rôle* et d'utilisateur en tant que *membre* d'un (ou plusieurs) groupe(s); finalement, le point de vue "processus" concerne les concepts de *processus* mis en œuvre par un groupe, d'*activité* coopérative et d'*objet partagé* par ce même groupe.

Points de vue	Concepts	Classes				
Espace	localisation	→	Localisation	Contexte physique		
		dispositif			→	Dispositif
					application	
Temps	calendrier	→	Calendrier			Contexte collaboratif
		Communauté			groupe	
rôle	→		Rôle			
	utilisateur			→		
Processus		processus	→	Processus		
	activité		→		Activité	
			objet partagé			

Figure 1. Points de vue et éléments identifiés, avec les classes correspondantes dans le modèle proposé.

Ces éléments peuvent être divisés en deux grands "sous-contextes" distincts (voir Figure 1). Le premier, appelé "*contexte physique*", est lié aux aspects physiques de l'utilisateur (sa localisation, son dispositif, sa localisation virtuelle, c'est-à-dire l'application qu'il utilise) ; le second, dit "*contexte collaboratif*", est lié au processus coopératif, ce qui inclut les notions de groupe, de rôle, d'activité, d'objet partagé, entre autres. La grande majorité des systèmes sensibles au contexte (par exemple [5], [12] ou [16]) n'adopte que la notion de contexte physique, en considérant l'utilisateur comme un individu isolé. Nous ne pouvons pas nous limiter à cette notion puisque nous traitons l'utilisateur comme membre d'un collectif, lequel coopère et possède des objectifs communs. Ce collectif (le groupe) et tout ce qui touche le processus de coopération sont, selon nous, des éléments clés pour bien cerner les circonstances dans lesquelles l'utilisateur accède au système. La notion de contexte collaboratif que nous proposons va donc dans ce sens.

Chacun des éléments énumérés ci-dessus est représenté dans le modèle proposé par une classe (voir Figure 1). Les relations entre

ces classes témoignent des liens existant entre les concepts qu'elles représentent (Figure 2).

Un utilisateur (classe "Membre") appartient à un ou plusieurs groupes à travers les rôles qu'il y joue. Nous considérons ici que : *i*) un groupe est un ensemble d'individus qui coopèrent avec un objectif commun ; *ii*) un rôle est la description des droits et responsabilités attribués à un membre du groupe ; et *iii*) un groupe admet plusieurs rôles. Par exemple, un comité de programme est un groupe dont les membres peuvent avoir un rôle de réviseur et/ou de président du comité.

De plus, nous considérons qu'un groupe peut définir un processus (classe "Processus"), ou encore adopter un processus qui a été défini auparavant (réutilisation du modèle de processus). Ce processus décrit les objectifs du groupe et les activités que ce groupe est censé réaliser. Ce processus est donc composé par un ensemble d'activités (classe "Activité"), lesquelles peuvent être à leur tour composées par d'autres activités. Un rôle (classe "Rôle") permet la réalisation d'une ou plusieurs activités, qui sont effectivement exécutées par un membre. De surcroît, chaque activité pourra manipuler, à travers une ou plusieurs applications, un ensemble d'objets partagés (classe "Objet Partagé"). Cette structure *processus/activité/objet partagé* est capable de représenter la structure adoptée par des équipes très hiérarchisées, qui utilisent souvent des outils de type "workflow", de même que celles qui adoptent un modèle de travail plus flexible et qui coopèrent surtout à travers les objets partagés.

