



Actes des septièmes journées francophones

Mobilité et Ubiquité 2011
UBIMOB'11

6-8 juin 2011

Muséum de Toulouse

Edition des actes : Emmanuel Dubois, Jean-Marc Pierson

ISBN : 978-2-917490-16-7

EAN : 9782917490167

Avant-Propos des Présidents du comité de programme d'Ubimob'11

L'édition 2011 de la conférence « Ubiquité et Mobilité », Ubimob'11, est la septième du genre.

La conférence a été organisée en sessions portées par des spécialistes des domaines considérés. Au total, ce sont 17 papiers qui seront présentés pendant les trois jours. Nous avons également le plaisir d'accueillir 4 démonstrations qui permettront de rendre tangible dans des scénarios applicatifs très pertinents les recherches menées par notre communauté.

Le programme de la conférence a été construit pour refléter le caractère pluridisciplinaire de notre communauté. Ce caractère cosmopolite se reflète dans l'organisation des sessions, qui feront intervenir des spécialistes des interfaces hommes-machines, des systèmes d'informations, des middlewares ; de plus les articles retenus présentent de manière équilibrée des contenus méthodologiques et d'autres plus applicatifs. En revendiquant cette approche, nous espérons créer des synergies allant au-delà de frontières communautaires des groupes de recherche ayant l'habitude de se côtoyer dans des conférences plus pointues dans des domaines très ciblés.

Le programme s'enrichit de deux conférences invitées. Marc-Olivier Kilidjian (LAAS-CNRS) présentera une intervention sur « Résilience des Systèmes Ubiquitaires Mobiles: l'exemple de la sauvegarde coopérative ». Augusto Celentano (Università Ca'Foscari, Venise, Italie) traitera quant à lui de la mobilité et sa relation à l'art dans sa présentation « Designing Mobile Applications for Unfamiliar Knowledge: Essays in the Art Domain ». Nous tenons à remercier les deux orateurs pour leur implication dans le programme d'UbiMob'11.

Nous tenons à remercier tous les membres du comité de programme et les responsables des sessions. Sans eux, UbiMob'11 n'aurait pas pu voir le jour.

Nous espérons que vous prendrez autant de plaisir à lire ces actes et/ou à participer à la conférence que nous en avons eu à présider ce comité de programme.

Emmanuel Dubois et Jean-Marc Pierson
Présidents du comité de programme de Ubimob'11.

Comités de Programme Ubimob'11

Présidents du Comité de Programme :

Emmanuel Dubois, Université Toulouse 3 - Tarbes, IRIT-ELIPSE

Jean-Marc Pierson, Université Toulouse 3, IRIT-ASTRE

Responsables des Démonstrations :

Christophe Bortolaso (IRIT, Toulouse)

Georges Da Costa (IRIT, Toulouse)

Membres du comité de Programme :

Cédric Bach, IRIT

Rémi Bastide, IRIT

Jean-Michel Bruel, IRIT

Nadine Couture, ESTIA

Thomas Devogele, LI

Fabrice Peyrard, IRIT

Philippe Roose, LIUPPA

Patricia Serrano Alvarado, LINA

Jean-Yves Tigli, I3S

Webmestre :

Philippe Truillet (IRIT, Toulouse)

Avec le soutien de :



ASR: Architecture, Systèmes, Réseaux
GDR 725 de l'INRSI et de l'INSIS du CHRS

Table des matières

Evaluation d'une technique d'interaction ubiquitaire pour le pointage de données complexes et spatialisées. Bénédicte Schmitt, Emmanuel Dubois, Mathieu Raynal, Cédric Bach et David Croenne	1
Sélection de services basée sur l'impact en environnement ubiquitaire. Benjamin Cogrel, Boubaker Daachi, Yacine Amirat, Abdelghani Chibani.....	9
MARTS : Un Système Mobile de Réalité Augmentée Dédié au Tourisme. Nehla Ghouaiel, Jean-Marc Cieutat, Jean-Pierre Jessel.....	16
Réseau de capteurs pour la maîtrise de la demande énergétique. André Lachaud, David Menga, Vincent Illionnet.....	21
GEPETO*: towards a platform for evaluating privacy aspects of mobility data. Sébastien Gambs, Marc-Olivier Killijian, Miguel Núñez del Prado Cortez.....	30
Conception et Evaluation d'un Système de Dialogue pour un Dispositif de Suppléance pour Déficiants Visuels. Slim Kammoun, Bernard Oriola, Philippe Truillet, Christophe Jouffrais.....	34
SysCARE : Système de Communication et d'Assistance Robotisée. Louis-Pierre Bergé, Cyril Briand, Agnan de Bonneval, Michel Taïx, Philippe Truillet, Olivier Lefebvre.....	38
Conception d'une interaction avancée pour stimuler les visiteurs d'un musée, Bénédicte Schmitt, Louis-Pierre Bergé, Emmanuel Dubois.....	42
Towards Bridging IoT and Cloud Services: Enabling Smartphones as Mobile and Autonomic Service Gateways, Roya Golchay, Frédéric Le Mouël, Stéphane Frénot, Julien Ponge,	45
Construction bottom-up d'applications ambiantes en environnements partiellement connus à priori, Christophe Vergoni, Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Michel Riveill.....	49
Vers un environnement de déploiement autonome, Sébastien Leriche, Amine Matougui.....	57
Sémantique pour les capteurs et effecteurs en environnement pervasif, B. Helena Rodriguez, Jean-Claude Moissinac.....	63
Un DSML pour l'automatisation du test d'applications mobiles, Youssef Ridene, Franck Barbier.....	68
Une approche de type optimisation pour l'auto-adaptation de services mobiles, Marcel Cremene, Kuderna-Iulian Benta, Michel Riveill.....	74
Requirement Language tooling with Xtext, Manzoor Ahmad, Jean-Michel Bruel.....	82
Scénarios pour un Micro Smart Grid Autonome. Sylvain Frey, François Huguet, Isabelle Demeure, Ada Diaconescu, David Menga, Cédric Mivielle.....	88
Adaptabilité des flux vidéos dans les logiciels collaboratifs: Test de qualité par référence prédéfinie. Jean-Baptiste Aupet, Jean-Christophe Lapayre.....	96

Technique d'Interaction Tactilo-Tangible en Environnement Virtuel 3D. Louis-Pierre Bergé, Emmanuel Dubois, Minica Houry-Panchetti, Mustapha Mojahid.....	109
Adaptation des processus de prise de décision au contexte dynamique des ressources en informatique ubiquitaire. Application en eSanté. Ebrahim Nageba, Jocelyne Fayn, Paul Rubel.....	117
Mise au point d'un fauteuil roulant manuel intelligent et communiquant pour le suivi et l'assistance de la personne à mobilité réduite : un projet pédagogique d'intelligence ambiante collaboratif, inter-filière et pluridisciplinaire. Nicolas Vuillerme, Quentin Mourcou, Pierre-Yves Gumery, Sophie Chareyron, David Eon, Didier Pradon, Nicolas Palix, Olivier Richard, Skandar Basrou, Didier Donsez.....	123
MARTS : Partage de données environnementales pour la découverte et l'observation de la terre. Nehla Ghouaiel, Jean-Marc Cieutat, Jean-Pierre Jessel.....	127

Evaluation d'une technique d'interaction ubiquitaire pour le pointage de données complexes et spatialisées

Bénédicte Schmitt^{1,2}Emmanuel Dubois²Mathieu Raynal²Cédric Bach²David Croenne¹¹Global Vision Systems

10 Avenue de l'Europe

31520 Ramonville Saint-Agne

benedicte.schmitt@global-vision-systems.com

²IRIT

Université Paul Sabatier

31062 Toulouse cedex 9

{prenom}.{nom}@irit.fr

RESUME

De nos jours, les systèmes proposent plusieurs vues utilisant différents types de données conjointement, comme 2D et 3D, afin de faciliter la compréhension et l'analyse des données. Cependant, les dispositifs classiques sont toujours utilisés pour interagir avec ces systèmes dont les données de natures différentes apparaissent comme étant complexes et spatialisées. Ces systèmes peuvent définir des situations d'interaction dites hétérogènes car plusieurs dimensions sont à considérer dans la réalisation de la tâche de l'utilisateur dans ce système. Notre objectif est de pouvoir adapter les interactions proposées à l'utilisateur avec cette tâche. Nous avons réalisé une première évaluation portant sur une technique ubiquitaire permettant d'interagir avec des données 3D, avant de l'évaluer dans le cas de situations hétérogènes. Les résultats de cette évaluation montrent que la technique est efficace et les utilisateurs ont apprécié de pouvoir réaliser des mouvements en 3D avec peu d'apprentissage, la technique étant facilement appréhendable.

Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: Input devices and strategies – Interaction styles.

General Terms

Design, Experimentation, Human Factors.

Mots clés

Techniques d'interaction mixte, tâche de pointage, loi de Fitts, étude d'utilisabilité, environnement virtuel 3D.

1. INTRODUCTION

Les récentes évolutions des technologies ont permis le développement des environnements virtuels 3D interactifs offrant des vues 3D toujours plus robustes et réalistes. Pour tirer pleinement avantage de ces vues 3D, des travaux ont porté sur l'utilisation conjointe de celles-ci avec d'autres types de vues, comme avec de la 2D [4], [24]. Ces travaux s'inscrivent dans les études menées sur les systèmes à multi-vues coordonnées, systèmes notamment utilisés pour les Systèmes d'Information de Géolocalisation (SIG) comme GoogleEarth [15] (cf Figure 1) ou ArcGIS [10], en industrie pour le monitoring (EasyMonitoring [13]) ou encore les bureaux d'études (CATIA [8], EasySensorManager [14]). L'utilisation de plusieurs vues a pour objectif de faciliter la manipulation et la compréhension de données complexes en proposant différents points de vue de ces données. Il émerge alors de ces systèmes une hétérogénéité due aux espaces d'affichage, données et représentations utilisées : ils sont de types variés et de tailles différentes, les données représentées forment alors des environnements de perception complexes. De plus, cette hétérogénéité des données visualisées

fait alors émerger une complexité dans les interactions de l'utilisateur.

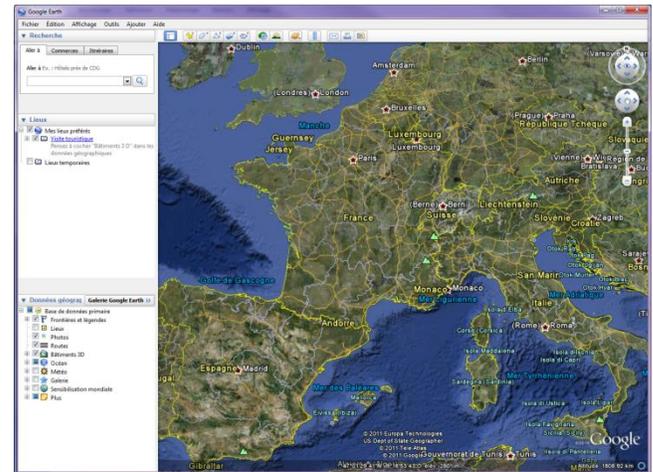


Figure 1 : GoogleEarth, exemple de SIG avec une vue 3D

Pourtant, les dispositifs classiques (le clavier et la souris) sont toujours les seuls à être mis à disposition des utilisateurs pour réaliser les différentes interactions, telles que la sélection ou encore la navigation. Il apparaît donc crucial de proposer des techniques d'interaction adaptées aux tâches de l'utilisateur. Ces techniques peuvent faciliter l'accès à ces représentations complexes et aux données qu'ils contiennent, et ainsi améliorer la compréhension des données manipulées. Des systèmes en émergence ont désormais pour objectif de combler le gouffre existant entre le monde numérique, ce qui est perçu par l'utilisateur, et le monde physique. Des solutions existent notamment dans le domaine des systèmes ubiquitaires et il pourrait être intéressant d'étudier de tels systèmes. En effet, ces systèmes ubiquitaires ont la particularité de s'intégrer aux contextes dans lesquels ils sont placés [31] mais aussi d'être intégrés à des objets du quotidien. Les utilisateurs étant familiers avec ces objets, ils rencontrent moins de difficultés à les manipuler puisque cette manipulation est usuelle pour eux et peut rendre l'accès plus simple aux données, quelle que soit leur dimension (2D ou 3D par exemple) ou encore leur espace d'affichage.

Parmi les différentes formes de systèmes ubiquitaires, nous privilégions l'étude et la mise en œuvre des Systèmes d'Interaction Mixte (SIM). Les SIM couvrent les interactions pervasives, mais ils regroupent également les systèmes de réalité mixte ou augmentée et les interactions tangibles. Un projet comme GeoTUI [7] utilisait, par ailleurs, les interactions tangibles dans un contexte ubiquitaire. Les SIM prennent

avantage des objets physiques pour interagir avec des entités numériques ; ces objets quotidiens, utilisés ici pour communiquer avec un système, constituent alors une véritable extension de l'utilisateur : ils font partie intégrante de la boucle d'interaction. Les SIM conduisent donc à un entrelacement des IHM avec les objets de la vie quotidienne et rejoignent en cela la définition de systèmes ubiquitaires fournie par Weiser [31]. En effet, Weiser abordait le fait de pouvoir enlever toute contrainte à l'utilisateur, « when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals ». Ce type de problématique rejoint celle des SIM pour faciliter l'interaction entre le monde physique et les systèmes numériques.

Dans cet article, nous proposons d'évaluer une technique d'interaction mixte mise en œuvre dans le contexte d'un système à multi-vues coordonnées. Nous présentons dans un premier temps des situations d'interaction hétérogènes induites par l'interaction avec de tels systèmes. Nous évaluons ensuite notre technique d'interaction mixte sur une tâche de pointage en 3D, cette tâche étant essentielle dans le contexte étudié.

2. CARACTERISATION DE L'HETEROGENEITE

2.1 Caractéristiques des systèmes à multi-vues coordonnées

L'application proposée par Global Vision Systems (GVS - cf. Figure 2) est un système à multi-vues coordonnées. Ce type de système utilise au moins 2 vues pour présenter un même concept. Les différentes vues rendent possible les comparaisons et permettent, entre autres, de faciliter la mémorisation des informations [29]. Les vues diffèrent par l'origine des données, qui peuvent être de types différents tels que la température, des positions, etc ; ou par leur représentation visuelle, des vues soit paramétriques, soit systèmes, ou encore 3D ou 4D sont ainsi proposées dans le cas de l'application de GVS [21]. L'utilisation des multi-vues est avantageuse lorsque le nombre de données à représenter est volumineux. Elles permettent ainsi d'avoir une meilleure vue d'ensemble de ces données selon plusieurs points de vue [23]. L'objectif est ainsi de faciliter l'accès aux données, leur compréhension et la prise de décisions.

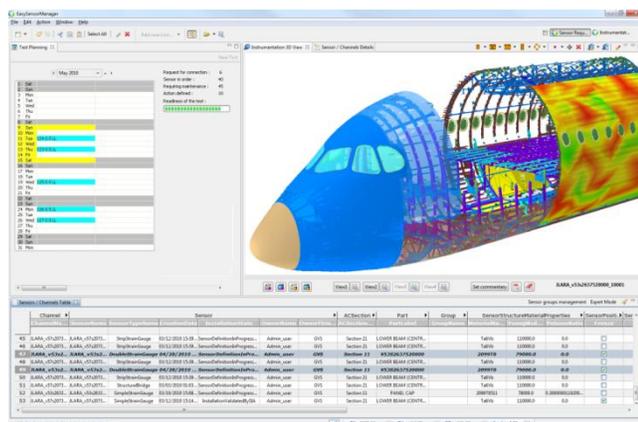


Figure 2 : EasySensorManager, systèmes à multi-vues coordonnées

Des études ont été réalisées sur les interactions possibles sur ces vues multiples afin de faciliter la manipulation et le traitement de ces données [23]. Seulement, l'interaction des utilisateurs avec ces données complexes sur un ensemble de tâches

enchaînées n'a pas été étudiée. Nous nous sommes alors intéressés à comprendre l'impact de la complexité de ces systèmes sur les interactions des utilisateurs.

2.2 Définition de situations d'interaction hétérogènes

Face à de tels systèmes, l'interaction avec le système s'avère plus riche et complète que dans une situation classique clavier – écran – souris. Plusieurs données sont représentées plusieurs fois, les lieux de perception sont multiples, les zones d'interaction également, etc. Sur la base des principales caractéristiques des systèmes à multi-vues coordonnées, nous avons défini la notion d'environnement d'interaction afin de clarifier le cadre dans lequel se déroule l'interaction avec un système à multi-vues coordonnées.

Un environnement d'interaction est constitué de 5 aspects :

- Modalité(s) d'entrée,
- Modalité(s) de sortie,
- Tâches utilisateur,
- Temporalité,
- Configuration spatiale.

Ces 5 aspects permettent de « cartographier » cet environnement (cf Figure 3).

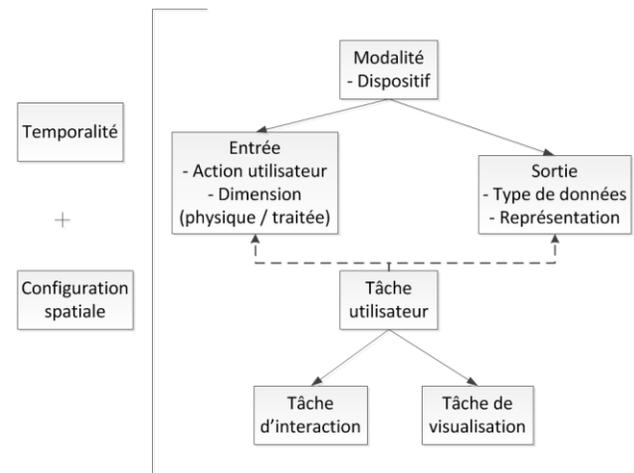


Figure 3 : Cartographie d'un environnement d'interaction

Les deux premières aspects sont des modalités d'entrée et de sortie utilisés par l'utilisateur. Nous avons adapté ces termes de la définition originale donnée par [20]. Dans notre cas, nous avons considéré qu'une modalité d'entrée avait pour attributs : un dispositif, une action utilisateur (pression, clic, etc) ainsi que deux types de dimension, la dimension physique décrivant les degrés de liberté du dispositif (par exemple dans une utilisation normale de la souris, 3 DDL existent : 2 translations et une rotation) et les dimensions traitées décrivant les dimensions réellement captées et transmises au système (par exemple la souris à 2 dimensions seulement transmises).

La modalité de sortie comprend également trois attributs : un dispositif (comme un écran), un type de données (ceux issus de la taxonomie de [25] et incluant notamment la 2D, la 3D, les données multidimensionnelles ou encore les arbres) ainsi que des représentations (celles issues de la taxonomie haut-niveau de [18] dont les graphes ou les tables par exemple). Les types de

données présents dans l'application de GVS sont par exemple des données sous forme d'arbre, en 3D, des données temporelles ou encore multidimensionnelles.

Ces deux modalités sont impliquées dans une tâche réalisée par un utilisateur, ce qui constitue le 3^{ème} aspect de l'environnement. Cette tâche peut être de deux types : soit d'interaction (taxonomies de [12] et [5] selon le type de données), soit de visualisation [27] (ces différentes tâches s'appliquant aux données à plusieurs dimensions). Dans le système de GVS, il est ainsi possible de réaliser des tâches de sélection en 1D/2D ou en 3D, de réaliser la tâche de navigation en 3D comme tâche d'interaction. Il est également possible d'identifier, de comparer et de configurer comme exemples de tâche de visualisation.

L'utilisateur ne communique pas seulement une fois avec le système mais au cours du temps ; il est ainsi amené à effectuer un enchaînement de tâches [2], [28], ceci correspond à la 4^{ème} aspect mise en œuvre dans l'environnement. L'enchaînement de tâches peut correspondre à une séquence, deux tâches réalisées l'une après l'autre sans délai, ou encore à un délai lorsque deux tâches sont réalisées l'une après l'autre, mais la seconde débute après un temps de délai (cf Figure 4). Ceci peut être le cas dans l'application de GVS.



Figure 4 : Exemples d'enchaînement de tâches

Le 5^{ème} aspect correspond à la configuration spatiale [9], [6] des modalités. Elle peut être soit intra-modalités, soit inter-modalités. La spatialisation pour une modalité d'un même type, soit d'entrée, soit de sortie, correspond à une configuration spatiale intra-modalités. Cette spatialisation peut se caractériser par le fait d'utiliser deux modalités d'entrée dans une même zone d'interaction : par exemple, un clavier et une souris utilisés dans un espace de type disjoint. Elle peut également se caractériser par l'utilisation d'une même modalité dans deux zones d'interaction : par exemple, manipuler une souris sur une table et en l'air correspond à un espace qui se rencontre (cf Figure 5).

D'autres exemples de configuration spatiale intra-modalités peuvent être donnés pour les modalités de sortie : l'utilisation de deux écrans dans une même zone d'interaction, pour un exemple de disjonction, ou l'interaction avec deux types de données formant deux zones d'interaction mais situés dans un même écran, pour un exemple de rencontre.

La spatialisation entre au moins deux modalités de types différents correspond à une configuration spatiale inter-modalités. La configuration spatiale inter-modalités comprend ainsi l'ensemble des zones d'interaction formées par ces différentes modalités.

Sur la base de ce cadre de définition d'un environnement d'interaction, nous établissons alors qu'une situation d'interaction est dite hétérogène s'il existe au moins 2 valeurs différentes pour l'un des quatre éléments suivants :

- (a) d'attributs pour les entrées / sorties,
- (b) d'enchaînements de tâches,
- (c) d'organisations spatiales,
- (d) de liaisons tâche-modalité.

Nous allons illustrer ces différents cas par des exemples issus de l'application de Global Vision Systems [14].

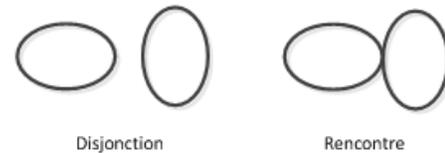


Figure 5 : Exemples de configuration spatiale

L'application de Global Vision Systems (cf Figure 2) propose l'utilisation conjointe de plusieurs vues. Il existe ainsi plusieurs valeurs de l'attribut type de données pour la modalité de sortie : dans cet exemple, des données en 3D, des données temporelles et des données multidimensionnelles sont présentes pour un même dispositif, qui se trouve être un seul écran. C'est un cas d'hétérogénéité de type (a).

Cette application propose également (b) la possibilité de réaliser une tâche d'interaction suivie en séquence par une tâche de visualisation (sélection d'un élément puis identification de celui-ci).

En ce qui concerne (c) les organisations spatiales, les utilisateurs interagissent avec le système avec un clavier et une souris qui ont une configuration intra-modalités disjointe en entrée tandis que les différentes zones d'interaction de la modalité de sortie forment une configuration intra-modalités qui se rencontre. Nous avons donc deux valeurs différentes de configurations spatiales intra-modalités ce qui constitue une forme d'hétérogénéité.

En ce qui concerne (d) les liaisons tâche-modalité, un exemple présent dans cette application est la possibilité de réaliser une même tâche d'interaction, comme celle de sélection dans deux types de données différents qui sont des données en 3D puis des données multidimensionnelles.

L'utilisation de vues 3D étant de plus en plus importante, et ce type de données étant le cœur de métier de Global Vision Systems, nous avons posé comme prédicat que nos travaux prendront toujours en compte des vues 3D couplées à d'autres types de données.

L'ensemble de ces exemples implique différentes dimensions de l'environnement d'interaction créant des situations d'interaction hétérogènes. L'utilisateur est alors amené à manipuler des données de natures différentes dans des espaces variés et interagit avec un système qui s'avère complexe. Notre objectif est de proposer à l'utilisateur une technique d'interaction adaptée à ces différentes situations d'interaction hétérogènes afin de faciliter son interaction avec le système.

2.3 Solution proposée

Notre objectif est de proposer une solution adaptée aux tâches réalisées par les utilisateurs dans le cadre d'une application industrielle. Proposer un système ubiquitaire ou d'interaction mixte consiste à intégrer notre solution dans un objet du quotidien lié à la tâche métier de l'utilisateur. Pour commencer nos travaux, nous nous sommes abstraits de la conception de l'objet métier du quotidien pour proposer une solution adaptable à n'importe quel objet et ainsi à plusieurs contextes autres que celui de notre cadre d'étude.

Pour répondre à la contrainte de prise en compte des vues 3D, nous avons comme besoin de pouvoir réaliser des mouvements dans au moins 3 degrés de liberté. Mais cette solution doit aussi

permettre de manipuler des données avec moins de degrés de liberté afin de répondre aux situations d'interaction hétérogènes identifiées dans cette application. Pour ce faire, nous avons choisi d'utiliser un dispositif isotonique, i.e. un dispositif dont la position dans l'espace sert à manipuler un pointeur. Ce choix s'est appuyé sur différentes études recensées dans [26] qui ont démontré les performances de la souris, un dispositif isotonique. Nous avons ainsi utilisé un dispositif isotonique avec au moins 3 degrés de liberté et ne comportant pas de fil : le *Polhemus Patriot* [22]. Il est donc possible de contrôler le curseur en 3D selon sa position dans l'espace récupérée par un capteur électromagnétique.

Pour anticiper l'intégration de ce dispositif dans un environnement d'interaction potentiellement hétérogène, nous avons au préalable conduit une évaluation de ce dispositif pour le pointage en 3D, tâche essentielle pour les activités de GVS. Cette évaluation traite conjointement la performance du dispositif et la satisfaction des utilisateurs. Avant même de pouvoir étudier la performance du dispositif dans le cas de situations d'interaction hétérogènes, nous avons évalué dans un premier temps les performances et la satisfaction de cette solution dans un environnement 3D. Nous définissons un environnement 3D comme : un espace où 1 ou plusieurs utilisateurs interagissent en temps réel par le biais de moyens physiques avec des données numériques 3D générées par un ordinateur. Notre cas d'étude, situation d'interaction présente dans l'application métier, porte sur la sélection et la modification de la position d'objets en 3D – sans modification de l'orientation [5]. Nous avons alors évalué la tâche de pointage dans ce contexte.

3. EVALUATION DE LA SOLUTION PROPOSEE

L'évaluation conduite s'appuie sur le protocole proposé par la norme 9241-9 [17]. Les adaptations de la norme que nous avons du apporter ont consisté à placer la tâche dans un environnement virtuel 3D mais également à proposer une évaluation de la satisfaction des utilisateurs. Celle-ci semble importante à évaluer afin de connaître des aspects liés à l'expérience utilisateur, point particulièrement prépondérant pour ce type d'interaction avancée. Nous détaillons ci-après les méthodes d'évaluation de la tâche de pointage ainsi que les caractéristiques principales de la norme ISO 9241-9.

3.1 Evaluation d'un dispositif de pointage

3.1.1 Loi de Fitts

La loi de Fitts [11] a été initialement conçue pour prédire le temps mis par un utilisateur pour atteindre une cible. Les expériences qui ont mené à la validation de cette loi consistaient à effectuer des tâches de pointage sur un plan et sur un seul axe soit en 1D. Ce temps est dépendant de la taille de la cible et de la distance à parcourir. La formulation couramment employée est :

$$MT = a + b \log_2(D/W + 1)$$

où a et b sont des constantes dépendantes du contexte d'interaction et sont déterminées de façon empirique. W correspond à la taille de la cible et D est la distance à parcourir pour atteindre la cible souhaitée. Le terme logarithmique est l'indice de difficulté (ID) mesuré en bits.

3.1.2 Extension de la loi de Fitts pour un pointage en 3D

Suite à cette première loi, des propositions de modèle ont été faites pour étendre cette formule à deux ou trois dimensions. Ainsi, MacKenzie et Buxton [19] ont proposé deux équations différentes pour le calcul de l'ID en deux dimensions. Mais, Zhai et Accot [1] ont montré par la suite certaines limitations à ces modèles, telles que le manque de prise en compte de la contrainte de direction ou encore une modification de hauteur comparée à la largeur, et ont eux-mêmes proposé un modèle pour la 2D. Toutefois, l'équation classique de la loi de Fitts demeure la plus utilisée pour les évaluations comparées à ces modèles.

Certaines adaptations de la loi de Fitts ont également été proposées pour la 3D, comme le modèle de Ware et Lowther [30] qui ont étendu le modèle 2D défini par MacKenzie et Buxton (où W est la taille de la cible, H la hauteur et D la profondeur).

$$IDe = \log_2(A/\min(W, H, D) + 1)$$

Cependant, ce modèle souffre des mêmes problèmes que ceux afférents au modèle de référence et il n'a pas été validé dans une expérience qui manipule explicitement les 3 dimensions de la cible. Par ailleurs, dans le cas d'utilisation d'une sphère en tant que cible, ce modèle reste similaire à la loi de Fitts classique. Un autre modèle proposé est celui de Grossman et Balakrishnan [16] où leur étude porte sur les mouvements physiques de l'utilisateur dans le plan XZ pour un affichage volumétrique 3D. Ils ont ainsi validé leur modèle pour un affichage volumétrique et non un Environnement Virtuel 3D. De plus, les mouvements étaient contraints sur l'axe (X,Z) avec ($Y=0$), tous les axes 3D n'ont donc pas été évalués.

3.1.3 Norme ISO 9241-9

La norme ISO 9241-9 nommée « Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV), Partie 9 : Exigences relatives aux dispositifs d'entrée autres que les claviers », fournit des exigences pour la conception et l'évaluation de dispositifs contrôlés par la main tels que souris, manettes, joysticks. Celle-ci propose 6 tâches élémentaires à déterminer selon l'utilisation du dispositif en vue de mesurer son débit. Par ailleurs, les différentes variables à mesurer sont recommandées en fonction de la tâche sélectionnée. La norme propose également une étude qualitative basée sur un questionnaire pour évaluer le confort et sur une échelle pour graduer l'effort requis lors de la tâche.

Cependant, ces exigences sont plus particulièrement à destination des dispositifs à 2 degrés de liberté et les tâches élémentaires ne concernent que des espaces 1D et 2D. Cela ne correspond donc pas à nos attentes et la norme ne nous apporte pas de recommandations pour évaluer et comparer des dispositifs de pointage à 6 degrés de liberté, ni dans un environnement 3D.

Désormais, nous allons présenter les différents éléments du protocole mis en place pour l'évaluation.

3.2 Méthodologie utilisée

3.2.1 Hypothèses

Une étude de Zhai [32] ayant déjà porté sur ce type de dispositif pour une tâche de positionnement, nous nous appuyons entre autres sur ses résultats pour formuler nos hypothèses.

Tout d'abord, peu de preuves de l'application de la loi de Fitts en 3D ont été apportées. Notre première hypothèse (H1) est donc que cette loi peut s'appliquer dans ce type d'environnement et pour ce type de dispositif. Notre deuxième hypothèse (H2) est que le dispositif isotonique entrainera de la fatigue pour les utilisateurs. D'un point de vue de la préférence utilisateur, notre troisième hypothèse (H3) est que le dispositif isotonique sera satisfaisant pour les utilisateurs. Son interaction habituelle et la nouveauté de ce type d'interaction seront ainsi appréciées par les utilisateurs. Nous avons formulé une 4^{ème} hypothèse (H4) portant sur les mouvements réalisés par l'utilisateur : les performances du dispositif seront moins bonnes pour des mouvements complexes effectués sur les 3 dimensions que pour des mouvements simples effectués sur un seul axe.

3.2.2 Participants

Quatorze utilisateurs (3 femmes et 11 hommes) ont participé à notre étude et ont testé le dispositif. Dans la suite de notre étude, nous n'avons pris en compte les résultats de seulement 13 utilisateurs, les logs d'un des utilisateurs étant erronés. Ces utilisateurs sont familiers de l'outil informatique, mais pas du périphérique évalué. Ils sont âgés entre 22 et 39 ans (moyenne : 27,5 ; écart-type : 5,26) et utilisent leur main dominante (12 droitiers et 1 gaucher).

3.2.3 Matériel utilisé

3.2.3.1 Plateforme d'évaluation

Pour l'évaluation de la tâche de pointage, nous avons étendu les recommandations de la norme ISO 9241-9 [17] proposées pour la 2D à de la 3D. Nous avons ainsi transformé les cercles représentant les cibles à atteindre par des sphères, réparties équitablement sur le contour d'une sphère au lieu d'un cercle. Les 14 sphères cibles ont été positionnées de manière à avoir une symétrie par rapport au centre de la sphère support (cf. Figure 6). La tâche de pointage demandée consiste alors à partir d'une des cibles et d'aller pointer le plus rapidement et le plus précisément possible son symétrique sur la sphère support. Ainsi, la distance à parcourir est toujours égale au diamètre de la sphère support.

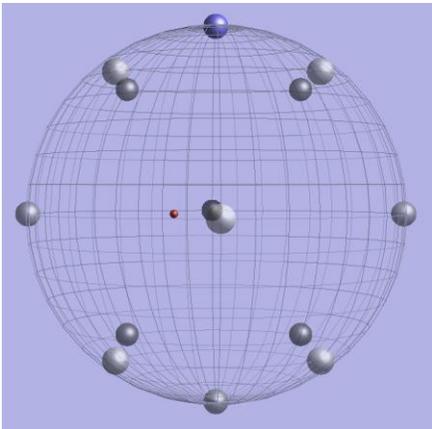


Figure 6 : Plateforme d'évaluation

La perception de la profondeur dans un environnement 3D est toujours difficile pour les utilisateurs. Nous avons alors utilisé un dégradé de couleurs sur les différentes sphères cibles ainsi qu'une représentation en fil de fer pour la sphère globale afin de faciliter cette perception. Les sphères cibles ont une transparence suffisante pour percevoir les sphères situées derrière et pour faciliter le positionnement du pointeur par

rapport à ces sphères cibles. Nous n'avons pas utilisé d'ombres pour ne pas surcharger la scène, ni la rendre illisible. La taille du curseur est également modifiée automatiquement en fonction de sa distance à la caméra de la scène : lorsque le curseur est devant, il est plus gros que lorsqu'il est derrière.

3.2.3.2 Dispositif testé

L'expérimentation a eu lieu dans un laboratoire des usages pour tester le *Polhemus Patriot* [22], un système de tracking isotonique (cf Figure 7). Nous avons appliqué un contrôle par position pour le *Patriot*.

L'expérimentation a été conduite sur un ordinateur portable *HP Elitebook 6930p* combiné à un écran LCD d'une résolution de 1920*1080. Les participants n'avaient face à eux que l'écran LCD et non pas le PC portable. Le système d'exploitation utilisé était *Windows 7*. Le processeur était un Intel Core Duo de 2.4 GHz avec 4GB de RAM. La carte graphique était une ATI Mobility Radeon HD 3450 de 256Mo. Le dispositif était branché à un port USB 2.0.



Figure 7 : dispositif isotonique, Polhemus Patriot

3.2.3.3 Données recueillies

Les événements ont été recueillis sous forme de logs au format XML. Ces événements étaient : le début et la fin de chaque bloc de tâche de pointage, les positions du curseur, la validation de la sélection, les erreurs ainsi que les temps d'exécution des événements.

3.2.4 Procédure

Après avoir signé un accord de consentement éclairé, les participants s'asseyaient devant un bureau où se situaient un écran d'ordinateur et un des deux périphériques à tester. En premier lieu, ils remplissaient un questionnaire de recueil des profils, puis ils prenaient connaissance du déroulement de l'expérimentation. Une démonstration de la tâche attendue ainsi qu'une phase d'apprentissage d'environ 10 minutes leur étaient proposées.

La tâche demandée à l'utilisateur est de sélectionner une cible fixe le plus rapidement possible et en faisant le moins d'erreurs possibles. L'action de sélection consiste à pointer la cible en déplaçant le curseur sur la cible à atteindre au moyen du dispositif de pointage. La tâche de sélection débutait dès lors que l'utilisateur avait sélectionné la cible de départ et que la cible d'arrivée était indiquée. Elle se terminait lorsque l'utilisateur avait validé la sélection de la sphère d'arrivée. Les utilisateurs pouvaient prendre une pause lorsqu'ils le souhaitaient entre deux tâches de pointage. Une fois tous les blocs réalisés, les participants remplissaient un questionnaire de satisfaction [3] afin d'évaluer l'utilisabilité du dispositif ainsi que l'échelle comparative issue de la norme 9241-9. Une fois cette procédure terminée avec le premier dispositif, ils recommençaient la même chose avec le second.

3.2.5 Plan expérimental

Les variables indépendantes étaient la taille de la cible à atteindre (15px, 30px, 45px) et la distance à parcourir pour pointer la cible (150px, 450px, 750px). Les combinaisons de taille et de distance donnaient des Indices de Difficultés (ID) compris entre 2,114 et 5,672 (par exemple taille=15px et distance=450px donne ID =4,95).

Le test du dispositif s'appuyait sur un exercice comportant 9 blocs et chaque bloc comportait une série de 14 tâches de pointage. Toutes les tâches de pointage d'un bloc avaient le même indice de difficulté : i.e., la cible à atteindre a la même taille et la distance à parcourir étaient identiques pour toutes les tâches d'un même bloc. Seuls la direction et le sens du pointage à réaliser changeaient. Chaque bloc d'un exercice correspondait à un des 9 indices de difficulté défini précédemment. Ainsi pour chaque exercice, les 9 indices de difficulté proposés ont été étudiés. L'ordre des blocs suivait un contrebalancement par carré latin.

Les variables dépendantes étaient la vitesse (temps d'acquisition de la cible, MT) et la précision (taux d'erreur). Nous avons également dépisté le degré d'utilisabilité perçue en proposant les items du questionnaire SUS et quelques-uns du SUMI ainsi que le questionnaire sur le confort issu de la norme 9241-9.

3.3 Résultats

3.3.1 Mesures quantitatives

Cette section présente les résultats quantitatifs mesurés lors de l'expérimentation qui permettent de déterminer l'efficacité et l'efficacité du dispositif.

3.3.1.1 Temps moyen et taux d'erreur

Nous avons calculé les temps moyens mis par les utilisateurs pour réaliser la tâche : le dispositif isotonique a un temps moyen de 6193ms.

Nous avons mené dans les mêmes conditions une première étude avec un souris 3D, dispositif isométrique plus classiquement utilisé en EV3D, et le temps mis pour réaliser la tâche est plus élevé (10585ms). Grâce à un test ANOVA, nous pouvons dire qu'il existe une différence significative entre les temps des deux dispositifs ($F_{(1,11)} = 25,6764, p = 0,0000008$). Le *Patriot* s'avère ainsi plus efficace puisqu'il a un meilleur temps de réalisation de la tâche.

L'analyse du taux d'erreur du dispositif peut fournir une indication sur la précision avec laquelle les utilisateurs ont réalisé la tâche et ainsi nous indiquer l'efficacité du dispositif. Le *Patriot* a un taux d'erreur de 7,64%. Les taux d'erreur maximum et minimum sont respectivement 35%, obtenu sur l'ID=5,672, et 7%. Si nous comparons ce résultat à celui obtenu avec le dispositif isométrique lors de notre étude préalable, le taux d'erreur du dispositif isométrique est de 4,4%. Grâce à un test ANOVA, nous pouvons dire qu'il existe une différence significative entre les taux d'erreur des deux dispositifs ($F_{(1,11)} = 8,263, p = 0,004$). Le *Patriot* s'avère moins efficace puisqu'il a un taux d'erreur plus élevé que celui du dispositif isométrique.

Le dispositif isotonique permet donc d'atteindre plus facilement et rapidement les cibles que le dispositif isométrique, mais finalement au détriment de la précision.

3.3.1.2 Analyse de la loi de Fitts

Nous avons analysé le temps moyen mis par les utilisateurs pour réaliser la tâche en fonction de l'ID. Dans la Figure 8, l'axe X et

l'axe Y représente respectivement l'Indice de Difficulté (ID, en bits) et le temps moyen réalisé par les utilisateurs et calculé par ID (en ms). La droite du dispositif isotonique a pour équation $MT = 2584,2 \times ID - 3811,6$ avec $R^2 = 0,8513$.

Cette analyse met en évidence un problème : la corrélation entre l'ID et le temps moyen (MT) pour le dispositif est faible ($R^2 < 0,9$). Le nuage de points révèle que les temps moyens sont au-dessus de la droite de régression linéaire pour les IDs dont les cibles avaient une distance de 750 (la plus longue distance) et dont les IDs sont 4.143, 4,7 et 5.672.

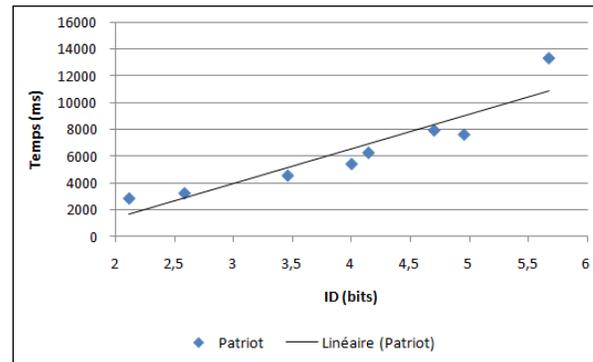


Figure 8 : Régression linéaire MT vs ID

Pour vérifier que cette distance est bien un problème révélé par l'analyse de régression, nous avons calculé une seconde droite de régression linéaire sans les IDs correspondant à la distance de 750 pixels (cf Figure 9). La nouvelle droite de régression linéaire est $MT = 1675 \times ID - 991,82$ avec $R^2 = 0,9765$ pour ce dispositif. Sans la distance la plus élevée, la corrélation entre MT et ID est ainsi plus forte. Il semble donc que cette distance de 750 pixels soit trop élevée et induise des perturbations dans les résultats.

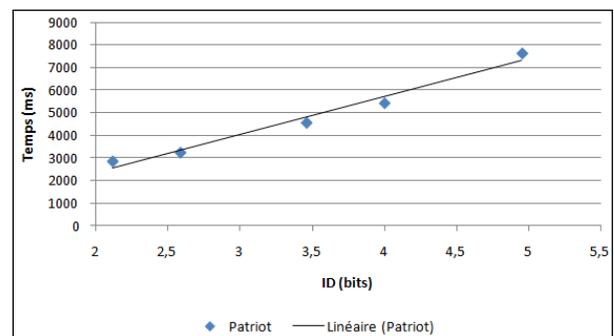


Figure 9 : Régression linéaire MT vs ID sans la distance de 750 pixels

Nous pouvons ainsi remarquer que plus les distances sont importantes, plus il est difficile d'atteindre les cibles avec le dispositif isotonique.

Cette analyse démontre que la loi de Fitts peut s'appliquer aux environnements 3D. Notre première hypothèse (H1) est donc validée.

3.3.1.3 Analyse des mouvements réalisés par les utilisateurs

Les mouvements des utilisateurs sont réalisés en 3D dans l'espace physique de l'utilisateur mais également dans l'espace numérique, et deux types de mouvements ont pu être effectués : des mouvements simples sur une dimension pour les cibles

situées sur les axes principaux, et des mouvements complexes sur trois dimensions pour les cibles hors axes principaux. Nous avons effectué des analyses de variance (ANOVA) sur le temps mis par les utilisateurs pour atteindre les cibles et les distances parcourues selon la disposition des sphères.

L'analyse des résultats montre qu'il n'existe pas de différence significative entre les mouvements simples et complexes réalisés avec le *Patriot* que ce soit pour le temps mis pour réaliser la tâche ou pour la distance parcourue. Ces résultats ne valident pas (H4), toutefois, ces résultats sont intéressants car ils montrent que les dimensions impliquées dans les mouvements réalisés par les utilisateurs n'ont pas affecté les performances des utilisateurs. Ces résultats sont également pertinents pour la suite de nos travaux car les performances étant similaires quelle que soit la dimension manipulée, ce dispositif semble être une solution qui convient à des interactions se situant dans différentes dimensions et spatialisations, et ainsi répondre à des situations d'interactions hétérogènes.

3.3.2 Mesures qualitatives

Plusieurs questionnaires et entretiens ont été proposés aux utilisateurs afin d'analyser leur satisfaction sur le dispositif. Cette section détaille les différents résultats collectés d'après les différentes dimensions évaluées, notamment l'utilisabilité et le confort.