Par ailleurs, nous considérons qu'un membre se trouve, à un instant donné, dans un espace présentant un caractère à la fois virtuel et réel (relation "est localisé dans" dans la Figure 2). Cette idée correspond à l'idée de session, reflétant ainsi la présence de

l'utilisateur dans le système. Ce membre se trouve donc dans un espace physique, composé par sa localisation dans le monde réel (classe "Localisation"), dans un espace d'exécution, qui indique le dispositif (classe "Dispositif") qu'il emploie, et dans un espace virtuel, comportant l'application (classe "Application") qu'il utilise. Ce concept d'application correspond, en fait, à la notion de *service*. Nous supposons ici qu'un collecticiel sur le Web est composé de plusieurs services, lesquels peuvent se présenter sous la forme d'applications Web interconnectées. La Figure 2 présente toutes ces classes et les associations qui les lient, en indiquant aussi quelques attributs possibles. Il faut cependant remarquer que ces attributs ne sont là qu'à titre indicatif, et que d'autres sont envisageables pour chaque classe et association, en fonction du système qui exploite le modèle.

Conformément à la définition de contexte donnée par Dey [7] (cf. section 2), nous envisageons un contexte comme un ensemble constitués des éléments présentés ci-dessus. Dans notre modèle, illustré par le diagramme de classes UML de la Figure 3, le concept de contexte est représenté par la classe "Description de Contexte". Cette classe est la composition des classes représentant les éléments de contexte identifiés précédemment. Ces classes sont les sous-classes d'une classe abstraite "Élément de Contexte". On notera dans la Figure 3 la présence de la relation "décrit contexte de" entre une description de contexte et un élément. Cette relation représente le fait qu'une *description de contexte* est toujours relative à un élément donné : contexte d'un utilisateur, contexte d'un dispositif, contexte dans lequel une application peut être exécutée...

Les diagrammes présentés par les Figure 2 et Figure 3 résument le modèle que nous proposons. Ce modèle doit être personnalisé au moment de sa mise en application à l'intérieur d'un système

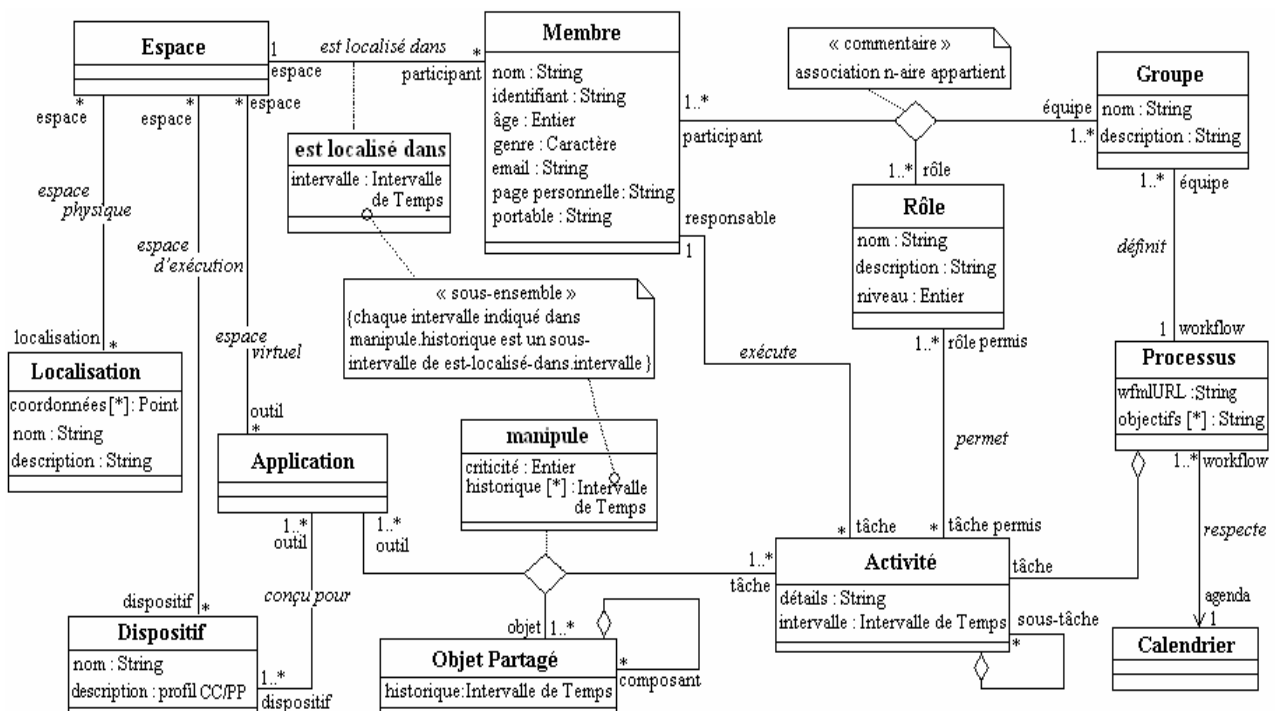


Figure 2. Les classes représentant des concepts de contexte et leurs relations.

sensible au contexte. Ainsi, les concepteurs d'un tel système doivent spécifier les classes de ce modèle conformément aux besoins du système à concevoir. Par exemple, les concepteurs peuvent décrire dans le détail les activités et les rôles autorisés par le système à travers la spécialisation des classes "Activité" et "Rôle" respectivement. Ils peuvent aussi ajouter une nouvelle classe "Réseau" (sous-classe d'Elément de Contexte) afin de représenter les caractéristiques et l'état du réseau utilisé par l'utilisateur pour se connecter au système.

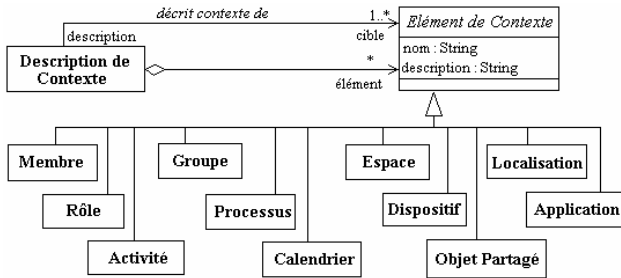


Figure 3. La description du contexte vu comme une composition d'éléments de contexte.

Un système sensible au contexte ainsi conçu manipulera donc les instances de ces classes à travers les instances de *description de contexte*. Ces dernières représentent, pour le système, le contexte d'un élément donné. Au fur et à mesure que les circonstances dans lesquelles se trouve cet élément évoluent, les éléments qui composent sa description de contexte pourront aussi évoluer par la mise à jour de ces instances. Cette évolution des instances garantit le caractère dynamique nécessaire à l'information de contexte (voir [6] et [4]).

Il est important d'observer que le modèle de contexte supporte l'utilisation de différentes techniques de détection pour chaque élément du contexte (pour la détection de la localisation il est possible d'utiliser, par exemple, le GPS [5], ou encore l'intensité du signal dans les réseaux sans fil [16]). Ainsi, le concepteur d'un système sensible au contexte qui utilise le modèle ici proposé peut, non seulement étendre ce modèle selon les besoins du système, mais aussi choisir les techniques de détection qui conviennent au mieux à chaque élément de contexte et au système.

De plus, le modèle, à travers notamment la relation de composition entre la *description de contexte* et les *éléments de contexte* (voir Figure 3), permet la description de contextes incomplets par omission des instances de certains éléments du modèle. En d'autres termes, si le système est incapable de détecter un ou plusieurs *éléments de contexte* (par exemple, la localisation ou les activités d'un utilisateur), il ne peut tout simplement pas créer les instances correspondantes, ni associer ces instances (si elles existent déjà) à la *description de contexte* correspondantes. En revanche, l'omission d'un élément entier de la représentation peut s'avérer inadéquate, surtout quand le système ne dispose que d'informations partielles sur l'élément en question. Dans ce cas, l'idéal est de laisser non évalués, dans ces instances, les attributs dont la valeur est inconnue. Ceci est possible en fonction de l'implémentation du modèle choisie. Par exemple, si une base XML est utilisée pour garder les instances

du modèle, l'omission d'un attribut dont la valeur est inconnue est parfaitement possible.