3.3.2.1 Utilisabilité et Satisfaction des utilisateurs

L'utilisabilité du dispositif a été estimée par les utilisateurs par le biais du questionnaire SUS [3]. Un score a pu être calculé et il est de 64,4 pour le dispositif isotonique. D'après [3], ce score traduit une utilisabilité tangente et montre que le dispositif peut encore être amélioré au niveau de son utilisabilité, même s'il peut déjà être utilisé (comme le montrent les sections précédentes). Des améliorations possibles ont été identifiées par le biais des questionnaires et des entretiens, celles-ci sont présentées dans la conclusion. En plus du questionnaire SUS, six items issus du SUMI ont également été posés aux utilisateurs pour avoir des résultats sur d'autres dimensions complémentaires. Nous avons retenu les items traitant de la frustration, du plaisir ou encore de la stimulation. Nous avons également évalué les items concernant l'utilisation quotidienne du dispositif, la simplicité de la réalisation de la tâche et la vitesse du dispositif. Ces points n'étaient pas traités par le questionnaire de la norme et nous ont donc permis de compléter les items déjà présents dans ce questionnaire. Il ressort de l'analyse de ces trois items que les utilisateurs ont trouvé que le dispositif était stimulant à utiliser, que la tâche était simple à réaliser et que sa vitesse d'utilisation était suffisamment rapide (moyenne > 3,6 pour une échelle de Lickert allant de « 1 : pas du tout d'accord » à « 5 : tout à fait d'accord »).

Nous avons demandé aux utilisateurs de déterminer les points positifs et négatifs du dispositif. Ces questions sont pertinentes pour indiquer des améliorations possibles. Les réponses les plus fréquentes pour le dispositif isotonique sont :

- Pour les 3 points les plus positifs : (a) « l'intuitivité » (b) le fait de pouvoir faire des mouvements en 3D (c) se repérer facilement dans l'espace.
- Pour les 3 points les plus négatifs : (a) la fatigue (b) le design (c) les difficultés à rester immobile ou le bras levé, points qui sont liés.

Ces résultats montrent que les utilisateurs ont avant tout retenu l'interaction facilement appréhendable du *Patriot*, mais

également la fatigue qu'il peut engendrer. Celle-ci peut être expliquée par le fait de garder le bras levé pour les mouvements en 3D, mais cela est contrebalancé par le fait que les utilisateurs ont pu se repérer facilement dans l'espace grâce à des repères qu'ils ont pu se construire. Il sera alors intéressant d'étudier davantage ce point.

L'ensemble de ces résultats semblent valider notre troisième hypothèse (H3) sur la satisfaction des utilisateurs. La section suivant complète l'analyse de la satisfaction par celle du confort d'utilisation, initialement l'unique dimension traitée par la norme 9241-9.

3.3.2.2 Confort des utilisateurs

Nous avons utilisé l'échelle d'évaluation du confort proposée par la norme ISO 9241-9 qui comprend deux types d'indices : les indices généraux et ceux sur la fatigue. Les résultats concernant la force d'activation, la souplesse d'utilisation et la précision sont bons pour le dispositif avec une moyenne supérieure à 3,5 (sur une échelle allant de « 1 : très négatif » à « 5 : très positif »). Les résultats ont pu mettre en évidence le fait que les utilisateurs ont ressenti une fatigue assez forte au bras et à l'épaule. Ces résultats confirment donc la 2^{ème} hypothèse (H2) concernant la fatigue. Nous pouvons toutefois ajouter que la fatigue pour les doigts, le poignet et le cou est quasi inexistant (moyenne > 4,2).

4. CONCLUSION

La manipulation de données spatialisées et complexes se manifestent dans nos travaux par des situations d'interaction hétérogènes. Nous avons donc dans cet article présenté l'ébauche d'un cadre permettant de cartographier un environnement d'interaction. Ce cadre identifie plusieurs dimensions permettant de décrire les situations d'interaction et surtout de déterminer les caractéristiques de leur hétérogénéité. Ce travail a été mené afin de pouvoir proposer des solutions d'interaction adaptées au mieux aux tâches des utilisateurs et améliorer l'accès aux données et leur manipulation.

En vue de son utilisation en réponse à une situation d'interaction hétérogène, nous avons dans un premier temps conçu et évalué une technique d'interaction ubiquitaire pour pointer dans un environnement virtuel 3D. Cette solution a été testée d'un point de vue performance et satisfaction des utilisateurs. Les résultats de cette évaluation montrent que le dispositif est efficace et permet de réaliser la tâche de pointage de façon performante. La solution est aussi performante pour des mouvements simples que pour des mouvements complexes, ce qui est intéressant pour la poursuite de nos travaux. Du point de vue des mesures qualitatives, les utilisateurs ont été satisfaits par l'utilisation de cette solution et ont mis en avant sa facilité de manipulation. Une certaine fatigue a pu être ressentie dans le bras ou l'épaule par les utilisateurs, et ceci reste donc un point à améliorer pour nos travaux futurs. D'autres améliorations comme la taille du dispositif, le fait de pouvoir rester immobile ou d'éviter de garder le bras levé pourront également être réalisées.

Les perspectives données à ces travaux conduiront à la mise en œuvre de cette technique dans un contexte hétérogène et à son évaluation en termes de performance et satisfaction. Un parallèle pourra donc être dressé entre la situation d'utilisation hétérogène et classique. Elle pourra également être mise en parallèle d'une technique d'interaction classique (par exemple une souris 3D) pour mesurer l'impact d'une technique ubiquitaire sur la résolution d'une tâche hétérogène.

5. REFERENCES

- [1] Accot, J. and Zhai, S. 2003. Refining Fitts' law models for bivariate pointing. In *Proceedings of CHI '03*. ACM, New York, NY, USA, 193-200.
- [2] Allen, J. F.. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26, 11 (November 1983), 832-843.
- [3] Bangor, A. Kortum, P-T.&Miller, J-T. 2008. An empirical evaluation of the System Usability Scale. In *Journal of human-computer interaction*. Taylor&Francis group, 574-594.
- [4] Bleisch, S. and Nebiker, S. 2008. Connected 2D and 3D visualizations for the interactive exploration of spatial information. XXI ISPRS Congress. Beijing, China.
- [5] Bowman, D. A., Kruijff, E, LaViola, J. J., and Poupyrev, I.. 2004. 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- [6] Cheylan J.-P. 2007. Les processus spatio-temporels: quelques notions et concepts préalables à leur représentation, *M@ppemonde*, n ° 87 (3-2007), <http://mappemonde.mgm.fr/num15/articles/art07303.html>, octobre 2007.
- [7] Couture, N. and Rivière, G.. 2006. *3è Journées Francophones Mobilité et Ubiquité, Démonstrations*, (Paris, France, 5-8 Septembre, 2006), 163-164. ACM Press.
- [8] Dassault Systems. CATIA. <http://www.3ds.com/fr/products/catia/welcome/>.
- [9] Egenhofer, M. J. 1994. Spatial SQL: A Query and Presentation Language. *IEEE Trans. on Knowl. and Data Eng.* 6, 1 (February 1994), 86-95.
- [10] ESRI. ArcGIS. <http://www.esrifrance.fr/arcgis.asp>
- [11] Fitts, P. 1954. *The information capacity of the human motor system in controlling the amplitude of movement*. *Journal of Experimental Psychology*. 47. 381-391
- [12] Foley, J. D., Wallace, V. L., and Chan, P. 1984. *The human factors of computer graphics interaction techniques*. IEEE Computer Graphics and Applications, 13-48.
- [13] Global Vision Systems. EasyMonitoring. http://www.global-vision-systems.com/index.php?fr_EasyMonitoring
- [14] Global Vision Systems. EasySensorManager. http://www.global-vision-systems.com/index.php?fr_EasySensorManager.
- [15] Google. GoogleEarth. <http://www.google.fr/intl/fr/earth/index.html>
- [16] Grossman, T., and Balakrishnan, R. Pointing at trivariate targets in 3D environments. In *Proceedings of CHI '04*. ACM, New York, NY, USA, 447-454.
- [17] ISO, 9421--9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. 2000. International Organization for Standardization.
- [18] Lohse, G. L., Biolsi, K., Walker, N., and Rueter, H. H.. 1994. A classification of visual representations. *Commun. ACM* 37, 12 (December 1994), 36-49.
- [19] MacKenzie, I.S. and Buxton, B. 1992. W. Extending Fitts' law to two-dimensional tasks. In *Proceedings of CHI '92*, Penny Bauersfeld, John Bennett, and Gene Lynch (Eds.). ACM, New York, NY, USA, 219-226.
- [20] Nigay L, Coutaz J, 1997. Multifeature Systems: The CARE Properties and Their Impact on Software Design. *Intelligence and Multimodality in Multimedia Interfaces*, AAAI Press.
- [21] North, C., Shneiderman, B. 1997. *A taxonomy of multiple window coordinations*. Technical Report CS-TR-3854, University of Maryland.
- [22] Polhemus. Patriot. http://www.polhemus.com/?page=Motion_PATRIOT%20Wireless.
- [23] Roberts, J. C.. 2007. State of the Art: Coordinated \& Multiple Views in Exploratory Visualization. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV '07)*. IEEE Computer Society, Washington, DC, USA, 61-71.
- [24] Schafhitzel, T, Weiskopf, D. and Ertl, T. 2005. Interactive Exploration of Unsteady 3D Flow with Linked 2D/3D Texture Advection. In *Proceedings of the Coordinated and Multiple Views in Exploratory Visualization (CMV '05)*. IEEE Computer Society, Washington, DC, USA, 96-105.
- [25] Shneiderman, B. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)*. IEEE Computer Society, Washington, DC, USA, 336-.
- [26] Soukoreff, R. W. and MacKenzie, I. S.. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *Int. J. Hum.-Comput. Stud.* 61, 6 (December 2004), 751-789.
- [27] Valiati, E. R. A, Pimenta, M. S. and Freitas, C. M. D. S.. 2006. A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization (BELIV '06)*. ACM, New York, NY, USA, 1-6.
- [28] Vernier F. 2001. La multimodalité en sortie et son application à la visualisation de grandes quantités d'information. Grenoble 1 University PhD Thesis, 271 pages.
- [29] Wang Baldonado, M. Q., Woodruff, A., and Kuchinsky, A.. 2000. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces (AVI '00)*. ACM, New York, NY, USA, 110-119.
- [30] Ware C. and Lowther K. 1997. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Trans. Comput.-Hum. Interact.* 4, 4 (December 1997), 309-322.
- [31] Weiser, M. 1991. The computer for the 21st century. *Scientific American*, 94-104.
- [32] Zhai, S. 1998. User performance in relation to 3D input device design. *SIGGRAPH Comput. Graph.* 32, 4 (November 1998), 50-54

Sélection de services basée sur l'impact en environnement ubiquitaire

Benjamin Cogrel
Laboratoire Images Signaux et
Systèmes Intelligents
122 rue Paul Armangot
94400 Vitry-sur-Seine
benjamin.cogrel@u-
pec.fr

Boubaker Daachi
Laboratoire Images Signaux et
Systèmes Intelligents
122 rue Paul Armangot
94400 Vitry-sur-Seine
daachi@u-pec.fr

Yacine Amirat
Laboratoire Images Signaux et
Systèmes Intelligents
122 rue Paul Armangot
94400 Vitry-sur-Seine
amirat@u-pec.fr

Abdelghani Chibani
Laboratoire Images Signaux et
Systèmes Intelligents
122 rue Paul Armangot
94400 Vitry-sur-Seine
chibani@u-pec.fr

ABSTRACT

Le contexte a une importance cruciale dans la manière dont les actions sont perçues et réalisées. Dans un environnement ubiquitaire, le contexte est riche et varié; aussi, il mérite d'être pleinement pris en considération par les dispositifs évaluant la performance des systèmes, ce qui est le cas de la sélection de services. Malheureusement, cette relation entre le contexte et la sélection n'a pas été pleinement considérée dans les travaux précédents. Premièrement, la plupart des modèles de sélection existants se basent sur des paramètres de Qualité de Service (QoS) qui ont été estimés à partir des exécutions précédentes. Cependant, deux exécutions consécutives peuvent se dérouler dans deux contextes distincts et donc se comporter différemment. Aussi, ces paramètres de QoS doivent être prédits à partir du contexte. Enfin, l'agrégation de ces paramètres de QoS en un score reflète les attentes vis-à-vis du service; elle doit donc être dépendante du contexte. La solution proposée dans ce papier se focalise sur les services auxiliaires. Ces services auxiliaires assistent d'autres services durant leur exécution, typiquement en leur délivrant un flux de données. Au lieu de se concentrer sur leur performance individuelle, la sélection s'intéresse à leur impact sur le service assisté. Nous proposons d'obtenir ce modèle à partir d'un perceptron multi-couches sous apprentissage supervisé non-adaptatif. Ce modèle est validé par un scénario de robotique ubiquitaire impliquant une sélection de services de localisation.

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Metrics—*Process metrics*;
D.2.11 [Software Architectures]: Service-Oriented architecture (SOA); I.2.6 [Learning]: Connectionism and neural nets; I.2.9 [Robotics]: Manipulators, Sensors

General Terms

Design, Performance

Keywords

Sélection de service, qualité de service, robotique ubiquitaire, apprentissage artificiel, réseaux de neurones artificiels

1. INTRODUCTION

Il y a près de vingt ans, Mark Weiser [18] a proposé les prémisses d'un nouveau paradigme d'interaction, dans lequel les ressources de calcul sont distribuées dans de nombreux objets communicants intégrés dans leur environnement. Cette informatique, dite ubiquitaire, vise à utiliser ces ordinateurs ensemble; elle se focalise donc sur leur interaction. Ces objets communicants sont des actionneurs agissant sur l'environnement ou des capteurs fournissant de nombreuses informations. On dispose alors qu'une richesse informationnelle que l'on appelle le contexte, et qui d'après la définition de Dey [6] "est n'importe quelle information pouvant être utilisée pour caractériser la situation d'une entité". Ces informations peuvent être aussi diverses que l'activité de l'utilisateur, les positions des objets, la présence d'obstacle ou bien simplement le temps. Ordinairement, le contexte est utilisé pour améliorer le comportement de l'action en l'adaptant à l'utilisateur. En effet, dans une perspective d'interaction homme-machine, il fournit des informations additionnelles implicites qui auraient été contraignantes voire difficiles à expliciter de la part de l'utilisateur [8]. Par ailleurs, puisque les entités sont situées dans un environnement, elles peuvent être influencées par celui-ci. Par exemple, en robotique ubiquitaire, les éléments tels que les robots et les capteurs sont soumis aux contraintes du monde physique. Ainsi, si l'on cherche à prédire leur comportement, l'information contextuelle apportera une aide précieuse.

L'informatique ubiquitaire se focalisant sur l'interaction d'éléments hétérogènes et dynamiques, elle a fréquemment recours à des architectures orientées services. Ces architectures s'intéressent principalement aux aspects fonctionnels et visent à assembler dynamiquement les éléments, vus comme des services, en entretenant entre eux des relations faiblement couplées. La dimension non-fonctionnelle est gérée par le module de sélection de services qui vise à sélectionner le service le plus approprié pour mener à bien une tâche. Plus précisément, la sélection de services compare des services fonctionnellement équivalents en se concentrant sur leur performance non-fonctionnelle comme par exemple leur précision et leur temps d'exécution.

De nombreux travaux ont été proposés en ce sens, allant du niveau local d'une tâche [2, 5, 12, 16] jusqu'à celui d'une composition entière [3, 10, 17]. Leur évaluation de la performance se concentre sur le service lui-même et est basée sur des métriques appelées paramètres de Qualité de Service (QoS). Par exemple, ces paramètres de QoS peuvent provenir du réseau (comme la latence, la bande passante et la disponibilité) ou de l'application (le temps d'exécution et la précision). Généralement, le processus d'évaluation consiste à acquérir les valeurs des paramètres de QoS et à les agréger en un score. Tout d'abord, ces valeurs sont proviennent généralement de mesures ou d'estimations basées sur les dernières exécutions. Étant donné que la plupart de ces propositions s'intéressent à des services sur le Web, cette approche a sa pertinence car les paramètres de QoS peuvent être mesurés en temps réel et ces services sont très utilisés ; on peut donc s'attendre à ce que le contexte ne change pas considérablement entre deux exécutions consécutives. Cependant, dans un environnement ubiquitaire, deux exécutions du même service peuvent se dérouler à deux instants différents et dans des contextes distincts. Pour cette raison, il ne semble pas adéquat de se reposer sur les dernières exécutions pour estimer le comportement que le service aura. Enfin, l'agrégation des paramètres de QoS doit refléter les attentes que l'on a vis-à-vis du service. Ces attentes proviennent du consommateur du service, qui peut être un autre service ou bien l'utilisateur lui-même. Elles peuvent inclure des effets de seuil qui requièrent un modèle non linéaire. Malheureusement, dans la majorité des propositions, les attentes sont statiques et linéaires ; peu de travaux les ont rendues dépendantes du contexte [16].

Face à ces limites, nous proposons d'étudier les points suivants. Premièrement, les paramètres de QoS ne doivent pas être estimés à partir des dernières exécutions mais prédits à partir du contexte courant. Par ailleurs, au lieu de se focaliser sur la performance du service lui-même, il peut être plus pertinent pour certains services d'évaluer leur impact sur le service qu'ils assistent. À titre d'exemple, considérons la sélection d'un système de localisation devant assister un robot pour franchir une pièce. Comme dans [9], les services de localisation et de déplacement du robot sont exécutés en parallèle et échangent des flux continus de données et sont donc étroitement couplés. De notre point de vue, il semble plus pertinent d'évaluer les services de localisation selon leur impact sur le score de QoS du service de déplacement que de se concentrer sur leur propre score. Dans une telle situation, les modèles de prédiction de QoS et d'agrégation peuvent être remplacés par un modèle prédisant l'impact

du service auxiliaire sur le service principal. Nous proposons d'apprendre ce modèle à partir de l'expérience en utilisant des outils d'apprentissage non linéaires.

La suite du papier est structurée de la façon suivante. Avant de se concentrer sur la sélection, la section 2 décrit une architecture dans laquelle ce module pourrait être inséré. La section 3 présente des travaux traitant de la sélection de services et met en évidence leurs limites quant à leur application dans un environnement ubiquitaire. Les sections suivantes traitent du cas spécifique de la sélection basée sur l'impact. La section 4 introduit le modèle proposé puis la section 5 le valide dans un scénario de localisation. Enfin, la dernière section conclut et énonce quelques perspectives.

2. ARCHITECTURE DE RÉOLUTION DE REQUÊTES

Au sein d'un environnement ubiquitaire doté d'une architecture orientée service, les actions sont menées par l'intermédiaire de requêtes émises par un utilisateur ou une application. La résolution d'une telle requête se base sur les opérations suivantes. Tout d'abord, une découverte de services est nécessaire pour lister les services disponibles et trouver ceux appropriés à partir de leur description. Ensuite, si aucun service n'a été trouvé pour mener entièrement à bien la requête, une composition de services doit être construite. Une étape importante dans cette construction est la sélection d'un service ou d'un ensemble de services parmi des alternatives équivalentes en se basant sur des informations non-fonctionnelles voir contextuelles. Enfin, la supervision de l'exécution doit détecter les changements de comportement des services et réagir si nécessaire. Habituellement, la supervision ne s'intéresse qu'à la détection d'échec et y réagit en recherchant une alternative.

Selon la relation entre la sélection et la composition, deux niveaux de sélection peuvent être distingués. Si la composition est composée de tâches (appelées services abstraits), la sélection consiste à sélectionner un service concret au niveau de chaque tâche ; on parle alors de sélection locale. Au contraire, la sélection peut se situer au niveau d'une composition de services concrets ; cette sélection est alors dite globale. Par exemple, dans le cas des services de localisation, les services concrets sont ceux délivrés par différents systèmes (comme la technologie Cricket [14, 11] ou un système exploitant le signal RSSI [4]) et sont regroupés en un service abstrait : le service de localisation.

Ce papier se concentrera principalement sur la sélection locale et présupera une architecture basée sur de la planification abstraite [16, 19]. Cette architecture est représentée par la figure 1. Son exécution se déroule de la façon suivante : tout d'abord, une tâche requise est envoyée au module de planification abstraite (étape 1). Le planificateur génère une composition de services abstraits à partir de ceux retenus par le module de découverte [13] (étape 2) et l'envoi au module chargé de son exécution (étape 3). Le module prend le premier service abstrait et se charge de son exécution. Pour cela, il doit obtenir un service concret ; il requête donc le module de sélection locale de service (étape 4). Ce module de sélection transmet le service abstrait à l'élément de découverte (étape 5) qui lui renvoie un ensemble de services concrets (étape 6). Il sélectionne ensuite le service concret le plus ap-

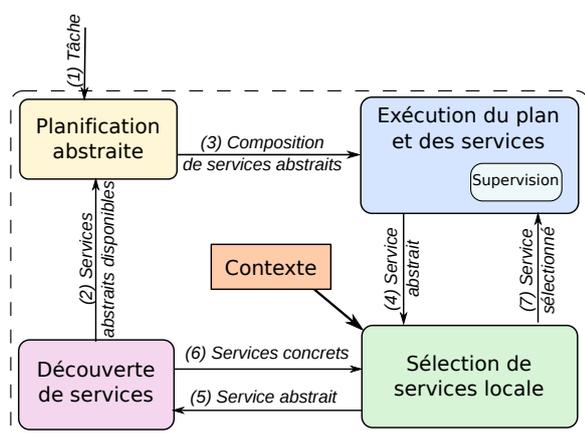


Figure 1: Architecture de résolution de requête proposée

propriété et le renvoi pour exécution (étape 7). Ensuite, les services restants sont exécutés en retournant à l'étape 3.

3. TRAVAUX EXISTANTS SUR LA SÉLECTION DE SERVICES

Dans cette section sont présentés les différents travaux selon les deux étapes du processus de sélection de services avant d'exposer leurs limites face à leur application à un environnement ubiquitaire.

3.1 Estimation de paramètres de QdS

Comme mentionné précédemment, la sélection dépend dans la plupart des propositions des paramètres de QdS. À un niveau local, leurs valeurs sont généralement publiées, mesurées ou estimées à partir des dernières exécutions. Dans un environnement dynamique, les valeurs publiées ne sont pertinentes que pour des valeurs conventionnelles restant constantes, comme le coût de facturation d'un service. Les mesures peuvent être obtenues à partir d'éléments environnants qui affectent le comportement du service, comme le réseau. Les paramètres de QdS sur le service lui-même ne peuvent être qu'estimés ou prédits, car les mesures ne sont faites qu'à l'exécution, c'est-à-dire après la sélection. Tandis que la plupart ne s'intéressent qu'à la dernière exécution, Tari *et al.* [16] ont proposé des règles d'agrégation de façon à avoir une mémoire plus longue. Les paramètres de QdS globaux sont dérivés des paramètres locaux grâce à des règles d'agrégation dépendant de la nature du paramètre. Par exemple, la bande passante n'est pas agrégée de la même façon que la durée d'exécution. Alternativement, Gao et Wu [7] ont proposé de prédire quelques paramètres de QdS tels que la durée d'exécution à partir du temps et de paramètres de QdS réseau. Cette prédiction a été réalisée avec un perceptron multi-couches sous apprentissage supervisé non-adaptatif.

3.2 Agrégation des paramètres de QdS

La grande majorité des propositions existantes reposent uniquement sur des paramètres de QdS. Les propositions peuvent être distinguées selon la présomption qu'ils font sur le modèle d'agrégation : certains utilisent un modèle analytique

déjà connu, d'autres ne le connaissent pas et essaient donc de l'apprendre.

Modèle analytique. Lorsque plusieurs paramètres de QdS interviennent, la sélection de services peut être vue comme un problème d'optimisation multi-objectif. Ce problème est résolu par l'agrégation des attributs, qui dans la majorité des propositions est réalisée par une somme pondérée. La décision repose sur les poids, qui généralement tendent à représenter les préférences du consommateur. Ces poids sont souvent directement fournis par l'utilisateur ou parfois indirectement via un processus d'analyse hiérarchique [5]. Malheureusement, la plupart du temps ceux-ci sont fixes. Cependant, certains ont suggéré d'exposer le modèle d'agrégation au contexte en utilisant des poids dynamiques [16]. Un module externe est alors responsable de changer les poids juste avant la sélection. Ces poids peuvent être dynamiques si la sélection est locale et se déroule à l'exécution.

À un niveau local, ce problème d'optimisation est trivial puisqu'il se limite à comparer un ensemble de services équivalents. Il devient un problème combinatoire complexe au niveau global où un nombre exponentiel de compositions concrètes doivent être évaluées. Aussi, de nombreux travaux traitent ce problème avec les techniques telles que des algorithmes génétiques [17] et des heuristiques [10, 3].

Modèle inconnu. Lorsque le modèle d'agrégation n'est pas entièrement connu, il doit être appris à partir d'expériences. De façon similaire au cas précédent, il peut s'agir d'un modèle linéaire dans lequel les poids doivent être appris. Toutefois, ce modèle peut aussi être présumé non-linéaire par rapport aux paramètres de QdS. Dans un tel cas, l'expression analytique est généralement inconnue : on peut alors utiliser des outils tels que d'un perceptron multi-couches, une machine à vecteurs supports ou des arbres de régression pour apprendre ce modèle.

Al-Masri *et al.* [2] et Mohanty *et al.* [12] ont utilisé ces outils non-linéaires pour apprendre leur modèle d'agrégation. Malencontreusement, leurs données ont été générées à partir d'une fonction linéaire statique appelée WsRF (Web service Reputation Function) [1].

3.3 Limites

Comme mentionné dans l'introduction, les estimations des paramètres de QdS d'un service à partir de ces dernières exécutions ne sont pas adéquates dans un environnement ubiquitaire car le contexte peut avoir changé de façon conséquente, rendant la performance du service différente. En effet, le contexte influence le comportement du service et les attentes que l'on a dessus. Il détermine les valeurs des paramètres de QdS et impacte leur agrégation en un score de QdS, de façon similaire aux niveaux locaux et globaux (voir la figure 2). Dans les travaux précédemment mentionnés, le contexte impliqué dans le processus de sélection est limité aux paramètres de QdS mesurés [7]. Cependant, ces mesures peuvent ne pas être disponibles ni suffisantes pour prédire les autres paramètres de QdS. En conséquence, le contexte doit être étendu à d'autres informations. De plus, l'impact du contexte sur l'agrégation des paramètres de QdS

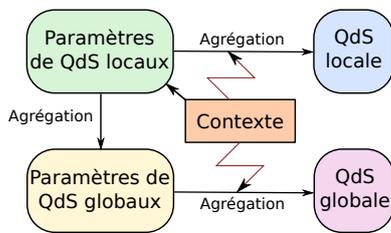


Figure 2: Impact du contexte sur la QoS

est généralement ignoré. Tandis que l'agrégation a été modélisée linéairement, une non-linéarité peut se justifier de deux façons. Premièrement, en dessous d'un certain seuil, un paramètre de QoS peut avoir un effet très négatif sur la performance du service. Enfin, l'intégration du contexte dans le modèle d'agrégation introduit de la non-linéarité. Ce dernier point peut être évité en externalisant l'effet du contexte [16].

La suite de cet article va s'intéresser à la mise en œuvre de ces remarques au travers d'une solution à destination de services spécifiques.

4. SÉLECTION DE SERVICES BASÉE SUR L'IMPACT

Dans la partie restante de cet article, nous nous intéressons à la sélection de services auxiliaires, qui ont pour rôle d'assister un service principal durant son exécution. Il peut aussi bien s'agir de services envoyant des flux de données que d'actionneurs. Par exemple, dans le scénario de validation proposé, les services auxiliaires fournissent des informations de localisation aux robots durant leur déplacement. Leur sélection doit dépendre des attentes du service assisté. Par exemple, à un moment donné, un service assisté préférera de la précision tandis que pour un autre le temps de réponse sera essentiel. Néanmoins, établir ce modèle d'agrégation analytiquement peut s'avérer difficile lorsque les services sont étroitement liés, comme c'est le cas entre des services de localisation et de déplacement. Il faut alors apprendre ce modèle à partir d'expériences.

Au lieu d'apprendre les modèles de prédiction et d'agrégation des paramètres de QoS séparément, nous nous proposons de les regrouper en un modèle unique. Ce modèle se focalise sur l'impact de ces services auxiliaires sur les services qu'ils assistent. Chaque instance de ce modèle est spécifique à un type de service auxiliaire mais pourra être utilisé pour plusieurs services assistés. Pour déterminer ce modèle, un apprentissage supervisé non-adaptatif est appliqué.

4.1 Formes de sélection

Pour ce modèle de sélection, trois formes de sorties peuvent être considérées :

1. Régression : chaque service auxiliaire reçoit un score ; le plus important est sélectionné. Ici, ce score représente le score de QoS du service assisté.
2. Classement : tous les services candidats sont triés.
3. Classification : seul le meilleur service est distingué.

Le classement et la classification comparent les services candidats entre eux et nécessitent donc d'être réalisés au niveau

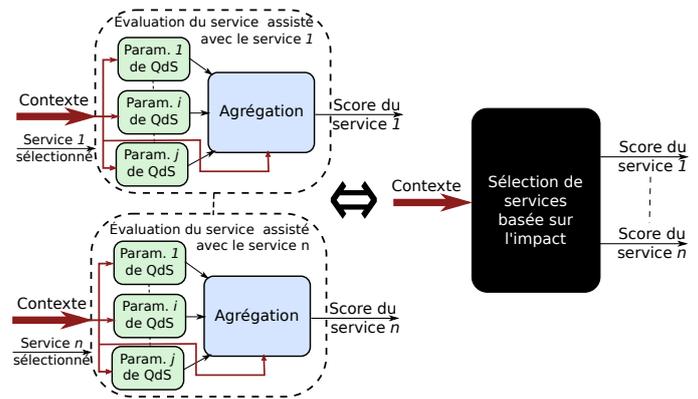


Figure 3: Correspondances entre les entrées et les sorties durant la génération d'exemples de régression

global. Cependant, le score de régression peut être calculé au niveau local d'un service individuel et ensuite extrapolé au niveau global d'un ensemble de services candidats. La régression fournit une information plus riche que respectivement le classement et la classification ; ils peuvent être dérivés à partir d'elle.

De façon à comparer ces trois formes, notre modèle de sélection basé sur l'impact adopte une architecture globale. Comme montré dans la figure 3, il reçoit en entrée le contexte incluant les détails sur le service assisté et des informations spécifiques au domaine des services auxiliaires. Ses sorties auront l'une des trois formes.

4.2 Génération d'exemples

Avant d'appliquer l'apprentissage supervisé hors-ligne, les exemples doivent être générés. Comme illustré dans la figure 3, nous proposons d'évaluer l'impact d'un service auxiliaire avec le score de QoS du service assisté. Les exemples sont générés sous la forme de régression puis sont dérivés dans les deux autres formes. Le processus de génération d'exemples se déroule de la façon suivante. Les entrées sont extraites du contexte initial. La génération d'un exemple se déroule en n exécutions du même service assisté dans le même contexte initial, à chaque fois avec un service auxiliaire avec l'un de n services auxiliaires. Durant chaque exécution, les valeurs des paramètres de QoS sont mesurés, puis ils sont agrégés en un score grâce au modèle d'agrégation du service assisté. La valeur du score est attribuée à la sortie associée au service auxiliaire exécuté. Toutes les exécutions d'un exemple doivent se dérouler dans le même contexte initial.

Une fois que plusieurs ensembles d'exemples ont été générés, l'apprentissage supervisé peut être appliqué.

4.3 Apprentissage supervisé non-adaptatif

L'apprentissage est non-adaptatif, c'est-à-dire qu'il est réalisé en hors-ligne à partir de l'ensemble des exemples à apprendre. Il se base sur un outil d'apprentissage non-linéaire, le perceptron multi-couches et utilise l'algorithme de rétro-propagation pour mettre à jour ses poids à partir de l'erreur. L'apprentissage s'effectue sous forme de validation croisée à n plis de façon à économiser les exemples car ils sont contrai-

gnants à générer. La phase d'entraînement utilise comme fonction de perte l'erreur quadratique tandis que la validation et le test évaluent l'apprenant selon son taux de bonne sélection. Cette fonction d'évaluation est spécifique à l'usage du modèle, c'est-à-dire à la sélection et est indépendante de chacune des trois formes.

5. SCÉNARIO DE SÉLECTION DE SERVICE DE LOCALISATION

Afin de valider notre modèle, nous avons mis en place un scénario de robotique ubiquitaire impliquant plusieurs robots communicants. La sélection de service basée sur l'impact est effectuée sur des services de localisation qui assistent les services de déplacements des robots.

5.1 Description du scénario

Le scénario se déroule dans un bâtiment composé de deux pièces équipées de systèmes de localisation. Le système Cricket [14, 11] n'est disponible que dans la pièce principale tandis que le système de localisation basé sur le signal RSSI [4] est déployé partout. Les deux robots sont placés initialement dans la pièce principale. Un d'entre eux, le robot assistant, est équipé d'un télémètre laser et d'une carte du bâtiment lui permettant de se localiser n'importe où. Il est ainsi capable de fournir un service de localisation en mesurant la position relative d'un robot grâce à sa caméra et leur boussole [9].

La modélisation des systèmes de localisation vise à donner un ordre de grandeur réaliste de leur comportement afin de démontrer la faisabilité de notre approche. Ces quatre systèmes sont modélisés de la façon suivante. Le système basé sur le signal RSSI a une précision constante mais très faible précision (60 cm) dans les deux pièces. Le système Cricket requiert que l'objet à localiser soit visible par au moins trois de ses capteurs ultrasons. Sa précision est de 5 cm si plus de trois capteurs voient l'objet ou de 10 cm dans le cas contraire. Le télémètre laser a la meilleure précision (1 mm) mais n'est disponible que pour le robot assistant. De plus, il a un faible coût énergétique et ne sollicite pas le réseau pour communiquer avec son robot ; son coût est donc minimal. Quant au système de localisation impliquant le robot assistant, l'erreur de précision de sa caméra vient s'ajouter à celle de sa propre position, qui est commise par un autre service de localisation. Les précisions sont modélisées comme les écart-types d'une fonction d'erreur gaussienne. Tandis que le temps de réponse des autres systèmes est constant, le système utilisant le robot assistant doit d'abord s'approcher de sa cible, ce qui peut évidemment nécessiter un temps important qui varie selon la distance à parcourir. De plus, il suit l'autre robot et se synchronise avec lui, rendant leur déplacement saccadé.

Afin d'introduire des attentes différentes sur un service assisté, nous les avons associées à différentes tâches. En ce sens, les services de déplacement des robots peuvent réaliser deux tâches : **déplacement simple** et **approche précise**. Les modèles d'agrégation associés à ces deux tâches sont différents : outre l'utilisation de poids différents, la seconde tâche ajoute une contrainte sur la précision de localisation.

L'exécution recherchée est la suivante : le robot principal,

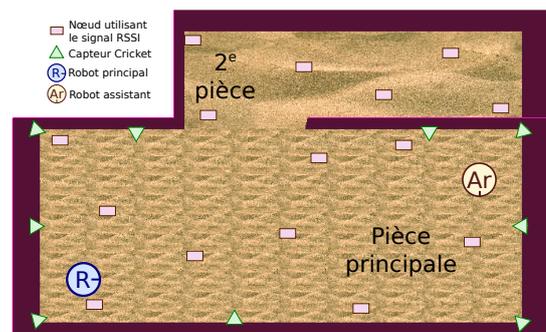


Figure 4: Contexte initial du scénario

situé au fond à gauche de la pièce principale doit réaliser une séquence de deux tâches (voir la figure 4). Il atteindra le point A en exécutant la tâche **déplacement simple** puis approchera précisément du point B en exécutant la tâche **approche précise**. Tout d'abord, le service Cricket est sélectionné pour la première tâche. Arrivé dans la seconde pièce au point M, ce service ne se trouve alors plus capable de localiser le robot. Une seconde sélection a alors lieu : le service basé sur le signal RSSI est sélectionné car il fournit une précision suffisante pour atteindre le point A et a bien un meilleur temps de réponse que le robot assistant situé dans la partie droite de la pièce principale. Le robot atteint le point A et la première tâche est achevée. Une nouvelle sélection intervient pour la seconde tâche : le service RSSI est rejeté car il ne fournit pas une précision moyenne inférieure à 15 cm ; le service restant fourni par le robot assistant est donc sélectionné. Le robot assistant approche le robot principal et l'aide à atteindre le point B.

Cette simulation a été implémentée sur la plateforme multi-agents ARéVi [15].

5.2 Génération des exemples

La génération d'exemples se déroule en plusieurs étapes. Tout d'abord, le contexte initial est généré. Ensuite, les quatre services auxiliaires sont exécutés et évalués un par un.

5.2.1 Génération du contexte initial

Dans ce scénario, le contexte initial est caractérisé par :

- les positions initiales des robots, générées aléatoirement.
- le choix d'un type de tâche et d'un robot. Dans ce scénario, seules les tâches **simple déplacement** et **approche précise** sont considérées.

De ce contexte, on en extrait les entrées suivantes :

- l'identifiant du type de tâche
- l'identifiant du robot
- les coordonnées x et y de la position du robot
- la distance du robot sollicité par rapport au robot assistant, cette distance pouvant être nulle.

5.2.2 Exécution et évaluation d'un service auxiliaire

Nous considérons l'exécution complète de la tâche menée par le service assisté. Une exécution pourra avoir utilisé plusieurs services de localisation car le service auxiliaire sélectionné peut échouer en cours d'exécution. Néanmoins, l'éva-

luation de la performance de la tâche est attribué au service auxiliaire testé, c'est-à-dire à celui qui a été le premier sélectionné. Si ce premier service auxiliaire n'est pas disponible ou s'il ne peut pas localiser le robot dans son contexte initial, son exécution est considérée comme échouée. Toutefois, l'échec d'un service assistant n'entraîne pas systématiquement l'échec de l'exécution. En effet, dans l'exécution illustrative, le service Cricket échoue lorsque le robot sort de sa portée sans que cela porte préjudice à l'exécution. Cependant, l'exécution s'arrête lorsque le service assisté échoue ou lorsqu'aucun service de localisation ne peut être sélectionné.

Les trois paramètres de QdS sont impliqués dans l'évaluation sont la précision ($QdS_{precision}$), la durée (QdS_{duree}) et le coût (QdS_{cout}). Leurs valeurs sont obtenues ou dérivées à partir de deux mesures : la précision moyenne p et la durée d . Le coût c est dérivé des coûts individuels des services exécutés et de leur durée d'utilisation.

Avant d'être agrégées, ces mesures doivent être normalisées afin d'être considérées comme des paramètres de QdS. La précision utilise une fonction gaussienne centrée avec comme écart-type $\sigma = 20$ cm. La valeur de l'écart-type résulte d'un compromis entre les précisions extrêmes fournies par l'ensemble des systèmes de localisation.

$$QdS_{precision} = \exp\left(-\frac{p^2}{2 * \sigma^2}\right) \quad (1)$$

La durée et le coût varient selon la distance parcourue durant l'exécution. Au lieu d'ajouter la distance $dist$ en entrée des exemples, ces mesures sont comparées à leur valeurs minimales prédites d_{min} et c_{min} . Ces valeurs prédites dépendent de plusieurs paramètres : la vitesse maximale du robot v_{max} , le coût individuel du robot c_{robot} et le coût minimal d'un service de localisation c_{eco} .

$$d_{min} = \frac{dist}{v_{max}} \quad (2)$$

$$c_{min} = d_{min} * (c_{robot} + c_{eco}) \quad (3)$$

Ces paramètres de QdS sont normalisés de la façon suivante :

$$\forall x \in [0; 1], f(x) = \frac{e^x - 1}{e - 1} \quad (4)$$

$$QdS_{duree} = f\left(\frac{d_{min}}{d}\right) \quad (5)$$

$$QdS_{cout} = f\left(\frac{c_{min}}{c}\right) \quad (6)$$

Enfin, nous utilisons une simple somme pondérée pour agréger ces paramètres avec des poids dépendant de la tâche (voir la table 1). Ce score est calculé à condition que l'exécution ait réussi et que la contrainte ait été validée; dans le cas contraire, ce score est nul. Pour la tâche **approche précise**, la contrainte est :

$$precision < 15 \text{ cm} \quad (7)$$

Par ailleurs, un bonus $b = 0.1$ est ajouté au score afin de distinguer une exécution réussie avec une très mauvaise per-

Type de tâche	$w_{precision}$	w_{duree}	w_{cout}	Contrainte ?
Déplacement simple	0.2	0.7	0.1	non
Approche précise	0.4	0.5	0.1	oui

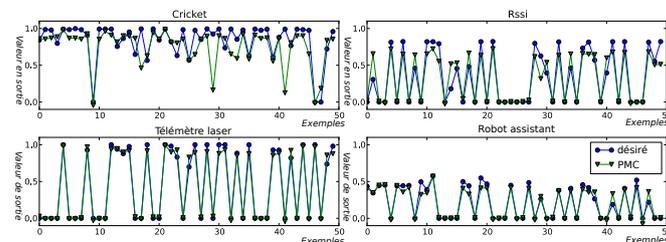
Table 1: Poids des paramètres de QdS

formance d'une exécution échouée. L'équation de ce score est :

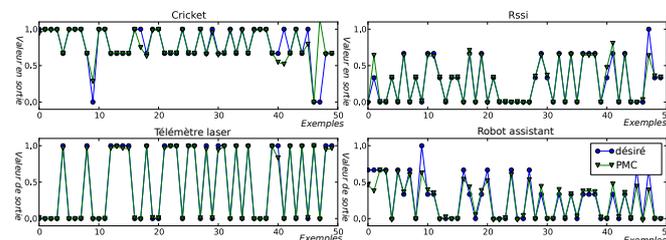
$$score = b + (1 - b) * \sum_1^3 w_i * QdS_i \quad (8)$$

5.3 Résultats

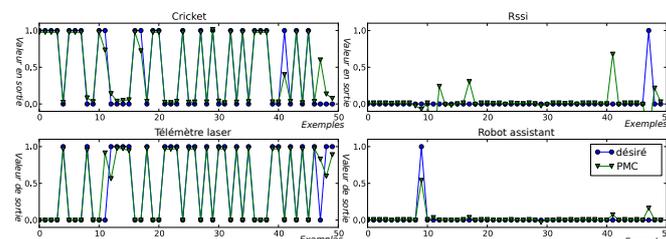
Les exemples sont générés sous forme de régression et sont ensuite dérivés vers les formes de classement et de classification. Le perceptron multi-couches (PMC) a des fonctions d'activation linéaires pour les entrées et les sorties et une tangente hyperbolique pour l'unique couche cachée. Les couches d'entrée, cachée et de sortie ont respectivement 5, 10 et 4 neurones.



(a) Régression



(b) Classement



(c) Classification

Figure 5: Sorties en généralisation

Une validation croisée à 5 plis a été réalisée pour l'apprentissage du PMC, à partir d'ensembles de 200 exemples. Comme montré dans la figure 5, le PMC semble être capable de suivre

Forme	Cricket	Rssi	Laser	Assistant	Taux de bonne sélection
Idéal	468	38	471	13	
Régression	477	32	467	14	92 %
Classement	455	22	494	19	94 %
Classification	454	22	495	19	94 %

Table 2: Résultats en généralisation

la sortie désirée en généralisation, sauf pour quelques évènements rares comme la sélection des services basés sur le Rssi et le robot assistant. La table 2 montre les résultats lorsque notre modèle est appliqué à la sélection. Toutes les formes ont un taux de bonne sélection similaire (entre 92 et 94%) et sont capables de sélectionner chaque service. Ces résultats ne nous permettent pas de distinguer ces formes.

6. CONCLUSION ET PERSPECTIVES

Un des aspects les plus intéressants du paradigme ubiquitaire est la prise en considération du contexte au sein des systèmes informatiques. L'information contextuelle peut être fournie directement aux services afin qu'ils s'y adaptent mais également aux outils de prédiction du comportement de service soumis à leur environnement. Ce second point est celui qui intéresse la sélection de services et afin de bien l'illustrer, nous avons proposé une application en robotique ubiquitaire, où les services subissent les contraintes du monde physique. Malheureusement, les travaux existants, trop centrés sur le réseau, ont peu considéré cette relation. Ils essaient d'estimer les paramètres de QoS du service à partir de ses dernières exécutions. Ces dernières exécutions pouvant s'être déroulées dans un contexte très différent de l'actuel, nous proposons d'utiliser plutôt l'état courant de l'environnement pour prédire ces paramètres de QoS. Par ailleurs, nous souhaitons également rendre l'agrégation dépendante du contexte.

La solution proposée dans cet article s'intéresse au cas spécifique des services auxiliaires qui assistent un service et elle propose de les sélectionner à partir de leur impact. Ce modèle de sélection a été appris en utilisant un perceptron multi-couches sous apprentissage supervisé non-adaptatif. Enfin, ces propositions ont été validées par un scénario de sélection de services de localisation.