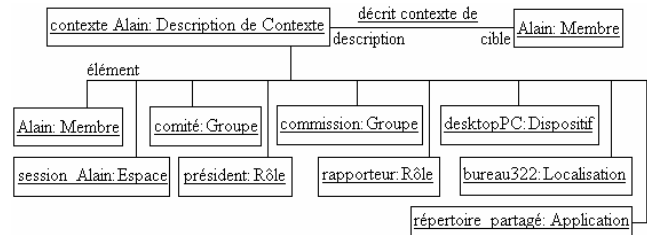


Figure 4. Exemple de description de contexte.

La Figure 4 présente un exemple d'instance de *description de contexte*. Dans cet exemple, le système, semblable à ToxicFarm [17], fournit à ses utilisateurs, entre autres, un service de répertoire partagé. Un utilisateur spécifique (appelé 'Alain'), participe à deux groupes distincts : le comité de programme d'une conférence, en tant que président du comité, et une commission d'évaluation, en tant que rapporteur. A un moment donné, nous supposons le contexte d'Alain composé par les instances présentées dans la Figure 4 (les relations entre ces instances sont omises pour simplifier la présentation), en considérant que l'utilisateur 'Alain' accède à l'application de répertoire partagé à partir du poste fixe de son bureau.

A l'aide d'une telle représentation de contexte, un système sensible au contexte est capable d'utiliser les instances de *description de contexte* afin d'adapter son comportement à l'utilisateur en tenant compte de son contexte courant. La Section 4 présente une façon de procéder à l'adaptation du contenu informationnel fourni à un utilisateur à l'aide du modèle ici proposé.

4. APPLICATION DU MODÈLE DE CONTEXTE

Les systèmes sensibles au contexte utilisent la notion de contexte pour adapter soit leur interface, soit leurs services, soit le contenu délivré à l'utilisateur. Le modèle proposé dans la section 3 s'applique facilement au dernier cas, pouvant être utilisé dans un processus de sélection du contenu informationnel. Un tel processus vise la présentation à l'utilisateur des informations qui sont pertinentes vis-à-vis de son contexte actuel. Outre la représentation du contexte actuel de l'utilisateur, ce modèle peut également être utilisé pour exprimer le contexte lié à une entité du système. Par exemple, un objet donné (un tableau, un rapport...) qui se trouve dans une localisation donnée et qui est utilisé dans une activité donnée peut être associé à une *description de contexte*. Celle-ci contient les éléments faisant référence à cette localisation et à cette activité. Ceci permet l'utilisation des composants de ce modèle (les instances liées à une description de contexte et ses attributs) pour la construction de requêtes en tant que paramètres de ces requêtes (trouver toute information dont la *description de contexte* inclut une certaine localisation, c'est-à-dire, toute information liée à cette localisation). Ainsi, les instances du modèle proposé peuvent intervenir dans l'expression de règles et de conditions visant à sélectionner parmi des informations disponibles celles qui sont pertinentes pour le contexte actuel d'un utilisateur.

Nous avons proposé dans [10] un processus de filtrage qui sélectionne un sous-ensemble des informations disponibles à l'aide d'un ensemble de *profils*. Ces profils servent à indiquer les préférences d'un utilisateur, notamment la pertinence des informations, par rapport à un contexte donné. Ceci permet l'expression de contraintes telles que "si l'utilisateur se retrouve chez lui, alors tel type d'information est prioritaire". Ainsi, chaque objet profil contient des conditions pour le filtrage des informations et est associé à une *description de contexte* (appelée *contexte d'application*), laquelle décrit les conditions pour lesquelles le profil s'applique. Ce processus de filtrage utilise donc les instances de *description de contexte* pour indiquer ces situations types, ainsi que pour représenter le *contexte actuel* de l'utilisateur. Ce processus repose sur l'hypothèse qu'un utilisateur nomade accède à un système donné à partir d'un ensemble limité et plus ou moins défini de situations : de son bureau, de chez lui, de l'aéroport, avec son ordinateur portable, son ordinateur de poche, etc. A partir du moment où ces situations types sont identifiées, il est possible pour l'utilisateur de définir ses préférences pour chaque situation, et de couvrir ainsi la majorité des cas d'utilisation du système.