Cet apprentissage supervisé non-adaptatif a pour limite de devoir être réalisé hors-ligne et de nécessiter une contraignante phase préalable de génération d'exemples. Aussi, nous allons poursuivre nos investigations vers un nouvel enjeu qui est de réaliser cet apprentissage en-ligne, directement au sein d'un système en cours de fonctionnement. Enfin, ces techniques pourront être appliquées à notre plateforme de robotique ubiquitaire.

7. REFERENCES

- [1] E. Al-Masri and Q. H. Mahmoud. QoS-based discovery and ranking of web services. In *IEEE International Conference on Computer Communications and Networks*, pages 529–534, Honolulu, HI, USA, 2007.
- [2] E. Al-Masri and Q. H. Mahmoud. Discovering the best web service : A neural network-based solution. In *IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, 2009.
- [3] M. Alrifai, T. Risse, P. Dolog, and W. Nejdl. A scalable approach for QoS-Based web service selection. In *ICSOB*, pages 190–199, 2009.
- [4] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta. A novel approach to indoor rssi localization by automatic calibration of the wireless propagation model. In *IEEE Vehicular Technology Conference*, page 1–5, 2009.
- [5] S. Dehousse, S. Faulkner, C. Herrensens, I. J. Jureta, and M. Saerens. Learning optimal web service selections in dynamic environments when many Quality-of-Service criteria matter. In *Machine Learning*, pages 207–229. InTech, Feb. 2009.
- [6] A. K. Dey. Understanding and using context. *Personal And Ubiquitous Computing*, 2001.
- [7] Z. Gao and G. Wu. Combining QoS-based service selection with performance prediction. In *IEEE International Conference on E-Business Engineering*, pages 611–614, Los Alamitos, CA, USA, 2005.
- [8] H. Lieberman and T. Selker. Out of context : Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3) :617–632, 2000.
- [9] R. Lundh, L. Karlsson, and A. Saffiotti. Autonomous functional configuration of a network robot system. *Robotics and Autonomous Systems*, 56 :819–830, 2008.
- [10] N. B. Mabrouk, S. Beauche, E. Kuznetsova, N. Georgantas, and V. Issarny. QoS-Aware service composition in dynamic service oriented environments. *Middleware 2009*, pages 123–142, 2009.
- [11] Memsic.
- [12] R. Mohanty, V. Ravi, and M. R. Patra. Web-services classification using intelligent techniques. *Expert Systems With Applications*, 2010.
- [13] S. B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers. EASY : efficient semAntic service discoverY in pervasive computing environments with QoS and context support. *Journal of Systems and Software*, May 2008.
- [14] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *IEEE International Conference on Computer Communications*, volume 1, 2005.
- [15] P. Reignier, F. Harrouet, S. Morvan, J. Tisseau, and T. Duval. ARéVi : a virtual reality multiagent platform. In *Virtual Worlds*, page 229–240, 1998.
- [16] K. Tari, Y. Amirat, A. Chibani, A. Yachir, and A. Mellouk. Context-aware dynamic service composition in ubiquitous environment. In *IEEE International Conference on Communications*, South Africa, May 2010.
- [17] Y. Vanrompay, P. Rigole, and Y. Berbers. Genetic algorithm-based optimization of service composition and deployment. In *International workshop on Services integration in pervasive environments*, pages 13–18, Sorrento, Italy, 2008.
- [18] M. Weiser. Ubiquitous computing. *Computer*, 1993.
- [19] A. Yachir, K. Tari, Y. Amirat, A. Chibani, and N. Badache. QoS based framework for ubiquitous robotic services composition. In *IEEE Intelligent Robots and Systems*, pages 2019–2026, 2009.

MARTS : Conception d'un Système Mobile de Réalité Augmentée Dédié au Tourisme*

Nehla Ghouaiel
ESTIA Recherche - IRIT
Technopole Izarbel
64600 Bidart
n.ghouaiel@estia.fr

Jean-Marc Cieutat
ESTIA Recherche - IRIT
Technopole Izarbel
64600 Bidart
j.cieutat@estia.fr

Jean-Pierre Jessel
IRIT
118 route de Narbonne
31062 Toulouse
jessel@irit.fr

ABSTRACT

Le but de cet article est de montrer comment les interfaces mobiles et la Réalité Augmentée (RA) peuvent s'avérer être une assistance précieuse à la navigation piétonne. Le support de l'étude est tout naturellement le Smartphone qui est devenu en quelques années l'appareil qui nous est le plus familier, qui nous accompagne quotidiennement partout où nous nous déplaçons. Après avoir examiné les différentes situations auxquelles peut être confronté un touriste qui visite un environnement urbain pour la première fois, nous proposons le concept d'un nouveau système multimodal, le système MARTS, d'aide à la navigation piétonne que nous prévoyons réaliser.

Categories and Subject Descriptors

H. [Information Systems]; H.5 [INFORMATION INTERFACES AND PRESENTATION]; H.5.1 [Multimedia Information Systems]: [Artificial, augmented, and virtual realities, Audio input/output, Video]

General Terms

Design

Keywords

Réalité Augmentée, interfaces mobiles, informatique ubiquitaire

1. INTRODUCTION

La navigation est le processus qui consiste à guider le mouvement du navigateur entre deux points spatiaux tout en lui permettant de connaître sa localisation à tout instant [15]. La navigation, dans son sens commun, signifie qu'une personne se déplace d'un endroit vers un autre, dont elle connaît la nature, même si elle n'arrive à le géo référencer

*(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM_PROC_ARTICLE-SP.CLS. Supported by ACM.

sans un outil d'aide à la navigation¹. Or, par exemple, ne pas avoir une destination précise dès le début, est en général le cas de tout touriste qui visite une ville historique pour la première fois [9]. Dans ce cas, la première question qui se pose à un touriste confronté à un environnement urbain non familier est : Que faut-il visiter dans cette ville ? Nous estimons que la réponse la plus appropriée à cette question dans une telle situation doit contenir au moins l'ensemble des POIs² (endroits les plus intéressants à visiter dans un environnement urbain dans notre cas) classés comme étant prioritaires. Les POIs prioritaires sont ceux situés dans le plus proche voisinage du visiteur et aussi ceux considérés comme symboles de la ville (c'est le cas de la tour Eiffel à Paris). Pour distinguer la navigation terrestre commune point à point (dans laquelle la destination est déterminée) de la navigation dans laquelle la destination est non connue d'avance, nous avons choisi d'appeler cette dernière la navigation multipoint. D'un autre côté, il existe dans un environnement urbain des POIs qu'un visiteur peut très facilement rater s'ils sont moins connus ou s'ils sont plus difficiles à localiser. Ce genre de POIs peut être qualifié de caché. D. McGookin [9] montre comment un visiteur peut passer à proximité de statues sans les voir. Une fois que le touriste choisit une destination précise, la deuxième question qui se pose alors est la suivante : Comment puis-je me rendre au POI A ?

Nous essayons à travers ce travail de recenser les différents enjeux liés à la mobilité et à la navigation d'un touriste dans un environnement urbain inconnu. Le but de cet article est de démontrer comment la navigation terrestre multipoint peut bénéficier des interfaces mobiles et de la réalité augmentée. Nous présentons dans la dernière partie de l'article le système que nous proposons pour répondre à la problématique avancée.

2. LES TRAVAUX ASSOCIÉS

Beaucoup de recherches ont été menées afin de guider une personne pour qu'elle atteigne une localisation déterminée [2] en environnement "outdoor". Cependant, quelques limites peuvent être identifiées. La première contrainte est d'ordre technique : elle est liée au problème de géo référencement dans les environnements urbains [18]. Une deuxième contrainte est plutôt d'ordre ergonomique. En effet, une personne préfère se promener dans un lieu urbain sans avoir à se concentrer sur une carte géographique ou toute autre interface gra-

1. Par exemple avec un GPS
2. Point Of Interest (POI)

phique. Si le visiteur concentre son attention sur le logiciel d'aide à la navigation, il accorde alors moins d'attention à l'environnement [7, 9]. Une telle situation fait appel à ces scénarios :

- Comment puis-je explorer un lieu urbain tout en conservant mes mains libres et sans gêner mon regard ?

- Comment puis-je explorer un lieu urbain que je ne connais pas sans la moindre surcharge mentale ?

2.1 Les Systèmes Ubiquitaires de RA

La réalité augmentée telle que vue par Azuma [5] est une technologie avancée dont le but est d'augmenter la perception du monde réel. Les systèmes de RA superposent des objets virtuels à la scène réelle pour améliorer l'expérience visuelle de l'utilisateur. L'une des limites des systèmes de RA est leur incapacité à réagir au contexte.

Pour améliorer l'interaction de l'utilisateur mobile avec son environnement, des informations relatives à la variation du contexte (les localisations des objets, les tâches courantes, le nombre d'utilisateur, le profil de l'utilisateur, l'activité passée, l'état affectif, etc.) [15, 16, 17] sont requises. Dans certaines recherches la connaissance du domaine et le comportement de l'utilisateur sont pris en compte pour construire le contexte [6]. Les performances d'un système mobile peuvent être mesurées par sa capacité d'assimiler la situation actuelle d'un utilisateur afin de lui fournir les services, les ressources, ou les informations les plus adaptés au contexte. Dans ce sens, plusieurs recherches ont été menées pour concevoir des méthodes d'interaction par la combinaison des technologies de la réalité augmentée et de l'informatique sensible au contexte [4, 19, 15]. Les données contextuelles peuvent être classifiées en deux types. Le premier type inclut les informations contextuelles de bas niveau qui sont fournies par des capteurs et le second regroupe les informations de haut niveau. Plusieurs capteurs sont utilisés pour déterminer les informations liées au contexte dites de bas niveau. Le système de positionnement global plus connu sous le sigle GPS, ce système, conçu par le département américain de la défense, permet de déterminer une position globale [1]. Il se base sur un positionnement par satellites. Equipé d'un récepteur GPS, la position, le temps et la vitesse de l'opérateur sont estimés à partir des différents signaux provenant des satellites. L'orientation de champ de vue de l'utilisateur est estimé par les capteurs inertiels. Trois types de capteurs inertiels sont généralement utilisés : les accéléromètres, les magnétomètres et les gyroscopes. Les capteurs inertiels mécaniques traditionnels demeurent encombrant. L'avènement des nanotechnologies a permis de miniaturiser ces dispositifs appelés MEMS (Micro-Electro-Mechanical Systems). Certains téléphones mobiles sont déjà équipés de capteurs de luminosité ambiante tel que l'iPhone 4 d'Apple. Le capteur est utilisé en conjonction avec un logiciel pour déterminer les conditions optimales d'éclairage de l'écran en fonction des conditions ambiantes permettant ainsi l'adaptation automatique de la luminosité pour un meilleur confort visuel. Le WSN³ [19, 12] a été largement utilisé dans les recherches menées sur l'informatique ubiquitaire pour contrôler un environnement et reconnaître les objets qui y sont présents. Il

3. Wireless Sensor Network

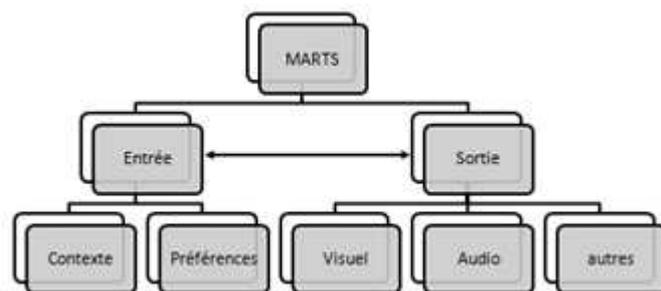


Figure 1: La structure du système MARTS.

est composé d'un grand nombre de capteurs intégrés dans des noeuds sans fils et qui communiquent à travers des liens radio multi-hop [19]. Les informations contextuelles dites de haut niveau sont en général relatives au profil de l'utilisateur, son milieu culturel, son milieu social, etc.

2.2 Les Interfaces Mobiles

Grâce aux interfaces mobiles, l'utilisateur peut se déplacer dans le monde physique tout en restant accompagné par les services informatiques (être servi, communiquer indépendamment de sa localisation). La prolifération des appareils mobiles et l'amélioration de leur capacité de calcul et de communication, a donné naissance à une série de nouvelles technologies : les systèmes distribués, les nanotechnologies, l'informatique mobile, etc. Toutes ces technologies sont regroupées sous le nom de l'informatique ubiquitaire [11]. Les téléphones mobiles intelligents (Smartphone) constituent la classe d'appareils mobiles qui sont les plus petits et les plus répandus. Contrairement à du matériel dédié comme un "ordinateur Backpack avec un HMD⁴", ou encore Tablet PC et PDA, ils se sont intégrés dans la vie de chacun. La plupart des téléphones mobiles ont des résolutions d'écran similaire à celle des PDA, mais leurs écrans sont plus petits. En général, ils possèdent également une plus petite puissance de calcul que les autres classes d'interfaces mobiles [10].

3. LE SYSTÈME MARTS

Afin de répondre aux problèmes de navigation dans un environnement urbain inconnu soulevés dans les parties précédentes de l'article, nous sommes en train de concevoir un système d'aide à la navigation terrestre multipoint dans un environnement urbain. Le système que nous appelons MARTS comme acronyme de Mobile Augmented Reality Touring System est susceptible d'attirer l'attention d'un visiteur, non familiarisé avec un environnement urbain, de l'existence de POIs. MARTS est destiné à fonctionner sur des plateformes mobiles. Les POIs gérés par le système sont ceux situés au voisinage de l'utilisateur qui sont susceptibles de l'intéresser en fonction de son profil. Pour étudier les profils des touristes visitant le pays-basque et répondre au mieux à leurs attentes, le cluster tourisme Goazen destiné à fédérer l'ensemble des corporations liées au secteur du tourisme a été

4. Head Mounted Display

créée à l'initiative de la Chambre de Commerce et d'Industrie de Bayonne Pays-Basque (<http://www.bayonne.cci.fr/index.php/clus-le-cluster-goazen-cest-quoi->). Une banque de données regroupant tous les POIs a été ainsi constituée; en fonction des types d'utilisateur, il peut être alors suggéré tel ou tel POI. MARTS fait appel aux technologies de la réalité augmentée et de l'informatique sensible au contexte pour suggérer à l'utilisateur des POIs et le guider tout au long de ses visites. En effet, le système devra s'adapter aux conditions ambiantes (vent, bruit, illumination, temps) ainsi qu'aux préférences de l'utilisateur pour basculer d'une modalité à une autre (sonore, visuelle, tactile, kinesthésique, etc.) [8]. Le système est conçu pour passer du mode standard au mode personnalisé afin de présenter à un touriste indécis les POIs qui correspondent le mieux à son profil.

L'utilisateur du MARTS n'est pas appelé à suivre un chemin prédéfini de telle sorte que la découverte d'un nouvel environnement soit la plus naturelle possible. Pour réaliser le prototype logiciel du système MARTS sous la plateforme mobile Android, nous envisageons utiliser le Kit de développement Android (SDK) fourni par Google. Le SDK Android comporte des outils, parmi lesquels un émulateur, permettant de couvrir quasiment toutes les étapes du cycle de développement d'une application mobile.

3.1 Mode Audio

Dans le cas où le système fonctionne en mode sonore, l'utilisateur reçoit des appels à visite de chaque POI. Un appel à visite relatif à un POI dure 15 secondes et il est divisé en deux parties : une partie parlée qui indique le nom du POI, puis une partie sonore spatialisée (non parlée) qui indique la direction du POI. La technique du son spatialisé consiste à faire varier les formes d'onde [9] qui atteignent l'oreille droite et l'oreille gauche pour traduire à l'utilisateur la position de la source sonore. Pour éviter le problème du chevauchement des sons relatifs aux POIs, un seul appel à visite est autorisé à la fois. L'utilisateur reçoit les sons relatifs à chaque point d'intérêt sous forme d'une boucle sonore. La boucle sonore commence par traiter le POI le plus proche jusqu'à arriver dans une zone d'activation de longueur et de largeur variables dépendant du nombre maximal de POIs autorisés. La zone d'activation ou zone de couverture du système est mise à jour en fonction de la position de l'utilisateur et de son déplacement. Donc, les POIs qui n'appartiennent pas à la zone d'activation se trouvent être désactivés; le système audio ne les gère pas pour éviter toute surcharge mentale de l'utilisateur.

3.2 Mode Visuel

L'un des objectifs de la réalité augmentée est d'améliorer la perception ou la visibilité du monde physique. L'écran du Smartphone se comporte comme une lucarne sur le monde réel dont on peut augmenter le flux vidéo. Nous nous appuyons sur les données géo-référencées des objets pour informer l'utilisateur sur sa localisation comme le montre la figure 2 par exemple où l'on voit l'information de localisation des différents POIs situés à proximité. Il est en effet impossible d'équiper de marqueurs artificiels tous les éléments du monde réel. Nous devons donc mettre en oeuvre un calcul de la pose de la caméra par un système sans marqueur ("markerless"). Le système calcule la position de l'utilisateur en utilisant les données GPS et celles de la boussole. Pour

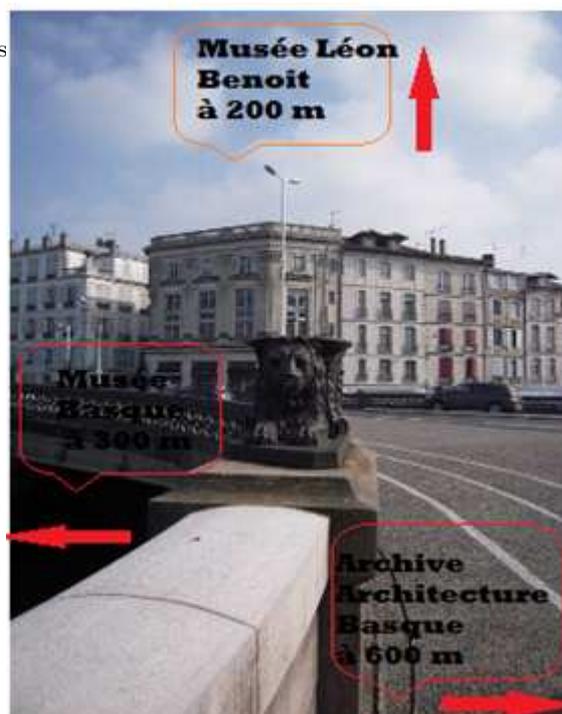


Figure 2: Interface visuelle prévue pour MARTS.

une reconnaissance spécifique (façade d'un bâtiment,), les techniques basées capteurs peuvent être complétées par les techniques basées image. Les caractéristiques SIFT [20] sont alors extraites des images de la vidéo traduisant le monde réel comme en [18] et comparées avec une base de caractéristiques. Pour l'affichage, l'adjonction des annotations dans la scène réelle, visibles depuis l'écran du Smartphone comme l'illustre la figure 2 est effectuée par la technique de "Vision See Through (VST)" [13] largement utilisée dans les applications de réalité augmentée. A l'image de la fonctionnalité de réalité documentée relative à la réalité augmentée, notre flux vidéo peut être enrichi d'informations identifiant ce qui est visible depuis la caméra.

3.3 Le Prototype Matériel

La figure 3 montre les composants matériels du notre système mobile d'aide à la navigation piétonne dans un environnement urbain. Le système comporte un casque audio appelé Audio Bone⁵ qui permet à l'utilisateur d'écouter les séquences audio à travers l'os de l'oreille et non à travers le tympan. Ce casque ne bloque pas l'oreille externe et donc il permet à l'utilisateur d'écouter en plus des sons synthétisés les sons ambiants. Audio Bone peut être considéré comme une alternative au casque ARA⁶ utilisé dans la réalité audio augmentée [3, 14]. Le système comporte aussi un Smartphone de type Samsung Galaxy II qui contient entre autres un récepteur AGPS, un accéléromètre 3 axes et un magnétomètre 3 axes.

5. Les détails de spécifications d'Audio Bone sont présents sur ce site : www.Audiobone.eu

6. Augmented Reality Audio

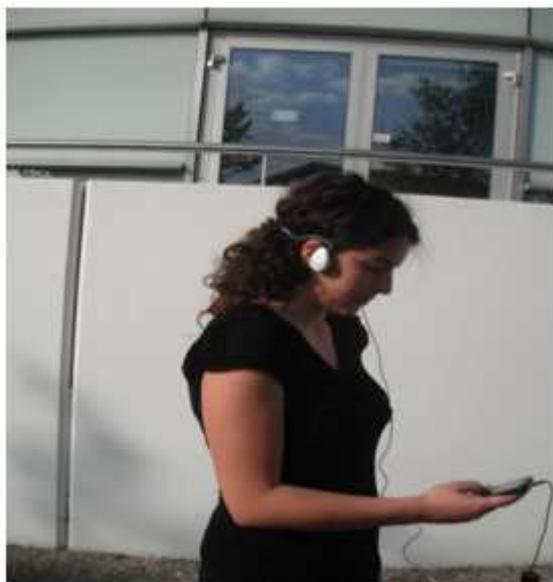


Figure 3: Usage prévu du MARTS

4. CONCLUSIONS

Dans un contexte de fortes évolutions des Interfaces Homme-Machine (IHM) des systèmes mobiles, nous explorons les possibilités offertes par la constitution d'un guide à la navigation urbaine multipoint suite à l'intégration dans une unique application, de plusieurs modalités de réalité augmentée. Nous essayons de répondre à la demande de l'utilisateur qui varie en fonction des conditions de navigation. Les fonctionnalités apportées par la réalité augmentée doivent donc différer suivant les personnes et les conditions ambiantes, d'où l'obligation d'apporter des informations contextuelles. Nous présentons dans des travaux futurs le prototype logiciel du MARTS ainsi que les résultats des expérimentations conduites sur ce système. Etant donné le caractère exploratoire de cette plateforme de navigation terrestre, nous souhaitons étendre notre plateforme afin qu'elle puisse se généraliser en plateforme de RA disposant de plusieurs modalités pouvant être utilisée autant à pied qu'en voiture ou à vélo.

5. REFERENCES

- [1] F.Evennou. *Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor*. Phd thesis, Université Joseph-Fourier, France, Mars 2007.
- [2] G.Reitmayr and D.Schmalstieg. Collaborative Augmented Reality for Outdoor Navigation and Information Browsing. In *Proceedings of the Symposium on Location Based Services and TeleCartography*, pages 31–41. Wiley, 2004.
- [3] A.Harma, J.Jakka, M.Tikander, M.Karjalainen, T.L.Tapio, and H.Nironen. Techniques and Applications of Wearable Augmented Reality Audio. In *Audio Engineering Society Convention 114*. Audio Engineering Society, March 2003.
- [4] A.Henrysson and M.Ollila. UMAR : Ubiquitous Mobile Augmented Reality. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 41–45. ACM, October 2004.
- [5] R. Azuma. A survey of augmented reality.
- [6] N. Bonnefoy. *Système Informationnel Mixte, Ubiquitaire et Intuitif - Support aux opérateurs nomades de maintenance aéronautique*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, septembre 2009.
- [7] B.Thomas, W. Piekarski, D.Hepworth, B.Gunther, and V.Demczuk. A Wearable Computer System with Augmented Reality to Support Terrestrial Navigation. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, pages 168–172. IEEE Computer Society, 1998.
- [8] P. R. D. R.Panagiotis and S. G.Antonis. Standards for augmented reality : a user experience. In *International AR Standards Meeting*, February 2011.
- [9] D.Mcgookin, S.Brewster, and P.Priego. Audio Bubbles : Employing Non-speech Audio to Support Tourist Wayfinding. In *Haptic and Audio Interaction Design - HAID*, pages 41–50. Springer-Verlag, July 2009.
- [10] D.Wagner. *Handheld augmented reality*. Phd thesis, Graz university of technology, Austria, October 2007.
- [11] F.Díez-Díaz, M.González-Rodríguez, and A.Vidau. An accesible and collaborative tourist guide based on augmented reality and mobile devices. In *Proceedings of the 4th international conference on Universal access in human-computer interaction : ambient interaction*, pages 353–362. Springer-Verlag, 2007.
- [12] J.Newman, G.Schall, and D.Schmalstieg. Modelling and handling seams in wide-area sensor networks. In *Wearable Computers, 2006 10th IEEE International Symposium*, pages 51–54. IEEE Computer Society, January 2006.
- [13] M.Kanbara, T.Okuma, H.Takemura, and N.Yokoya. A Stereoscopic Video See-through Augmented Reality System Based on Real-time Vision-based Registration. In *Proceedings of the IEEE Virtual Reality 2000 Conference*, pages 255–262. IEEE Computer Society, March 2000.
- [14] M.Tikander, M.Karjalainen, and V.Riikonen. An augmented reality audio headset. In *Proceedings of the th International Conference on Digital Audio Effects*, September 2008.
- [15] S.Choonsung, L.Wonwoo, S.Youngjung, Y.Hyoseok, L.Youngho, and W.Woontack. CAMAR 2.0 : Future Direction of Context-Aware Mobile Augmented Reality. In *Proceedings of the 2009 International Symposium on Ubiquitous Virtual Reality*, pages 21–24. IEEE Computer Society, 2009.
- [16] S.Lchiro. Mobile applications in ubiquitous computing environments. *IEICE Transaction in Communication*, 88(3) :1026–1033, March 2005.
- [17] R. J. Weiss and J.P.Craiger. Ubiquitous computing. *The Industrial-Organizational Psychologist*, 39(4) :44–52, April 2002.
- [18] W.Zhang and J.Kosecka. Image based localization in urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing*,

Visualization, and Transmission (3DPVT'06), pages 33–40. IEEE Computer Society, 2006.

- [19] X.Li, D.Chen, and S.Xiahou. Ubiquitous Augmented Reality System. *2009 Second International Symposium on Knowledge Acquisition and Modeling*, 3(2) :91–94, December 2009.
- [20] Y.Ke and R.Sukthankar. PCA-SIFT : A More Distinctive Representation for Local Image Descriptors. In *Computer Vision and Pattern Recognition, Proceedings of the 2004 IEEE Computer Society Conference*, pages 506–513. IEEE Computer Society, July 2004.

RESEAU DE CAPTEURS POUR LA MAITRISE DE LA DEMANDE ENERGETIQUE

André Lachaud
Laboratoire CEDRIC du CNAM

292, rue Saint-Martin
75003 PARIS
+33 6 11 42 41 79
alac@free.fr

David Menga
EDF R&D (ICAME)

1, Av. du Général de Gaulle
92141 CLAMART CEDEX
+33 1 47 65 56 68
david.menga@edf.fr

Vincent Illionnet
EDF R&D (ICAME)

1, Av. du Général de Gaulle
92141 CLAMART CEDEX
+33 1 47 65 41 80
vincent.illionnet@edf.fr

RESUME

Afin de participer à la mise en œuvre la maîtrise de la demande énergétique, dans le cadre de la maison communicante, nous avons élaboré le projet Pi. Dans cet article nous étudierons la réalisation d'un réseau de capteurs-actionneurs architecturé en relation maître à esclaves. Nous proposons une interface homme machine destinée à l'utilisateur, ainsi qu'un lien vers une passerelle ouverte sur internet. Les éléments du réseau communiquent localement par courant porteur en ligne, effectuent la relève synchronisée des consommations électriques par usage, et autorisent la coupure ou la mise en fonction des usages connectés. L'accent est mis sur l'acquisition synchronisée au pas de la seconde, l'identification des usages, la possibilité de contrôler depuis l'extérieur ou localement chacun des actionneurs.

ABSTRACT

To implement the energy efficiency in the context of the smart home, we developed the Pi project. In this paper we study the realization of this network of sensors-actuators organized in master-slave relationship. We made a human machine interaction interface for the user, and a link to open a gateway to the internet. The network elements communicate locally through power line current, perform synchronized acquisition of power consumption by usage, and authorize the cutting or setting function for the connected appliances. Emphasis is placed on the synchronized acquisition with a pitch of one second, the identification of usages, the ability to control locally or remotely each of the actuators.

Categories and Subject Descriptors

H.m [Information Systems]: Miscellaneous; H.4.m [Information Systems Applications]: Miscellaneous

General Terms

Saving energy, Consumption by usage, Network sensors, Load shedding.

MOTS CLES

Économie d'énergie, Consommation par usage, Réseau de capteurs, Délestage.

1. INTRODUCTION

Nous suivons les recommandations de l'ADEME [1] « La loi de programme fixant les orientations de la politique énergétique [5] impose de réduire de 2 % par an d'ici à 2015 et de 2,5 % d'ici à 2030 l'intensité énergétique finale, c'est-à-dire le rapport entre la consommation d'énergie et la croissance économique. »

De nombreux accords internationaux traitant de l'économie d'énergie, de même que la prise en compte des changements

climatiques, imposent aux fournisseurs d'électricité la mise en œuvre de dispositifs visant à minimiser et à optimiser tant la production que la consommation d'énergie. C'est dans ce cadre que s'inscrit le projet Pi, acronyme de Prise Intelligente, que nous décrivons dans ce papier. Nous y proposons un ensemble d'outils matériels et logiciels facilitant la gestion de la demande énergétique dans le cadre de la maison communicante.

2. LE PROJET Pi

2.1 Architecture générale

Pour mener à bien une gestion intelligente et efficace de l'énergie en milieu résidentiel, nous avons construit un réseau de prises esclaves dotées de capteurs et d'actionneurs dialoguant par courant porteur en ligne (CPL) avec une prise maître. Cette dernière, figurée au centre de la Figure 1, assure une liaison série, à travers un tunnel USB, vers un PC sous Linux. Celui-ci, doté d'une application graphique, permet à l'utilisateur d'interagir avec les appareils de son domicile. Grâce à la connectivité vers internet de cet ordinateur, un lien IP permet de remonter les informations collectées vers une « usine MDE » (Maîtrise de la Demande Énergétique), située chez le fournisseur d'énergie, qui synthétise les données détaillées provenant de ses clients, et peut en fonction des besoins agir individuellement sur n'importe quel actionneur de chacun de ses abonnés.

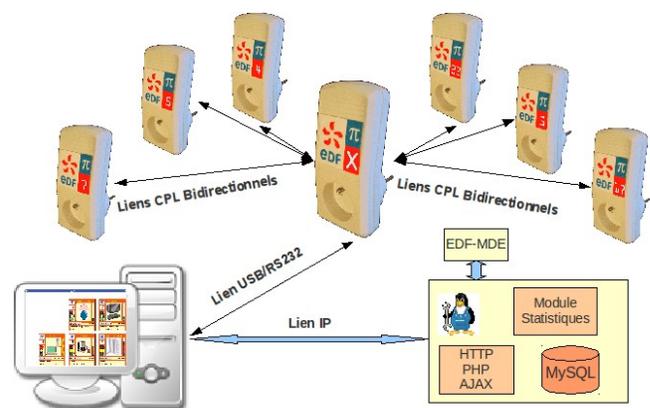


Figure 1: Vue d'ensemble du projet Pi

2.2 Mesure de la consommation

L'acquisition des consommations par usage s'effectue au moyen de prises gigognes, qui s'intercalent entre une prise murale existante et l'appareil qui y est connecté. Cette mesure d'une précision de 2% est acquise en temps réel et fournit une donnée puissance exprimée en watt. Le cumul de ces puissances au cours

du temps évalue l'énergie consommée sur cette prise pendant une période déterminée. Son unité est le watt-heure.

2.3 Organe de coupure

Si nous souhaitons activer ou désactiver un usage, il faut un organe permettant d'interrompre ou d'autoriser le passage du courant vers l'objet qui est connecté à la prise gigogne. Pour cela nous utilisons un relais électromécanique capable de couper 20 ampères sur une charge de type résistif.

2.4 Identification des usages

Nous souhaitons identifier les appareils qui peuvent être branchés librement sur nos prises, afin de garantir une isolation galvaniquement, nous avons recours à un dispositif sans contact.

Chaque prise esclave est pourvue d'un lecteur de tag RFID (Radio Frequency Identification). Une inductance de 45mm de diamètre est formée de spires de cuivre enroulées autour du réceptacle femelle dans chaque prise gigogne esclave.

C'est la partie dorée que l'on voit en bas à droite, dans la figure 2. Elle fait partie d'un circuit oscillant accordé, permettant grâce à un circuit spécialisé, la lecture de tags RFID.

Toutes les fiches mâles des appareils de la maison sont équipées d'un petit tag circulaire qui vient se loger entre les deux contacts acheminant le 230V. Sur la figure 3 on peut voir au centre de la prise, isolée et protégée par deux minces pellicules de plastique, la bobine plate constituée de nombreuses spires de fil très fin, associée à un condensateur et à un circuit spécialisé. A coté, le même ensemble est encapsulé dans un jeton de plastique noir plus dur et plus résistant que nous collons au centre des fiches mâles.

Une fois par seconde, nous faisons osciller le circuit primaire, présenté Figure 2, sur une fréquence de 120kHz. Disposé autour de la prise femelle, il transmet l'énergie pour alimenter le tag RFID fixé sur la prise mâle de l'appareil connecté. Le circuit secondaire va alors transférer le code contenu dans la mémoire du tag, vers la prise.

Chaque appareil de la maison est ainsi identifiable, Nous avons réservé 120 classes d'usages possibles, profils liés principalement à l'ampleur de l'énergie réclamée. En priorité les éléments chauffants : chauffage électriques tels convecteurs ou rayonnants à infra-rouge, les fer à repasser, plaques de cuisson, bouilloires, sèche-linge, puis les aspirateurs, lave vaisselle, lave linge, enfin les congélateurs, réfrigérateurs, TV/Vidéo, ordinateur, box internet et l'éclairage.

2.5 Interface homme machine

Nous avons développé une application graphique écrite en C++ sous Linux avec la bibliothèque Qt. Cette interface autorise l'utilisateur final à interagir avec notre système de gestion de l'énergie, en accédant aux données essentielles qui reflètent l'état du réseau de capteurs-actionneurs et offre la fonctionnalité de commander la coupure ou la mise en activité de chaque prise, au moyen d'un simple clic.



Figure 2: Prise avec son lecteur de tags RFID

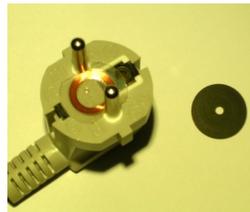


Figure 3: Tags RFID

2.6 Passerelle internet

L'ouverture sur le réseau internet permet au fournisseur d'énergie de récupérer les données de consommations ainsi que l'état du système. Le site distant peut en retour renvoyer des consignes pour optimiser la gestion globale de l'énergie.

2.7 Outil de décision

La décision de couper ou d'enclencher un usage peut dépendre de nombreux critères: paramètres physiques, périodes horaires, confort de l'utilisateur, réglementation, besoins du fournisseur d'énergie en rapport avec « l'usine MDE ». Ces fonctionnalités peuvent s'implémenter sous forme d'algorithmes qui dépendent de critères figés, ou au contraire, rester ouvert à des modifications asynchrones pouvant provenir de diverses sources. Nous nous contentons à ce stade de l'expérimentation, de garder le système ouvert et laissons le libre choix à l'utilisateur d'agir manuellement sur ses prises au moyen de l'interface que nous détaillerons dans le paragraphe 6.

2.8 Modèle maître-esclave

2.8.1 Le maître

Le maître a pour principale fonction de coordonner les prises esclaves. Il intègre également un lien série vers un PC, servant de passerelle vers internet. C'est la prise maître qui va déclencher de façon synchrone la mesure des consommation sur chacune des prises esclaves. Il est similaire à la « Smart Gateway » proposée par Markus Weiss et Dominique Guinard dans[6]. Nous verrons plus en détail dans le paragraphe 5, les mécanismes de synchronisation et la manière de récupérer les informations.

2.8.2 L'esclave

Chaque prise communique exclusivement avec le « maître ». Elle dispose d'un instrument de mesure de l'énergie absorbée à travers elle, d'une sonde de température et d'un relais pour commander l'appareil connecté. En outre afin d'assurer le suivi de consommation par usage, nous avons intégré un lecteur de tag RFID décrit au paragraphe 2.4, qui permet de connaître ainsi la nature de l'appareil connecté à la prise.



Figure 4 : Prise esclave.

3. MESURE DE LA CONSOMMATION

La mesure de consommation est le résultat du produit $U \cdot I \cdot T$, comme nous sommes en courant alternatif, la relation est plus précisément :

$$\text{Energie appareil} = \int u_a i_a dt$$

Si l'acquisition de la tension ne pose aucun problème, un simple diviseur résistif ramène la tension alternative à un niveau adéquat pour la numérisation du signal, il en va différemment pour la mesure du courant. Afin de récupérer une image du courant, on utilise soit un transformateur de courant, soit un shunt. Nous avons opté pour ce dernier, moins volumineux, moins coûteux et ne nécessitant pas de correction due au déphasage. Il en résulte dans tous les cas une tension de très faible amplitude, délicate à manipuler. Nous avons retenu un circuit spécialisé destiné à équiper les compteurs électriques, produit référencé SA9903B, extrait de la gamme de la société SAMES basée en Afrique du Sud. La réalisation de la fonction complète au moyen de composants discrets aurait été trop complexe et délicate.

Le circuit SA9903B est certifié pour l'équipement des compteurs électriques et répond aux normes IEC 61036 et IEC 61268. Le synoptique de la figure 5 montre qu'il est composé de 2 amplificateurs différentiels, un pour la tension, l'autre pour le courant. Il sont suivis de convertisseurs analogique numérique et

de filtres passe-haut. Le produit courant-tension pilote un oscillateur commandé en tension (VCO), qui produit 320 000 impulsions pour une heure d'énergie à pleine échelle que nous avons fixée à $230V * 16A$ soit 3680 Wh. On en déduit que nous avons environ $320000/3680 = 87$ impulsions par Wh soit une résolution de 11,5 mWh par impulsion. Cette échelle peut être légèrement ajustée pour obtenir une résolution de 12,5Wh soit 80 impulsions par Wh.

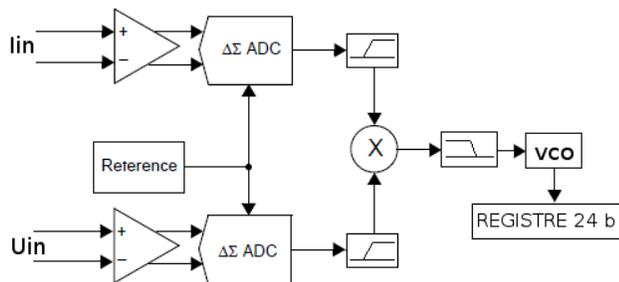


Figure 5 : Synoptique du circuit d'acquisition des consommations

Nous avons choisi ce circuit, car il est le seul à offrir un registre de 24 bits accumulant ces impulsions. Nous venons l'interroger très exactement chaque seconde, grâce à une interface SPI, puis nous totalisons ces mesures successives pour reconstituer la consommation totale sur cette prise. Nous pouvons en déduire la puissance moyenne sur la seconde écoulée : $N * 3600 * 0,0125$ exprimée en W, où N représente le nombre d'impulsions, 0,0125 la résolution en Wh, 3600 le nombre de secondes dans l'heure.

Le circuit de mesure nécessite une alimentation symétrique par rapport au neutre du secteur (+2.5V et -2.5V), alors que nos circuits logiques sont alimentés uniquement en +3.3V, ce qui implique le recours à une translation de niveau pour accéder aux signaux du bus SPI.

4. COURANT PORTEUR EN LIGNE

Comme vecteur de communication entre la prise maître et les prises esclaves nous utilisons le courant porteur en ligne. Toutes les prises sont reliées sur une seule phase du réseau électrique fourni par Électricité de France (EDF). Les technologies CPL bas débit s'appuient sur la modulation d'amplitude, de fréquence ou de phase, et peuvent aller jusqu'à gérer des protocoles de haut niveau tel ceux proposés par Echelon, et le protocole Lonworks[3]. Nous avons fait le choix du compromis: simplicité, robustesse et facilité de mise en œuvre.

Ceci nous conduit à retenir un circuit intégré de référence ST7540 de STMicroelectronics, que nous paramétrons pour travailler dans la bande B définie par la norme CENELEC EN 50065-2, bande libre n'imposant pas de protocole. Les données sont envoyées et reçues au moyen d'une interface sérielle en mode caractère asynchrone. Les flux de données modulent une porteuse de 110Khz à 2400 bauds. Un caractère étant constitué d'un bit de début, de 8 bits de données, et d'un bit de fin, nous disposons de 240 octets par secondes, soit une durée de 4,2ms/octet.

Afin d'arbitrer l'usage de l'unique ligne de transmission et éviter les collisions, nous avons eu recours, dans un premier temps, au mode d'accès CSMA/CD que nous avons rapidement abandonné, car trop consommateur de temps. De même nous avons abandonné le protocole MODBUS créé par Modicon[7], au profit de notre propre méthode. En effet le mode question/réponse de MODBUS impose des trames de 6 octets minimum soit au total

12 octets pour un échange. De plus il n'offre pas nativement de procédure de synchronisation temps réel. Le protocole que nous avons élaboré, s'avère plus efficace car il exploite de façon optimale le peu de bande passante qui nous est dévolue, et permet une parfaite synchronisation, de l'ensemble des prises.

5. LA COLLECTE SYNCHRONISÉE

5.1 La synchronisation

Comme nous sommes sur le réseaux électrique EDF à 50 hertz, nous allons nous caler sur le passage par zéro de la tension. Nous nous inspirons de la méthode proposée par Anthony Rowe[2]. A cet effet, nous utilisons un simple écrêteur à diode, qui produit une transition logique vers le niveau haut lorsque la tension secteur devient positive, de même la transition vers le niveau bas s'effectue lorsque la tension sinusoïdale redevient négative. Nous récupérons ainsi une horloge distribuée sur tout notre réseau de prises, comme illustré sur la figure 6.

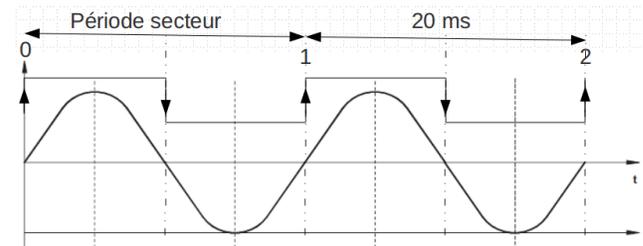


Figure 6 : La synchronisation sur la tension secteur

5.2 La collecte des données

Considérons la figure 7. La prise maître après un « zéro » initial, au point 0, active sa porteuse en A et émet en B une trame de synchronisation de trois caractères spécifiques sur la ligne CPL, indiquant à toutes les prises esclaves qui sont obligatoirement en écoute, qu'elle doivent se synchroniser sur le premier passage par zéro qui suit, en réinitialisant leur compteur de périodes secteur à la valeur 1. Chacune lance alors la lecture de la consommation d'énergie écoulée depuis la dernière seconde, et la température fournie par un capteur oneWire durant la plage C, puis la lecture du tag RFID en D. Tout ceci est effectué simultanément par l'ensemble des prises esclaves au cours des 7 périodes qui suivent le début de la synchronisation.

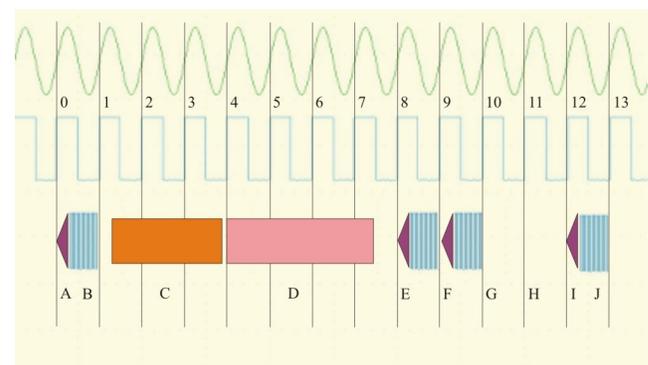


Figure 7 : La synchronisation et la collecte des données

A tour de rôle, chaque prise va envoyer son numéro, sa consommation, l'identifiant de son tag RFID, la température, et son état, ainsi qu'un numéro de séquence destiné à détecter d'éventuelles pertes de trames. A chaque prise est attribué un numéro, N de 1 à 32, chacune transmet dans sa case temporelle définie par la formule $8 + (N-1)$, à partir des points E, F et I pour

les prises 1, 2 et 5 respectivement. Dans notre exemple les prises 3 et 4 sont absentes. Ces réponses mobilisent 1 période par prise soit un total de 32 et nous laisse donc 10 périodes libres chaque seconde.

Durant une période secteur de 20 ms, nous avons besoin de 5ms pour que la fréquence et l'amplitude de la porteuse se stabilisent, il reste 15 ms. Nous disposons donc au maximum de 3 caractères pour renvoyer la totalité des informations de chaque prise dans le laps de temps imparti. Comme nous sommes très contraints il convient d'utiliser au mieux l'espace disponible.