Pour ce processus de filtrage, nous avons implémenté le modèle sous la forme d'une base de connaissances AROM [15]. AROM est un système de représentation de connaissances par objets que nous avons retenu pour plusieurs raisons. Tout d'abord, le fait qu'il soit un système de représentation de connaissances par objets a facilité la transformation du modèle vers une base de connaissances. Cette transformation n'a rien perdu en pouvoir de représentation car le système AROM permet la représentation des associations au même titre que celle des classes. En effet, dans le modèle proposé, les associations entre les classes ont une forte signification puisque elles représentent les relations qu'entretiennent les concepts dans le monde réel. D'autre part, AROM permet la modification de la base de connaissances à l'exécution, aussi bien à travers l'introduction et la modification d'objets (des instances de classes et des tuples d'associations), qu'à travers la création et la modification du schéma de la base de connaissance (des classes et des associations). Il permet aussi la gestion des attributs non évalués (des attributs dont la valeur est inconnue). Ceci nous permet de suivre et de respecter le caractère dynamique du contexte. Par exemple, lorsque l'utilisateur se déplace ou que le niveau de la batterie de son dispositif baisse, les composants du système responsables de la détection du contexte peuvent détecter ces changements et procéder aux mises à jour nécessaires dans la base de connaissances à travers la modification des instances de *localisation* et de *dispositif* correspondantes dans la *description de contexte* de l'utilisateur. Finalement, le système AROM propose un "parser" XML, lequel constitue un atout pour la transmission de ces instances sur le Web pour les systèmes partagés entre plusieurs sites ou même pour une éventuelle communication avec d'autres systèmes.

Le processus de filtrage utilise donc une base de connaissances AROM dans laquelle sont stockées les instances de *description de contexte* liées aux profils, ainsi que les instances qui représentent les contextes actuels des utilisateurs actifs dans le système. Tout l'intérêt du processus de filtrage est d'adapter les informations délivrées à un utilisateur à travers la sélection de celles qui semblent être les plus pertinentes pour son contexte actuel. Le processus s'effectue ainsi en deux étapes [10] : d'abord il sélectionne les profils qui peuvent être appliqués au contexte

actuel d'un utilisateur donné, puis il sélectionne les informations (représentées par des événements) à partir de ces profils.

La première étape sélectionne les profils en comparant le contexte d'application de chaque profil au contexte courant de l'utilisateur. Cette comparaison représente un problème de mise en correspondance ou "matching" entre deux instances de *description de contexte*, l'une liée au profil, l'autre à l'utilisateur. L'objectif est de vérifier si tous les éléments (et les relations qui les unissent) qui composent la *description de contexte* lié au profil existent aussi dans la *description du contexte* associée à l'utilisateur. En d'autres termes, nous vérifions si toutes les circonstances décrites dans le contexte d'application de chaque profil sont présentes dans le contexte actuel de l'utilisateur. Si elles le sont, le profil sera sélectionné. Ce "matching" est fait à travers un opérateur "contient" ("contains") qui vérifie si l'on retrouve dans une instance de *description de contexte* toute la structure (les instances et les tuples) définie dans une autre instance. Ce processus est donc similaire à un algorithme qui cherche à déterminer une relation de type "est un sous-graphe" entre deux graphes.

La deuxième étape du processus filtre les informations selon l'ensemble de profils déterminé par la première étape. Pour chaque profil sélectionné, cette étape filtre les événements qu'il signale comme pertinents et vérifie si chaque événement correspond aux conditions déclarées dans le profil. L'aspect temporel est privilégié dans l'expression des conditions : chaque profil peut indiquer un intervalle de temps, tous les événements choisis devront alors s'être produits dans cet intervalle.

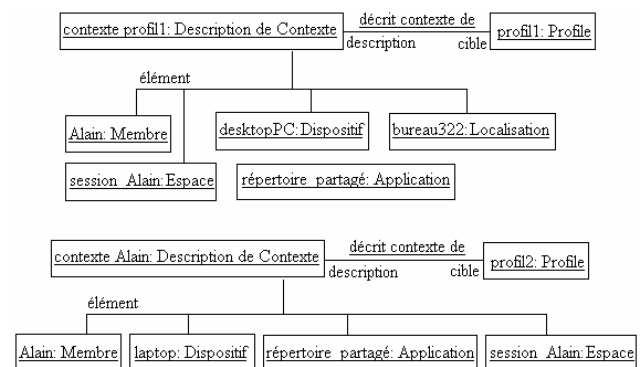


Figure 5. Deux instances de *description de contexte* associés à deux profils différents.

Reprenons l'exemple de la section 3. Supposons que l'utilisateur 'Alain' ait créé pour lui deux profils différents : un lorsqu'il travaille dans son bureau, et un autre lorsqu'il est en déplacement avec son ordinateur portable. Ces profils pourront être associés aux instances de *description de contexte* présentées dans la Figure 5. Lorsqu'Alain se trouve dans la situation décrite par la Figure 4, seul le premier profil sera sélectionnée par la première étape du processus de filtrage (le deuxième profil sera rejeté car son contexte d'application contient un élément, le *dispositif laptop*, qui n'appartient pas au contexte d'Alain). Donc, si Alain avait indiqué dans ce profil qu'il ne voulait récupérer que les événements qui ont été produits depuis hier, le système filtre les événements disponibles dans la deuxième étape du processus, en

rejetant les événements qui sont plus anciens et en ne présentant que ceux qui respectent les conditions associées au profil sélectionné.

Ce processus de filtrage illustre la manière dont un système sur le Web peut exploiter le modèle proposé en vue de l'adaptation des informations délivrées à un utilisateur nomade. Cependant, la généralité du modèle nous permet d'imaginer son utilisation à d'autres fins. Nous pensons ici notamment à l'adaptation des services proposés par le système selon des contextes précis (par exemple, des services qui ne sont disponibles que dans un site déterminé, ou pour un certain groupe d'utilisateurs, ou encore pour des utilisateurs qui jouent des rôles précis).

5. TRAVAUX VOISINS

Par rapport aux travaux menés dans le domaine de l'informatique sensible au contexte, comme ceux de Burrell *et al.* [5], Lemlouma et Layaïda [12], et Rubinsztejn *et al.* [16], notre proposition se distingue par deux aspects principaux : d'abord une représentation d'un modèle du contexte qui n'est pas présente dans ces travaux. Puis, par l'adoption d'une notion de contexte plus large, qui englobe les aspects relatifs à l'utilisateur en tant que membre d'un groupe, et qui n'est pas limitée aux traditionnels concepts de "localisation" et de "dispositif".

D'autres travaux, comme ceux de Chaari *et al.* [6] et Dey [7], s'intéressent à l'architecture des systèmes sensibles au contexte. Ces travaux proposent notamment des éléments pour la détection et l'interprétation des informations contextuelles, mais ils n'imposent pas de modèle de représentation pour ces informations. C'est pourquoi ces travaux nous semblent compatibles avec le modèle proposé, en considérant qu'un concepteur peut les combiner. Par exemple, Dey [7] propose trois éléments principaux dans son "Context Toolkit" : les "widgets", pour la capture des données contextuelles ; les "interpreters" pour interpréter les données collectées ; et les "aggregators". Ceux derniers rassemblent les informations de contexte sur une entité (par exemple, un utilisateur), agissant ainsi comme une passerelle entre le contexte et le système [7]. Ce concept correspond très exactement à la notion de *description de contexte* que nous proposons ici, ce qui atteste de la compatibilité entre ces deux propositions.