Le codage du numéro identifiant la prise tient sur 5 bits, le numéro de séquence de 0 à 7 nécessite 3 bits, l'ensemble est regroupé dans le premier octet. L'énergie en 1 seconde est au maximum de l'ordre du watt ce qui s'exprime par un nombre d'impulsions inférieur à 127. Le deuxième octet sera complété par un bit qui indiquera l'état de la prise : activée ou désactivée. Notons qu'en cas de désactivation, la consommation sera nulle, ce qui peut aussi se produire si l'appareil connecté est arrêté, nous avons donc bien besoin d'un bit d'état, la valeur nulle n'est pas un critère suffisant.

Le troisième octet contiendra alternativement la température, et l'identifiant du tag qui renvoie à une classe d'appareils électriques, comme définie plus loin, un codage sur 7 bits est suffisant le huitième indiquant le paramètre attendu. Nous n'utilisons que trois octets, car il faut laisser à la porteuse CPL le temps de se stabiliser avant d'envoyer les données.

5.3 Validation des échanges

Pour conclure sur la synchronisation et le contrôle des données, nous vérifions l'envoi séquentiel des prises par leur numéro, et son adéquation avec la période temporelle dans laquelle elles sont attendues. Pour détecter la perte éventuelle d'une trame, on se réfère au numéro de séquence émis par chaque esclave et vérifié par le maître. En cas de perte d'une trame, on résout en faisant une moyenne depuis la dernière bonne trame reçue.

5.4 Transmission des commandes

Pendant les 10 périodes secteur restantes nous pouvons acheminer des ordres de coupure ou d'activation, soit en adressant individuellement les prises au moyen de leurs identifiants, soit par groupes grâce à des numéros précédemment réservés, ou encore une seule commande destinée à l'ensemble du réseau. Notons que ces changements d'état sont peu fréquents.

6. APPLICATIF VISUEL DE GESTION

Nous avons développé une application graphique qui autorise l'utilisateur final à interagir avec notre système en accédant aux données de chaque capteur-actionneur et offre la possibilité de commander la coupure ou la mise en activité de chaque prise, représentée par un avatar. Il existe également des commandes pour agir sur des groupes de prises, ou adresser la totalité du réseau.



Figure 8 : Visualisation des prises

Dans le cadre de ce prototype, nous avons relié le maître à un PC sous Linux, au moyen d'une liaison série à travers un tunnel USB. Nous rapatrions les données issues des prises et les portons à la connaissance de l'utilisateur qui a également la possibilité d'acheminer des ordres vers les prises. De plus, le PC grâce à son

lien IP vers l'internet, peut renvoyer vers le fournisseur d'énergie les informations en temps réel sur l'état du réseau de son client.

Nous ne décrivons pas ici les échanges effectués avec l'usine MDE qui sont en cours d'élaboration.

6.1 L'avatar

Nous avons attaché à chaque prise un avatar, objet virtuel qui reflète l'état complet de la prise en temps réel, considérons un décalage maximum de l'ordre de la seconde.

L'accent a été mis sur l'identification visuelle immédiate et intuitive de trois paramètres : l'ordre de grandeur de l'énergie consommée, le type d'usage connecté à la prise et son état actif ou inactif. Une autre facilité est disponible à l'utilisateur qui peut actionner la charge à sa convenance, au moyen d'un simple clic.

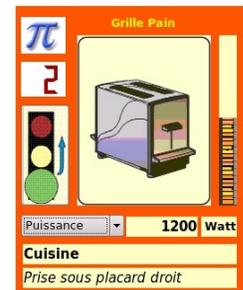


Figure 9 : L'avatar

6.1.1 Le « plug and play » électrique

Si l'on branche une nouvelle prise sur le réseau électrique, celle-ci apparaîtra automatiquement dans l'interface, de même en cas de retrait, elle cessera de s'afficher. Le délai de réaction est celui du rafraîchissement des données, soit une seconde.

6.1.2 L'identification de la prise

Un numéro logique, identifiant unique de chaque prise dans le réseau, apparaît dans une case sous le logo représentant la lettre grecque PI. Mémorisé dans une mémoire non volatile, on peut le modifier par logiciel en cas de nécessité.



Figure 10 : Le numéro

6.1.3 La jauge à énergie

La jauge fournit une information très redondante, en effet la taille de la jauge verticale est proportionnelle à la consommation, de même que la vitesse de l'animation, de plus la couleur de la jauge qui va du vert au rouge en passant par la couleur jaune puis orange, donne une indication supplémentaire.

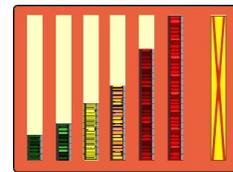


Figure 11 : Les jauges

6.1.4 Le type d'usage

Le tag RFID équipant les fiches des cordons d'alimentation de chaque appareil, délivre cette information, et permet de suivre par exemple, l'aspirateur sur chaque prise esclave où il est branché. Il est important de connaître quel appareil consomme quelle quantité d'énergie, afin de pouvoir couper en toute efficacité et sécurité les usages adéquats. En effet il est déconseillé de couper le congélateur, ce qui engendrerait la perte de son contenu, ou le chauffage dans la chambre d'un enfant malade. Pour exprimer intuitivement le type d'usage, nous avons relié les codes des tags RFID à une galerie d'images, ainsi qu'à un message présenté dans la partie supérieure de l'avatar indiquant le type d'appareil: grille pain, bouilloire, lampe, chauffage, télévision ...

6.1.5 L'état de la prise

Afin de présenter de façon intuitive l'état de la prise, nous avons recours à l'image et la symbolique des feux tricolores utilisés pour la circulation automobile. Le feu rouge indique un arrêt, renforcé par une jauge à énergie particulière, visible à droite dans la figure 11 : une grande croix rouge sur fond jaune. Le feu vert signifie un état passant associé à la jauge reflétant la consommation actuelle.

De plus l'indication précise de la puissance moyenne, est disponible dans une case, susceptible de présenter également, la consommation ou son coût sur une période déterminée, selon le choix de l'utilisateur.

6.1.6 L'organe de commande

La commande induisant le changement d'état de la prise, utilise également le feu tricolore décrit précédemment, mais comme il sert à la fois de commande et de témoin d'état, nous avons recours à la logique du sémaphore pour garantir un fonctionnement correct.

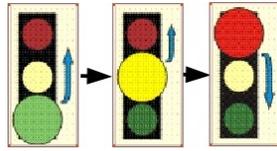


Figure 12 : Commande de délestage

L'action est réalisée par un simple clic, qui dans tout les cas fait passer immédiatement le feu au jaune, signifiant qu'il ne peut plus recevoir de nouvelle commande et qu'il est en attente de l'accusé de réception de commande, jusqu'à l'apparition du nouvel état stable, renvoyé par la prise concernée.

6.1.7 Lecture de la puissance

Une case à usage multiple reçoit au choix la valeur de : la puissance moyenne soit l'énergie absorbée sur une période déterminée soit le coût moyen (ou depuis une certaine date), selon le souhait de l'utilisateur.

6.1.8 Localisation de la prise

Une première case de texte à remplir librement affiche la dénomination de la pièce dans laquelle se trouve la prise, la seconde indiquant plus précisément sa situation à l'intérieur de la pièce, elle ne sont pas utilisées par le système, et ne sont qu'une facilité offerte à l'utilisateur.

7. CONCLUSION ET PERSPECTIVES

Le projet Pi a pour objectif de fournir des outils matériels et logiciels dans le but d'acquérir des données de consommations par usages, synchronisées, avec une granularité de une seconde.

Le projet que nous avons développé a rempli son objectif et prouvé son efficacité dans la relève synchronisée de consommations par usage, en temps réel, au pas de la seconde.

Cet outil a été utilisé conjointement avec les données issues du compteur bleu électronique pour valider d'autres travaux, comme ce projet interne à EDF R&D « Reconnaissance d'usages et estimation de leurs consommations d'énergie à partir de la courbe de charge ».

La collecte d'informations associant usage, consommation et température approvisionne une base de données dont l'étude permettra de dégager des schémas de délestage minimisant la perte de confort. Un autre projet en cours de développement, est destinée à l'assistance aux personnes âgées ou handicapées, à partir de leurs « activités électriques ».

Nous projetons des améliorations du réseau Pi, visant en particulier à réduire la consommation intrinsèque du réseau.

Considérant que dans notre système, au moins une prise émet à chaque instant 24h sur 24h et 7jours sur 7, la consommation intrinsèque du système est de 14 watts. Si l'on considère que généralement 10 prises suffisent (au lieu du maximum de 32 possibles), nous consommons environ 5 Watts.

Bien que chaque prise esclave ne fonctionne que pendant 20 ms par cycle, il faut quand même dimensionner pour chacune

l'alimentation nécessaire pour l'étagage de puissance CPL, ce qui est lourd, volumineux et coûteux.

Une alternative prometteuse consisterait à employer le module WPC[8], produits par la société Watteco et qui est en cours d'homologation par le CENELEC. Ce module utilise également les fils du réseau EDF, mais grâce à une technologies reposant sur des impulsions très courtes, il divise la consommation par un facteur 20, comparé aux CPL classiques.

Une autre piste de réflexion nous fait envisager une migration vers des solutions « radio » autour de la norme 802.15.4, plus précisément le 6LoWPAN[4], mais nous devons adapter la méthode de synchronisation.

Enfin, une nouvelle interface graphique, hébergée sur un serveur web, utilisant Ajax est en cours d'étude, ainsi que des versions destinées aux terminaux iPhone et Android.

8. REMERCIEMENTS

Remerciements à EDF et son département ICAME qui finance et soutien cette étude, dans le cadre du projet STABILO.

9. REFERENCES

- [1] ADEME;Certificats économie d'énergie; <http://www2.ademe.fr/servlet/KBaseShow?sort=1&cid=96&m=3&catid=15024>
- [2] Anthony Rowe, Vikram Gupta, and Ragunathan (Raj) Rajkumar. 2009. Low-power clock synchronization using electromagnetic energy radiating from AC power lines. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*. ACM, New York, NY, USA, 211-224. DOI=10.1145/1644038.1644060 <http://doi.acm.org/10.1145/1644038.1644060>
- [3] Echelon corporation; LonWorks; http://www.echelon.com/products/lonworks_control_networking.htm
- [4] Geoff Mulligan, « *The 6LoWPAN architecture* », dans *Proceedings of the 4th workshop on Embedded networked sensors, 2007*, 78-82 <http://dx.doi.org/http%3A%2F%2Fdoi.acm.org%2F10.1145%2F1278972.1278992>
- [5] Legifrance; Loi n°2005-781 du 13 juillet 2005 ; <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000813253&dateTexte=>
- [6] Markus Weiss and Dominique Guinard. 2010. Increasing energy awareness through web-enabled power outlets. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (MUM '10)*. ACM, New York, NY, USA, , Article 20 , 10 pages. DOI=10.1145/1899475.1899495 <http://doi.acm.org/10.1145/1899475.1899495>
- [7] Modicon MODBUS Protocol Reference Guide, June 1996, MODICON, Inc., Industrial Automation Systems One High Street, North Andover, Massachusetts 01845
- [8] Watteco ;modem WPC ; datasheet: http://www.watteco.com/images/stories/download/watteco_wpc_description_170310.pdf

Démo 1

GEPETO*: towards a platform for evaluating privacy aspects of mobility data

Sébastien Gambbs

Univ. de Rennes 1 - INRIA / IRISA

Campus Universitaire de Beaulieu, 35042 Rennes, France

sgambbs@irisa.fr

Marc-Olivier Killijian

LAAS - CNRS

7 avenue du Colonel Roche Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS F-31077 Toulouse, France

marco.killijian@laas.fr

Miguel Núñez del Prado Cortez

LAAS-CNRS

7 avenue du Colonel Roche Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS F-31077 Toulouse, France

mnunezde@laas.fr

Résumé

In this paper, we describe our ongoing work on GEPETO (for G_EoPrivacy Enhancing TOolkit), a software platform for evaluating the privacy aspects of mobility data. For instance, GEPETO can be used to perform inference attacks on geolocated data such as home identification through clustering algorithms or even to derive more structured knowledge such as learning a compact model of the mobility behaviour of an individual. We briefly review the inference attacks and sanitization methods currently available within GEPETO and we report on the preliminary experiments conducted on geolocated datasets.

Démo 2

Conception et Evaluation d'un Système de Dialogue pour un Dispositif de Suppléance pour Déficients Visuels

Slim Kammoun
Bernard Oriola
Philippe Truillet
Christophe Jouffrais
IRIT, CNRS & Université de Toulouse 3
118 Route de Narbonne, 31062, France
{Kammoun, Oriola, Truillet, Jouffrais}@irit.fr

Résumé

Avec l'essor des nouvelles technologies de l'information et de la communication, la question de l'utilisation des dispositifs mobiles par tous les utilisateurs devient cruciale. Dans cet article nous nous intéressons à l'utilisabilité d'un dispositif de suppléance (NAVIG) dédié aux déficients visuels. Ce système a comme objectif d'améliorer l'autonomie des utilisateurs en restaurant deux fonctions visuelles fondamentales à savoir la localisation de soi dans l'environnement (ce qui permet de fournir une aide aux déplacements) et la localisation d'objets d'intérêt (par exemple dans le but de les saisir). NAVIG1 est doté d'un système de positionnement par satellite et d'un système de vision artificielle embarqué. L'interaction avec le dispositif a été déterminée suivant un processus de conception centrée utilisateur, qui a abouti à une interaction vocale uniquement. L'évaluation du prototype a été réalisée par 8 sujets déficients visuels, et a montré l'efficacité des choix et des stratégies adoptées à savoir la reconnaissance vocale en entrée et la synthèse vocale en sortie.

Démo 3

SysCARE : Système de Communication et d'Assistance Robotisée

Louis-Pierre Bergé#, Cyril Briand* Agnan de Bonneval*, Michel Taïx* Philippe Truillet#
Université de Toulouse (LAAS-CNRS* et IRIT UMR CNRS#)
118, Route de Narbonne 31400 Toulouse, France
<briand, agnan, taix>@laas.fr
<berge, Philippe.Truillet>@irit.fr

Olivier Lefebvre
Magellium
24, Rue Hermès BP 12113 31521 Ramonville Saint-Agne, France
olivier.lefebvre@magellium.fr

Résumé

Nous présentons dans cet article SysCARE, un système de communication et d'assistance robotisée à destination des personnes âgées. SysCARE s'inscrit dans le domaine de la e-santé, et correspond à une étude de faisabilité d'une solution concrète. L'objectif de ce projet est la réalisation d'un démonstrateur composé d'un ensemble de services d'aide aux personnes âgées ou dépendantes à domicile.

Démo 4

Conception d'une interaction avancée pour stimuler les visiteurs d'un musée

Bénédicte Schmitt
2Global Vision Systems
10 Avenue de l'Europe
31520 Ramonville Saint-Agne
benedicte.schmitt@global-vision-systems.com

Louis-Pierre Bergé, Emmanuel Dubois
IRIT - Université Paul Sabatier
31062 Toulouse cedex 9
{prenom}.{nom}@irit.fr

Résumé

Au cours de ces dernières années, un nombre important d'expériences interactives liées au domaine culturel, et notamment à la muséographie, ont vu le jour. L'objectif est de rendre le visiteur acteur de sa visite en lui proposant d'interagir directement avec des concepts. Ceux-ci sont difficiles à comprendre car ils sont complexes ou abstraits. L'utilisation d'une technique d'interaction avancée, combinant artefacts physiques et numériques, a pour objet de stimuler le visiteur sur ce type de phénomène complexe. Nous avons conçu une technique d'interaction avancée qui vulgarise le phénomène d'eutrophisation dans le but de faciliter sa compréhension et ses origines.

GEPETO* : towards a platform for evaluating privacy aspects of mobility data

Sébastien Gams
Univ. de Rennes 1 - INRIA /
IRISA
Campus Universitaire de
Beaulieu, 35042 Rennes,
France
sgams@irisa.fr

Marc-Olivier Killijian
LAAS - CNRS
7 avenue du Colonel Roche
Université de Toulouse ; UPS,
INSA, INP, ISAE ; LAAS
F-31077 Toulouse, France
marco.killijian@laas.fr

Miguel Núñez del Prado
Cortez
LAAS-CNRS
7 avenue du Colonel Roche
Université de Toulouse ; UPS,
INSA, INP, ISAE ; LAAS
F-31077 Toulouse, France
mnunezde@laas.fr

ABSTRACT

In this paper, we describe our ongoing work on GEPETO (for G_{Eo}Privacy Enhancing T_Ookit), a software platform for evaluating the privacy aspects of mobility data. For instance, GEPETO can be used to perform inference attacks on geolocated data such as home identification through clustering algorithms or even to derive more structured knowledge such as learning a compact model of the mobility behaviour of an individual. We briefly review the inference attacks and sanitization methods currently available within GEPETO and we report on the preliminary experiments conducted on geolocated datasets.

Keywords

Privacy, Geolocated data, Inference attacks, Sanitization, Mobility Markov chain.

1. INTRODUCTION

Nowadays, there are many sources by which it is possible to access the mobility data of individuals such as data collected through geolocated devices (e.g., mobile phones, GPS-equipped vehicles...), geosocial networks such as Foursquare, Gowalla or Facebook Places or public repositories gathering geolocated datasets (e.g., Crawdad [6]). Once this data is publicly released, it can be used to infer many personal information beyond its first intended purpose such as the home and place of work of the individuals contained within it, the Points Of Interests (POIs) that characterize these individuals or even their social networks [2, 3].

In this paper, we describe our ongoing work on GEPETO (for G_{Eo}Privacy Enhancing T_Ookit), a software platform for evaluating the privacy aspects of mobility data. For instance, GEPETO can be used to perform inference attacks

*G_{Eo} Privacy Enhancing T_Ookit

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Ubimob 2011

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

on geolocated data such as home identification through clustering algorithms or even to derive more structured knowledge such as learning a compact model of the mobility behaviour of an individual. Once this structure has been learnt, it can be used to conduct more complex attacks such as the de-anonymization of sanitized geolocated data.

The outline of the paper is the following. First in Section 2, we briefly review the concept of geolocated data as well as the three mobility datasets that we have used. Afterwards in Section 3, we describe the architecture of GEPETO as well as the currently implemented algorithms. Finally, we briefly report on experiments performed in Section 4 before concluding in Section 5.

2. GEOLOCATED DATA

Regarding the type of geolocated data, we differentiate mainly between mobility traces and contact traces. A *mobility trace* is characterized by:

- An *identifier*, which can be the real identifier of the device (e.g. "Alice's phone"), a pseudonym or even the value "unknown" (when full anonymity is desired). A pseudonym is generally used when we want to protect the true identity of the system while still being able to link different actions performed by the same user.
- A *spatial coordinate*, which can be a GPS position (e.g. latitude and longitude coordinates), a spatial area (e.g. the name of a neighbourhood in a particular city) or even a semantic label (e.g. "home" or "work").
- A *time stamp*, which can be the exact date and time or just an interval (e.g. between 9AM and 12AM).
- Additional information such as the speed and direction for a vehicle, the presence of other geolocated systems or individuals in the direct vicinity or even the accuracy of the estimated reported position. For instance, some geolocated systems are able to estimate the precision of their estimated location as a function of the number of GPS satellites they are able to detect.

Contact traces are a specific form of mobility traces which consist in the recording of encounters between different devices. This kind of trace is composed of the identifiers of the devices and a time stamp. It may be recorded for instance by a device which has no integrated capacity for geopositioning

but is capable of probing his neighbourhood to detect the presence of other devices (e.g. using Bluetooth neighbour discovery).

The geolocated datasets that we have currently imported within GEPETO to test the different inference attacks and sanitization methods are:

- The aim of the *Phonetic project* [5] is to build realistic mobility models out of real data as well as to study the privacy risks associated with this type of data. Therefore, the goals of this project are closely related to the ones of GEPETO. In this project, Nokia 5800 smartphones have been distributed to registered participants. These smartphones are equipped with a GPS chip, an accelerometer, a compass, a WiFi and a bluetooth interface. The Phonetic software installed on the smartphones measures every minute the GPS position of the owner of the smartphone as well as the bluetooth neighbourhood. This experiment has begun in October 2010 and 7 individuals are currently participating to this study.
- The *taxi cabs dataset* (available from the Crowdad repository [6]) contains the trails of traces of approximately 500 taxis of San Francisco Bay recorded over a 30 days period. This experiment was originally performed to study the economic, social, political and cultural issues as revealed by these taxi traces but of course as this dataset is publicly available, it can also be used also to test the efficiency of inference attacks. These traces were collected through a GPS equipment installed on the taxis.
- The *Geolife dataset* [7] has been collected by Microsoft Research Asia and contains the trails of traces of 165 users over a period of 2 years (April 2007-August 2009). This dataset contains the users daily movements movements, which range from life routines like going home or going to work but also other outdoor movements such as entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling.

3. GEOPRIVACY ENHANCING TOOLKIT (GEPETO)

3.1 Architecture

The design of GEPETO follows a classical client-server architecture. More precisely, the database stored on the server processes and provides all the data needed to the “heavy” clients, which have been implemented in Java JDK 1.6. This client is organized into three layers:

- The Data layer, is in charge of communicating with the database to insert, update, delete and select geolocated data.
- The Logical layer, makes all computation necessary to perform inference attacks, sanitization methods as well as measurement of utility of data.
- The Graphical User Interface layer, allows the application to display raw data as well as to visualize results graphically (for instance on a worldmap).

GEPETO is organized into 14 packages containing 101 classes (which corresponds to approximately 42000 lines of code) and uses 4 different libraries; Colt¹ for matrix operations, JXMapView² for the cartography with open street maps, Jcoord³ for operations dealing with coordinates and Connector-J⁴ for the communication with the database.

GEPETO can parse a geolocated dataset composed of mobility traces into SQL sentences that will be inserted into a geolocated database (cf Figure 1). The grammar of this parser was implemented by using the Java Compiler Compiler (JCC) version 1.5. We run a MySQL 5.1.37 database server engine. The database contains 26 tables that allows to store raw mobility data as well as more refined knowledge issued from sanitization methods and inference attacks. Currently, we have approximately 27 millions raw mobility traces and 50 millions of traces learnt from the different experiments coming from 331 different users.

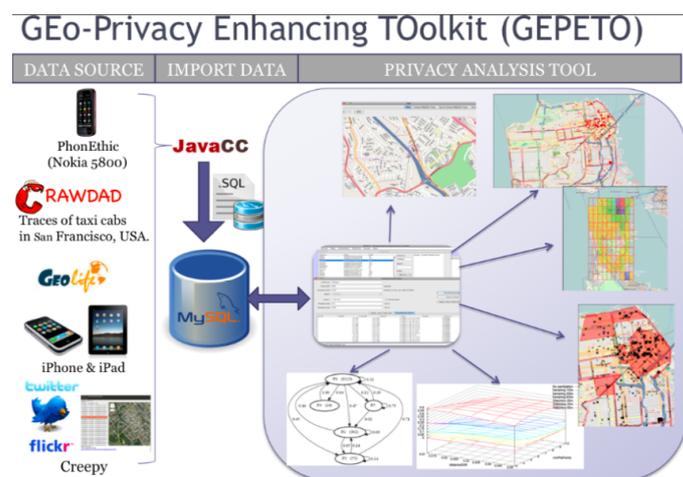


Figure 1: Privacy analysis process in GEPETO.

3.2 Inference attack and sanitization methods

GEPETO can process and analyze geolocated mobility data through inference attacks, sanitization methods as well as measurements of utility. Thus, in this subsection we present a list of the algorithms that have been implemented in GEPETO [2].

- An *inference attack* is an algorithm that takes as input some geolocated data, possibly with some auxiliary information, and produces as output some additional knowledge. For example, an inference attack may consist in identifying the house or the place of work of an individual. The auxiliary information reflects any *a priori* knowledge that the adversary might have gathered (for instance through previous attacks and by accessing some public data source) and which may help him in conducting an inference attack. The inference attacks that we have currently implemented in GEPETO are:

¹<http://acs.lbl.gov/software/colt/>

²<http://wiki.openstreetmap.org/wiki/JXMapView>

³<http://www.jstott.me.uk/jcoord/>

⁴<http://www.mysql.com/products/connector/>

- The *begin-end attack*, is an heuristic that find gaps in trail of traces, which often corresponds to POIs such as home, place work or leisure.
- The *collocation attack*, takes as input the trail of traces of two different individuals and returns as output the period and locations in which they have been in the same vicinity according to a distance threshold given as parameter.
- The *identification of POIs* is an inference attack that relies on a clustering algorithm (such as Density Joinable clustering [1] or GEPETO clustering [3]) to identifies the POIs characterizing the interests of an individual. Clustering is an unsupervised learning technique that groups together objects that are similar in the same cluster while putting objects that are dissimilar into different cluster. The clustering algorithms currently available in GEPETO are : *k*-medians clustering, Density Joinable (DJ) clustering, Time-based clustering and GEPETO clustering.
- A *sanitization algorithm* takes as input a geolocated dataset, introduces some uncertainty and removes some information from this dataset to increase the privacy of individuals whose movements are contained in the dataset. The sanitization algorithm produces as output a sanitized version of the original dataset. The main idea behind sanitization is that, for a potential adversary, performing an inference attack is harder when working on the sanitized version than with the original one. A sanitization procedure usually comes with some privacy guarantees and a trade-off between the uncertainty added and the utility of the information preserved often have to be set.

In GEPETO, we have currently implemented sanitization algorithms working both on the temporal and spatial aspects of data. For instance, a downsampling technique can be used to summarize several mobility traces over some period of time into a single trace while the perturbation technique randomly disturbs the mobility traces by moving them from their original location to a perturbed one according to uniform or Gaussian noise. We have also implemented a pseudonymization procedure replacing the identifier of mobility traces by a random string. This pseudonymization procedure is not sufficient to protect the privacy of individuals but is often the first step of the sanitization process.

- In order to evaluate the resulting impact of the sanitization process on the quality of the data, we have also implemented several *mechanisms for measuring the utility of the data*. For instance, the utility of the data can be measured by computing some global properties of the dataset such as the average speed and distance and verifying how the sanitization process will affect this value compare to the one obtained on the original dataset. Currently in GEPETO, we can compute some statistical properties (e.g. average speed and standard deviation), physical quantities (e.g., radius of gyration and moment of inertia), distance measures (e.g. Euclidean distance, short time series distance as

well as trajectory distance) as well as matrix metrics (e.g., grid density, occupancy rate).

- *Mobility Markov chains* [4] are a precise and compact way to represent the mobility behaviour of an individual in a form of a probabilistic automaton, where each state corresponds to POI (or several POIs) and transitions between states represent movements from one POI to another. This mobility model can be represented either in the form of transition matrix or as a directed graph (cf Figure 2).

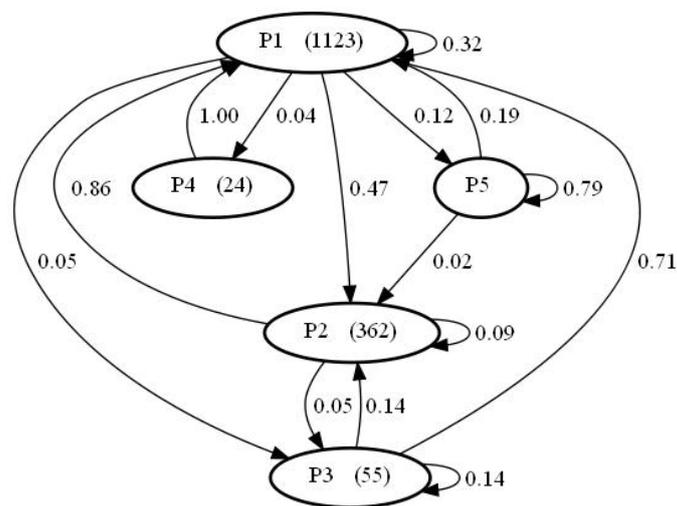


Figure 2: Mobility Markov chain of an individual. Each state represents a POI while transitions correspond to movement between states (i.e. POIs).

4. EXPERIMENTATIONS

First, we have tried the “begin-end” heuristic as well as clustering algorithms on the taxi cabs dataset in order to identify the POIs of taxi drivers [3]. Apart from the traditional touristic places, we were able to infer the homes of some taxi drivers (which use their taxis for work but also for their personal movements) as well as the headquarter of the taxi company. See Figure 3 for instance. We also have tried the same experiments with Phonetic and this has lead to the inference of many precise and highly significant POIs (which were later confirmed by the users themselves) such as home, work, houses of relatives, place of vacation, favorite supermarket or even the school of their children. In the Geolife dataset, we were able to identify the headquarters of Microsoft Labs as well as Universities in Beijing. However, as we are not that familiar with the geography of this Chinese city, it was sometime difficult to assess the validity and relevance of a discovered POI.

We also ran the clustering algorithms on several trails of traces at once to identify the hotspots of San Francisco city, which has lead to the (re)discovery of the airport, the train station, downtown, the red light district as well as some other famous neighborhoods. We also performed the same type of attacks on sanitized data obtained through downsampling and perturbation. In this situation, we observed that as expected the efficiency of the inference attacks degrades gradually as the magnitude of the noise increases (for

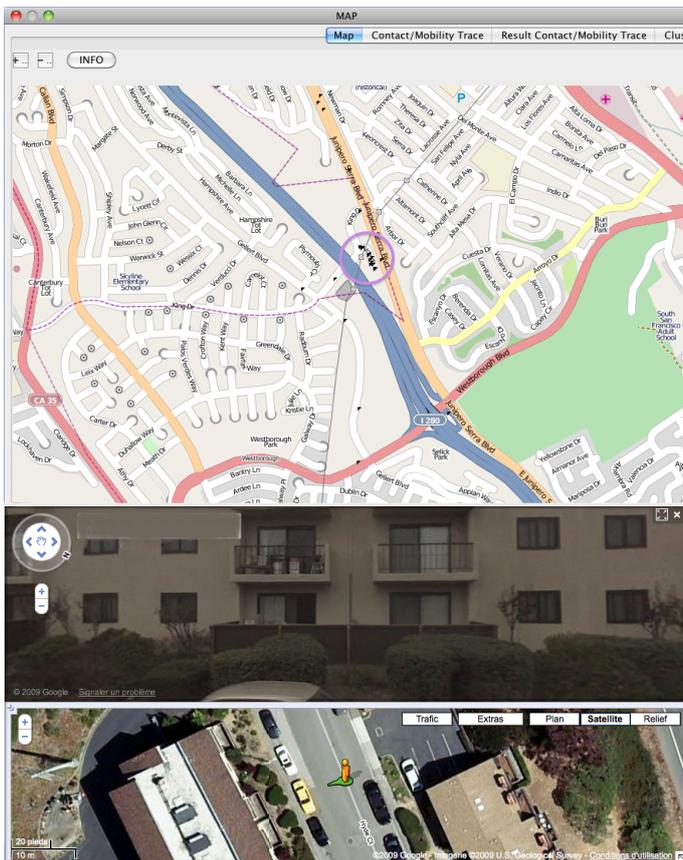


Figure 3: Successful inference attack for identifying the home of a taxi driver.

instance the perturbation level or the size of the sampling window). For instance, in Phonetic when we apply a perturbation of a magnitude of 50 meters, this greatly decreases the probability of identifying a particular POI (cf Figure 4). See [3] for more details of the experiments.

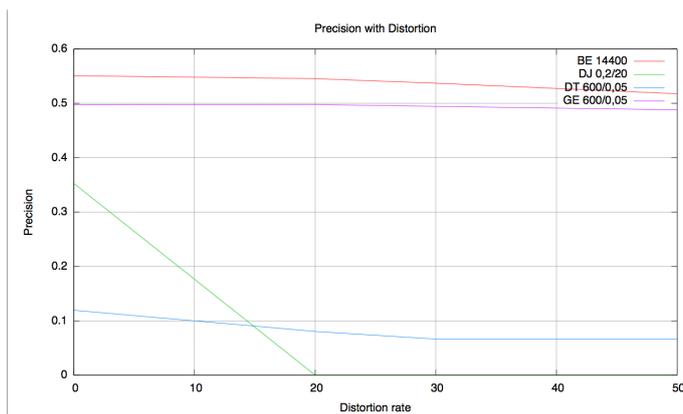


Figure 4: Precision evaluation in Phonetic under perturbation and for different clustering algorithms. The “Begin-end” attack is in red, DJ clustering in green, Time-based clustering in blue and GEPETO clustering in purple.

Finally, we have learnt the mobility Markov chains of users of Phonetic and we have observed that just from the structure of the Markov chain, it is easy to identify the home (i.e. the state that can be reached from almost all the states) and the work (i.e. the state that can be reached with the heaviest transition from the home). Of course, when we performed the same experiments on the taxi cabs, we observed that these rules of thumb cannot be used anymore as most of the states of their mobility Markov chains correspond to states frequently visited by tourists.

5. CONCLUSION

In this paper, we have given an overview of our ongoing work for developing GEPETO (for GEOPrivacy Enhancing TOOLkit), a software platform for evaluating the privacy aspects of mobility data. We have also briefly described some inference attacks as well as sanitization methods currently implemented within GEPETO and reported on preliminary experiments. As future work, we plan to integrate more sophisticated inference attacks and sanitization methods (such as spatial cloaking and mix-zone) within GEPETO as well as to continue to study the applicability of mobility model such as Markov chain for performing inference attacks.

6. REFERENCES

- [1] Z. Changqing, D. Frankowski, P. Ludford, S. Shekhar and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. *Proceedings of the ACM International Workshop on Geographic Information Systems, Washington, USA*, pages 266–273, 2004.
- [2] S. Gambs, M. Killijian, and M. N. del Prado. GEPETO: a GEO Privacy Enhancing TOOLkit. *Proceedings of the International Workshop on Advances in Mobile Computing and Applications: Security, Privacy and Trust, held in conjunction with the 24th IEEE AINA conference, Perth, Australia*, pages 1071–1076, 2010.
- [3] S. Gambs, M. Killijian, and M. N. del Prado. Show me how you move and I will tell you who you ar.. *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL'10), San Jose, USA*, pages 34–41., 2010.
- [4] S. Gambs, M.-O. Killijian, and M. N. del Prado. Show me how you move and I will tell you who you are (extended version). *Submitted to a journal*.
- [5] M. Killijian, M. Roy, and G. Trédan. Beyond San Francisco cabs: Building a *-lity mining dataset. *Proceedings of the Workshop on the Analysis of Mobile Phone Networks (netmob), Sattelite of NetSci Cambridge, Cambridge, USA*, 2010.
- [6] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from <http://crawdad.cs.dartmouth.edu/epfl/mobility>, February 2009.
- [7] Y. Zheng, Q. Li, Y. Chen, and X. Xie. Understanding mobility based on GPS data. *Proceedings of ACM conference on Ubiquitous Computing (UbiComp), Seoul, Korea*, page 312–321, 2008.

Conception et Evaluation d'un Système de Dialogue pour un Dispositif de Suppléance pour Déficients Visuels

Slim Kammoun Bernard Oriola Philippe Truillet Christophe Jouffrais

IRIT, CNRS & Université de Toulouse 3
118 Route de Narbonne, 31062, France
{Kammoun, Oriola, Truillet, Jouffrais}@irit.fr

RESUME

Avec l'essor des nouvelles technologies de l'information et de la communication, la question de l'utilisation des dispositifs mobiles par tous les utilisateurs devient cruciale. Dans cet article nous nous intéressons à l'utilisabilité d'un dispositif de suppléance (NAVIG) dédié aux déficients visuels. Ce système a comme objectif d'améliorer l'autonomie des utilisateurs en restaurant deux fonctions visuelles fondamentales à savoir la localisation de soi dans l'environnement (ce qui permet de fournir une aide aux déplacements) et la localisation d'objets d'intérêt (par exemple dans le but de les saisir). NAVIG¹ est doté d'un système de positionnement par satellite et d'un système de vision artificielle embarqué. L'interaction avec le dispositif a été déterminée suivant un processus de conception centrée utilisateur, qui a abouti à une interaction vocale uniquement. L'évaluation du prototype a été réalisée par 8 sujets déficients visuels, et a montré l'efficacité des choix et des stratégies adoptées à savoir la reconnaissance vocale en entrée et la synthèse vocale en sortie.

MOTS CLES

Interaction vocale, prototypage rapide, déficients visuels, autonomie, suppléance.

ABSTRACT

With the rise of new information and communication technologies, the use of mobile devices by all users is crucial. In this paper we focus on the usability of an assistive device (NAVIG) for the visually impaired people (VIP). The aim of the device is to enhance autonomy of VI users by restoring basic visual functions namely the location of self in the environment (e.g. to provide travel aid) and the location of objects of interest (e.g. in order to grasp). NAVIG is equipped with a satellite positioning system and an artificial vision system. The interaction with the device was determined in a user-centered design, which resulted in voice interaction only. The evaluation of the prototype was conducted by 8 blind subjects, and showed the effectiveness of choices and strategies adapted to speech recognition input and speech output.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: User-Centered Design - Prototyping
K.4.2 [computer and society]: Social issues – Assistive Technologies for person with disabilities.

¹ <http://navig.irit.fr/>

General Terms

Design, experimentation, human factors.

Keywords

Speech interaction, rapid-prototyping, visually impaired persons, assistive device.

1. INTRODUCTION

En 2002², plus de 161 millions de personnes dans le monde étaient atteintes de déficiences visuelles, 124 millions souffraient d'une baisse de vision et 37 millions étaient aveugles selon l'Organisation Mondiale de la Santé (OMS). De plus, 65% des personnes aveugles interrogées par l'institut national canadien pour les aveugles ont déclaré qu'ils avaient besoin d'une aide pour leurs déplacements quotidiens. C'est dans cet environnement que s'inscrit le projet NAVIG [3] (Navigation Assistée par VIsion embarquée et Géopositionnement) qui vise à développer et évaluer un système embarqué d'aide à la navigation pour les déficients visuels. L'objectif du projet est d'offrir une aide à l'intérieur (indoor) comme à l'extérieur (outdoor) en se basant sur un système de positionnement (type GPS) et une centrale inertielle pour améliorer la précision du positionnement, ainsi que sur un système de vision artificielle embarqué. Le module de vision artificielle embarqué [2] aura comme double objectif d'améliorer le positionnement GPS en détectant des cibles environnementales dont les coordonnées géographiques sont connues, et de détecter des cibles proches utiles à l'utilisateur (e.g. boîte à lettres, distributeur de billets, etc.). Dans ce papier, nous nous focaliserons plus particulièrement sur la méthode de conception et le prototypage de l'interface utilisateur dans le but de permettre une interaction simple et efficace par des déficients visuels.

2. INTERACTION AVEC LE SYSTEME

Permettre l'accessibilité des dispositifs mobiles aux déficients visuels consiste tout d'abord à substituer les fonctions assurées par la modalité visuelle. On la remplace par exemple par la modalité tactile ou la modalité sonore pour une personne aveugle comme dans le projet SWAN [14] avec la sonification par conduction osseuse ou encore Drishty [6], embarquant un système de reconnaissance et de synthèse vocales. De nombreux dispositifs d'aide à la navigation pour déficients visuels existent sur le marché. On peut distinguer deux principales catégories de

² L'étude la plus récente de l'Organisation Mondiale de la Santé.

dispositifs : les systèmes physiques dédiés comme le Trekker³ ou Kapten⁴ et les applications logicielles installées sur des téléphones mobiles ou des PDA (comme MobileGéo⁵ ou StreetTalk VIP⁶).

La réussite de ces dispositifs repose sur le bon choix des techniques d'interaction à savoir la saisie de commandes en entrée (destination, demande de renseignements, ...) et la méthode de restitution en sortie.

2.1 Saisie de données

Des espaces de classification pour l'interaction sur dispositifs mobiles ont été proposés ([9], [12]). Néanmoins, ces classifications ne se sont pas intéressées précisément aux techniques utilisables par les déficients visuels. Concernant les modalités utilisables par les déficients visuels, on trouve la modalité tactile avec des boutons physiques comme le clavier réduit ou fortement réduit, (utilisé par exemple dans la technique NAVTAP [7], en conjonction avec les écrans tactiles comme dans NavTouche [5], ou dans l'agenda VoCal [11]). On trouve aussi la modalité vocale en utilisant des systèmes de reconnaissance automatique de la parole.

2.2 La restitution des données

Pour les déficients visuels, l'utilisation de la modalité sonore comme moyen de restitution de données au travers d'une synthèse vocale semble a priori intéressante. On peut aussi citer la modalité somesthésique. Les sensations perçues par l'utilisateur peuvent alors se décomposer en deux catégories : soit par une stimulation tactile (affichage braille par exemple), soit par une stimulation proprioceptive (retour d'effort).

3. METHODE DE CONCEPTION

Le choix de la technique de saisie de données et de la méthode de restitution de l'information est primordial étant donné l'impact sur l'architecture globale de NAVIG et l'utilisabilité du système par les déficients visuels. Afin de définir les techniques les plus adaptées à leur déficience, nous avons adopté une méthode de conception participative adaptée à des déficients visuels [1]. Nous pensons qu'il est fondamental d'associer les usagers à toutes les étapes du développement du projet, a fortiori lorsque l'objet de la conception s'adresse à des utilisateurs déficients. Pour cela, nous avons organisé un brainstorming [10] avec les utilisateurs et l'ensemble de l'équipe de travail à la suite duquel nous avons élaboré un scénario de conception relatif à l'usage des modalités d'interaction. Le principal résultat a été que tous les utilisateurs ont évoqué la modalité vocale en entrée et en sortie comme modalité principale, à savoir un système de reconnaissance de la parole comme technique de saisie de données et la synthèse vocale pour la restitution de l'information. Cela présente plusieurs avantages, car en effet l'utilisation de la parole est intuitive et ne nécessite aucune phase d'apprentissage de l'usage de la modalité. De plus, il s'agit d'une approche main libre ce qui permet aux déficients visuels d'utiliser leur canne ou leur chien-guide. Néanmoins, une question centrale reste en suspens. La technologie est-elle suffisamment mature pour être adaptée dans le cadre de l'aide au déplacement ? Pour répondre à cette question, une étape de prototypage rapide s'est avérée nécessaire.

³ <http://www.humanware.com/en-europe/home>

⁴ <http://www.kapsys.com>

⁵ <http://www.codefactory.es>

⁶ <http://www.freedomscientific.com>

Il s'agissait ici plus particulièrement de tester l'utilisation d'un système, mais aussi d'améliorer les connaissances sur les besoins et le comportement de l'utilisateur, ainsi que les conséquences sur les fonctionnalités du système (en termes de modalités d'interaction, stratégies de dialogue, et vocabulaire utilisé).

4. LE PROTOTYPAGE RAPIDE

En tenant compte du fait que NAVIG était un projet pluridisciplinaire faisant intervenir plusieurs entités (laboratoires de recherche et partenaires industriels), nous avons opté pour une architecture distribuée basée sur la notion d'entités (agents) favorisant le prototypage rapide et la réutilisation d'entités existantes. Cette distribution permet notamment d'étudier les l'interaction, même si la totalité du système n'est pas fonctionnelle, en facilitant la réalisation de systèmes en Magicien d'Oz. Nous avons choisi le middleware Ivy [4] du CENA⁷.

Dans ce contexte, nous avons exploité deux agents Ivy préalablement codés dans l'équipe, pour l'entrée, un agent de reconnaissance vocale (SRA - Speech Recognition Agent) compatible MS-SAPI [8] permettant d'utiliser tous les moteurs de reconnaissance du marché. Pour la sortie, nous avons utilisé un agent de synthèse vocale lui aussi compatible MS-SAPI. Pour notre prototype, nous avons intégré le moteur de reconnaissance Dragon Naturally Speaking 10 (DNS10) en entrée et Elan RealSpeak (ERS) comme synthèse vocale. Pour compléter notre architecture, nous avons développé un contrôleur de dialogue (nommé ci-après CD) qui assure le séquençage du dialogue système-utilisateur en se basant sur une stratégie de dialogue constituée de séquences questions/réponses.

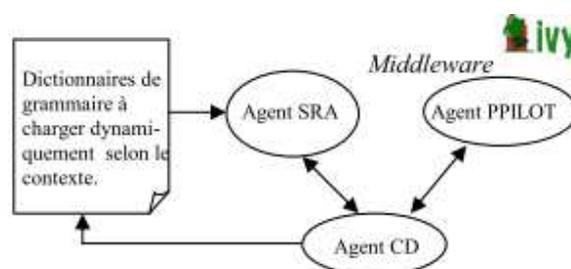


Figure 1: Architecture de l'interface utilisateur

L'ensemble des agents utilisés sont connectés entre eux (cf. Figure 1). Nous avons enfin implémenté un menu vocal qui représente les différentes possibilités imaginées et citées lors du brainstorming pour exploiter NAVIG. Afin d'améliorer les performances de la reconnaissance vocale, nous avons limité le contexte relatif à chaque question en se basant sur des grammaires manipulées dynamiquement par SRA et CD de façon à charger à chaque question du système le dictionnaire qui contient l'ensemble du vocabulaire attendu relatif à cette question.

⁷ Centre d'Etudes de la Navigation Aérienne, actuelle DTI R&D

4.1 Les dictionnaires

L'utilisabilité d'un système vocal est fortement liée à sa capacité à répondre rapidement à une sollicitation de l'utilisateur tout en proposant un taux de reconnaissance satisfaisant [13]. De ce fait, nous avons procédé par dictionnaires. Cette approche consiste à créer des grammaires qui contiennent l'ensemble des mots clés susceptibles d'être reconnus au cours du dialogue et pouvant être modifiées dynamiquement suivant le contexte du dialogue.

4.2 EXPERIMENTATION

4.2.1 Objectifs

Pour notre première expérimentation, nos hypothèses de départ étaient que l'interaction vocale est une modalité efficace et simple à utiliser par un déficient visuel. Nous avons vérifié cette hypothèse au travers de divers scénarii d'utilisation. Il ne s'agit pas ici d'une étude comparative avec d'autres modalités, mais bien de tester l'appropriation de l'interface par les utilisateurs. Nous nous sommes attachés à vérifier que les messages fournis par la synthèse vocale étaient suffisamment précis et intelligibles et que le temps de saisie d'un itinéraire restait raisonnable (il ne doit pas dépasser quelques dizaines de secondes).

4.2.2 Protocole

L'évaluation du prototype réalisé a été menée dans le laboratoire des usages⁸ de l'IRIT. Le groupe test était composé de 6 hommes et 2 femmes tous non-voyants âgés de 22 à 60 ans. Tous les sujets étaient des personnes disposant d'une autonomie correcte dans les déplacements, en utilisant un chien-guide ou une canne. Les tests se sont déroulés individuellement sur une durée d'une heure. Avant chaque test, une explication du fonctionnement et une démonstration étaient réalisées pour chaque sujet. Aucune phase d'apprentissage préalable n'a été réalisée. Le matériel disponible était un PC portable Core2duo doté de 1Go de RAM, un micro casque de la marque PLANTRONICS muni d'un filtre anti-bruit.

4.2.3 Déroulement de l'expérimentation

Chaque sujet devait effectuer 3 tâches relatives aux trois principaux modes d'utilisation de NAVIG à savoir 1) effectuer un parcours en saisissant une adresse de destination fournie aux sujets (imprimée en braille) ; 2) simuler un parcours en entrant deux adresses : un point de départ et une destination ; et 3) déclencher le guidage en champ proche. Dans cette dernière tâche, le sujet devait saisir le nom d'un objet particulier (comme une boîte aux lettres par exemple) qu'il voulait atteindre avec sa main.

4.2.4 Critères d'évaluation

La mesure des performances du prototype réalisé portait sur deux variables jugées critiques pour un système de suppléance : le taux d'erreur de la reconnaissance vocale et la vitesse de saisie d'une adresse complète. Ces deux mesures étaient effectuées automatiquement par le système. Par contre, la mesure de l'intelligibilité des messages vocaux était réalisée via un questionnaire post-expérimental.

5. RESULTATS ET ANALYSE

5.1 Taux d'erreur

L'interface de NAVIG devait pouvoir reconnaître les différents types de voix (hommes et femmes) dans n'importe quelle situation

⁸ <http://www.irit.fr/labo-usages/>

sans être perturbée par des facteurs acoustiques liés à la position de l'utilisateur (bruit ambiant, etc.), ni par des facteurs relatifs à l'état de l'utilisateur (stress, fatigue). Nous avons constaté que malgré l'hétérogénéité des voix de notre groupe d'utilisateurs (accents, intonations), le moteur de reconnaissance vocale utilisé (Dragon Naturally Speaking 10) s'est montré particulièrement performant en atteignant un taux de reconnaissance moyen de 97%. Ce taux a été calculé sur la base de 33 mots clés nécessaires pour effectuer l'ensemble des tâches demandées. Il faut signaler qu'aucune phase d'apprentissage n'a été réalisée et que tous les sujets ont accompli toutes les tâches sans aucune intervention extérieure. La différence entre le nombre de commandes prononcées par chaque utilisateur et le nombre nominal pour effectuer la tâche, est due aux hésitations car il s'agissait de leur première utilisation de l'interface.

5.2 Temps de saisie d'une adresse

Le temps de saisie d'une adresse est une variable très importante pour NAVIG, ce temps devant rester « raisonnable » et ne pas dépasser quelques dizaines de secondes. Nous avons défini l'adresse comme étant de la forme « Numéro, Rue, Ville ». Le sujet le moins rapide a mis 29 secondes en moyenne pour la saisie d'une adresse, et pour le plus rapide, ce temps n'a pas dépassé 15 secondes. La différence peut être expliquée par l'habitude à utiliser ce type de système et par l'utilisation de la fonction de barge-in implémentée dans notre prototype (la reconnaissance vocale est prioritaire sur la synthèse). Cette performance pourrait sans doute être encore améliorée en utilisant une stratégie de dialogue éliminant tous les messages de confirmation explicite de la saisie. La Figure 2 représente la variation des temps de saisie par sujet et par essai. On peut constater une amélioration du temps de saisie pour chaque sujet (sauf pour le sujet 4 dont le temps – le meilleur de tous - reste constant sur les trois essais).

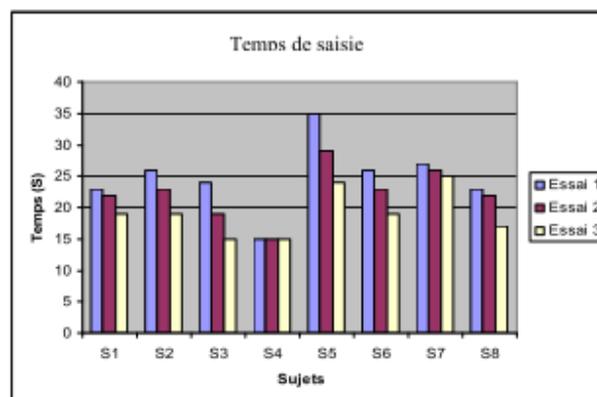


Figure 2: Variation des temps de saisie d'une adresse selon les sujets

5.3 La synthèse vocale

La synthèse vocale a été évaluée à travers le questionnaire post-évaluation. Les utilisateurs se sont montrés très intéressés par le système. Lors des tests, ils ont été capables de naviguer dans les menus sans aucune aide extérieure. Ils sont arrivés à bien comprendre les messages vocaux et à bien se repérer dans l'arborescence des menus. Les utilisateurs ont en revanche été critiques sur le rendu de la synthèse vocale, jugée parfois « tremblante » ou inintelligible. De plus, certains utilisateurs ont

trouvé les messages vocaux trop longs. Ils préféreraient des messages plus courts même s'ils sont moins explicites. Enfin, concernant le débit de la synthèse vocale, l'interface disposait de 3 niveaux différents de vitesse à savoir : lent, moyenne et rapide ce qui a été apprécié par l'ensemble des sujets.

6. CONCLUSION

Nous avons montré dans cet article la méthode de conception utilisée pour définir les interactions homme-système de NAVIG jusqu'au prototypage. Ce dernier est pleinement opérationnel et fait actuellement l'objet de modifications suivant les remarques exprimées par les usagers lors des tests. Nous travaillons actuellement sur de nouvelles stratégies de dialogue et essayons d'étendre la grammaire sans diminuer les performances de la reconnaissance. Nous travaillons enfin sur les autres composantes du projet pour entamer une nouvelle série de tests en situation réelle.

7. REMERCIEMENTS

Ce travail est soutenu par l'Agence Nationale de Recherche (ANR) au travers du programme TecSan (projet Navig n° ANR-08-TECS-011) et la Région Midi-Pyrénées par le biais du programme APRRTT.

BIBLIOGRAPHIE

- [1] Brock, A., Vinot, J.L., Oriola, B., Kammoun, S., Truillet, P., et Jouffrais, C., "Méthodes et outils de conception participative avec des utilisateurs non-voyants," The 22th French-speaking conference on HCI, Luxembourg, 2010.
- [2] Dramas, F., Thorpe, S. J., et Jouffrais, C. "Artificial Vision For The Blind: A Bio-Inspired Algorithm For Objects And Obstacles Detection," International Journal of Image and Graphics Vol. 10, No. 4, p. 531-544, 2010.
- [3] Katz, B.F.G., Truillet, P., Thorpe, S., et Jouffrais, C., "NAVIG: Navigation Assisted by Artificial Vision and GNSS," Multimodal Location Based Techniques for Extreme Navigation, Pervasive, helsinki, Finland, 2010
- [4] Buisson, M., Bustico, A., Chatty, S., Colin, F.R., Jestin, Y., Maury, S., Mertz, Ch., Truillet, P., "ivy : un bus logiciel au service du développement de prototypes de systèmes interactifs". Proceedings of the 14 th French-speaking conference on HCI, Poitiers, pp. 223-226.
- [5] Guerreiro, T., Lagoá, P., Nicolau, H., Santana, P., Jorge, J., "Mobile text-entry models for people with disabilities". Proceedings of the 15 th European conference on Cognitive ergonomics.
- [6] Helal, A., Moore, A., et Ramachandran, B. "Drishti: an integrated navigation system for visually impaired and disabled". dans Wearable Computers. Proceedings. Fifth International Symposium on, 2001, pp. 149-156, 2001.
- [7] Lagoá, P., Santana, P., Guerreiro, T., Gonçalves, D., Jorge, J. (2007). Blono: a New Mobile Text-entry Interface for the Visually Impaired. In Springer LNCS, Universal Access in HCI Part II, pp. 908-917, 2007.
- [8] Microsoft Corporation, Microsoft Speech and SAPI 5, <http://www.microsoft.com/speech>.
- [9] Nigay, L., et Coutaz, J., "Espaces conceptuels pour l'interaction multimédia et multimodale". TSI, spéciale Multimédia et collectif, AFCET & HERMES, 1996.
- [10] Norman, A., D., Draper, W., S. (Eds.). "User-centred system design: New perspectives in Human Computer Interaction", Hillsdale, NJ: LEA, 1986.
- [11] Oriola, B., Hermet, A., et Raynal, M., "User Centered Design Applied to Non Visual Interaction". UAH-CI'05, Las Vegas (U.S.A), 2005.
- [12] Roudaut, A., Lecolinet, E., "Un espace de classification pour l'interaction sur dispositifs mobiles". French-speaking conference on HCI, ACM Press, pp. 99-106. 2007
- [13] Truillet, Ph. Bothorel, G. VOICE : une plate-forme vocale pour la formation au contrôle aérien. French-speaking conference on HCI, ACM Press, pp. 67-74, 2005.
- [14] Walker, B.N. et Stanley, R. M. Thresholds of audibility for bone conduction headsets. Proceedings of ICAD 05- Eleventh Meeting of the International Conference on Auditory Display, Limerick, Ireland, pp. 218-222, 2005.

SysCARE : Système de Communication et d'Assistance Robotisée

Louis-Pierre Bergé[#], Cyril Briand^{*}
Agnan de Bonneval^{*}, Michel Taix^{*}
Philippe Truillet[#]

Université de Toulouse
(LAAS-CNRS^{*} et IRIT UMR CNRS[#])
118, Route de Narbonne
31400 Toulouse, France
<briand, agnan, taix>@laas.fr
<berge, Philippe.Truillet>@irit.fr

Olivier Lefebvre

Magellium
24, Rue Hermès
BP 12113
31521 Ramonville Saint-Agne, France

olivier.lefebvre@magellium.fr

RESUME

Nous présentons dans cet article *SysCARE*, un système de communication et d'assistance robotisée à destination des personnes âgées. *SysCARE* s'inscrit dans le domaine de la e-santé, et correspond à une étude de faisabilité d'une solution concrète. L'objectif de ce projet est la réalisation d'un démonstrateur composé d'un ensemble de services d'aide aux personnes âgées ou dépendantes à domicile.

MOTS CLES : télé-présence, monitoring à distance, technologies d'assistance.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 [User Interfaces]: Interaction Styles, Prototyping, Voice I/O. K.4.2 [computer and society]: Social issues – Assistive Technologies for persons with disabilities.

GENERAL TERMS: Design, Human Factors

KEYWORDS: tele-presence, remote monitoring, assistive technologies.

INTRODUCTION

Selon l'INSEE [4], le nombre de personnes âgées dépendantes devrait augmenter de 50% à l'horizon 2040. Le vieillissement rapide de la population pose un défi majeur à nos sociétés. En effet, avec l'augmentation inexorable du nombre de personnes âgées, on constate en parallèle une exclusion sociale de plus en plus prononcée et les problèmes de prise en compte de la santé et de la gestion des soins de longue durée deviennent cruciaux.

Ce défi est corrélé avec une demande sociétale d'une plus grande qualité de soin. Néanmoins, le manque de personnels formés et qualifiés pour dispenser ces soins et

plus généralement l'augmentation du nombre de personnes dépendantes pose la question du financement. Qu'elle soit physique ou mentale, la dépendance se caractérise par l'incapacité ou simplement la difficulté d'une personne à vivre seule dans un logement ordinaire. Cela implique la présence d'un soutien à domicile, ou d'un placement en institution spécialisée. De nouvelles solutions explorées sont issues des Technologies de l'Information et de la Communication (TIC) et constituent le domaine que l'on appelle e-Santé. Ce domaine d'application est en plein essor et c'est dans ce contexte que s'inscrit le projet *SysCARE* présenté ici.

SysCARE s'appuie sur une situation assez classique qui est celle de personnes âgées vivant seules, diminuées physiquement et mentalement, qui ont besoin d'un suivi médical régulier, mais qui ne nécessitent pas pour autant une hospitalisation. Ces personnes sont généralement visitées régulièrement par du personnel médical avec les inconvénients suivants :

- un coût financier important,
- un mode de suivi non suffisant si la personne présente des risques d'accidents (chute, AVC, etc.),
- un temps de présence globalement insuffisant pour lutter contre l'isolement de la personne.

SYSCARE – UN PROJET ... PEDAGOGIQUE

Ce projet, réalisé sur cinq mois, a été proposé à la promotion 2010/2011 du Master 2 de l'IUP Systèmes Intelligents (SI) de l'Université de Toulouse. [8] par la société Magellium [7] dans le cadre du projet de fin d'études nommé Projet de Grande Envergure (PGE). Le but est de concevoir et mettre en œuvre un système complexe nécessitant l'interaction et la coopération des compétences multiples vues au cours des trois ans de formation des étudiants de l'IUP SI autour d'un cahier des charges fourni par un industriel qui agit comme un véritable client tout au long du projet.

Organisés comme une réelle petite entreprise, les étudiants font face à un projet industriel avec les contraintes qui y sont associées, notamment celles de temps et de

qualité. Ce projet est innovant à la fois par les techniques mises en œuvre et aussi par son partenariat qui associe l'ensemble d'une formation et un industriel autour d'une thématique sociétale.

L'équipe du projet est divisée en trois catégories :

1. l'équipe « Encadrants » qui comprend 10 enseignant-chercheurs de l'université Paul Sabatier (UPS) qui appartiennent à deux laboratoires toulousains : le LAAS-CNRS [6] et l'IRIT [5],
2. l'équipe « Entreprise » qui comprend 3 ingénieurs de Magellium, dont un chef de projet
3. et enfin l'équipe « Etudiants » qui comprend 27 étudiants, dont 3 chefs de projet choisis par leurs pairs.

Le prototype est développé par la formation et il permet à l'industriel de tester en réel une solution en phase amont avant de décider un développement plus précis en interne dans la société. C'est donc un partenariat d'étude de faisabilité qui s'est développé dans le cadre de SysCARE et qui en démontre toutes les potentialités.

SYSCARE – UN PROJET ... INNOVANT

Une solution pour prolonger le maintien à domicile d'une personne âgée peut consister à lui fournir un système offrant des fonctionnalités :

- d'assistance physique et cognitive, caractérisée entre autre par la recherche d'objets dans une pièce ou par le rappel de prise de médicaments et la possibilité d'enregistrer un pense-bête vocal.
- de monitoring de la personne (détection de comportements anormaux, de chute, etc.)
- et de lien avec du personnel médical et des membres de la famille, grâce à une communication audio et vidéo avancée permettant la mise en relation avec des membres de la famille ou du personnel médical, transmissions de données de suivi, etc.

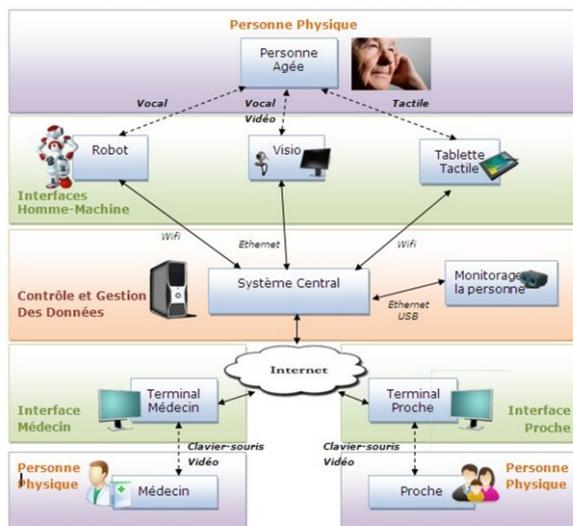


Figure 1: architecture de SysCARE

Ainsi, du point de vue matériel, SysCARE est composé de nombreuses technologies soit éprouvées soit émergentes de plus en plus utilisées (cf. Figure 1) permettant de mettre en œuvre les fonctionnalités listées.

Tout d'abord, le système repose sur de la vision artificielle à l'aide de caméras (système MagBot de Magellium) afin de suivre la personne et détecter des anomalies. La calibration du système perceptuel est faite lors de l'instrumentalisation du domicile.

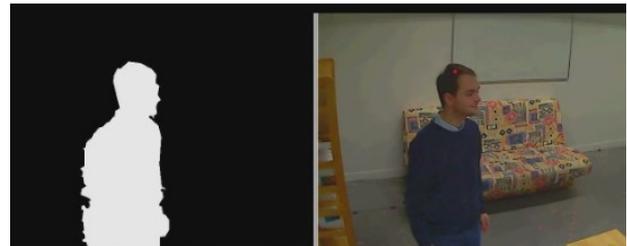


Figure 2: détection vidéo des utilisateurs

A l'aide d'une tablette tactile du commerce (Archos 101 internet tablet [3]), la personne peut interagir avec son environnement (action du robot), répondre ou lancer une visio-conférence sur un écran de télévision (équipe médicale ou familiale). L'interface a été développée sous Android OS 2.2 FroYo [2] et peut facilement évoluer en fonction d'un retour d'expérience.

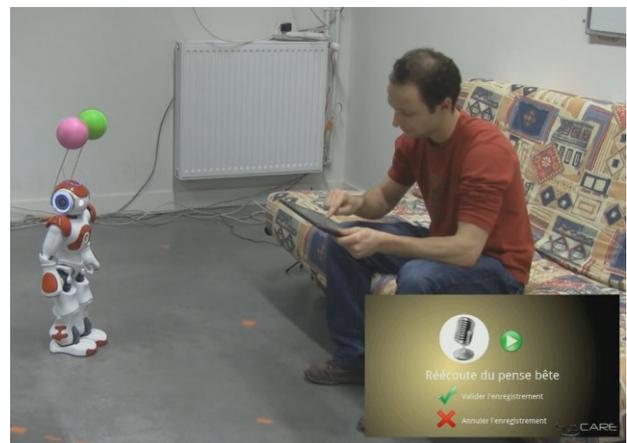


Figure 3: usage de la tablette tactile et du pense-bête

Enfin, SysCARE utilise le robot humanoïde Nao (Aldebaran Robotics [1]), accompagnant la personne. L'intérêt d'un robot humanoïde est qu'il est plus facilement accepté par une personne et qu'il offre beaucoup de fonctionnalités embarquées (vision, parole, reconnaissance d'objet ou de face, ...) et qu'il peut engendrer une action dans l'environnement. Il offre plusieurs possibilités de programmation qui ont été utilisées dans ce projet. De nombreux travaux en apprentissage et notamment en interaction gestuelle et verbale avec l'humain permettent d'envisager à court terme des applications cognitives intéressantes dans le cadre du maintien à domicile avec un robot humanoïde.

L'ensemble de ces services mis en œuvre représentent des applications à fort potentiel dans un futur proche.

Du point de vue logiciel, les étudiants ont développé les différents systèmes suivants :

- un système complet de visio-conférence,
- un système de surveillance médicale (détection de chute et d'inactivité),
- l'interface graphique de la tablette,
- les activités du robot,
- et les logiciels du système central nécessaires aux échanges de communication entre ces différents composants.

Plusieurs séquences d'interaction (par grande classe de fonctionnalités) ont été proposées et implémentées dans le prototype *SysCARE*.

DES EXEMPLES D'USAGE

Nous détaillons ci-après deux scénarii : celui de l'établissement de connexion à la visioconférence (cas nominal) et celui de la détection de chute.

Pour activer la visioconférence, la personne âgée a à sa disposition une tablette tactile (cf. visuels pour l'interface) et un écran de télévision. Le contexte de départ est l'écran d'accueil de l'interface. La personne sélectionne sur l'interface un contact à appeler, le système affiche alors la fiche du contact. La personne peut alors cliquer sur un bouton pour lancer l'appel. Le système établit la connexion, affiche la visioconférence sur l'écran TV, affiche la photo du contact sur la tablette et un bouton pour terminer l'appel.



Figure 4: usage de télé-présence

Concernant la détection de chute, le système de vision ainsi que le robot sont sollicités. Quand la personne tombe par terre (<http://youtu.be/H1nKt7kauWA>), le système de supervision se met en alerte (dite de niveau 1). Le système émet un ordre de déplacement pour le robot vers la personne, le robot confirmant l'ordre lors-

qu'il arrive à un mètre de la personne. Un compte à rebours est déclenché durant lequel le robot essaye de contacter (dialogue) la personne et de vérifier (vision) son état.



Figure 5: intervention du robot lors d'une chute

Si la personne ne répond pas durant cette période ou que la réponse de la personne nécessite une intervention, le robot indique au système qu'il faut élever le niveau d'alerte et déclenche une télé-opération. Le système envoie alors un SMS et un mail d'alerte au médecin contenant une photo de la chute. Dans le cas contraire, le robot émet une requête au système pour sortir du niveau d'alerte.

PROCHAINES ETAPES ET CONCLUSION

Ce prototype-support montre la faisabilité des concepts de base et la maturité de certaines technologies pour les mettre en œuvre à court terme.

De plus, il confirme l'intérêt d'engager des « expérimentations » plus poussées afin de mieux définir l'interface de communication et les services minimum à offrir à la personne.

BIBLIOGRAPHIE

1. Aldebaran Robotics, <http://www.aldebaran-robotics.com>
2. Android OS, <http://www.android.com>
3. Archos 101 internet Tablet
http://www.archos.com/products/ta/archos_101_it/index.html
4. Duée Michel, Rebillard Cyril, *La dépendance des personnes âgées : une projection à long terme*, INSEE, avril 2004.
5. IRIT, <http://www.irit.fr>
6. LAAS-CNRS, <http://www.laas.fr>
7. Magellium, <http://www.magellium.fr>
8. Master Systèmes Intelligents,
<http://mastersi.ups-tlse.fr>

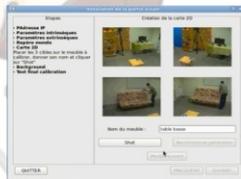
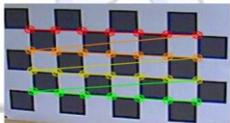


Système de vision

Ensemble de 4 caméras
 Localisation 3D de la personne dépendante
 Localisation 3D du robot d'assistance
 Détection de chute et d'inactivité prolongée



Interface de calibration des caméras
 Interface de création d'une carte 2D via une localisation des objets de l'environnement



Robot d'assistance

Assistance physique et cognitive chez la personne dépendante
 Rappel de la prise de médicaments
 Rappel de pense-bêtes vocaux
 Dialogue avec la personne
 Contrôlable à distance
 Planification de trajectoire, évitement d'obstacles



Tablette tactile



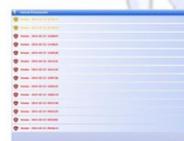
Destinée à la personne dépendante
 Interface de visioconférence avec les proches et médecins de la personne dépendante
 Inspection d'une zone à l'aide du robot d'assistance
 Enregistrement de pense-bêtes vocaux



Logiciel médecin et proche



Accès au travers d'Internet
 Communication en visioconférence avec la personne dépendante
 Téléopération du robot d'assistance
 Accès aux fiches patients
 Réception et visualisation d'alertes
 Prescription de médicaments



Conception d'une interaction avancée pour stimuler les visiteurs d'un musée

Bénédicte Schmitt
Global Vision Systems
10 Avenue de l'Europe
31520 Ramonville Saint-Agne
benedicte.schmitt@global-vision-
systems.com

Louis-Pierre Bergé, Emmanuel Dubois
Université Paul Sabatier
118 route de Narbonne
31062 Toulouse cedex 9
{prenom}.{nom}@irit.fr

ABSTRACT

During past years, a lot of cultural interactive experiments, and in particular in museography, appeared. The goal is to make visitor actor of the visit by interacting directly with concepts. These concepts are hard to understand as they are complex or abstract. The use of advanced interactive technique, combining physical and digital artifacts, aims at stimulating visitor about this kind of phenomenon. We designed an advanced interactive technique that explains the eutrophication phenomenon in order to facilitate its understanding and its background.

RESUME

Au cours de ces dernières années, un nombre important d'expériences interactives liées au domaine culturel, et notamment à la muséographie, ont vu le jour. L'objectif est de rendre le visiteur acteur de sa visite en lui proposant d'interagir directement avec des concepts. Ceux-ci sont difficiles à comprendre car ils sont complexes ou abstraits. L'utilisation d'une technique d'interaction avancée, combinant artefacts physiques et numériques, a pour objet de stimuler le visiteur sur ce type de phénomène complexe. Nous avons conçu une technique d'interaction avancée qui vulgarise le phénomène d'eutrophication dans le but de faciliter sa compréhension et ses origines.

Categories and Subject Descriptors: H.5.2. [User Interface]: *Prototyping/ Evaluation/Methodology/ User-centered design/ Theory and methods/*.

General Terms

Design, Experimentation, Human Factors.

Keywords

Mixed Interactive Systems, Advanced Interactive Experience, co-design, museology, eutrophication

1. INTRODUCTION

L'enjeu actuel des musées est de rendre l'expérience culturelle interactive de façon à engager les visiteurs dans leur visite. Les

visiteurs deviennent ainsi acteurs de leur propre expérience, et ne restent plus seulement passifs devant des illustrations, des objets ou des textes. Des outils comme des guides interactifs [5] ont été introduits pour fournir des informations numériques supplémentaires aux visiteurs. Des jeux [5], [7] ont également été proposés pour favoriser l'apprentissage et développer l'intérêt des visiteurs pour les expositions. Des systèmes tels que des Systèmes Interactifs Mixtes (SIM) ont été développés pour également rendre l'expérience des visiteurs interactive. Les SIM sont une forme de systèmes ubiquitaires qui regroupent les systèmes de réalité mixte ou augmentée et les interactions tangibles, telles GeoTUI [1] ou des systèmes destinés aux musées [3]. Les SIM prennent avantage des objets physiques pour interagir avec des entités numériques ; ces objets quotidiens, utilisés ici pour communiquer avec un système, constituent alors une véritable extension de l'utilisateur. Les SIM rejoignent la définition des systèmes ubiquitaires fournies par Weiser [6] dans le fait de pouvoir manipuler des objets familiers intégrés au contexte et permettant de se focaliser sur le but de l'interaction et non l'interaction elle-même. Par ailleurs, ce type de préoccupations fait écho avec les recommandations muséographiques récentes fournies par Wagensberg [4], qui conseille de placer les objets au centre des expositions des musées.

La problématique du projet est de proposer la manipulation d'objets du quotidien afin de vulgariser le phénomène d'eutrophication pour stimuler les visiteurs sur cette thématique. Celle-ci est un concept complexe (i.e. système en équilibre dynamique au cours du temps) à transmettre aux visiteurs, qui ne sont ni en mesure de se représenter mentalement ce concept, ni d'en faire concrètement l'expérience. Il faut donc proposer un outil de communication interactif adapté à la fois à la complexité de la thématique et aux ressources cognitives des visiteurs. Dans ce cadre, il s'agit bien d'apporter à l'utilisateur une solution interactive avancée par le biais de Systèmes Interactifs Mixtes (SIM). Les SIM sont ainsi intéressants à mettre en place dans un musée car ils permettent la manipulation d'objets physiques, les plaçant au cœur de la visite. Leur avantage réside dans le fait de pouvoir sortir du schéma habituel des expositions muséales où les objets de la visite sont seulement exposés.

2. CONSIDÉRATIONS MUSEOGRAPHIQUES

Une collaboration a été mise en place avec le Muséum d'Histoire Naturelle de Toulouse afin de sensibiliser les visiteurs sur le phénomène d'eutrophication. Celui-ci est la rupture d'un équilibre dynamique, rendant un lac inapte à la vie aquatique, par un enrichissement de l'eau en nutriments conduisant à un

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'11, Juin 6-8, 2011, Toulouse, France.

Copyright 2011 ACM 1-58113-000-0/00/0010...\$10.00

comblement par une accumulation de biomasse. Les eaux s'enrichissent en sels minéraux et si le phénomène est trop important, les algues prolifèrent, assimilant ces matières nutritives.

Le comblement est un processus naturel amenant un lac à mourir. Ce processus se déroule sur une longue période, comme une centaine d'années. Les visiteurs ne peuvent donc pas faire l'expérience de ce comblement. Par ailleurs, celui-ci est accéléré par des phénomènes biologiques complexes et l'activité humaine joue un rôle sur l'eutrophisation. Elle peut soit accélérer le comblement, avec par exemple l'ajout de désherbant ou d'engrais, soit le ralentir, grâce au curage ou à la surveillance de l'équilibre de l'écosystème. L'utilisation d'un SIM semble être une solution adéquate pour répondre aux considérations muséographiques. En effet, le comblement du lac peut être simulé par un environnement numérique 3D et le phénomène peut être représenté sur une longue période. De plus, des objets physiques peuvent être manipulés par les visiteurs de façon à observer ensuite l'impact de leurs activités sur le lac numérique.

3. PROTOTYPE S3MP (Système Mixte en Milieu Muséal avec Palette)

3.1 Processus de conception

Pour concevoir ce système, nous avons mis en place un processus de co-conception dédié [1] aux SIM en milieu muséal. Ce processus est composé de quatre phases. La phase d'analyse préliminaire sert à analyser le domaine en recensant ses activités, ses contraintes, ses utilisateurs, etc. La phase d'analyse des principes interactifs vise à définir comment rendre interactif un des besoins du domaine identifié lors de la phase précédente, en considérant les contraintes inhérentes. La phase d'optimisation a pour objectif de concevoir l'interaction avec le système en tenant compte de l'analyse précédente et de dimensions complémentaires telles que la dimension sociale, pédagogique etc. La dernière étape conduit à l'industrialisation d'un des prototypes conçus et issus de la phase d'optimisation.

Ce processus est adapté car il prend en compte la multidisciplinarité des intervenants lors de la conception et facilite la communication entre ceux-ci. Il permet également de mettre l'accent sur les attentes muséales plutôt que sur des contraintes techniques.

3.2 Architecture

L'architecture peut ainsi être flexible et évolutive, telle une architecture basée sur ARCH. Elle est ainsi composée de 5 éléments qui sont le noyau fonctionnel qui est la source de données comprenant l'ensemble des données liées à des paramètres environnementaux du lac et à leurs évolutions. L'adaptateur au noyau fonctionnel modifie les données du noyau. Le contrôleur de dialogue interprète les événements et séquence les interactions en entrée et sortie. La présentation joue le rôle d'interface avec le visiteur et correspond à la scène 3D. Puis l'interaction prend en compte les événements du visiteur qui manipule les objets physiques.

Les différents composants ont été conçus de façon indépendante et communiquent entre eux via le bus logiciel Ivy.

3.3 Interactions proposées

Le visiteur a la possibilité de manipuler deux objets physiques : une palette et une figurine humaine. La palette présente les activités humaines que le visiteur peut sélectionner. Il peut ainsi

ajouter de l'engrais, pomper de l'eau ou nettoyer la vase. La figurine est la métaphore de l'humain agissant sur le lac. Les visiteurs manipulent l'humain pour sélectionner une activité sur la palette, ce qui a pour effet d'activer l'objet numérique correspondant dans l'environnement 3D. Les visiteurs manipulent ensuite la figurine pour placer cet objet dans la zone où l'activité humaine est réalisée. Les visiteurs interagissent directement avec l'objet numérique dans l'environnement 3D car cet objet suit les positions données par le visiteur à la figurine.

Une fois le positionnement de l'activité dans la zone de réalisation, les visiteurs peuvent observer les impacts de l'activité sélectionnée sur le lac 3D. Celui-ci est intégré à un environnement comprenant un jardin et un champ. Le visiteur peut également visualiser l'espérance de vie du lac ainsi que des paramètres environnementaux, tels le niveau d'eau, le niveau de vase, le taux de phosphate, le taux de biomasse, etc.

4. CONCLUSION

Le prototype S3MP a fait l'objet d'une étude pilote avec six utilisateurs de façon à savoir s'ils étaient stimulés par le système sur la thématique. Les premiers résultats sont encourageants et montrent que les utilisateurs ont compris que leurs activités avaient un impact sur l'environnement. Ils se sont sentis présents lors de l'interaction et ont apprécié leur expérience.

5. REFERENCES

- [1] Bach C., Salembier P. and Dubois E. 2006. Co-conception d'expériences interactives augmentées dédiées aux situations muséales. In Proceedings of IHM'06 (Canada). IHM'06. ACM Press, New York, NY, 11-18.
- [2] Couture, N. and Rivière, G.. 2006. *3è Journées Francophones Mobilité et Ubiquité, Démonstrations*, (Paris, France, 5-8 Septembre, 2006), pp. 163-164. ACM Press.
- [3] Hornecker, E. and Stifter, M. 2006. Learning from Interactive Museum Installations About Interaction Design for Public Settings. In Proceedings Australian Computer-Human Interaction Conference OZCHI'06.
- [4] Wagensberg, J. 2005. The "total" museum, a tool for social change. *História, Ciências, Saúde – Manguinhos*, v.12 (supplement), 309-21.
- [5] Wakkary, R., Hatala, M., Muise, K., Tanenbaum, K., Corness, G., Mohabbati, B. and Budd, J. 2009. Kurio: a museum guide for families. In Proceedings of the 3rd International Conference on Tangible and Embedded Interaction, ACM Press, New York, NY, 215-222.
- [6] Weiser, M. 1991. The computer for the 21st century. *Scientific American*, 3(265):94-104.
- [7] Yiannoutsou, N., Papadimitriou, I., Komis, V., and Avouris, N. 2009. "Playing with" museum exhibits: designing educational games mediated by mobile technology. In Proceedings of the 8th international Conference on interaction Design and Children (Como, Italy, June 03 - 05, 2009). IDC '09. ACM, New York, NY, 230-233.



Figure 1. S3MP : les deux objets physiques utilisés pour interagir.

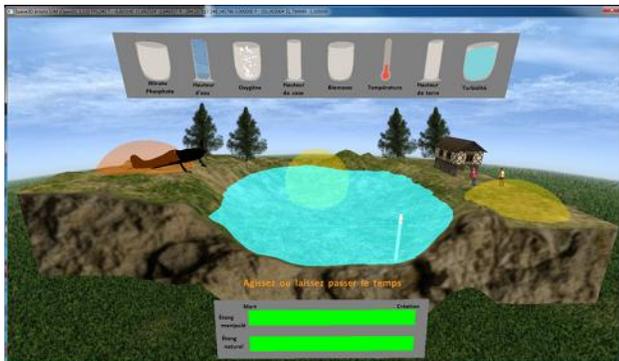


Figure 2. S3MP : la représentation numérique de l'environnement simulé.

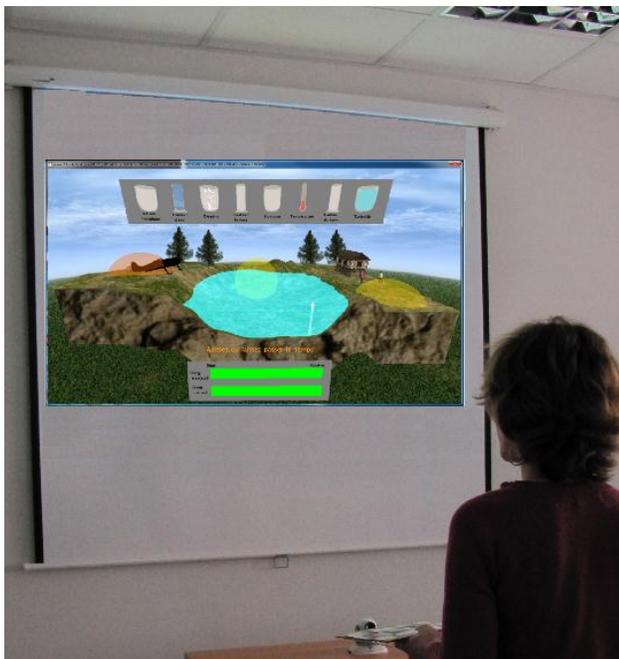


Figure 3 : S3MP : vue d'ensemble du système

Towards Bridging IoT and Cloud Services: Proposing Smartphones as Mobile and Autonomic Service Gateways

Position Paper

Roya Golchay
 Université de Lyon, INRIA
 INSA-Lyon, CITI
 F-69621, Villeurbanne, France
 roya.golchay@insa-lyon.fr

Frédéric Le Mouël
 Université de Lyon, INRIA
 INSA-Lyon, CITI
 F-69621, Villeurbanne, France
 frederic.le-mouel@insa-lyon.fr

Stéphane Frénot
 Université de Lyon, INRIA
 INSA-Lyon, CITI
 F-69621, Villeurbanne, France
 stephane.frenot@insa-lyon.fr

Julien Ponge
 Université de Lyon, INRIA
 INSA-Lyon, CITI
 F-69621, Villeurbanne, France
 julien.ponge@insa-lyon.fr

ABSTRACT

Computing is currently getting at the same time incredibly in the small with sensors/actuators embedded in our everyday objects and also greatly in the large with data and service clouds accessible anytime, anywhere. This Internet of Things is physically closed to the user but suffers from weak run-time execution environments. Cloud Environments provide powerful data storage and computing power but can not be easily accessed and integrate the final-user context-awareness. We consider smartphones are set to become the universal interface between these two worlds. In this position paper, we propose a middleware approach where smartphones provide service gateways to bridge the gap between IoT services and Cloud services. Since smartphones are mobile gateways, they should be able to (re)configure themselves according to their place, things discovered around, and their own resources such battery. Several issues are discussed: collaborative event-based context management, adaptive and opportunistic service deployment and invocation, multi-criteria (user- and performance-oriented) optimization decision algorithm.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Internet of Things*; C.2.4 [Computer-Communication Networks]: Distributed Systems—*Cloud computing*; D.2.11 [Software Engineering]: Software Architectures—*Service-oriented architecture (SOA)*

General Terms

Algorithms, Management, Performance, Reliability

Keywords

Cloud Computing, Internet of Things, Service-oriented Applications, Middleware, Smart and Autonomic Gateways, Smartphone

1. INTRODUCTION

Computing is currently getting at the same time incredibly in the small with sensors/actuators embedded in our everyday objects and also greatly in the large with data and service clouds accessible anytime, anywhere. This Internet of Things is physically closed to the user but suffers from weak run-time execution environments. Cloud Environments virtualize and provide powerful data storage and computing power but can not be easily accessed and integrate the final-user context-awareness.

In parallel with these run-time environment evolutions, the service-oriented programming and architectures are also important evolutions in software engineering. These paradigms allow to easily split applications, enable a great reuse and (sometimes) handle (some) communication protocol issues. New run-time challenges are still raised especially with IoT and Cloud environments such as service deployment mechanisms, service placement, service invocation scheduling, etc.

We consider smartphones are set to become the universal interface between Internet of Things and Cloud Computing worlds, as shown in Figure 1. In this position paper, we propose a middleware approach where smartphones provide service gateways to bridge the gap between IoT services and Cloud services. Since smartphones are mobile gateways, they should be able to (re)configure themselves according to their place, things discovered around, and their own resources such battery. Several issues are discussed to implement a smart and autonomic service gateway: collaborative event-based context management, adaptive and opportunistic

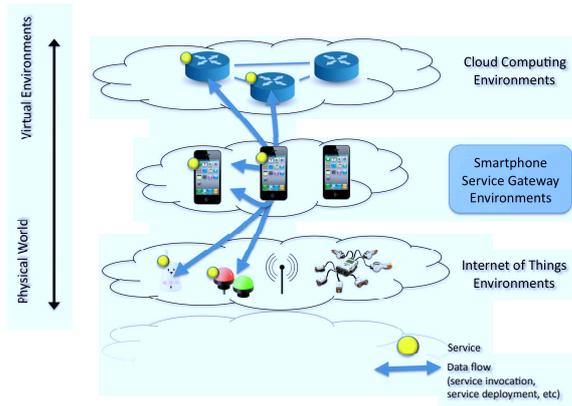


Figure 1: Smartphone Service Gateways Bridging IoT and Cloud Services

tic service deployment and invocation, multi-criteria (user- and performance-oriented) optimization decision algorithm.

The rest of this paper is organized as follow. Next section presents related works. An overview of our proposed solution is given in section 3 and smart and autonomic mechanisms are detailed in section 4. Finally the last section states our current and future works.

2. RELATED WORK

While talking about mobile environment the first thing that attracts attention is resource-poor devices compared with fixed stations. Mobile phones are limited in computation power, available memory and battery life. The wireless connections with different available bandwidth and probability of disconnection and reconnection are another important point in these environments. There is a considerable amount of research on how to address these mobility issues by migrating data and computation to distributed systems [12]. This issue is now raised by Internet of Things where things are even more constrained. To address the low energy power issue, [11] proposed using cloud computing. According to this article sending computation to the cloud will save energy on smart phone. The amount of saved energy is depends on the wireless bandwidth, the amount of data to be transmitted to cloud and also the application type if it is real-time data application or not. In order to use the cloud more efficiently, how to distribute the application is very important. Wishbone [13] is a system that takes a data-flow graph of operators and produces an optimal partitioning. The partitioning problem is to find a cut of data-flow graph, where its vertices are stream operators and its edges are streams. Edge and vertex weights represent bandwidth and CPU utilisation respectively. The partitioning algorithm models the program as an integer linear program that minimizes a linear combination of network bandwidth and CPU load. The notion of data-flow graph is also used in [8] where the presented middleware platform can automatically distribute different layers of an application between the phone and the cloud. In this graph the vertexes are software mod-

ules where service dependency between them represents by the edges. The optimal cut of the graph will minimizes or maximizes the objective function that can be the end-to-end interaction time between a phone and a server, the amount of exchanged data or complete the execution in less than a predefined time. The layered architecture is also considered in [9] where presented tenet architecture for tiered sensor networks networks simplified application development. This architecture composed of two tiers: a lower tier consisting of motes, which enable flexible deployment of dense instruction, and upper tier containing fewer, relatively less constrained nodes. Another three tiers management architecture is used in [7] where a service-oriented framework is introduced to simplify the development and run-time adaptive support of autonomic pervasive applications. A middleware is proposed to serve as a framework to host autonomic home applications. A distributed resource model and management tailored for deployment of adaptive services in a mobile environment is also discussed in [1]. This research is focuses on the resources directly surrounding one mobile user. The presented distributed resource management framework is used to monitor the context and dynamically respond to changes by selecting the most appropriate application variant which called self-adaption mechanism. Although the distributed powerful resources in the cloud seems useful to cover the resource constraints of small components in Internet of Things, absence of context-awareness in the cloud make it difficult to communicate with it for the users. Several bridging approaches exist [5] but a link is required to handle communication between almost physical environment of IoT and virtual environment of the cloud.