Par ailleurs, il existe des travaux directement liés au domaine de recherche TCAO, comme celui de Alarcón *et al.* [1] [2]. Dans ses travaux, Alarcón *et al.* [1][2] introduisent une représentation du contexte à l'aide d'un langage à ontologies. Les ontologies constituent une méthode de représentation de connaissances très puissante, qui permet la représentation des concepts et de relations existant entre eux, au même titre que les représentations par objets. Néanmoins, les ontologies présentent une faiblesse pour la conception des logiciels : elles sont plus difficiles à comprendre et à manipuler que les représentations par objets. Alarcón *et al.* [1] le concèdent, en soulignant que cette méthode de représentation exige un large effort de la part des concepteurs (et des utilisateurs qui sont invités à personnaliser certaines ontologies dans [2]), lesquels ne sont pas forcément familiarisés à ce type de représentation. De surcroît, les ontologies ne permettent pas toujours la définition de contraintes fortes dans les relations (*i.e.* les relations transitives, les quantificateurs, etc.), ce qui peut être négatif lors de la construction dynamique du contexte [2].

6. CONCLUSION ET PERSPECTIVES

Nous avons proposé ici un modèle en UML pour la représentation du contexte dans un système coopératif sur le Web. Ce modèle se présente comme une représentation formelle, mais simple et extensible, du contexte, notamment grâce à l'utilisation de l'approche objet qui le rend compréhensible pour les concepteurs de systèmes sensibles au contexte. La séparation entre les informations de contexte et celles propres au système facilite la conception d'un nouveau système ainsi que l'intégration de notre proposition dans un système existant. De surcroît, nous croyons que la structure du modèle proposé (en particulier la *description de contexte*) assure la généralité nécessaire à d'éventuelles extensions et ainsi que son application à des domaines différents du TCAO. Nous pouvons imaginer, par exemple, l'introduction, à l'échelle d'une entreprise, des *éléments de contexte* liés à la culture et à l'organisation de l'entreprise, et donc l'utilisation d'un modèle de contexte commun à tous les systèmes de cette entreprise.

Nous avons également présenté un processus de filtrage qui utilise le modèle pour l'adaptation de contenu. Nous croyons que ce modèle peut aussi être utilisé pour l'adaptation des services fournis par le système, notamment à travers la sélection de ceux qui sont en adéquation avec un contexte donné. Nous étudions actuellement plus particulièrement le cas de l'adaptation des services dans un système composé par plusieurs "Web Services" coopérants, à l'aide du processus de filtrage présentés dans l'article.

Finalement, ce modèle a été implémenté dans une base de connaissances par objets, en utilisant le système AROM [15]. Les tests qui ont été faits avec le processus de filtrage en utilisant cette base de connaissances n'ont montré aucun impact significatif dans la performance de l'application au fur et à mesure que le nombre d'instances grandit. Cependant, nous voulons pratiquer d'autres tests sur des systèmes dans des conditions réelles d'utilisation afin de valider notre approche.

7. REMERCIEMENTS

Manuele Kirsch-Pinheiro remercie le gouvernement brésilien pour son soutien dans le cadre des accords CAPES-COFECUB (BEX 2296/02-0).