3. SOLUTION OVERVIEW

We consider service-oriented applications modeled as a graph where nodes are services and vertices are service dependencies. Two service dependencies are possible: static deployment dependency and dynamic run-time invocations.

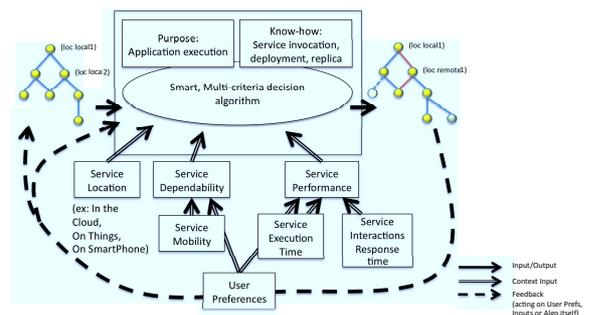


Figure 2: Smartphone Service Gateway - Conceptual Model

For the smartphone service gateway we propose, as shown in Figure 1, the major challenge consists in deploying, invoking, (and eventually replicating) services of an application by

integrating (i) existing light-services on communicating objects from IoT, (ii) mid-services on the smartphones closer to the end-users, (iii) heavy-services distributed in the Cloud.

The proposed conceptual model in Figure 2 has this service graph in input, applies a multi-criteria decision algorithm and realizes deployment/invocation/(replication) actions to schedule/coordinate the service choreography. The decision algorithm takes into account different context inputs extracted from the services (size, performance, etc), from platforms (mobility, location, etc) and from the user (preferences, profile, etc).

This process is autonomic because we think a smartphone has especially to be self-managed, without the user intervention. It is however important the decision algorithm to be user-oriented because a smartphone is really an end-user terminal. The process includes so a retro-action loop from service uses/monitoring feedbacks. This retro-action loop allows to automatically modify the user profile, to change the algorithm inputs by reannotating the service graph, or to adapt the decision algorithm it-self.

4. SMART AND AUTONOMIC GATEWAY MECHANISMS

After analysing the properties of autonomic systems, we propose several mechanisms implemented in a service-oriented middleware approach (cf Figure 3).

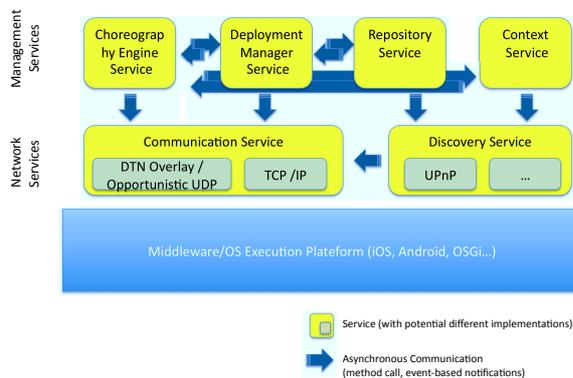


Figure 3: Smartphone Service Gateway - Architecture Overview

Aware property

1. **Event-based context management:** The context management is crucial in an autonomous system while these inputs are conditioning the decision. In our approach, we focus the context monitoring on services: their performance, location, mobility, etc. In very highly changing environments such as IoT and Cloud ones, to have a constantly reevaluating algorithm, we use a *Listener/Event Notifier* approach for each changes. Pertinently qualifying the moment to send an event is critical on a smartphone because a

too active monitoring leads to a battery run off and a too passive monitoring do not produce a accurate view of the physical world. We will examine context aggregation and filtering such as in [14].

2. **Collaborative services repositories:** Services running on a smartphone can be available for other smartphones in vicinity. Each smartphone holds a local *Repository Service* that can be discovered by using different discovery protocols. Smartphones can then collaborate, for example, to use all the same service and reduce battery consumption of participants. Aimed collaborative approaches are not those of distributed systems with strong consensus or transaction problems, either are geographically-closed and short-lived collaborations with only local stability.

Adaptive property

3. **Adaptive, delayed and opportunistic service deployment and invocation:** Deployment and coordination mechanisms need to be adaptive to allow migration or call redirection according to smartphones mobility and/or disconnections. We will use a DTN communication stack implementation (Disrupted-Tolerant Network) in the *Communication Service* to provide on-demand deployment and most of all - spontaneous and opportunistic deployment such as in [10]. The opportunistic deployment can for instance be implemented by caching services prioritarily from well-known users or from users sharing common interests such as in Social Networks.

Automatic property

4. **Smart multi-criteria decision algorithm:** Our Service Gateway bridging several worlds - IoT, Cloud, Smartphones and End-users -, the decision algorithm implemented in the *Choreography Engine Service* has to consider contextual events from the physical environment such as the service location or mobility, from the virtual environment such as the service execution performances, from platforms such as the battery level and from user such as the service use statistics. This problem of solving multiple constraints is NP-complete but several approaches seems very interesting for mobile environments such as approximation algorithms for constrained knapsack problems [6] or biologically-inspired algorithms [15, 2]. Using the feedback control loop, we will particularly enrich them with a smart learning of the user profile.

5. CURRENT AND FUTURE WORK

A prototype - AxSeL¹, A conteXtual Service Loader - was implemented based on a Felix², one Java implementation of the OSGi service-oriented specification³ [3, 4]. This prototype can run on constrained devices and locally manages

¹<http://amazones.gforge.inria.fr/>, INRIA Amazones Team Forge

²<http://felix.apache.org/>

³<http://www.osgi.org/>

service-oriented application deployment according to hardware resources. It has been tested with a PDF reader application.

As immediate extensions, we will improve the prototype and focus on Smart Building domain with the INSA Lyon project: "The Smart Chappe Building: a Platform for Contextual Services"⁴. This platform will allow us to demonstrate the integration and detail (i) the interaction between smartphones and temperature/presence sensors in the building, (ii) the interaction between smartphones and static Wifi hotspots and/or mobile service gateways and (iii) the storage and computation migration of sensed data collected in the building to a cloud.

6. REFERENCES

- [1] M. Alia, S. Hallsteinsen, N. Paspallis, and F. Eliassen. Managing distributed adaptation of mobile applications. In *Proceedings of the 7th IFIP WG 6.1 international conference on Distributed Applications and Interoperable Systems*, DAIS '07, pages 104–118, Berlin, Heidelberg, 2007. Springer-Verlag.
- [2] E. Altman, P. Dini, D. Miorandi, and D. Schreckling, editors. *Paradigms for Biologically-Inspired Autonomic Networks and Services – The BIONETS Project eBook*. BIONETS Consortium, 2010. Deliverable D0.2.3.
- [3] A. Ben Hamida, F. Le Mouël, S. Frénot, and M. Ben Ahmed. A graph-based approach for contextual service loading in pervasive environments. In *Proceedings of the 10th International Symposium on Distributed Objects and Applications, DOA '2008*, volume 5331 of *Lecture Notes in Computer Science*, pages 589–606. Springer Verlag, Monterrey, Mexico, Nov. 2008.
- [4] A. Ben Hamida, F. Le Mouël, S. Frénot, and M. Ben Ahmed. Déploiement adaptatif d'applications orientées services sur environnements contraints. *Computer Science and Technology / Technique et Science Informatiques*, 30(1):59–91, 2011.
- [5] T. Bissyandé, L. Réveillère, Y.-D. Bromberg, J. Lawall, and G. Muller. Bridging the gap between legacy services and web services. In *Proceedings of the ACM/IFIP/USENIX 11th international conference on Middleware, Middleware '10*, volume 6452 of *Lecture Notes in Computer Science*, pages 273–292. Springer Berlin / Heidelberg, 2010.
- [6] G. Borradaile, B. Heeringa, and G. T. Wilfong. Approximation algorithms for constrained knapsack problems. *CoRR*, abs/0910.0777, 2009. Revised version 2 (Feb 2010).
- [7] J. Bourcier, A. Diaconescu, P. Lalanda, and J. A. McCann. AutoHome: An autonomic management framework for pervasive home applications. *ACM Trans. Auton. Adapt. Syst.*, 6:8:1–8:10, February 2011.
- [8] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso. Calling the cloud: Enabling mobile phones as interfaces to cloud applications. In *Proceedings of the ACM/IFIP/USENIX 10th international conference on Middleware, Middleware '09*, volume 5896 of *Lecture Notes in Computer Science*, pages 83–102. Springer Berlin / Heidelberg, 2009.
- [9] O. Gnawali, K.-Y. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th international conference on Embedded Networked Sensor Systems, SenSys '06*, pages 153–166, New York, NY, USA, 2006. ACM.
- [10] F. Guidec, N. L. Sommer, and Y. Mahéo. Opportunistic software deployment in disconnected mobile ad hoc networks. *IJHCR*, 1(1):24–42, 2010.
- [11] K. Kumar and Y.-H. Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, april 2010.
- [12] F. Le Mouël, F. André, and M. Segarra. AeDEn: An adaptive framework for dynamic distribution over mobile environments. *Annals of Telecommunications*, 57(11–12):1124–1148, Nov. 2002.
- [13] R. Newton, S. Toledo, L. Girod, H. Balakrishnan, and S. Madden. Wishbone: profile-based partitioning for sensor network applications. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI '09*, pages 395–408, Berkeley, CA, USA, 2009. USENIX Association.
- [14] D. Riboni and C. Bettini. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.
- [15] H. Wada, J. Suzuki, Y. Yamano, and K. Oba. Evolutionary deployment optimization for service-oriented clouds. *Softw. Pract. Exper.*, 41:469–493, April 2011.

⁴<http://telecom.insa-lyon.fr/content/plateforme-smart-chappe-building>

Construction Bottom-up d'applications ambiantes en environnements partiellement connus à priori

Christophe Vergoni
GFI Informatique et I3S
930 Route des Colles - BP 145
06903 Sophia-Antipolis France
vergoni@polytech.unice.fr

Gaëtan Rey
I3S (UNSA - CNRS)
930 Route des Colles - BP 145
06903 Sophia-Antipolis France
gaetan.rey@unice.fr

Jean-Yves Tigli
I3S (UNSA - CNRS)
930 Route des Colles - BP 145
06903 Sophia-Antipolis France
tigli@polytech.unice.fr

Stéphane Lavirotte
I3S (UNSA - CNRS)
930 Route des Colles - BP 145
06903 Sophia-Antipolis France
stephane.lavirotte@unice.fr

ABSTRACT

La construction d'applications comme composition de services est largement utilisée en Informatique Ambiante. Traditionnellement, ces applications sont spécifiées au moment du développement sous forme de workflow enrichis de politiques d'adaptation. Elles sont rendues effectives par une architecture logicielle capable de rechercher et coordonner l'ensemble des services nécessaires. Toutefois, dans le cadre d'environnements complètement dynamiques, ces approches se heurtent à une forte méconnaissance et une "imprévisibilité" quant à l'environnement cible dans lequel évoluera l'application. Dans un premier temps, nous étudierons l'impact sur le design et sur l'architecture logicielle d'un environnement partiellement connu à priori et fortement dynamique. Puis, dans un second temps, nous déduirons de l'analyse précédente une approche basée sur une méthodologie ascendante (dite Bottom-up) dans le design et la construction de telles applications.

Keywords

Ubiquitous computing, service oriented architecture, service composition, mobile Ad Hoc network, bottom-up engineering.

1. INTRODUCTION

Notre environnement quotidien se voit de plus en plus peuplé d'artefacts électroniques communiquant. Des dispositifs mobiles, aux capteurs embarqués augmentés de réseaux ubiquitaires, l'informatique ambiante (IAm) se propose de fournir à l'utilisateur un environnement informatisé uniforme et pertinent s'appuyant sur cet ensemble hétérogène de technologies. Dans cet objectif, l'utilisation de services et la

construction d'applications comme composition de services est largement utilisée [2]. Du domaine des systèmes distribués, les services permettent d'encapsuler des données, ressources, ou fonctionnalités en entités logicielles découpées, réutilisables et interopérables grâce à des descriptions déclaratives et des protocoles standardisés. Les problématiques d'interopérabilité technologique ainsi abstraite, l'IAm peut se concentrer sur des problématiques liées à la mobilité de l'utilisateur ou des dispositifs, à la prise en compte du contexte, ou encore à l'ambition d'une technologie discrète [20].

L'IAm peut être envisagée comme le décrit [14] dans des environnements physiques différents. On peut, par exemple, distinguer les environnements physiquement bornés comme une maison ou le hall d'une gare, des environnements complètement mobiles centrés cette fois sur l'utilisateur plutôt que sur un lieu particulier. Dans cette deuxième catégorie d'environnements dits mobiles, l'infrastructure d'une application gravite autour de l'utilisateur [14] et se forme de manière spontanée ou Ad Hoc [3]. Nous pouvons nommer ce type d'environnements *mobile Ad Hoc* ou *sans infrastructure fixe* [3].

La projection des principes de l'IAm dans ces environnements mobiles Ad Hoc à un impact sur les trois phases du cycle de vie d'une application. La phase de développement où un développeur crée ou design, maintiens ou améliore une application. La phase de déploiement où l'application est mise en oeuvre ou synthétisée par l'architecture logicielle. Et enfin, la phase d'exécution où l'utilisateur utilise l'application qui s'exécute [1, 5].

En architecture orientée services, la construction d'une application s'articule autour de trois composantes. Les services(i), comme briques de bases présentes dans l'environnement, la spécification déclarative(ii) sous forme de langages ou modèles de l'application, et les systèmes ou architectures logicielles(iii) capable de synthétiser à l'exécution et automatiquement l'application décrite [13]. Le design d'une application d'IAm dans ce cadre reviens donc à décrire une spécification de l'application qui sera mis en oeuvre lors de la phase de déploiement comme une composition de services présents dans l'environnement [2]. Cette spécification peut être aussi explicite que la description de l'assemblage des services entre eux (l'orchestration, workflow statique) ou plus

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

abstrait et mise en œuvre par des techniques telle que l'AI planning (workflow dynamique) [13],[10].

L'impact d'un environnement mobile Ad Hoc sur l'architecture logicielle est largement traité dans la littérature [1, 3, 18, 14]. La décentralisation de l'architecture, la composition dynamique automatisée et la gestion de l'éventualité sont traditionnellement prônés. En revanche, l'impact sur la phase de design reste un sujet, à notre connaissance, très peu traité.

Traditionnellement, les applications sont designées au moment du développement pour répondre à des besoins préalablement identifiés. La construction d'applications par composition de services en IAM reprend ce même schéma. Ces constructions suivent généralement une approche méthodologique descendante ou Top-down. Sur des besoins identifiés, on spécifie une application donnée. Cette spécification globale est ensuite décomposée, au moment du déploiement, en sous problèmes jusqu'à l'obtention de tâches réalisables dans l'environnement présent. Ces tâches réalisables sont la sélection de tous les services de l'environnement nécessaires à la réalisation de l'application globale décrite. Cette sélection s'appuie sur des mécanismes de recherche et découverte de services aidés d'algorithmes de matching entre les spécifications de l'application d'une part et la description des services de l'environnement d'autre part [21]. Ces approches descendantes ou Top-down sont adaptées à la mise en œuvre d'une application donnée en réponse à un besoin spécifique, défini à priori pour un type d'utilisateur dans un environnement physique donné.

Dans les environnements complètement mobiles, la mobilité de l'utilisateur comme celle des entités de l'infrastructure d'une éventuelle application sont un frein à l'identification d'un besoin précis et complet. En effet, chaque entité informatisée, dispositif ou capteur, est mobile et évolue suivant une dynamique, à l'exécution, qui lui est propre. L'infrastructure matérielle et logicielle d'une application est donc fortement dynamique et imprévisible. L'hypothèse de l'existence d'une entité particulière à un moment donné, ne peut donc être formulée. De plus, la mobilité de l'utilisateur rajoute encore un degré d'incertitude à l'environnement cible. La prise en compte du contexte dans un processus automatisé de construction de l'application [19] ne peut pas non plus être envisagé complètement. D'une part, la perception de l'environnement est permise sous conditions de l'existence d'un ensemble de capteurs dans ce même environnement. La forte variabilité qui caractérise l'infrastructure dans ce type d'environnement invalide cette première hypothèse. D'autre part, la combinatoire des configurations possibles de l'environnement est trop importante pour que soit envisagée, à priori, la spécification des réactions d'adaptations [1, 19]. La spécification d'une application d'IAM dans ce type d'environnement, partiellement connu à priori et fortement variable, empêche une identification précise et complète des besoins. La spécification d'une application pouvant y répondre ne peut donc pas non plus être envisagée de manière complète.

En partant de la problématique du design des applications d'IAM dans ces environnements mobiles et Ad Hoc nous avons appliqué les principes d'une méthodologie Bottom-up dans la spécification et la construction d'une application d'IAM. La contribution de ce papier est double. Pour des environnements dits mobiles, particulièrement dynamique et imprévisible, d'une part la proposition de la spécification

d'une application menée suivant une méthodologie ascendante. D'autre part, la modification de l'architecture SLCA [17] actuelle pour répondre aux besoins de décentralisation de l'architecture logicielle. Nous détaillerons dans une première section, les caractéristiques d'un environnement fortement dynamique et imprévisible, qui représente le cadre de nos travaux. A cette occasion nous expliciterons d'abord les problématiques soulevées puis leurs impacts sur la phase de design et sur l'architecture logicielle supportant le déploiement et l'exécution d'une application. Dans la section suivante, nous détaillerons l'approche mise en œuvre pour la construction ascendante d'une application dans ces environnements. Nous expliciterons ensuite dans une nouvelle section les avantages et faiblesses de notre approche pour finir par détailler les perspectives envisagées et conclure.

2. CADRE ET PROBLÉMATIQUES

La projection des principes de l'IAM dans des environnements différents à un impact sur l'architecture logicielle de ces applications [14]. Le cadre dans lequel nous proposons de placer nos travaux est un environnement mobile Ad Hoc. Ce cadre particulier soulève des problématiques ayant un impact sur les architectures logicielles capables de synthétiser les applications [3], mais aussi sur la phase de design d'une application. Cette section détaille, dans un premier temps, le cadre d'un environnement mobile Ad Hoc pour se concentrer ensuite sur les problématiques soulevées par la projection des principes de l'IAM dans ces environnements.

2.1 Environnement mobile Ad Hoc

Comme énoncé précédemment, deux grands types d'environnements cibles peuvent être distingués. Les environnements qui possèdent une infrastructure plutôt fixe, liés à un lieu physique particulier et les environnements sans infrastructure fixe, cette fois plutôt centrés sur l'utilisateur [14]. L'IAM dans ces environnements dits physiquement bornés permet d'enrichir le lieu de comportements intelligents [14]. Ainsi, l'espace délimité par un appartement ou un musée peut se voir augmenté de comportements d'IAM. Dans ces environnements, un certain nombre de dispositifs, capteurs ou actuators sont déployés et l'infrastructure réseau, matérielle comme logicielle est majoritairement fixe. Le reste de l'infrastructure peut être mobile, par exemple porté par l'utilisateur, et intégré par l'architecture logicielle de l'environnement. Cette intégration peut être effectuée suivant la localisation des entités mobiles déterminée par les différents capteurs présents dans l'environnement.

Le développement d'applications d'IAM dans ces environnements fixes peut faire l'hypothèse de l'utilisation de telle ou telle partie de l'infrastructure dans un but particulier. On peut par exemple supposer l'existence d'un nœud d'exécution capable de supporter la coordination de tous les services en jeu dans une application. De la même manière, un ensemble de capteurs disséminés dans l'environnement et connus au moment du développement, peut être pris en compte pour envisager les variations de l'application en fonction de l'environnement perçu. L'IAM s'attache, dans ces environnements physiquement bornés, à offrir à l'utilisateur un ensemble fini d'applications spécifiques proposées par l'environnement physique lui-même. On peut parler d'environnements intelligents [14].

Les environnements mobiles Ad Hoc sont quant à eux centrés sur l'utilisateur plutôt que sur un lieu particulier. Les

dispositifs ne sont plus liés à un environnement physique donné mais sont plutôt mobiles. Chaque entité, dispositif ou capteur possède une dynamique qui lui est propre. L'infrastructure d'une éventuelle application se forme de manière spontanée entre l'ensemble des entités qui gravitent autour de l'utilisateur. La communication de ces entités mobiles est rendue possible par des réseaux mobiles spontanés ou Ad Hoc [12].

Ces environnements mobiles Ad Hoc sont caractérisés par :

- Une infrastructure de topologie fortement *variable et imprévisible* due à des entités possédant une *dynamique propre* [3].
- Un environnement physique *imprévisible et partiellement accessible* dû à des capteurs potentiellement inexistant [1].
- Un caractère *non borné*. L'ensemble des entités de l'infrastructure, les utilisateurs, l'environnement physique et ses situations contextuelles sont potentiellement non bornés [1].

Ces caractéristiques additionnées à la mobilité de l'utilisateur entraînent une forte méconnaissance de l'environnement cible au moment du développement d'une application. De plus, le caractère non borné de ces environnements accentue d'autant plus ce phénomène en rendant incertain, à un moment donné, l'environnement (physique, logique, matériel) ou les utilisateurs en jeux. D'autre part, le nombre potentiellement important d'entités et l'impossibilité de supposer l'existence de nœuds fixes et robustes dans l'infrastructure pour la coordination de l'application soulèvent des problématiques propres à l'architecture logicielle.

2.2 Problématiques propres à l'architecture

Le cadre d'un environnement mobile Ad Hoc fait naître des problématiques propres à l'architecture logicielle. D'une part, la forte variabilité des entités de l'infrastructure a un impact sur l'architecture logicielle. D'autre part, l'hypothèse d'une infrastructure connue et robuste capable de supporter l'éventuelle charge ne pouvant pas être formulée, la gestion des services comme leur coordination ne peuvent être effectuées de manière centralisée.

2.2.1 Forte variabilité

La forte dynamique et le caractère imprévisible des entités qui composent l'infrastructure entraîne le besoin, dans l'architecture logicielle, d'une gestion de la disparition et l'apparition impromptue de ces entités [2]. L'architecture logicielle doit donc mettre en œuvre des comportements robustes dans la coordination et le remplacement éventuel d'entités pour maintenir au mieux l'application effective.

2.2.2 Centralisation impossible

L'architecture logicielle doit être capable de supporter le déploiement et l'exécution d'une ou plusieurs applications pouvant faire appel à un nombre potentiellement important d'entités. La gestion de l'infrastructure de services doit donc être capable de gérer la forte variabilité et la charge potentiellement importante. Traditionnellement, dans des environnements mobiles Ad Hoc, cette gestion est supportée par des services de découverte [21]. Un annuaire décentralisé ainsi que la gestion de l'apparition ou de la disparition spontanée des services peuvent aussi être mise en œuvre par ces systèmes.

Enfin, la coordination de l'ensemble des services en jeu

dans une application ne peut pas être déléguée à un nœud donné de l'infrastructure a priori. Elle doit donc être rendue possible spontanément malgré la forte variabilité des entités et de manière décentralisée pour éviter goulot d'étranglement et point central de défaillance [3, 14].

2.3 Problématiques propres au design

Le caractère fortement variable et imprévisible de l'infrastructure, ajouté à l'incertitude et au caractère partiellement accessible de l'environnement physique, a un impact direct sur la phase de design d'une application. En premier lieu, la faible connaissance a priori de l'environnement dans lequel déployer une application est un frein à une phase d'analyse des besoins traditionnelle et donc un frein au développement de l'application qui en résulte. D'autre part, toutes les configurations possibles de l'infrastructure ou toutes les situations contextuelles ne peuvent pas être envisagées au développement à cause de la combinatoire trop importante des solutions possibles. Enfin, le caractère non borné d'un tel environnement s'oppose à une phase de design finie ayant pour but une application monolithique.

2.3.1 Connaissance faible de l'environnement

D'abord, l'environnement physique, fortement dépendant de la mobilité de l'utilisateur, ne peut qu'être partiellement envisagé. La spécification d'un besoin complexe et précis dans un environnement aussi incertain est donc difficile. Ensuite, la mobilité et l'imprévisibilité qui caractérise l'environnement logique et matériel entraînent une forte méconnaissance de l'infrastructure logicielle et matérielle effective à un moment donné. Cette forte incertitude est donc un frein au design d'une application. Enfin, la spécification exhaustive des réactions aux changements de l'environnement peut difficilement être établie a priori [1, 19]. En premier lieu, parce que l'environnement physique est partiellement accessible dû à l'existence incertaine des capteurs nécessaires à sa perception. Les réactions peuvent être établies mais risquent de ne pas s'opérer. Ensuite parce que dans ces environnements partiellement connus a priori, il est difficilement envisageable de spécifier des réactions précises et pertinentes. Dans ces conditions, le design d'une application complète répondant à un ensemble de besoins difficilement identifiable ne peut pas être envisagé.

2.3.2 Combinatoire trop importante

Le nombre potentiellement important d'entités de l'infrastructure et la forte variabilité qui les caractérisent entraînent une combinatoire trop importante pour que soit envisagée la spécification a priori de toutes les configurations possibles. De la même manière, le caractère partiellement accessible de l'environnement physique [1] et sa forte méconnaissance a priori empêche la spécification exhaustive des situations contextuelles et réactions associées. Toutes les réactions aux changements de contexte physique comme logique ne peuvent donc pas être déterminées a priori [1, 19].

2.3.3 Environnement non borné

Ces environnements étant non bornés en terme d'entités, d'utilisateurs ou de situations contextuelles, l'apparition continue de nouveaux besoins doit pouvoir être pris en compte. Le développement d'une application ne peut donc pas s'appuyer sur un processus de design unique, borné en temps et ayant pour but une application satisfaisant un

unique objectif, complet et monolithique.

Enfin, dans ces environnements mobiles Ad Hoc, le design d'une application comme un processus fini est incompatible avec le caractère non borné du cadre dans lequel elles évoluent. De plus, la méconnaissance à priori de l'environnement physique et logique/matériel dans lequel sera déployée l'application empêche l'identification d'un ensemble de besoins spécifiques. Les approches méthodologiques descendantes ou Top-down traditionnelles ne sont pas correctement adaptées car elles proposent un processus de raffinement d'un objectif identifié, complet et fini. Dans ces environnements, de nouveaux besoins peuvent apparaître continuellement et devraient pouvoir être pris en compte en enrichissant l'application de nouveaux comportements. Cet enrichissement devrait pouvoir être mené de manière incrémentale et dynamique sans que l'application ne soit stoppée puis redémarrée [9]. Enfin, la combinatoire trop importante des variations de l'environnement empêche la spécification des réactions d'adaptation à l'environnement physique comme logique ou matériel. Ainsi, le design d'applications d'IAM, dans le cadre d'un environnement mobile Ad Hoc, telle une planification descendante ou Top-down de la composition de services n'est pas envisageable.

3. NOTRE APPROCHE ASCENDANTE

Notre approche suppose la construction d'une application de manière ascendante ou Bottom-up. Ainsi, au lieu de spécifier l'application comme un workflow spécifique, l'application est décrite à travers un ensemble de règles indépendantes mise en œuvre de manière tardive par des entités autonomes.

3.1 SLCA et Aspects d'Assemblage

3.1.1 Orchestration locale

L'architecture logicielle de notre approche met en œuvre une orchestration de services à travers un assemblage de composants. Les composants représentent soit des proxies de services logiciels, soit des composants logiciels traditionnels. Le flow d'exécution, entre services logiciels (sous forme de proxy dans l'assemblage) et composants logiciels, est donc supporté par un container de composants. Le modèle de composants utilisé LCA (Lightweight Component Architecture) [17] est un modèle dérivé des beans. Il décrit les composants logiciels comme un ensemble d'entrées, de sorties et de propriétés publiques. Les entrées représentent des méthodes alors que les sorties représentent des événements. Un assemblage de composants dans ce modèle est donc une description du flow d'événements entre sorties des uns et entrées des autres. D'autre part, le container qui supporte l'exécution de l'assemblage permet la modification dynamique de cet assemblage. On peut, par exemple à l'exécution, supprimer une liaison ou un composant, créer une nouvelle liaison ou instancier un nouveau composant. L'orchestration de services pouvant être décrite devient donc dynamiquement modifiable.

Par soucis de réutilisation des orchestrations construites, nos containers peuvent s'exporter eux même en tant que services dits composites (Fig. 1). On parle alors du modèle d'architecture SLCA [17]. Les services dans ce modèle proposent, comme pour le modèle LCA, des événements, des propriétés publiques et des méthodes. Chaque service composite possède deux interfaces. L'interface fonctionnelle du container

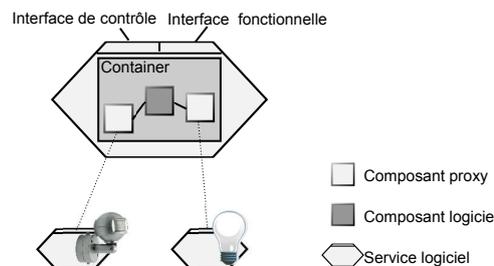


Figure 1: Container LCA encapsulé en tant que service

qui permet à travers des sondes et des puits, l'export d'événements de l'assemblage ou l'externalisation de méthodes internes. Et l'interface de contrôle qui permet d'exporter, sous forme d'un service, les capacités du container à modifier dynamiquement l'assemblage dont il est responsable.

Le modèle d'architecture SLCA décrit aussi une infrastructure Ad Hoc de services. L'infrastructure propose donc la découverte spontanée et une gestion des services sans annuaire centralisé. L'implémentation du modèle SLCA est aujourd'hui rendue possible par la plateforme WComp. WComp propose donc un container de composants LCA et l'export des interfaces fonctionnelles et de contrôles en tant que services d'un réseau spontané [12]. Le choix du service de découverte UPnP¹ dans cet objectif a été motivé notamment par la stabilité de la pile Intel² utilisée. La découverte spontanée et la gestion de l'infrastructure de services décentralisée s'appuie donc sur l'infrastructure de services UPnP.

SLCA définit aussi une approche multi-design qui décrit la capacité donnée à une ou plusieurs entités externes de créer ou modifier l'assemblage interne d'un container. Ceci est rendu possible à travers l'interface de contrôle du container. Ces entités externes sont nommées des *designer*.

3.1.2 Spécification de l'application

Le processus de spécification d'une application a pour objectif une description déclarative d'un assemblage de composants. Comme énoncé précédemment, les composants sont soit logiciels soit des proxies vers des services présents dans l'environnement. On spécifie ainsi une orchestration de services comme un assemblage de composants. Dans notre approche ce processus n'est pas borné en temps et peut être mené de façons indépendantes ou conjointes par différents développeurs.

La spécification est décrite par un ensemble de règles déclaratives et indépendantes. Ces règles peuvent être spécifiées au moment du développement et être mises en œuvre lors du déploiement ou bien être déployées dynamiquement de manière incrémentale dans une application existante.

Nos règles reprennent la métaphore des aspects [8]. Traditionnellement, la programmation orientée aspect permet d'externaliser, d'un code source, des préoccupations dites non fonctionnelles. L'ensemble du code responsable de la sécurité ou la journalisation peut, de cette manière, être séparé du code source fonctionnel. Le développeur peut ainsi se concentrer sur le caractère fonctionnel du code sans se soucier des préoccupations non fonctionnelles. Dans une phase

1. Universal Plug n Play. <http://www.upnp.org/>
2. <http://opentools.homeip.net/dev-tools-for-upnp>

de tissage préalable à la compilation, l'intégration ou le *tissage* des aspects est automatisé. Pour cela, les aspects décrivent à la fois, le code qui devra être tissé (*greffon*) et une expression régulière décrivant l'endroit dans le code où il doit être tissé (*point de coupe*). L'endroit effectif où l'aspect sera tissé est le *point de jonction*.

Le processus de spécification a donc pour objectif la production d'aspects appelés *Aspects d'assemblage* (AA). Ces aspects ne s'appliquent pas sur du code source mais sur un assemblage de composants. Les greffons sont eux même des sous-assemblages de composants. Les points de coupe, quant à eux, décrivent sur quels composants de l'assemblage le tissage d'un AA donné sera effectué. L'*identification des points de jonction* est le mécanisme qui détermine, pour un AA donné, sur quels composants d'un assemblage initial sera tissé un sous-assemblage. L'expression des points de coupe autorise l'utilisation des types de composants ou des meta-données sémantiques. On peut ainsi exprimer un sous-assemblage reliant tout type d'éclairage à un interrupteur donné (Fig. 2).

```
//Pointcut :
CapteurPresence = system.proximitySensor*
Eclairage = system.light*

//Greffon :
Advice EclairageAutomatique (
    CapteurPresence, Eclairage):
seuil : system.seuil
CapteurPresence.^proximityDetected -> (
    seuil.inputValue)
seuil.^valueChanged -> (Eclairage.on)
```

Figure 2: Aspect d'assemblage

3.1.3 Construction dynamique

Le processus de construction d'une application met en jeu trois entités. Ces entités représentent notre architecture logicielle capable de synthétiser l'application décrite sous forme d'AA :

- Le container.
- L'*UPnP designer*.
- Le *tisseur* d'AA.

Le container est l'entité qui supporte l'exécution de l'assemblage de composants. Comme énoncé précédemment, il permet à travers l'interface de contrôle de rendre accessible à d'autres entités sa capacité de modifier dynamiquement l'assemblage. L'*UPnP designer* comme le *tisseur* d'AA vont tous deux utiliser cette interface.

L'*UPnP designer* est un point de contrôle UPnP. Pour tous les services qui apparaissent spontanément dans l'environnement, un proxy correspondant est généré dans un container donné. Dans un premier temps, le designer crée automatiquement un nouveau type de composant représentant le proxy du service apparu grâce au contrat fourni lors du processus de découverte. Il demande ensuite à un container donné l'instanciation de ce proxy. L'*UPnP designer* réagit de manière spontanée à la disparition des services UPnP et retire en conséquence les proxies des services disparus.

La modification de l'assemblage initial d'un container donné va déclencher un cycle de tissage du *tisseur* d'AA. Le *tisseur*

d'AA est responsable, comme son nom l'indique, du tissage des aspects d'assemblages. Des détails sur le cycle de tissage peuvent être trouvés dans [7]. Les aspects sont applicables dès lors que l'ensemble des points de jonctions sont découverts dans l'assemblage initial. Le tisseur va donc tisser ou composer les sous-assemblages (greffons) à l'assemblage initial, pour tous les aspects applicables. Le tissage ou la composition se fait donc de manière incrémentale sur un assemblage initial donné. Le mécanisme du tissage est déterministe grâce au caractère symétrique des AA entre eux. Cette caractéristique prouvée dans [4] permet, pour un même ensemble d'AA tissés dans des ordres différents, l'obtention d'une même application.

La construction de l'application s'effectue de manière dynamique et incrémentale en fonction des services qui apparaissent dans l'environnement. L'apparition ou la disparition de services dans l'infrastructure entraînent respectivement la réaction spontanée de tissage ou de dé-tissage d'AA. On parle de *construction opportuniste* d'applications car l'on crée une application en réaction à l'apparition opportune des services dans l'environnement. L'application en tant que composition ou orchestration est donc automatiquement modifiée à l'exécution sans que celle-ci soit stoppée. On parle alors d'adaptation structurelle de l'application [16]. L'application ainsi dynamiquement construite ou adaptée est le résultat de la *composition tardive* de tous les sous-assemblages (greffons) des AA mis en œuvre. Cette composition incrémentale de workflow est déterministe quelque soit l'ordre dans lequel on les compose [7].

3.2 Vers une décentralisation de l'architecture

L'orchestration créée dynamiquement par l'architecture logicielle est dite locale. Elle s'exécute sur un nœud donné de l'infrastructure matérielle. Dans des environnements mobiles Ad Hoc, on ne peut pas supposer l'existence d'un nœud d'exécution précis. Dans ce cadre nous proposons une architecture logicielle décentralisée.

Chaque entité matérielle ou logicielle de l'environnement supporte la construction dynamique d'une orchestration locale. Le triplet [container, perception de l'infrastructure de services, tisseur] qui offre les capacités de construction dynamique est supporté par chacune de ces entités. Ces entités sont appelées *systèmes locaux autonomes* (Fig. 3). Chacun de ces systèmes possède ses propres règles de construction locales et réagit donc différemment suivant sa propre perception de l'environnement en construisant dynamiquement ou en adaptant son orchestration. Ces systèmes locaux autonomes sont dit *auto-organisés* car ils s'adaptent dynamiquement sans intervention extérieure[6].

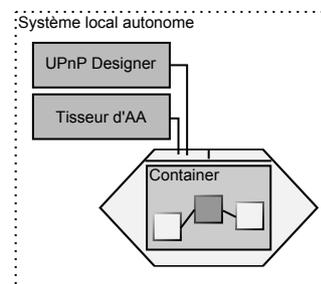


Figure 3: Système local autonome

Les orchestrations locales des systèmes autonomes mettent en œuvre une chorégraphie dès lors qu'ils communiquent ensemble [11] (Fig. 4). A un niveau micro, les systèmes locaux autonomes décrivent un ensemble d'orchestrations locales. A un niveau macro, l'ensemble des interactions entre systèmes locaux autonomes fait émerger le comportement de l'application [6].

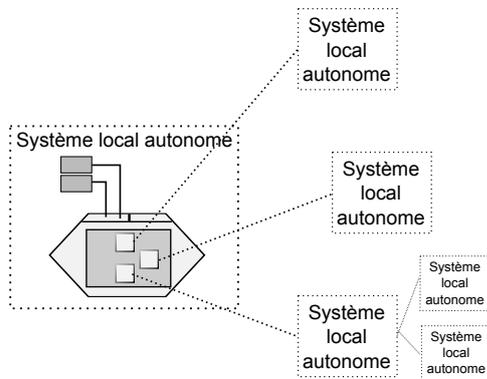


Figure 4: Architecture décentralisée

4. AVANTAGES ET FAIBLESSES

L'approche méthodologique descendante ou Top-down montre ses limites dans l'établissement d'une application d'IA en environnement partiellement connu à priori et fortement variable. La méthodologie ascendante semble, quant à elle, répondre correctement aux besoins identifiés d'une phase de design non fini et d'une décentralisation de l'architecture logicielle. Cependant cette approche soulève de nouvelles problématiques. Nous faisons état dans cette section des avantages qu'apporte cette approche et des nouvelles problématiques qu'elle soulève.

4.1 Avantages

4.1.1 Réponse aux problématiques du design

En premier lieu, la spécification d'une application dans un environnement complètement mobile est plus aisée. En effet, l'identification d'un besoin précis et complet dans ces environnements n'étant pas facilement envisageable à priori, l'établissement d'un ensemble de règles répondant chacune à un besoin simple donc plus facilement identifiable à priori est plus adapté.

Ensuite, la construction d'une application adaptée à son environnement est permise sans avoir eu à traiter la combinatoire des configurations possibles à priori. En effet, l'expression d'une règle est aujourd'hui déterminée par l'infrastructure perçue. La construction est donc sensible à un certain contexte limité aujourd'hui à celui de l'infrastructure. La combinatoire des configurations possibles de l'application n'est donc pas entreprise à priori mais toutefois rendue possible par l'expression distincte de chacune des règles de composition suivant l'infrastructure perçue. La composition tardive va ainsi construire dynamiquement une configuration donnée grâce à l'expression indépendante de chacune des règles de construction.

Enfin, la spécification d'une application à travers un ensemble de règles dynamiquement extensibles permet de ré-

pondre aux problématiques d'un environnement non bornés. En effet, l'application peut être dynamiquement étendue par l'ajout de nouveaux aspects. Le processus de design de l'application n'est donc plus un processus fini et monolithique. Ces nouveaux aspects seront tissés dans l'application à l'exécution l'étendant ainsi incrémentalement. De plus, l'indépendance entre les AA offre plus de flexibilité dans le processus de design qui peut être ainsi mené de manière indépendante et découplée par plusieurs développeurs à des moments différents. La réutilisation des règles offre aussi une modularité dans la spécification et donne tout son sens à la séparation des préoccupations. Enfin, le caractère symétrique des AA et leur composition tardive déterministe rend consistante, pour un même ensemble d'AA, la construction.

4.1.2 Réponse aux problématiques de l'architecture

D'abord la forte variabilité de l'infrastructure logicielle ou matérielle est traitée par l'utilisation d'une infrastructure réseau spontanée [12]. Ensuite, cette forte variabilité est aussi prise en compte par la réaction spontanée d'adaptation de l'application par notre architecture logicielle dès l'apparition ou la disparition des services.

Enfin, l'utilisation d'une infrastructure réseau spontanée et la mise en place d'une architecture logicielle toutes deux décentralisées permet d'éviter à la fois goulots d'étranglement et point central de défaillance dans l'annuaire des services comme dans la coordination des services. L'autonomie donnée à nos systèmes locaux met en œuvre une construction de l'application Bottom-up sans coordination centralisée.

4.2 Faiblesses

Parmi les faiblesses nous pouvons noter en premier lieu le caractère opportuniste de la construction qui a pour résultat l'émergence d'un comportement applicatif non maîtrisé. Enfin, l'autonomie des systèmes locaux met en œuvre une perception locale et forcément limitée.

4.2.1 Application opportuniste non pertinente

Au niveau micro, le caractère opportuniste des compositions locales met en œuvre des orchestrations de services que l'environnement peut supporter. Toutefois les comportements des orchestrations locales qui peuvent être supportés par l'infrastructure peuvent ne pas être adéquate avec l'environnement physique ou avec l'utilisateur. En effet, l'expression opportuniste des règles de chacune des constructions permet la mise en œuvre d'un comportement réalisable mais pas forcément pertinent à un moment donné dans l'environnement. Au niveau macro, l'ensemble des comportements locaux opportunistes fait émerger une application dont la pertinence ne peut pas être assurée. L'émergence de l'application n'est pas maîtrisée.

4.2.2 Perception locale limitée

Le caractère autonome donné aux systèmes locaux s'appuie sur la capacité de ces systèmes à percevoir l'environnement. Cette perception est locale donc limitée. Cette limitation est souhaitée pour rendre totalement autonomes les systèmes locaux mais toute la chaîne de la gestion du contexte doit donc être supportée par les systèmes locaux jusqu'à la décision. D'une part, les systèmes locaux autonomes sont susceptibles de ne pas pouvoir mener à bien cette tâche à cause de potentielles ressources limitées. D'autre part un cer-

tain nombre de calculs sur l'information contextuelle perçue peuvent être effectués par les systèmes autonomes de manière redondante. Enfin, l'interprétation d'informations de plus haut niveau par une entité suppose qu'elle soit capable de retrouver toutes les informations contextuelles nécessaires potentiellement importantes.

5. PERSPECTIVES ET CONCLUSION

5.1 Perspectives

Deux perspectives sont envisagées pour répondre aux faiblesses de l'approche. Un moyen d'atteindre l'émergence pertinente d'une application ainsi que la mise en place d'une perception globale.

5.1.1 Recherche de la pertinence

L'application qui émerge de l'ensemble des comportements des systèmes locaux autonomes est aujourd'hui opportuniste. Elle est la résultante de toutes les orchestrations de services créés de manière spontanées en réaction à l'apparition et disparition opportuniste des services dans l'environnement. La pertinence de l'application sous entend son adéquation avec son environnement physique, logique et utilisateur. Seule l'adéquation de l'application avec son environnement logique est aujourd'hui mise en œuvre.

Plusieurs pistes peuvent être envisagées dans la recherche de l'adéquation de l'application avec son environnement. Au niveau micro tout d'abord, la pertinence des règles locales doit être recherchée. Pour cela, la condition d'applicabilité d'une règle doit être enrichie de la prise en compte de l'environnement physique et utilisateur. Les règles effectivement appliquées seront donc pertinentes localement vis à vis de l'infrastructure, de l'environnement physique et de l'utilisateur. Pour cela, soit le tisseur d'AA détermine lui-même la pertinence à l'application d'un AA, soit les AA déterminent eux même s'ils ont du sens à être appliqués. Au niveau macro, la pertinence locale des règles ne détermine pas forcément la pertinence du comportement global. Ainsi, la coordination des comportements locaux doit être rendu possible dans l'objectif du contrôle de l'émergence du comportement global (de l'application). On peut pour cela, s'appuyer sur des travaux relatifs aux systèmes complexes qui proposent par exemple un *contrôle global* ou une *influence globale* [15].

La mise en œuvre d'un contrôle global rendrait obsolète le caractère auto-organisé des systèmes locaux. La notion d'auto-organisation contrôlée apparaît alors [15]. Le système global n'est plus décentralisé mais multi-niveau. Chaque niveau possède son propre contrôle. Celui de l'application (niveau global) contrôle celui des sous-systèmes locaux. On peut envisager le contrôle global comme ayant un effet sur l'application des règles locales de construction ou sur la distribution de ces règles. Les systèmes locaux perdraient cependant en autonomie et la spécification explicite d'une politique globale mise en œuvre par le contrôle resterais, dans ces environnements mobiles Ad Hoc, une difficulté.