8. REFERENCES

- [1] Alarcón, R., Collazos, C. and Guerrero, L.A. Distributed shared contexts. In: Karmouch, A., Korba, L. and Madeira, E. (Eds.), *1st International Workshop on Mobility Aware Technologies and Applications - MATA 2004, LNCS 3284*, Springer-Verlag, 2004, 27-36.
- [2] Alarcón, R. and Fuller, D. Intelligent awareness in support of collaborative virtual work groups. In Haake, J.M. and Pino, J.A. (Eds.), *8th International Workshop on Groupware: Design, Implementation and Use - CRIWG 2002, LNCS 2440*, Springer-Verlag, 2002, 168-188.
- [3] Brézillon, P. *Modeling and using context: past, present and future*. Rapport de Recherche LIP6 2002/010, 2002. <http://www.lip6.fr/reports/lip6.2002.010.html>.
- [4] Brézillon, P. Expliciter le contexte dans les objets communicants. In Kintzig, C., Poulain, G., Privat, G., and

- Favennec, P-N. (Eds.), *Objets Communicants*, Hermes Science Publications, Paris, 2002, 293-303.
- [5] Burrell, J., Gray, G.K., Kubo, K. and Farina, N. Context-aware computing: a text case. Borriello, G. and Holmquist, L.E. (Eds.), *4th Int. Conference on Ubiquitous Computing - Ubicomp 2002*, LNCS 2498, Springer-Verlag, 2002, 1-15.
- [6] Chaari, T., Laforest, F. and Celentano, A. *Design of context-aware applications based on web services*. Technical Report RR-2004-033, LIRIS, Lyon, France, 2004. http://liris.cnrs.fr/publis/rr_html.
- [7] Dey, A.K. Understanding and using context. *Personal and Ubiquitous Computing*, 5, 1, 2001, 4-7.
- [8] Dourish, P. Seeking a foundation for context-aware computing. *Human Computer Interaction*, 16, 2-4, 2001, 229-241.
- [9] Dourish, P. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8, 1, 2004, 19-30.
- [10] Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J. and Martin, H. Context-Aware Filtering for Collaborative Web Systems: Adapting the Awareness Information to the Users Context. *20th ACM Symposium on Applied Computing (SAC 2005)* (Santa Fe, New Mexico, March 2005) 1668-1673.
- [11] Leiva-Lobos, E.P. and Covarrubias, E. The 3-ontology: a framework to place cooperative awareness. In Haake, J.M. and Pino, J.A. (Eds.), *8th Int. Workshop on Groupware: Design, Implementation and Use - CRIWG 2002*, LNCS 2440, Springer-Verlag, 2002, 189-199.
- [12] Lemlouma, T. and Layaïda, N. Context-Aware Adaptation for Mobile Devices. *IEEE Int. Conference on Mobile Data Management* (Jan. 2004) IEEE Computer Society, 106-111.
- [13] Moran, T. and Dourish, P. Introduction to this special issue on context-aware computing. *Human-Computer Interaction*, 16, 2-3, 2001.
- [14] Mostéfaoui, K., Pasquier-Rocha, J. and Brézillon, P. Context-aware computing: a guide for the pervasive computing community. *Proceedings of the IEEE/ACS International Conference on Pervasive Services (IPCS'04)*, IEEE Computer Society, 2004, 39-48.
- [15] Page, M., Gensel, J., Capponi, C., Bruley, C., Genoud, P. and Ziébelin, D. Représentation de connaissances au moyen de classes et d'associations : le système AROM. *6^{èmes} Journées Langages et Modèles à Objets 2000* (Jan. 2000) 91-106. <http://www-lsr.imag.fr/Les.Groupes/sigma/articles/page00a.pdf>.
- [16] Rubinsztein, H.K., Endler, M., Sacramento, V., Gonçalves, K. and Nascimento, F. Support for context-aware collaboration. In: Karmouch, A., Korba, L. and Madeira E. (Eds.), *1st Int. Workshop on Mobility Aware Technologies and Applications - MATA 2004*, LNCS 3284, Springer-Verlag, 2004, 37-47.
- [17] Skaf-Molli, H., Molli, P., Oster, G. and Godard, C. Toxic farm: a cooperative management platform for virtual teams and enterprises. *Proceedings of 5th Int. Conference on Enterprise Information Systems (ICEIS'03)* <http://www.loria.fr/~molli/rech/iceis03/>.