A l'inverse d'un contrôle global, la mise en place d'une influence globale permettrait quant à elle de garder le caractère autonome des systèmes locaux. Au lieu de contrôler l'application des règles, on influence leur applicabilité. Pour cela, d'une part, la spécification d'un attracteur doit être mise en œuvre. D'autre part la perception de cet attracteur doit être permis par les systèmes locaux. Cette perception permettrait à chacun des systèmes locaux d'orienter la

construction de l'application dans le sens décrit par l'attracteur. L'émergence du comportement n'est alors pas contrôlée mais plutôt influencée par la perception de cet attracteur.

5.1.2 Perception globale

La perspective d'une perception partagée par toutes les entités permettrait de résoudre le problème d'une perception locale limitée. L'ensemble de ces informations contextuelles pourrait être géré de manière décentralisée en pair à pair. Cette perspective suppose évidemment un modèle commun des informations contextuelles comme une ontologie.

5.2 Conclusion

Le cadre d'un environnement mobile Ad Hoc peut mettre en jeux un nombre potentiellement important d'artéfacts communicants. Tous ces dispositifs, services logiciels, capteurs, actuateurs ou moyens d'interactions sont mobiles et leur présence dans notre environnement n'est déterminée que par leur dynamique propre. La spécification d'une application unique et monolithique s'appuyant sur ces entités est donc difficilement envisageable. De plus, la construction d'une application comme un processus fini s'oppose au caractère continuellement évoluant de l'environnement.

L'approche méthodologique de construction d'une application Bottom-up ou ascendante au cœur de notre contribution propose l'émergence d'un comportement applicatif issu de l'expression indépendante de sous-objectifs. Le comportement global de l'application n'est ainsi pas spécifié a priori mais apparaît spontanément dans l'environnement suivant l'expression des sous-objectifs.

Cette approche est rendue possible par deux aspects. D'une part la spécification de règles locales et indépendantes répondant à des sous-objectifs simples, ainsi plus facilement identifiables. D'autre part une architecture logicielle décentralisée rendant autonome les entités supportant la mise en œuvre de ces règles. L'application résultante est l'expression de tous les comportements auto-organisés des systèmes locaux autonomes.

Le comportement global de l'application, aujourd'hui opportuniste, doit être rendu pertinent vis à vis de l'environnement et de l'utilisateur. Cet axe de recherche est, à présent, au centre de nos travaux. Et le caractère auto-organisé donné aux systèmes locaux ainsi que l'émergence d'un comportement de haut niveau suggèrent un rapprochement avec le domaine des systèmes complexes qui met typiquement en œuvre ces principes.

6. REFERENCES

- [1] G. Banavar and A. Bernstein. Challenges in design and software infrastructure for ubiquitous computing applications. *Advances in computers*, 62 :179–202, 2004.
- [2] J. Bronsted, K. M. Hansen, and M. Ingstrup. Service Composition Issues in Pervasive Computing. *IEEE Pervasive Computing*, 9(1) :62–70, Jan. 2010.
- [3] D. Chakraborty, A. Joshi, T. Fimin, and Y. Yesha. Service Composition for Mobile Environments. *Mobile Networks and Applications*, 10(4) :435–451, Aug. 2005.
- [4] D. Cheung-Foo-Wo. *Adaptation dynamique par tissage d'aspects*. PhD thesis, Université de Nice-Sophia

Antipolis, 2009.

- [5] C. A. da Costa, A. C. Yamin, and C. F. R. Geyer. Toward a General Software Infrastructure for Ubiquitous Computing. *IEEE Pervasive Computing*, 7(1) :64–73, Jan. 2008.
- [6] T. De Wolf and T. Holvoet. Emergence versus self-organisation : Different concepts but promising when combined. *Engineering Self-Organising Systems*, pages 1–15, 2005.
- [7] N. Ferry, V. Hourdin, S. Lavirotte, G. Rey, J.-y. Tigli, and M. Riveill. Models at Runtime : Service for Device Composition and Adaptation. In *4 th Workshop on Models@ run. time at MODELS 09*, page 10, 2009.
- [8] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. Griswold. An overview of AspectJ. *ECOOP 2001 Object-Oriented Programming*, pages 327–354, 2001.
- [9] T. Kindberg and a. Fox. System software for ubiquitous computing. *IEEE Pervasive Computing*, 1(1) :70–81, Jan. 2002.
- [10] J. Peer. Web Service Composition as AI Planning - a Survey. *Language*, (March), 2005.
- [11] C. Peltz. Web services orchestration and choreography. *Computer*, 2003.
- [12] S. Preuß and C. Cap. Overview of Spontaneous Networking-Evolving Concepts and Technologies. *Rostocker Informatik-Berichte*, 24 :113—123, 1999.
- [13] J. Rao and X. Su. A Survey of Automated Web Service Composition Methods. *Semantic Web Services and Web Process Composition*, 2005.
- [14] G. Schiele, M. Handte, and C. Becker. Pervasive Computing Middleware. *Handbook of Ambient Intelligence and Smart Environments*, pages 201–227, 2010.
- [15] H. Schmeck, C. Müller-Schloer, E. Çakar, M. Mnif, and U. Richter. Adaptivity and self-organization in organic computing systems. *ACM Transactions on Autonomous and Adaptive Systems*, 5(3) :1–32, Sept. 2010.
- [16] J.-y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. Cheung-Foo-Wo, E. Callegari, and M. Riveill. WComp middleware for ubiquitous computing : Aspects and composite event-based Web services. *Annales des Telecommunications*, 2009.
- [17] J.-y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, and M. Riveill. Lightweight Service Oriented Architecture for Pervasive Computing. *International Journal of Computer Science*, 4(1) :1–9, 2009.
- [18] A. Urbietta, G. Barrutieta, J. Parra, and A. Uribarren. A Survey of Dynamic Service Composition Approaches for Ambient Systems. *Proceedings of the First International Conference on Ambient Media and Systems*, 2008.
- [19] M. Vukovic. Context aware service composition, 2007.
- [20] M. Weiser and J. S. Brown. THE COMING AGE OF CALM TECHNOLOGY. *Distributed Computing*, 2001.
- [21] F. Zhu, M. W. Mutka, and L. M. Ni. Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, pages 81–90, 2005.

Vers un environnement de déploiement autonome

Mohammed El Amine Matougui
 Institut Télécom ; Télécom SudParis
 UMR 5157 CNRS SAMOVAR
 Évry, France
 mohammed_el_amine.matougui@it-
 sudparis.eu

Sébastien Leriche
 Institut Télécom ; Télécom SudParis
 UMR 5157 CNRS SAMOVAR
 Évry, France
 sebastien.leriche@it-sudparis.eu

Résumé

Le déploiement autonome de logiciel dans les systèmes répartis à grande échelle (grilles, systèmes pervasifs, P2P...) est une problématique actuelle ouverte. Quelques solutions existent aujourd'hui, mais ne sont exploitables que dans le cadre d'architectures figées. De ce fait, elles ne prennent pas en compte la réalité des évolutions dynamique (variation de QoS, pannes, apparition et disparition de nJuds, de services...) des environnements ouverts.

Nous présentons une approche originale, basée sur nos précédents résultats de recherche dans le domaine des agents mobiles adaptables (AMA). Les propriétés de ces agents logiciels (autonomie, proactivité, adaptation dynamique au contexte...) nous permettent d'envisager la construction d'un intergiciel capable de déployer des applications dans les environnements les plus instables, en limitant au minimum les interventions humaines.

Mots-clés

Intergiciel, déploiement logiciel, gestion autonome, auto-déploiement, auto-reconfiguration.

ABSTRACT

Autonomic software deployment in large scale distributed systems (grids, pervasive systems, P2P systems) is an open issue. Some solutions exist, but they are usable only in the context of static architectures. They do not take into account the dynamic changes of QoS, hosts failures, appearance and disappearance of nodes and services occurred in open environments.

In this paper we present an original approach, based on our previous research on adaptable mobile agents (AMA). The properties autonomy and dynamic adaptation to context of mobile agents allow us to consider building a middleware for deploying applications in the most unstable environments, with minimal human intervention.

Keywords

Middleware, Software deployment, autonomic deployment, self-deployment, self-reconfiguring

1. INTRODUCTION

Le déploiement de logiciel est défini comme un processus complexe qui comporte un ensemble d'activités de déploiement [3], à l'heure actuelle il n'existe pas de consensus autour des activités de déploiement. Le cycle de vie de déploiement regroupe toutes les activités depuis la validation du logiciel par le producteur jusqu'à sa désinstallation des sites cible de déploiement, cela inclue les activités de mise en place sur les sites cible comme l'installation et l'activation ainsi que les activités de maintenance du logiciel comme reconfiguration, mise à jour. La figure 1 représente le cycle de vie du déploiement [2].

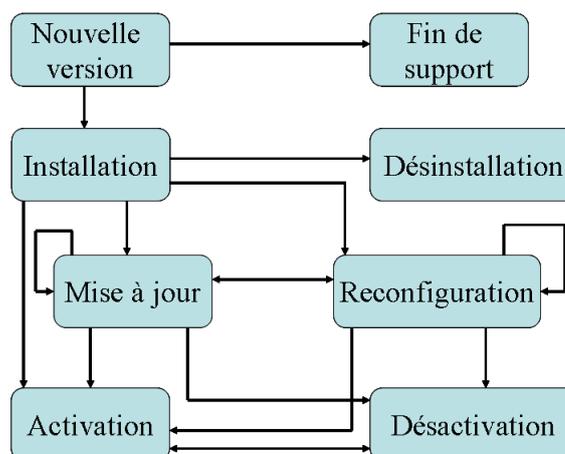


Figure 1: Cycle de vie du déploiement logiciel

Les principales activités de déploiement sont l'installation, la désinstallation, l'activation, la désactivation, la mise à jour et la reconfiguration.

L'activité d'installation [2] a pour objectif la préparation de l'environnement du logiciel à déployer. Elle permet d'insérer un logiciel sur les sites cible de déploiement. Elle comporte plusieurs opérations telles que la résolution des dépendances logicielles et matérielles du logiciel, le transfert du logiciel

vers les sites cible et la configuration du système afin que le logiciel déployé soit activable.

La désinstallation [2] est une activité qui consiste à remettre à l'état initial le système et d'enlever toutes les modifications introduites sur les sites cible de déploiement par les activités de déploiement.

L'activité d'activation [2] regroupe toutes les opérations nécessaires pour que le logiciel installé puis être exécuté. Elle consiste au lancement de l'exécution du logiciel déployé. Dans certains cas l'activation d'un certain nombre d'éléments logiciels peut être nécessaire pour l'exécution du logiciel (des serveurs web, autres logiciels installés, E) avant de pouvoir activer le logiciel déployé.

L'activité de désactivation [2] est une activité utilisée en général avant les activités de mise à jour, de reconfiguration et de désinstallation. Elle consiste à interrompre l'exécution du logiciel ou une partie du logiciel déployé.

L'activité de mise à jour [2] consiste à introduire des modifications sur le logiciel installé en utilisant la configuration courante. Un exemple de cette activité est le déploiement d'une nouvelle version du logiciel déployé.

La reconfiguration [2] modifie un logiciel déjà déployé mais en utilisant une configuration différente à la configuration existante. Ces modifications peuvent être la suppression, l'ajout ou le remplacement d'une entité logiciel et la réparation d'une panne.

Le terme déploiement de logiciel est souvent utilisé et laisse penser que l'unité de déploiement est le logiciel, tandis que souvent les éléments qui sont effectivement déployés sont des unités de déploiement (des composants, des services, E). Une unité de déploiement est une entité logicielle packagée sous la forme d'un package exécutable dans un environnement d'exécution sans besoin d'intervenir pour le modifier au moment du déploiement afin de le rendre prêt à être déployé [3]. Notons qu'un logiciel peut être une unité de déploiement en lui-même ou être constitué de plusieurs unités de déploiement.

Dans notre proposition de déploiement de logiciel nous nous intéressons au déploiement autonome de logiciel à grande échelle. Les systèmes ciblés par ce déploiement sont du type grille de calcul, systèmes P2P et systèmes pervasifs. Ces systèmes sont caractérisés par un grand nombre de sites équipés d'environnements matériels et logiciels hétérogènes. Une partie de ces systèmes est caractérisée aussi par la mobilité des sites (cas des systèmes pervasifs) et des changements de topologie du réseau (disparition et apparition des pairs dans les systèmes P2P).

Il existe plusieurs plate-formes de déploiement dédiées à ce type de systèmes tels que Software Dock [6], DeployWare [5] et JADE [8], mais elles ne sont exploitables que dans le cadre de topologies statiques et ne prévoient pas des solutions d'auto-adaptation et de reconfiguration prenant en compte les variations de QoS et/ou par exemple les situations de pannes de machines et des liens du réseau qui caractérise les environnements ouverts. En plus l'utilisateur de

ces plates-formes de déploiement doit gérer manuellement la plupart du temps les activités de déploiement ainsi que les opérations de reconfiguration ce qui représente une intervention humaine très importante dans le processus de déploiement. Ces interventions deviennent très vite une tâche très complexe qui présente un besoin de rendre autonome le processus de déploiement pour réduire au minimum les interventions humaines.

Dans cet article, après avoir présenté deux scénarios motivants nos travaux (section 2), nous présentons notre ébauche de proposition de déploiement autonome de logiciel à grande échelle qui se base sur un système d'agents mobiles adaptable pour la réalisation des activités de déploiement (section 3). Dans la section 4 nous présentons l'état d'avancement de l'implémentation du prototype. Enfin, dans la section 5 nous présentons une conclusion et les perspectives de notre proposition.

2. SCÉNARIOS MOTIVANTS

Pour pouvoir exprimer au mieux les différents problèmes et les spécificités du déploiement de logiciel à grande échelle nous présenterons deux scénarios de déploiement. Le premier scénario consiste au déploiement d'une application de supervision de l'état d'activités pour un ensemble d'équipements connectés à un réseau WiFi en utilisant l'intergiciel de gestion de contexte COSMOS[7], développé dans notre équipe. Le deuxième scénario consiste au déploiement d'une application de simulation sur une grille.

2.1 Scénario 1

Considérons un logiciel réparti de supervision de l'état d'activités d'un ensemble d'équipements pour un ensemble de participants à une conférence. Ce logiciel fournit à l'ensemble des participants des informations statistiques sur leurs équipements connectés à un réseau WiFi au moment de l'expérimentation au moyen de l'intergiciel de gestion de contexte COSMOS. Le nombre de participants à cette expérimentation est imprévisible, il peut être à l'échelle d'une dizaine ou d'une centaine de participants.

COSMOS comporte plusieurs versions, une version installable sur des téléphones mobiles, une version installable sur des Smartphones équipés d'Android et une version java standard installable sur des ordinateurs équipés d'une machine virtuelle java standard.

Les informations récoltées dans le cadre de cette expérimentation sont la taille de la RAM disponible, le type de l'OS utilisé, le pourcentage de l'occupation du CPU et la taille du disque disponible pour chaque équipement impliqué. Les équipements impliqués dans l'expérimentation sont de natures différentes, ils peuvent être des ordinateurs portables, des Smartphones, des netbooks et des Macbook...

Chaque équipement connecté au réseau WiFi est potentiellement un site cible de déploiement.

L'application à déployer est constituée de 8 composants (noeud de contexte COSMOS), chaque composant de l'application représente une unité de déploiement. La figure 2 présente l'application sous la forme de noeuds de contexte COSMOS. Le composant OS-Type permet de retourner le

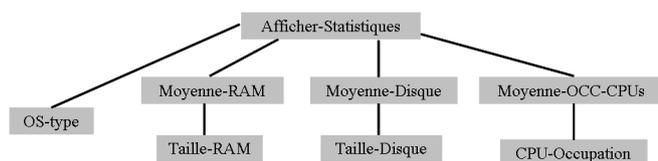


Figure 2: Composition de nœuds de contexte de l'application

type du système d'exploitation utilisé. Le composant Taille-RAM permet le calcul de la taille de la RAM disponible dans chaque site. Le composant Taille-Disque permet de calculer la taille disque disponible dans chaque site. Le composant CPU-Occupation permet de calculer le pourcentage de l'occupation des processeurs. Les composants Moyenne-RAM, Moyenne-Disque et Moyenne-CPUS-Occupation permettent de calculer respectivement la moyenne de RAM disponible, la moyenne de l'espace disque disponible et la moyenne du pourcentage d'occupation des processeurs. Enfin le composant Afficher-Statistiques a pour rôle l'affichage des informations statistiques calculées au courant de cette expérimentation.

Le plan de déploiement souhaitable de cette application est le suivant : on distingue une machine dans lequel on installe tous les composants de l'application puis on installe les composants OS-type, Taille-RAM, Taille disque et CPU-occupation sur toutes les autres machines impliquées.

Pour initialiser le fonctionnement global du système, nous aurons besoin d'un mécanisme de "bootstrap" qui aura pour rôle la préparation de l'environnement de l'exécution pour le système de déploiement, ainsi que pour résoudre le problème de l'administration multiple. Cela pourrait être réalisé par l'intermédiaire d'un programme dédié, volontairement installé par l'administrateur d'une machine, et mis à sa disposition par d'autres moyens (push sur bluetooth, publicité, SMS...)

Le système de déploiement doit couvrir toutes les activités de déploiement. Il doit permettre l'installation du logiciel selon le plan de déploiement défini. Dans cette activité le système de déploiement doit résoudre les dépendances logicielles et matérielles, puis transfère les composants sur les sites cible et enfin il procède à leurs mises place dans les sites cible de déploiement.

Le système de déploiement doit aussi permettre l'activation du logiciel déployé. Une fois le logiciel activé il récupère les informations sur les sites (taille de la RAM et du disque...), puis il calcule les moyennes et affiche les résultats obtenus.

Le système de déploiement doit aussi fournir les mécanismes nécessaires pour la désactivation, la désinstallation et la mise à jour du logiciel déployé. Enfin le système de déploiement doit aussi prévoir un mécanisme de reconfiguration dynamique au moment de l'exécution pour prendre en charge les situations de pannes de machine, les déconnexions et les nouvelles connexions.

2.2 Scénario 2

Le deuxième scénario présente le déploiement d'une application de simulation à grande échelle sur une grille. Le processus de simulation commence par la préparation de la simulation, cette étape consiste à développer le modèle de simulation ainsi que les différents scénarios de tests de ce modèle. La deuxième étape consiste au déploiement du modèle de simulation et les différents scénarios de tests sur une grille. La troisième étape consiste à analyser les résultats de la simulation.

Dans ce scénario l'étape qui nous intéresse est la deuxième. Pour déployer cette application de simulation, la majorité des outils de déploiement existants exige que la personne en charge du déploiement fournisse un ou plusieurs fichiers de description dans lesquelles il déclare un certain nombre d'informations. L'utilisateur doit spécifier les propriétés du logiciel à déployer, par exemple le chemin vers l'archive de l'application, les ports de communication utilisés et les dépendances vers d'autres logiciels. L'utilisateur doit aussi souvent fournir un plan de déploiement précis de son application, il doit décrire comment installer et activer son application. Enfin l'utilisateur doit lister à la main tous les nœuds physiques dans lesquels les unités de déploiement doivent être déployées. La déclaration des sites cible et la mise en place du plan de déploiement manuellement une tâche très lourde à réaliser par un opérateur humain dans le cadre d'un système de grosse taille, tel que les grilles de calcul.

Une fois que l'utilisateur a réalisé toutes les déclarations, le processus de déploiement peut commencer. Dans ce type de système, la personne en charge du déploiement n'a pas besoin d'un mécanisme de détection de sites cible de déploiement par ce qu'elle connaît au préalable les différents sites. De même, le problème d'administration multiple posé par le premier scénario n'est plus pertinent, étant donné que l'on est dans le cadre d'un système mono-administrateur.

Les outils de déploiement existants pour ce type de système ne couvrent pas en général toutes les activités de déploiement, ces dernières sont souvent réalisées manuellement ou d'une manière semi-automatique à l'aide de scripts. En plus la majorité de ces systèmes n'offrent pas généralement des mécanismes de reconfiguration et d'auto-adaptation pour la prise en charge des situations de pannes de machines et des liens du réseau.

2.3 Synthèse

Dans cette section nous avons présenté deux scénarios de déploiement de logiciel sur des environnements répartis à grande échelle. Le premier scénario traite le déploiement d'une application dans un environnement ouvert. Le déploiement de logiciel dans ce type d'environnement engendre des activités supplémentaires.

1. La première activité est la découverte du réseau. Dans ce type de système (P2P, pervasif) on ne connaît pas au préalable les sites de déploiement. Donc on a besoin de détecter les sites cible de déploiement pour pouvoir déployer un logiciel.
2. La deuxième activité est la satisfaction des problèmes d'administration et de droit d'accès pour que notre

logiciel soit déployable sur les sites détectés.

3. Enfin les outils de déploiement pour ce type d'environnement doivent prévoir des mécanismes d'auto-adaptation et de reconfiguration pour la prise en charge des différentes situations de pannes et de déconnexions.

Le deuxième scénario présente le déploiement d'une application de simulation à grande échelle. Le déploiement de ce logiciel avec les plates-formes de déploiement existantes tel que DeployWare, exige la fourniture de plusieurs descripteurs de déploiement dans lesquelles l'utilisateur déclare les propriétés du logiciel à déployer, la liste des sites cible et son plan de déploiement. En plus l'utilisation de ces outils nécessite un effort supplémentaire non négligeable qui se manifeste dans la gestion manuelle des activités de déploiement et les opérations de reconfiguration.

De ce fait, la plate-forme de déploiement qui répond aux problématiques et spécificités de ce type d'environnement (grille, P2P, pervasif) doit être :

1. Capable de détecter, gérer, et accéder aux sites cible d'une manière automatique.
2. Capable de gérer les hétérogénéités logicielles et matérielle des sites détectés.
3. Fournir un moyen pour la déclaration des dépendances logicielles, les préférences matérielles et les contraintes de déploiement.
4. Capable de calculer un plan de déploiement qui satisfasse les contraintes de déploiement d'une manière automatique.
5. Capable d'exécuter les activités de déploiement avec un minimum d'intervention humaine.
6. Capable de s'auto-adapter et offre des mécanismes de reconfiguration automatiques au moment de l'exécution pour prendre en charge les situations de pannes de machine et des déconnexions.
7. Utilisables dans des topologies à grande échelle.

3. PROPOSITION POUR LE DÉPLOIEMENT AUTONOMIQUE

Dans cette section nous allons présenter brièvement l'architecture de notre intergiciel pour le déploiement autonome de logiciel. Nous proposons un intergiciel qui utilise OSGi [9] comme support de déploiement, un langage dédié pour l'expression des contraintes de déploiement, un solveur de contraintes pour le calcul d'un plan de déploiement qui satisfasse les contraintes et un système d'agents mobiles adaptables qui prend en charge l'exécution et la supervision des activités de déploiement.

Les objectifs de l'intergiciel sont de permettre le déploiement réparti à grande échelle en réduisant au minimum les interventions humaines dans le processus de déploiement.

3.1 Support de déploiement

Notre intergiciel utilise le Framework OSGi comme support de déploiement. OSGi [9] est une spécification qui définit une plate-forme java qui permet le déploiement d'unités de déploiement appelées bundle dans l'environnement d'exécution OSGi. Un bundle est une archive java au format JAR, contenant également un fichier de méta-données qui décrit le JAR, les classes et les bibliothèques de code de l'application et une classe d'activation.

Les activités de déploiement qu'on peut réaliser avec le Framework OSGi sont l'installation, l'activation, la désactivation, la mise à jour et la désinstallation des bundles. En plus l'environnement prend en charge la gestion des dépendances de code qui doivent être satisfaites après l'installation du bundle afin de pouvoir l'activer. Les applications OSGi sont construites à partir des bundles connectés à travers des services. Les services peuvent dynamiquement apparaître ou disparaître au courant de l'exécution de l'application.

L'utilisation d'OSGi comme support de déploiement va nous permettre le déploiement de logiciel sur plusieurs types d'équipements dotés d'environnements matériels et logiciels hétérogènes y compris des matériels de petite taille (type smartphones / tablettes PC / UMPC...). En plus la réutilisation des fonctionnalités de déploiement telles que l'installation et la désinstallation offertes par l'environnement d'exécution OSGi nous permettent de nous concentrer sur les autres aspects.

3.2 L'unité de déploiement

L'unité de déploiement est actuellement celui d'un bundle OSGi. Le déploiement d'autres types d'unités de déploiement sera envisagé ultérieurement.

3.3 Description des contraintes de déploiement

Comme toutes les plates-formes de déploiement existantes, l'utilisateur de notre intergiciel doit fournir des informations sur l'application à déployer et un ensemble de contraintes de déploiement. Les informations (propriétés du logiciel à déployer) et les contraintes de déploiement seront utilisées par la suite pour le calcul d'un plan de déploiement qui satisfasse les contraintes de déploiement.

Les informations de l'application à déployer ainsi que ses différentes contraintes de déploiement seront déclarées en utilisant un langage dédié pour description de contraintes de déploiement. Ce langage est fortement inspiré du langage DELADAS [4].

Les informations à déclarer sont respectivement le chemin vers l'archive de l'application, le nom et la version du logiciel, les composants qui forment l'application, les services requis et fournis par l'application, les dépendances vers d'autres logiciels et enfin les préférences matérielles et logicielles des sites cible de déploiement (la taille de la RAM et du disque, le type de l'OS, l'état de la batterie, ...).

Les contraintes de déploiement sont le nombre minimum des sites cible de déploiement et les contraintes liées aux activités de déploiement, par exemple les contraintes d'installation et

les contraintes d'activation. La figure 3 présente un exemple de déclaration des contraintes d'installation.

```
ConstraintsSet {
    InstallContraintes {
        Install exemple1 forall hosts;
        Install exemple2 in 2 hosts;
    }
    ...
}
```

Figure 3: Déclaration de contraintes d'installation

Dans cet exemple les contraintes d'installation sont, installer le composant exemple1 sur tous les hôtes détectés et installer le composant exemple2 sur seulement deux hôtes détectés.

3.4 Système d'agents mobiles adaptable

Dans notre intergiciel de déploiement, l'entité qui prend en charge l'exécution des activités de déploiement est un système d'agents mobiles adaptables. Un agent mobile adaptable est une entité autonome capable de communiquer et de se déplacer disposant des connaissances et d'un comportement privé et d'une capacité d'exécution propre, en plus il dispose des mécanismes d'adaptation au contexte de l'exécution [1].

À chaque activité de déploiement on associe un ou plusieurs agents spécialisés. Les opérations de supervision et de contrôle du processus de déploiement seront assurées par un autre agent. Donc dans notre système on a des agents d'installation, des agents de désinstallation, des agents d'activation, des agents de désactivation, des agents de mise à jour et des agents de supervision.

L'agent d'installation a pour objectif l'installation de bundle selon le plan de d'installation calculé. Les tâches confiées à cet agent sont : le téléchargement du package de l'application sur les différents sites cible de déploiement, résolution des dépendances de l'application, l'installation physique du package et enfin notifier à l'agent de supervision la fin de l'installation.

L'agent de désinstallation a pour objectif d'enlever toutes les modifications introduites par les différentes activités de déploiement. Cet agent doit être en mesure de détecter si une dépendance installée est en cours d'utilisation par une autre application afin de ne pas la désinstaller et avoir causé des situations de conflits.

L'agent d'activation a pour but de démarrer l'exécution de l'application selon le plan d'activation à la demande de l'agent de supervision.

L'agent de désactivation arrête sur la demande d'un autre agent tout ou une partie de l'application déployée.

L'agent de mise à jour réalise les mises à jour de l'application sur la demande de l'agent de supervision.

L'agent superviseur a pour rôle, le contrôle du processus

de déploiement et la reconfiguration de l'application dans le cas de pannes et de déconnexions. L'agent de supervision offre des interfaces d'échanges avec les utilisateurs du système de déploiement. Ces interfaces permettent de consulter l'état du processus de déploiement et d'intervenir dans une ou plusieurs activités de déploiement si l'utilisateur le souhaite.

L'exécution et la création ainsi que les communications des agents mobiles sont assurés par un système d'accueil d'agents installé et exécuté sur l'environnement d'exécution d'OSGi.

3.5 Exécution du processus de déploiement

La figure 4 résume toutes les étapes de l'exécution du processus de déploiement.

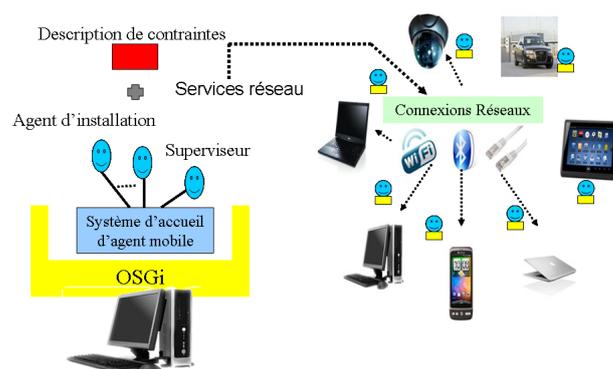


Figure 4: Processus de déploiement

Le processus de déploiement commence dès que l'utilisateur fournit ça description de contraintes de déploiement.

L'intergiciel commence par le lancement du service de découverte du réseau pour la détection des sites cible de déploiement. Une fois les sites détectés l'intergiciel lance le solveur de contraintes en lui passant comme argument les contraintes de déploiement. Si le solveur réussit à calculer un plan de déploiement qui satisfasse les contraintes, le système d'agents mobiles est déclenché, les agents de déploiement sont créés et envoyés vers les sites cible afin d'exécuter les différentes activités de déploiement. Le contrôle du processus de déploiement est assuré par les agents superviseurs.

4. PROTOTYPE

Pour valider notre approche, nous avons procédé au développement d'un prototype. Dans ce prototype le système d'accueil d'agents mobile utilisé est JavAct¹.

Nous avons transformé JavAct sous la forme d'un bundle OSGi pour permettre son installation et son activation dans l'environnement d'exécution OSGi. Le bundle de JavAct est testé et exécuté sur les Framework Equinox² du projet Eclipse et le Framework Apache Felix³.

¹<http://www.javact.org>

²<http://www.eclipse.org/equinox/>

³<http://felix.apache.org/site/index.html>

Nous avons aussi développé les agents d'installation, d'activation, de désactivation, de désinstallation et de mise à jour. L'ensemble de ses agents permet le déploiement d'applications composées d'un ou plusieurs bundles sur l'environnement d'exécution OSGi. La version actuelle du prototype permet d'envoyer nos agents de déploiement à d'autres systèmes d'accueil JavAct exécutés sur d'autres machines virtuelle java afin d'exécuter des activités de déploiement.

Nous allons poursuivre l'amélioration du prototype en développant les agents de supervisions de déploiement afin de contrôler tout le processus de déploiement et offrir la possibilité de reconfiguration et d'auto-adaptation à notre système de déploiement.

Entre autres nous avons commencé le développement d'un service de découverte du réseau ainsi que l'étude et l'extension du langage de description de contraintes de déploiement DELADAS et son solveur de contraintes qui seront intégré dans la prochaine version du prototype.

5. CONCLUSION

Dans cet article nous avons présenté une approche pour le déploiement autonome de logiciel à grande échelle. Cette approche consiste à un assemblage de technologies pour le développement d'un intergiciel pour le déploiement autonome de logiciel à grande échelle.

L'intergiciel est constitué d'un langage de description de contraintes de déploiement, d'un service de découverte de réseau, d'un service de bootstrap pour la préparation de l'environnement de l'exécution, d'un solveur de contraintes de déploiement pour calculer un plan de déploiement et d'un système d'agents mobiles adaptable qui exécute les activités de déploiement.

Nous avons choisi le Framework OSGi comme plate-forme de déploiement pour permettre le déploiement de logiciels sur les matérielles de petite taille comme les Smartphones et les téléphones portables et même d'autres types d'équipement comme les calculateurs des voitures.

Grâce aux caractéristiques des agents mobiles (comportement autonome et capacité de déplacement) nous pouvons réaliser des reconfigurations et des adaptations dynamiques au moment de l'exécution sans aucune intervention de l'utilisateur de notre système de déploiement. Nous envisageons par l'assemblage des différentes technologies répondre aux problématiques des environnements ouverts comme les systèmes P2P et les systèmes pervasifs en réduisant au minimum les interventions humaines dans le processus de déploiement.

Nous poursuivons actuellement nos travaux sur une nouvelle version du prototype. Nous travaillons sur la génération du plan de déploiement à partir d'une description de contraintes écrite à l'aide du langage de description de contraintes de déploiement. Nous travaillons aussi sur les comportements des agents de superviseurs et des algorithmes de déploiement par des agents pour résoudre au mieux les problématiques de déploiement à grande échelle et offrir un système qui permet de réaliser des opérations d'auto-adaptation dynamique au

moment de l'exécution.

6. REFERENCES

- [1] J.-P. Arcangeli, S. Leriche, and M. Pantel. Un framework à composants et agents pour les applications réparties à grande échelle. *L'OBJET*, 12(4):103–132, 2006.
- [2] A. Carzaniga, A. Fuggetta, R. S. Hall, D. Heimbigner, A. van der Hoek, A. L. Wolf, A. V. Der, E. L. Wolf, and E. L. Wolf. A characterization framework for software deployment technologies. Technical report, Dept. of Computer Science, University of Colorado, 1998.
- [3] A. Dearle. Software deployment, past, present and future. In *FOSE*, pages 269–284, 2007.
- [4] A. Dearle, G. N. C. Kirby, and A. J. McCarthy. A framework for constraint-based deployment and autonomic management of distributed applications. In *ICAC*, pages 300–301, 2004.
- [5] A. Flissi, J. Dubus, N. Dolet, and P. Merle. Deploying on the grid with deployware. In *CCGRID*, pages 177–184, 2008.
- [6] R. S. Hall, D. Heimbigner, and A. L. Wolf. A cooperative approach to support software deployment using the software dock. In *Proceedings of the 21st international conference on Software engineering, ICSE '99*, pages 174–183. ACM, 1999.
- [7] R. Rouvoy, D. Conan, and L. Seinturier. Software architecture patterns for a context-processing middleware framework. *IEEE Distributed Systems Online*, 9(6):1, 2008.
- [8] C. Taton, S. Bouchenak, N. De Palma, D. Hagimont, and S. Sicard. Self-sizing of clustered databases. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, WOWMOM '06*, pages 506–512, Washington, DC, USA, 2006. IEEE Computer Society.
- [9] The OSGi Alliance. OSGi service platform core specification, release 3. version 4.2, 2009.

Sémantique pour les capteurs et effecteurs en environnement pervasif

B. Helena Rodríguez, Jean-Claude Moissinac

Institut Telecom / Telecom ParisTech
CNRS LTCI
46, rue Barrault
75634 PARIS Cedex 13

{bhrdrig} {moissinac}@telecom-paristech.fr

Résumé

Les technologies permettant l'interaction de l'homme avec l'ordinateur ont évoluées vers une forme d'abstraction qui facilite l'intégration de diverses ressources techniques dans un environnement connu par l'utilisateur.

Nous allons montrer comment l'informatique pervasive peut être vue dans la perspective des abstractions déjà utilisées par les pilotes –drivers- génériques. Nous allons présenter l'approche proposée par le W3C avec l'architecture MMI dans un contexte de multimodalité. Enfin, nous montrerons pourquoi il nous paraît aujourd'hui nécessaire de définir un niveau complémentaire d'abstraction des technologies d'interaction et pourquoi une dimension sémantique des descripteurs de ces technologies nous paraît nécessaire. En particulier, nous introduirons la nécessité de prendre en compte des informations sociales du contexte et de disposer des descripteurs correspondants pour la découverte des modes d'interactions disponibles dans une situation donnée.

Termes généraux

Design.

Mots clés

Service, interaction multimodale, architecture MMI, Ontologies.

1. INTRODUCTION

Après un temps où l'on interagissait avec un ordinateur à l'aide de cartes ou de bandes perforées, sont apparus des moyens plus conviviaux: l'écran, le clavier, puis la souris. D'abord, ceux-ci étaient spécifiques de la machine sur laquelle ils fonctionnaient. Puis, les pilotes –drivers- génériques sont apparus: ils permettaient de fournir à un système d'exploitation une vision unique de familles d'équipements: pointeur/souris, clavier...

Cette approche est encore utilisée actuellement. Dès qu'un équipement fournit un tel driver, il s'intègre facilement à l'environnement d'une machine en fournissant une série de services prédéfinis: ainsi, un trackpad, une manette de jeu et une

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 2011, Juin 6–8, 2010, Toulouse, France.

Copyright 2011 ACM

souris peuvent tous être vus comme un équipement fournissant une suite de positions (x,y) et un état correspondant généralement à l'appui de boutons; chacun de ces équipements peut fournir d'autres services spécifiques, mais chacun peut être vu via son pilote de façon abstraite, unifiée, comme étant un pointeur 2D.

Cette approche est employée pour permettre d'utiliser une grande variété d'équipements connectés à un ordinateur; elle est notamment utilisée pour:

- Les pointeurs 2D,
- Les écrans, qui affichent des images numériques avec une fréquence de rafraîchissement élevée,
- Les scanners qui permettent une capture d'image,
- Les claviers qui permettent la saisie de texte,
- Les caméras qui permettent la capture de séquences d'images,
- Les imprimantes.

Pour chacune de ces catégories, les systèmes d'exploitation définissent un pilote générique qui permet de connecter une grande variété d'équipements.

L'évolution technique actuelle nous permet d'envisager de découvrir de tels services non plus par connexion directe à l'ordinateur, mais par découverte dans l'environnement. Une telle approche est à rapprocher des technologies développées pour l'informatique distribuée et qui permettent en particulier la déclaration, la description et la découverte de services.

Par exemple, le standard UPnP [4] /DLNA [5] permet à des dispositifs en réseau de se déclarer comme capables d'être une source de données audio ou vidéo, et à d'autres de se déclarer comme capables de lire - jouer de telles données; des dispositifs 'points de contrôle' permettent de mettre en relation une source et un lecteur. Ainsi, sans liaison initiale autre que la présence sur un même réseau, une telle technique permet de jouer une vidéo sur l'écran de son choix dans l'environnement où l'on se trouve.

Nous allons voir dans la suite de cet article comment l'architecture MMI, proposée par le W3C, et à laquelle nous contribuons, propose une architecture générique pour gérer des interactions multimodales, notamment en environnement pervasif. Nous allons présenter une approche SOA sémantique permettant d'enrichir l'instanciation et découverte des services unimodaux et multimodaux selon les activités des utilisateurs dans l'espace et selon le contexte d'utilisation. Ensuite nous présenterons l'implémentation qui en découle.

2. L'ARCHITECTURE MMI DU W3C

MMI est un acronyme pour Multimodal Interaction, qui est l'objet d'un groupe de travail au W3C, dédié aux interactions multimodales. Ce groupe a défini en janvier 2011 une architecture [3] pour faciliter l'intégration d'interactions multimodales dans un environnement informatique.

Conceptuellement, l'architecture suppose l'existence d'un bus, d'une couche de transport d'évènements, qui peut être implémentée dans plusieurs mécanismes concrets de transport, principalement utilisés pour le transport des données produites par des capteurs ou envoyées à des effecteurs.

Les capteurs et les effecteurs –en général constitués d'interfaces matérielles- sont gérés par des modules nommés Composant de Modalité -Modality Component-, noté MC par la suite (voir Figure 1). Le MC est en quelque sorte une vision étendue du pilote (driver) : il sert d'interface entre un dispositif et une application via l'architecture MMI ; son mode de communication avec MMI est décrit par la spécification de l'architecture.

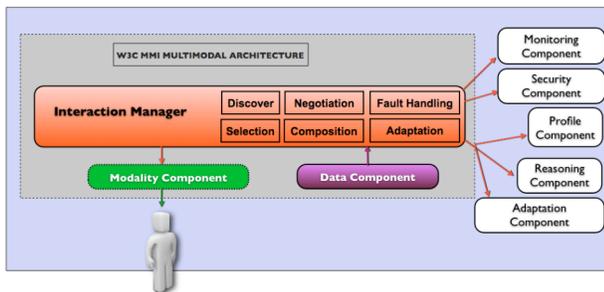


Figure 1 - Architecture Multimodal du W3C

L'architecture introduit un Manager d'Interactions –Interaction Manager, IM- (voir Figure 1). Tous les évènements produits par le MC passent par l'IM. Tous les évènements envoyés aux MC viennent de l'IM. L'IM est donc au cœur de l'interaction multimodale. L'IM gère des comportements associés aux évènements ; il dispose soit de comportements par défaut, soit de comportements spécifiques. De cette manière, chaque application configure un IM pour définir la logique d'interaction associée.

Bien sûr, cette logique peut s'appuyer sur des comportements par défaut, mais elle peut les préciser ou les changer. La façon de spécifier ces comportements est dépendante de l'IM, par exemple à l'aide de langages comme CXML [6] ou d'EcmaScript [7].

Le fait que tous les évènements d'interaction passent par l'Interaction Manager facilite l'implémentation d'un ensemble de fonctionnalités telles que : assurer la cohérence entre plusieurs effecteurs fournissant une perception d'un état de l'application ; assurer la meilleure utilisation des modalités en fonction d'un contexte d'utilisation ; rendre une application utilisable dans un environnement donné par découverte des modalités disponibles.

La spécification laisse hors de son champ la définition de ce qu'est le contexte et la façon d'accéder à sa description, ainsi que la découverte des MC. Cependant, le groupe de travail assure un suivi des meilleures pratiques [30] dans ces domaines afin de fournir, éventuellement, des recommandations relatives à MMI.

Néanmoins, MMI n'a pas prévu d'enrichissement sémantique des informations qui circulent, sont produites et sont traitées par un système MMI.

Nous présentons dans la suite de cet article les motivations d'un ajout du support d'informations sémantiques dans MMI, afin d'offrir de meilleurs services d'interactions multimodales. Notre propos n'est pas d'imposer à tout système MMI des traitements sémantiques, mais d'assurer que la spécification MMI permette l'utilisation d'informations sémantiques au niveau des Composants de Modalité tout comme à un niveau d'abstraction plus haut : le niveau des services multimodaux complexes.

3. SEMANTIQUE ET MODALITES

Dans notre proposition, chaque service de modalité (en tant que système capteur ou en tant que système effecteur) porte des métadonnées concernant sa participation dans des services plus complexes et à plus haut niveau pour un type spécifique d'interaction utilisateur avec le système multimodal.

Par exemple, dans un environnement d'entreprise, une application pourra être bâtie sur un service de haut niveau de guidage, pour lequel le comportement de diverses modalités aura été défini dans le Manager d'Interactions du système.

Pour notre recherche, un système multimodal [18] [19] [20] est :

- Un système avec plusieurs capteurs de données brutes sensorielles à partir de plusieurs canaux physiques, capable d'encoder ces données dans de multiples modes structurés sous la forme d'une ou plusieurs modalités (voir Figure 2).
- Un système avec plusieurs effecteurs décodant les données dans des multiples modes et sous la forme de multiples modalités et exécutant des stimulations sensorielles sur de multiples canaux physiques.
- Un système bidirectionnel capable de combiner des multiples capteurs et effecteurs sous divers modes et modalités.
- Un système dans lequel les données d'entrée ou de sortie peuvent être fusionnées ou dissociées afin d'identifier le sens du comportement de l'utilisateur ou afin de composer en retour un message significatif en utilisant de multiples canaux physiques, modes et modalités.

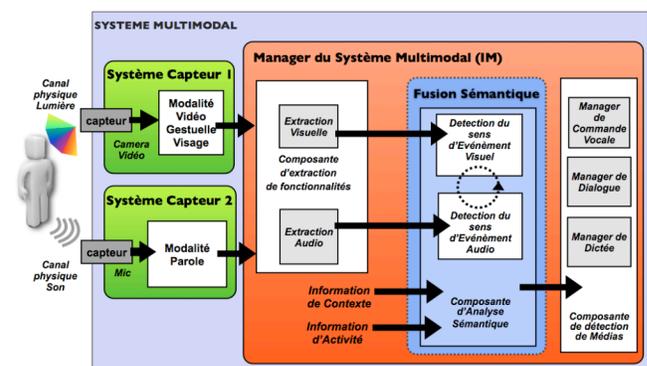


Figure 2 - Système Multimodal Sémantique

Pour identifier le sens du comportement de l'utilisateur, le service fournit des descriptions basées sur les technologies de description sémantique des fonctionnalités multimodales tout comme sur la description des situations d'utilisation (les activités) associées au service dans le contexte d'assistance à un utilisateur final (voir Figure 2).

Nous considérons que l'IM doit prendre en compte les caractéristiques relatives au contexte d'utilisation de l'application ou du service plus complexe de haut niveau tout comme il doit aussi prendre en compte les caractéristiques des modalités selon chaque contexte.

Un exemple simple concerne l'utilisation d'une modalité du type acoustique en environnement bruyant ou dans un environnement où l'espace sonore est 'réservé', comme c'est le cas pendant une conférence.

Il s'agit donc, de produire et d'utiliser des métadonnées capables de décrire les services complexes de haut niveau composés à la volée (dans notre exemple, le service de guidage) et les fonctionnalités de modalité disponibles dans les dispositifs se trouvant dans l'espace du système pervasif [17].

Nous proposons de décrire l'ensemble de services de deux manières: en termes du comportement du service et du contexte d'utilisation du service (activités engagées).

Nous proposons d'utiliser l'attribut `modelReference` (voir Figure 4) prévu originalement dans SAWSDL [2] et repris par le standard WSDL 2.0 [1]. En fonction de l'élément qui sera ainsi annoté, des valeurs issues de différents espaces de nommage seront utilisées pour exprimer différents niveaux de représentation.

Dans le premier mode de description, nous pouvons décrire le comportement du service avec des ontologies OWL-S [8] sur les éléments de l'arbre WSDL qui représentent l'interface d'invocation.

Dans le deuxième mode de description, nous proposons d'utiliser le même attribut mais cette fois-ci pour décrire les possibles situations d'utilisation associées au service et à son exécution avec des ontologies basées sur un ensemble de taxonomies [12] et d'ontologies fondamentales prévues dans la bibliothèque DOLCE Descriptions & Situations [8].

Par ce biais, des informations fonctionnelles et non fonctionnelles peuvent être liées à des accords communs sur la description des services pervasifs multimodaux dans un contexte spécifique.

L'ontologie de contexte prévue dans DOLCE Descriptions & Situations, a déjà prouvé être une base pour une bonne modélisation des ontologies multimédia de base [10], des ontologies de description des événements sociaux [11] et des ontologies en systèmes pervasifs car elle est une démarche de modélisation des situations, sujets, plans, croyances en tant qu'entités et sous la forme d'axiomes.

Cette ontologie de base est un support de référence pour les catégories de haut niveau de la vie réelle qui sont indépendantes d'un domaine applicatif précis comme c'est le cas des événements, les connexions spatiales et temporelles, les situations sociales et les objets informationnels (présentations multimédias, modèles des documents ou widgets GUI par exemple).

Notre hypothèse est que l'utilisation de DOLCE Descriptions & Situations nous donne alors la possibilité de modéliser sémantiquement des informations utiles pour la composition de services multimodaux (fusion) [13] [15] [16], ainsi que pour la restitution multimodale (fission) [14] et les objectifs finaux des utilisateurs lors d'une activité précise dans un espace contrôlé.

Cette modélisation implique trois types d'abstractions qui sont nécessaires pour le bon fonctionnement du système:

1. L'abstraction de l'interface (API) entre les services de haut niveau et les possibles applications clientes et la définition de ces services, comme c'est le cas du service d'agenda ou de notification et ses interfaces (Voir Figure 3).
2. L'abstraction et définition des services matériels de bas niveau unimodaux ou multimodaux au niveau des points de contrôle. Il s'agit des services d'entrée (systèmes capteurs) et des sorties (systèmes effecteurs) associés aux divers dispositifs techniques disponibles dans l'environnement de l'utilisateur.
3. L'abstraction des documents multimédias appropriés pour la situation d'utilisation concernée, le dispositif disponible, la modalité pertinente et le service d'application de haut niveau en jeu. Cette abstraction est à la base des choix d'adaptation de document selon le contexte.

Ces abstractions sont l'apport sémantique nécessaire pour une adaptation au contexte et une composition à la volée plus fine et plus proche des besoins courants de l'activité engagée par l'utilisateur dans un espace donné.

Dans la section suivante, nous allons présenter les choix d'implémentation que nous avons fait en accord avec la modélisation des informations nécessaires pour l'exécution multimodale des fonctionnalités des services de haut niveau.

4. IMPLEMENTATION

Nous avons mis en place une architecture multimodale souple, en mode SOA et enrichie par des services multimodaux pour certains dispositifs basés sur les protocoles UPnP ou Bonjour.

Dans cette architecture - inspirée de la recommandation MMI - nous proposons un composant contrôleur (l'Interaction Manager), un composant qui gère les données (Data Component) et un ou plusieurs composants de modalité (Modality Component) dans chaque nœud de service matériel (voir Figure 3).

Notre modèle adopte de l'architecture MMI la communication asynchrone des événements en mode non ciblé, le couplage lâche et la mise en œuvre du paradigme Modèle Vue Contrôleur (MVC).

Dans notre proposition, les capacités et les caractéristiques des dispositifs sont traitées en tant que services (Modality Component). Ces services sont des composants indépendants avec des interfaces décrites sémantiquement qui peuvent être requêtées comme des boîtes noires, sans aucune connaissance sur leur comportement sous-jacent.

Pour la modélisation des documents multimédia en sortie (composés dans la Vue) nous utilisons le travail de Bernsen [12], qui propose une taxonomie des modalités de sortie, liée aux capacités cognitives et sensorielles humaines avec les notions de profil et de média.

Nous avons donc choisi de nous appuyer sur cette taxonomie pour la composition des documents de sortie où l'on a besoin de décider quelle est la représentation optimale des informations que le système et l'utilisateur sont amenés à échanger au cours de leur

tâche commune.

Ceci nous a permis d'implémenter une boîte à outils de 'templates' de documents adaptés au contexte qui servent pour un paramétrage « par l'exemple ».

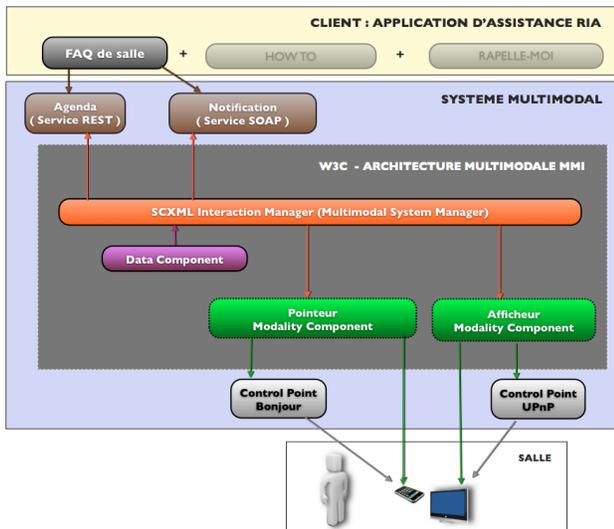


Figure 3 - Architecture Proposée

Lors de l'implémentation nous avons aussi étendu un framework OpenSource de génération de webservices en PHP (NuSOAP) pour pouvoir implémenter les services annotés sémantiquement avec les dernières normes SOAP et WSDL 2.0 et la technologie REST.

Grâce à ce framework nous pouvons maintenant assez rapidement générer un service de base qui tient compte des contraintes de publication et découverte sémantiques et des besoins d'évaluation du système, à savoir, les traces d'exécution, la validation de fonctionnalités et des performances, l'extensibilité ou la montée en charge.

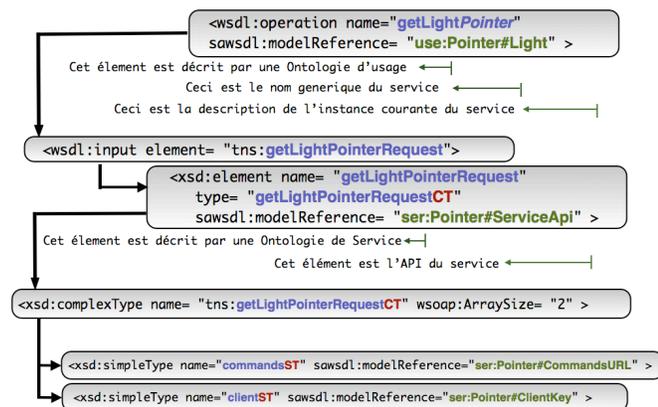


Figure 4 - Exemple d'annotation sémantique

Basés sur ce générateur, nous avons développé un certain nombre de services de base dont la description est annotée sémantiquement selon deux types d'annotations : la sémantique fonctionnelle (Ontologie de service) et la sémantique non fonctionnelle de haut niveau qui apporte les informations sur les activités individuelles ou collectives concernées par le service (Ontologie d'usage). Notre approche est donc basée sur l'utilisation de deux types d'ontologies lors des annotations SAWSDL dans la description du service avec WSDL 2.0 (voir Figure 4).

Finalement nous avons aussi implémenté un point de contrôle pour la découverte de services abstraits que nous avons proposés. Il s'agit d'une passerelle de découverte de services Bonjour/UPnP orientée sémantiquement.

Ce point de contrôle gère aussi un compagnon d'assistance, qui est une application client pour iPhone. Cette application d'assistance fait appel à deux services de haut niveau pour composer une fonctionnalité multimodale de FAQ de salle : les services d'agenda de la salle et le service de notification. Chacun de ses services de haut niveau pouvant à son tour se servir des services matériels de bas niveau proposés par les Composants de Modalité.

Le travail qui reste à faire concerne principalement la tâche de décrire les activités et les situations avec des ontologies et de valoriser pleinement l'interopérabilité avec une composante externe de prise de décision. Nous espérons que ce travail nous permettra de tester notre approche d'abstraction et d'évaluer les résultats définitifs.

5. CONCLUSION

Dans cet article, nous avons présenté une implémentation SOA de l'architecture MMI enrichie par une approche sémantique.

L'architecture MMI se révèle être un support efficace pour intégrer des capteurs et des effecteurs et créer des interfaces multimodales dans des services et des applications pervasifs.

Nous avons proposé les bases de la prise en compte dans cette architecture d'informations sémantiques permettant d'affiner l'usage de la multimodalité en s'appuyant sur les technologies de Web Services.

La faisabilité d'une telle approche est montrée par l'implémentation présentée dans cet article. La démonstration de la pertinence de l'approche sera l'objet de nos prochains travaux.

6. REMERCIEMENTS

Cette recherche est faite dans le cadre du projet SOA2M du laboratoire commun Ubimedia entre Alcatel-Lucent Bell Labs et l'Institut Telecom.

7. REFERENCES

- [1] Web Services Description Language (WSDL) Version 2.0, <http://www.w3.org/TR/wsd20/>
- [2] Semantic Annotations for WSDL and XML Schema, <http://www.w3.org/TR/sawSDL/>
- [3] Multimodal Architecture and Interfaces, <http://www.w3.org/TR/mmi-arch/>
- [4] Universal Plug and Play activity, <http://www.upnp.org/>
- [5] Digital Living Network Alliance, <http://www.dlna.org/>
- [6] Call Control Extensible Markup Language, <http://www.w3.org/TR/ccxml/>

- [7] Specification officielle EcmaScript, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [8] Semantic Markup for Web Services, <http://www.w3.org/Submission/OWL-S/>
- [9] MASOLO C., BORGO S., GANGEMI A., GUARINO N., OLTRAMARI A. and SSCHNEIDER L. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology. WonderWeb Deliverable D18, Final Report, vr. 1.0, 2003. <http://wonderweb.semanticweb.org>
- [10] SCHERP, A., FRANZ, T., SAATHOFF, C. and STAAB, S. « F--a model of events based on the foundational ontology dolce+DnS ultralight » In : *Proceedings of the fifth international conference on Knowledge capture*, September 01-04, 2009, Redondo Beach, California, USA doi>10.1145/1597735.1597760
- [11] OBERLE D., ANKOLEKAR A., HITZLER P., CIMIANO P., SINTEK M., KIESEL M., MOUGOUÏE B., BAUMANN S., VEMBU S., ROMANELLI M., BUITELAAR P., ENGEL R., SONNTAG D., REITHINGER N., LOOS B., ZORN H-P., MICELLI V., PORZEL R., SCHMIDT C., WEITEN M., BURKHARDT F. and ZHOU, J. « DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO) » In : *Web Semantics: Science, Services and Agents on the World Wide Web*, v.5 n.3, p.156-174, September, 2007 doi>10.1016/j.websem.2007.06.002
- [12] BERNSEN, O. "Foundations of multimodal representations. A taxonomy of representational modalities". In: *Interacting with Computers*, 1994, Vol. 6, No. 4, p. 347-371. [http://dx.doi.org/10.1016/0953-5438\(94\)90008-6](http://dx.doi.org/10.1016/0953-5438(94)90008-6)
- [13] DASARATHY, B. "Sensor Fusion Potential Exploitation-Innovative Architectures and Illustrative Applications" In: *Proceedings of the IEEE*, 1997. Vol. 85. No.1. pp 24-38. <http://dx.doi.org/10.1109/5.554206>
- [14] FOSTER, M.E. State of the art review: Multimodal fission. COMIC project Deliverable 6.1, September 2002. Project Number IST-2001-32311. University of Edinburgh.
- [15] HALL, D.L and LLINAS J. "An Introduction to Multisensor Data Fusion", In: *Proceedings of the IEEE*, 1997. Vol. 85. No 1. pp. 6-23 <http://dx.doi.org/10.1109/5.554205>
- [16] NIGAY, L. and COUTAZ, J. "A Desing Space for Multimodal Systems Current Proceession and Data Fusion" In: *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems, 1993*. Amsterdam, April 24-29. pp. 172-178. <http://dx.doi.org/10.1145/169059.169143>
- [17] SINGH, S., PURADKAR, S., and LEE, Y. "Ubiquitous computing: Connecting pervasive computing through semantic web". In: *Information Systems and e- Business Management Journal*. Springer Berlin / Heidelberg 2005 <http://dx.doi.org/10.1007/s10257-005-0003-8>
- [18] NIGAY, L. and COUTAZ, J. "Espaces conceptuels pour l'interaction multimédia et multimodale" In: *TSI, spécial Multimédia et Collecticiel*. AFCET & Hermes Publ., 1996. Vol. 15 No.9. pp. 1195-1225 <http://iihm.imag.fr/publs/1996/TSI96-Mu.pdf>
- [19] JAIMES, A. and SEBE, N. "Multimodal human-computer interaction: A survey". In: *Computer Vision and Image Understanding*. Elsevier Science Inc. New York, NY, USA . 2007 <http://dx.doi.org/10.1016/j.cviu.2006.10.019>
- [20] DUMAS, B., LALANNE, D. and OVIATT, S. "Multimodal Interfaces: A Survey of Principles, Models and Frameworks" In: *Human Machine Interaction: Research Results of the MMI Program*, Springer-Verlag, Berlin, Heidelberg, 2009 http://dx.doi.org/10.1007/978-3-642-00437-7_1
- [21] GARLAN, D., SIEWIOREK, D., SMAILAGIC, A. and STEENKISTE, P. "Project Aura: Toward Distraction-Free Pervasive Computing", In: *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22-31, Apr.-June 2002. <http://doi.ieeecomputersociety.org/10.1109/MPRV.2002.1012334>
- [22] NIGAY, L. and COUTAZ, J. "A Desing Space for Multimodal Systems Current Proceession and Data Fusion" In: *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems, 1993*. Amsterdam, April 24-29. pp. 172-178. <http://dx.doi.org/10.1145/169059.169143>
- [23] OVIATT, S.L., COHEN, P.R., WU, L., VERGO, J., DUNCAN, L., SUHM, B., BERS, J., HOLZMAN, T., WINOGRAD, T., LANDAY, J., LARSON, J. and FERRO, D. "Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions." In: *Carroll, J. (ed.) Human-Computer Interaction in the New Millennium*. Addison-Wesley Press, Reading, 2001. Ch. 19, pp. 421-456. doi:10.1207/S15327051HCI11504_1
- [24] VAZQUEZ, J. I., LOPEZ DE IPIÑA, D. and SEDANO, I. "Soam: A web-powered architecture for designing and deploying pervasive semantic devices". In: *International Journal of Web Information Systems*, 2007 <http://dx.doi.org/10.1108/17440080780000301>
- [25] MASUOKA R., PARSIA B., and LABROU, Y. "Task computing - the semantic web meets pervasive computing". In: *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia - MUM04*. pages 866-881, 2003. <http://doi.acm.org/10.1145/1052380.1052416>
- [26] GRIPAY, Y., PIERSON J., PIGEOT C.E. "Une architecture pervasive sécurisée : PerSE", In: *UbiMob'06*, ACM ed., 2006, Paris. pp. 147-150
- [27] DEES, W. and SHRUBSOLE, P. 2007. "Web4CE: accessing web-based applications on consumer devices". In: *Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007)*. WWW '07. ACM, New York, NY, 1303-1304. <http://dx.doi.org/10.1145/1242572.1242820>
- [28] CONCOLATO C., LE FEUVRE J. and DUFOURD J. C. "Declarative Interfaces for Dynamic Widgets Communications" In: *Proceedings of the 9th ACM Symposium on Document Engineering*, Munich, Germany, pp. 241-244. <http://dx.doi.org/10.1145/1600193.1600245>
- [29] PELLAN, B. and CONCOLATO, C. "Authoring of scalable multimedia documents" In: *Multimedia Tools and Applications* 43, 3 (July 2009), pp. 225-252 <http://dx.doi.org/10.1007/s11042-009-0268-x>
- [30] Best Practices for creating MMI Modality Components, <http://www.w3.org/TR/2011/NOTE-mmi-mcbp-20110301>

Un DSML pour l'automatisation du test d'applications mobiles

Youssef Ridene
Neomades
Technopole Izarbel
64201 Bidart, France

youssef.ridene@univ-pau.fr

Franck Barbier
LIUPPA
Avenue de l'Université
64013 Pau, France

franck.barbier@univ-pau.fr

ABSTRACT

L'engouement du grand public pour les applications mobiles, dont le nombre ne cesse de croître, a rendu les utilisateurs de plus en plus exigeants quant à la qualité de ces applications. Seule une procédure de test efficace permet de répondre à ces exigences. Dans le contexte des applications embarquées sur téléphones mobiles, le test est une tâche coûteuse et répétitive principalement à cause du nombre important de terminaux mobiles tous différents les uns des autres. Nous présentons dans cet article le langage MATeL (Mobile Application Testing Language), un DSML (Domain-Specific Modeling Language) qui permet de décrire des scénarios de test spécifiques aux applications mobiles. Par ailleurs, il permet d'enrichir ces scénarios avec des points de variabilité afin de définir des variations dans le test en fonction des particularités d'un mobile ou d'un ensemble de mobiles.

Categories and Subject Descriptors

D.2.5 [Testing and Debugging]: Testing tools D.2.6 [Software Engineering]: Programming Environments - graphical environments

General Terms

Design, Languages, Verification.

Keywords

DSML, Model-Driven Testing, Embedded Mobile Applications, Software Product Lines, Variability.

1. INTRODUCTION

Le développement d'applications mobiles est une tâche complexe principalement à cause du syndrome de la « fragmentation » [1]. En effet, le développeur est obligé de produire un code spécifique de la même application pour s'adapter aux spécificités de chaque terminal mobile afin que cette application ait un comportement uniforme sur tous les terminaux. La gestion de code distinct pour répondre aux besoins de chaque mobile est extrêmement coûteuse en particulier pour la phase de test [2]. Malgré le slogan Java *Write Once, Run Anywhere*, la pratique montre que le fonctionnement de programmes Java ME¹ (Mobile Edition) n'est pas le même *in situ* (une fois déployés sur un périphérique donné). C'est principalement pour cette raison que le test sur des

simulateurs tels que WTK² ou Android SDK³ n'est pas suffisant [3-4]. Même si une partie importante de la vérification du programme peut être effectuée pendant le développement grâce à l'utilisation des simulateurs, le test d'applications mobiles ne peut être exhaustif que sur les plateformes de déploiement finales. En effet, les simulateurs ne prennent pas en charge toutes les caractéristiques des terminaux et les extensions de chaque plateforme.

Au-delà de la plateforme Java, nous constatons que les spécificités matérielles des appareils telles que les propriétés des écrans, la mémoire disponible, l'espace de stockage, la vitesse du processeur ou encore les protocoles de communication supportés imposent des adaptations particulières pour chaque application. Pour atteindre le plus grand nombre de clients, les développeurs sont contraints de proposer une application compatible avec les plateformes mobiles les plus répandues comme Android, iOS, BlackBerry, Bada ou encore Java ME. Ces plateformes sont souvent basées sur des langages de programmation différents, des API différentes (avec des versions différentes). Ainsi, la fragmentation des appareils transforme l'idée de créer et de distribuer une application mobile en un problème complexe. Dans l'optique d'apporter une solution à la fragmentation mobile, certains travaux ont vu le jour. Il s'agit principalement d'outils qui permettent à partir d'un langage d'entrée unique de générer des binaires compatibles avec différentes plateformes mobiles [5-7]. Ces outils se basent sur l'utilisation d'une base de données renseignée avec toutes les caractéristiques des mobiles qui sont prises en compte lors de la génération des binaires.

Une fois déployée sur des centaines d'appareils mobiles différents, l'application doit avoir « le comportement attendu et souhaité par le développeur ». Des bugs résiduels, qui ne sont pas détectés au moment du développement, peuvent persister. Ils sont souvent liés aux spécificités de l'environnement de déploiement, *i.e.* principalement l'opérateur de télécommunications et les spécificités du périphérique. Ainsi, les différentes versions du logiciel doivent être testées en relation forte avec les téléphones réels, de sorte que le « comportement attendu » soit contrôlé et donc testé lors de l'exécution de l'application par l'utilisateur.

L'achat de centaines d'appareils et la réalisation, de façon industrielle et à grande échelle, des activités de test pour valider

¹ Java ME: <http://www.oracle.com/technetwork/java/javame>

² Wireless Toolkit de Sun

³ SDK Android: <http://developer.android.com/sdk/>

chaque application sur chaque mobile, nécessitent un investissement important en termes de coûts et d'organisation (équipe de testeurs, processus de test, outils de saisie de bug, etc.).

Afin d'apporter une solution à la problématique de la répétitivité et du coût du test, nous présentons dans ce papier, MATeL [8] ; un DSML, *i.e.*, un méta-modèle et son environnement (un banc de test et son infrastructure logicielle) comme une approche possible pour l'automatisation de tests d'applications mobiles. MATeL utilise des modèles graphiques pour exprimer non seulement des scénarios de test élémentaires, mais aussi des points de variabilité. L'environnement complet de MATeL permet également l'exécution en ligne, le suivi et la gestion (enregistrement, suivi...) des tests et des résultats associés. Les mobiles sont accessibles et utilisables à distance via Internet. Ils sont physiquement connectés au banc de test qui est contrôlé par un *middleware* dédié qui se charge aussi de l'exécution des instructions MATeL.

Le document est structuré comme suit : la section 2 présente les solutions existantes pour tester les applications mobiles et leurs limitations. Dans la section 3, nous développons une vision qui met l'accent sur le *Model-Driven Testing* comme une technique adéquate pour exprimer des tests de manière intuitive. Nous expliquons le rôle, le positionnement et les avantages d'un DSML dans un environnement de test professionnel. Dans la section 4, nous montrons l'apport de MATeL. Enfin, dans la section 5 nous discutons les perspectives pour conclure dans la section 6.

2. Solutions existantes et limitations

Nous présentons dans cette section les différentes approches possibles pour le test d'applications mobiles ainsi que leurs limitations.

2.1 Offshore vs. Crowdsourcing Testing

La validation d'applications mobiles est principalement basée sur des actions effectuées par les utilisateurs (appuyer sur une suite de touches, cliquer sur des zones de l'écran, basculer le mobile en mode paysage, etc.). Elle s'appuie également sur des interprétations visuelles et/ou sonores pour vérifier l'IHM, la connectivité et les communications réseaux ou encore pour s'assurer que la musique est jouée correctement. Souvent, le nombre d'actions et leurs propriétés varient d'un mobile à un autre (la touche à utiliser, la zone d'écran à toucher...).

De nos jours, la solution la plus utilisée, pour réduire le coût des tests, est de déléguer cette activité à des sociétés offshore à faible coût. Pour les projets mobiles externalisés, le problème majeur est lié aux caractéristiques du réseau mobile. En effet, certaines applications mobiles peuvent fonctionner correctement sur certains réseaux, et pas sur d'autres en raison de restrictions techniques (ports bloqués, limitations de bande passante, etc.). De plus, à cause de la délocalisation, la difficulté de recueillir et de réinterpréter les résultats des corrections est élevée. L'externalisation des tests ne peut pas être la panacée en tenant compte de la forte réactivité du marché et des tendances.

La deuxième solution est connue sous le nom *Crowdsourcing Testing*. Le principe est de s'appuyer sur une communauté de testeurs. Ces derniers sont des utilisateurs normaux qui sont payés pour effectuer des tests sur leur propre mobile [9]. Cette solution peut effectivement être efficace pour une première phase de beta test dont le but est de valider certains points clés d'une application tels que la facilité d'utilisation, la fluidité ou encore le *Game Play*

s'il s'agit d'un jeu. Toutefois, elle ne peut servir de test final rigoureux et complet. En effet, les membres de la communauté n'étant pas des testeurs professionnels, le développeur ne pourra pas compter sur leur contribution pour avoir des tests exhaustifs.

2.2 Cloud Testing

Afin de résoudre la problématique de procuration des téléphones mobiles et de suivre l'évolution effrénée du marché, certaines entreprises ont lancé des plateformes de test à distance (banc de test en ligne). L'idée générale est de fournir un service Internet qui permet d'accéder et de manipuler de vrais téléphones mobiles via le Web (cf. Fig. 1). Dans cet article, nous proposons une approche qui enrichit ces plateformes de tests telle que celle proposée par [10].

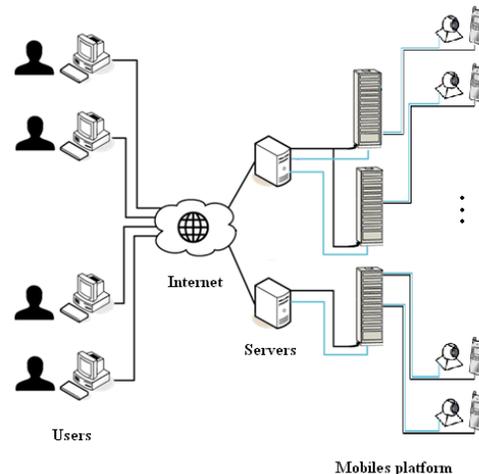


Fig. 1 Architecture globale d'une plateforme de *Cloud Testing*

Fig. 1 montre l'architecture globale d'une solution de test en ligne. L'idée est basée sur une infrastructure logicielle dédiée (*middleware*) avec des téléphones réels connectés à des serveurs Web via des bancs de test. Une caméra vidéo est placée en face de chaque appareil pour filmer son écran, les kits mains libres des mobiles sont utilisés pour capturer les sorties audio et les touches sont électroniquement câblées afin de pouvoir effectuer des appuis (Fig. 2). Une application Web permet de se connecter au banc de test et utiliser n'importe quel téléphone disponible.

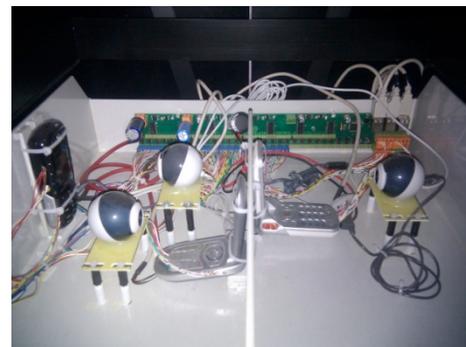


Fig. 2 Quelques mobiles branchés au banc de test

Une fois la connexion au serveur du banc de test établie, l'application affiche à l'utilisateur le téléphone mobile et joue, en temps réel, l'audio et la vidéo transférés à partir du banc de test. L'utilisateur est capable d'envoyer des commandes (manipuler le

clavier, toucher un écran tactile, brancher/débrancher le chargeur ou un câble USB...) vers le téléphone mobile en utilisant la souris et le clavier de son ordinateur. Les téléphones du banc de test sont connectés aux réseaux mobiles, ceci permet de tester des applications connectées, mais également d'analyser le comportement des applications lors de la réception d'événements externes comme les appels entrants, SMS, MMS, etc.

Les plateformes de *Cloud Testing* existantes, telle que [10] permettent aux testeurs d'enregistrer une séquence d'actions en les réalisant sur un téléphone donné. Un scénario basé sur un langage propriétaire est alors généré. L'utilisateur ne peut le (re)jouer que sur un téléphone mobile similaire. Ceci est justifié par les raisons que nous avons mentionnées dans la première partie, *i.e.* chaque mobile a ses propres caractéristiques matérielles et logicielles. Par exemple, si l'utilisateur enregistre un test sur un téléphone mobile en utilisant les touches de validation gauche et droite, le scénario généré ne fonctionnera pas sur un mobile n'ayant pas ces touches comme par exemple la majorité des mobiles de marque BlackBerry ou encore les téléphones à écran tactile. Les deux points faibles de ces produits industriels sont l'utilisation d'un langage propriétaire (loin des standards du marché et des langages de modélisation répandus tel que UML (*Unified Modeling Language*) [11]) et l'absence de la gestion de la variabilité dans la conception des scénarios alors que le besoin de partager les scénarios entre les mobiles est élevé.

3. MATeL

3.1 Description

Nous proposons MATeL, un DSML qui permet la description de scénarios de test dans lesquels les points communs et les points de variabilité entre les différents téléphones mobiles peuvent être exprimés d'une manière intelligible. Différents chemins d'exécution sont alors générés en fonction de ces points de variation. La syntaxe concrète de MATeL ressemble à celle des diagrammes de séquences UML. Ces diagrammes ont pour but de montrer la séquence des interactions entre objets ou acteurs (entre testeur et mobile dans notre cas) sur un axe de temps donné. Le diagramme est constitué de deux dimensions. La dimension verticale montre la séquence des messages dans le temps. La dimension horizontale montre les instances des objets auxquelles sont envoyés les messages. Le temps est représenté du haut vers le bas le long des lignes de vie (*Lifetime Line* en anglais). Les messages sont représentés par des flèches d'un acteur vers un autre (de testeur à mobile ou de mobile à mobile). Afin de résoudre la problématique de la répétitivité des tests et de l'absence de valeurs ajoutées, MATeL aide le testeur dans la description d'un scénario de test et dans son exécution sur n'importe quel téléphone connecté au banc de test.

MATeL se base sur un métamodèle et un ensemble de règles OCL. Les scénarios de test sont des modèles conformes à ce métamodèle, ceci est équivalent au principe classe/instance dans les technologies orientées objet. Disponible dans l'IDE Eclipse, MATeL bénéficie des apports d'*Eclipse Modeling Framework* (EMF) et de *Graphical Editing Framework* (GEF) [12]. Ce dernier permet la génération d'un langage graphique basé sur des icônes, des images ou encore des formes toutes choisies par le concepteur du langage. L'environnement Eclipse implémente de nombreux standards ouverts et regroupe une grande communauté ce qui facilite l'adoption, l'extension ou encore la personnalisation de MATeL.

3.2 Composants de MATeL

MATeL est composé de différentes parties qui permettent aux concepteurs de test d'avoir un grand nombre de fonctions et de concepts lui offrant la possibilité d'enrichir ses scénarios.

Éléments de base

Pour un scénario de test de base, un testeur, un mobile et les actions effectuées sur ce dernier sont indispensables (cf. Fig. 3). MATeL fournit une liste complète des actions possibles en fonction des capacités de chaque téléphone mobile (appui sur une touche, toucher l'écran, glisser le pointeur, etc.). En pratique, un scénario contient au moins deux lignes de vie et au plus trois :

- **Un testeur** : il s'agit d'une ligne de vie obligatoire qui désigne l'utilisateur qui exécute les actions sur le ou les téléphones mobiles. MATeL fournit une liste de toutes les actions qui peuvent être effectuées. Ces actions peuvent être « atomiques » comme par exemple appuyer sur une touche ou « combinées » comme écrire un texte ;
- **Un premier mobile** : une ligne de vie obligatoire qui correspond aux téléphones mobiles sous test ;
- **Un deuxième mobile** : une ligne de vie optionnelle qui peut être utilisée pour générer des interruptions sur le mobile sous test (SMS, appels entrants, connexion Bluetooth, etc.).

3.3 Apports de MATeL

L'approche *Model-Driven Testing* améliore la qualité des logiciels en général [13]. En effet, l'utilisation d'un DSML facilite grandement la conception de scénarios de test car ces derniers se basent sur des éléments étroitement liés au domaine ciblé [14]. De plus, aucune connaissance en programmation n'est requise. En effet, MATeL fournit des éléments du domaine du test d'applications mobiles (mobiles, testeurs, événements entrants... et surtout les interactions entre eux) pour permettre aux utilisateurs de se concentrer sur la conception du test et non plus le répéter sur des centaines de terminaux.

4. Gestion de la variabilité dans MATeL

La gestion de la variabilité a été explorée dans divers travaux comme une solution à la fragmentation des terminaux mobiles [15-17]. Ces travaux ont pour but d'apporter des solutions pour les fabricants de mobiles afin de réduire les coûts et accélérer le temps de mise sur le marché de nouveaux périphériques. MATeL se base sur une approche similaire appliquée à la conception de scénarios de test.

4.1 La variabilité au niveau des mobiles

Un scénario de test comprend les étapes suivantes:

1. Configurer le téléphone
 - a. Insérer la carte SIM
 - b. Démarrer le téléphone
 - c. Régler l'heure et la date
2. Configurer les connexions nécessaires (GPRS, Wi-Fi, Bluetooth, USB...)
3. Installer l'application à tester
4. Lancer l'application et commencer le scénario de test

5. Générer des interruptions en utilisant un autre téléphone
6. Supprimer l'application et effacer toutes les données propres à l'utilisateur

Cette variabilité est également avérée lorsque l'utilisateur essaye de naviguer dans les menus du téléphone ou lorsqu'il essaye de le configurer. Pour cette raison, nous proposons une liste de fonctions incontournables pour le test :

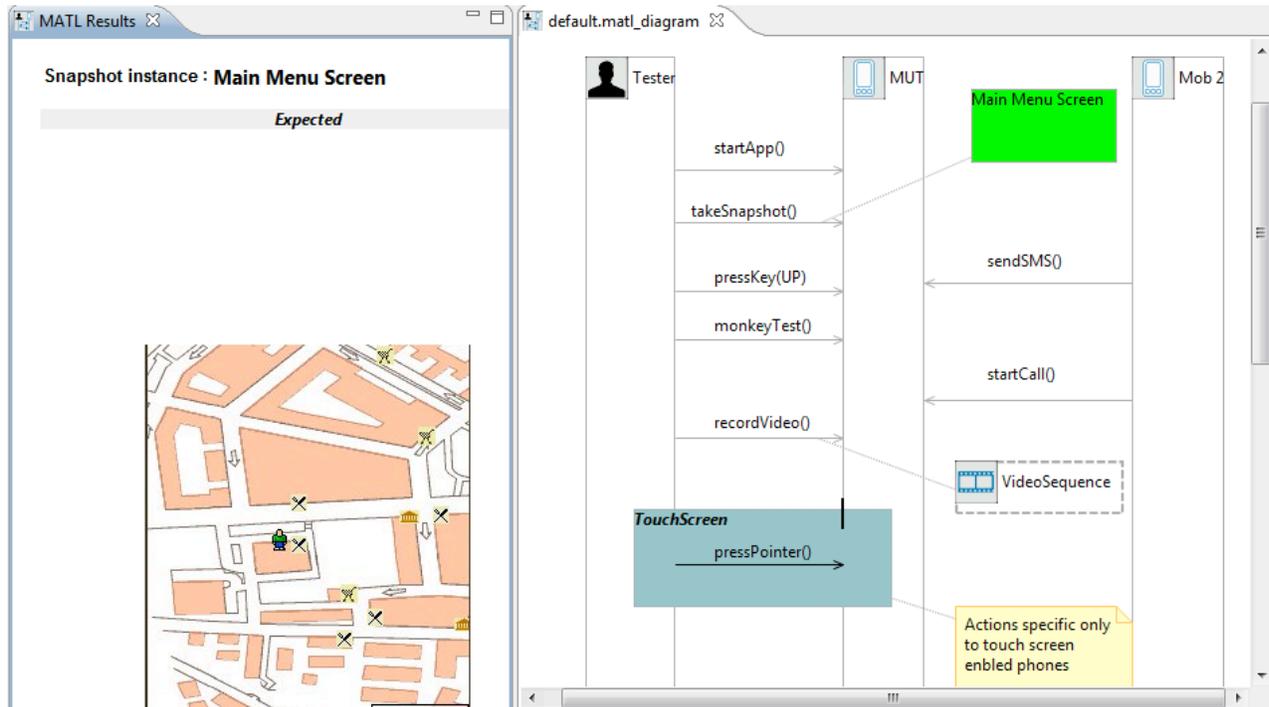


Fig. 3 Exemple de scénario

Selon le modèle du mobile à tester, les actions nécessaires pour accomplir ces différentes étapes varient (pas les mêmes menus, pas les mêmes touches...). C'est la raison pour laquelle MATeL fournit des « fonctions automatisées » qui permettent de réduire le temps nécessaire avant de démarrer le test effectif des applications (étape 4). Par exemple, nous fournissons la fonction *SendSMS* où le testeur ne fournit que le texte qui sera envoyé. Grâce à une base de données renseignée avec toutes les caractéristiques de chaque téléphone du banc (propriétés de l'écran, type de clavier, mappage de claviers, entrées/sorties disponibles, capteurs (mode paysage, rotation...), les menus de navigation, etc.), nous sommes en mesure d'adapter la fonction d'envoi de SMS à chaque mobile.



Fig. 4 Exemple de quelques claviers

Fig. 4 présente trois différentes configurations de clavier mobile, le premier (de gauche à droite) est un clavier virtuel pour téléphones Android, le second est un clavier classique d'un Sony Ericsson G502 et le dernier est un clavier QWERTY d'un Nokia E61. Il est évident que la séquence de touches sur lesquelles il faut appuyer et les zones de l'écran à toucher, pour écrire le même texte, seront différentes d'un mobile à un autre.

- **SetWiFi** : permet de connecter le mobile à un réseau Wi-Fi décrit par l'utilisateur ;
- **SetGPRS** : connecte automatiquement le mobile au réseau GPRS. Cette fonction est particulièrement utile vu que l'utilisateur n'a plus besoin d'apprendre toutes les informations sur les *Access Point Names* (APN), les paramètres de passerelle WAP et éventuellement les paramètres de connexion exigés par l'opérateur de télécommunication ;
- **SetDateAndTime** : cette configuration est essentielle pour le test d'applications utilisant des connexions sécurisées (HTTPS par exemple) ;
- **SetApplicationsConnectivity** : définit automatiquement les autorisations nécessaires pour les connexions Internet ;
- **SendSMS** ou **SendMMS**, **WriteText** ou **Call** : de nombreuses applications demandent à l'utilisateur de saisir un texte, le banc de test est en mesure de produire les séquences claviers nécessaires en se basant sur le texte à écrire ou le numéro de téléphone à composer ;
- **ConnectBluetooth** : lance une connexion Bluetooth ;
- **MonkeyTest** : il s'agit de plusieurs appuis successifs sur des touches différentes (choisies par le concepteur du scénario). Ceci sert à vérifier la robustesse de

l'application et sa capacité à gérer des entrées inattendues ;

- **StartApp** : cette fonction parcourt le système de fichiers du mobile et lance l'application à tester ;
- **RemoveApp** : une fois le test terminé, l'utilisateur peut utiliser cette fonction pour désinstaller l'application ;
- **ClearBrowserHistory, ClearCookies, ClearCache** : afin de ne pas divulguer les données privées de chaque utilisateur du banc de test, nous fournissons des fonctions pour les effacer automatiquement.

4.2 La variabilité au niveau de l'application

Comme nous l'avons mentionné dans l'introduction de cet article, la même application mobile doit être adaptée selon les téléphones ciblés. Dans l'esprit des lignes de produit logiciel, cinq différents types de variabilité sont définis dans MATeL :

- Le premier est relatif aux modèles des mobiles. Par exemple : Nokia N8, BlackBerry 9800 Torch, etc. ;
- La deuxième repose sur les spécificités de chaque mobile. L'éditeur de MATeL fournit une liste prédéfinie (*isTouchScreen, hasGPS, hasQWERTYKeyboard, hasAZERTYKeyboard, hasLandscapeMode* et *hasBluetooth*) à partir de laquelle l'utilisateur peut spécifier un ou plusieurs types ;
- La troisième est dédiée aux résolutions d'écran. L'utilisateur peut spécifier la résolution qu'il souhaite. Ceci est possible en fournissant la largeur et la hauteur de l'écran du mobile ;
- La quatrième est liée à la version de l'application elle-même (pour la gestion des modifications ou évolutions entre versions) ;
- La dernière est liée à la plateforme mobile (Android, Java ME, iOS, BlackBerry, Windows Phone, etc.).

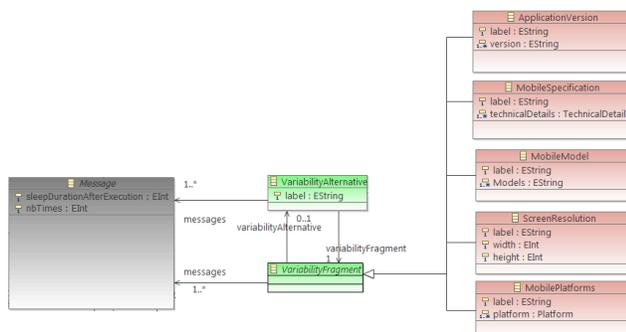


Fig. 5 Métatypes pour la gestion de la variabilité

Ces cinq types de variabilité sont fournis par des spécialistes en développement d'applications mobiles. Selon eux, une application peut être adaptée principalement en raison de ces variations. En pratique, l'utilisateur est en mesure d'attacher des actions spécifiques en fonction de ces points de variabilité. Ces concepts sont disponibles dans MATeL, grâce aux métatypes *ApplicationVersion, MobileSpecification, MobileModel, ScreenResolution* et *MobilePlatforms* qui héritent tous du métatype générique *VariabilityFragment* (Fig. 5). L'utilisateur

peut manipuler directement ces concepts dans un scénario de test. Un ou plusieurs messages peuvent être assignés à une instance de *VariabilityFragment*.

Par exemple, considérons une application dans laquelle le développeur a décidé d'ajouter un écran de publicité pour inciter les utilisateurs à télécharger d'autres applications. Cet écran ne sera ajouté que pour les téléphones mobiles ayant une résolution de 240x320 pixels. Cela signifie que, lors du test, une action (pour annuler ou valider cet écran) doit être omise pour les téléphones mobiles ayant une résolution différente de 240x320 pixels. Grâce à la gestion de la variabilité fournie par MATeL, c'est assez simple de concevoir un tel point de variation.

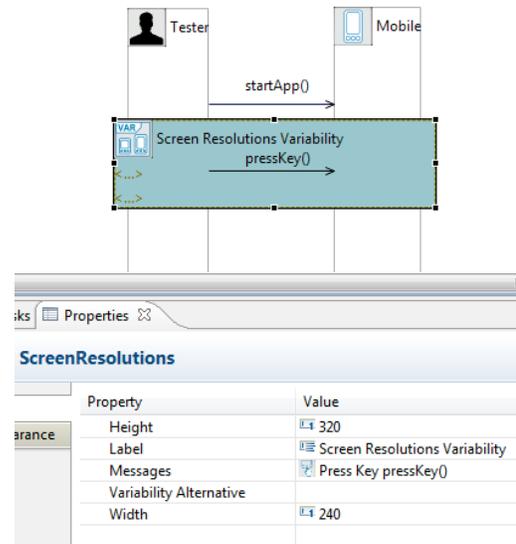


Fig. 6 Exemple de modélisation de variabilité

Dans la Fig. 6, nous voyons une variabilité appelée *Screen Resolutions Variability* (rectangle grisé) à laquelle nous avons attribué une action *PressKey*. Cette variabilité est une instance du métatype *ScreenResolution* paramétrée par une résolution de 240x320 pixels. Grâce à la base de données, le banc de test est capable de réaliser cette action uniquement sur les mobiles ayant la bonne résolution d'écran.

En pratique, l'action *PressKey* est attachée à la variabilité *Screen Resolutions Variability* qui est une instance du métatype *ScreenResolutions*. En utilisant l'onglet *Propriétés* de l'éditeur (Fig. 6, partie basse), nous pouvons spécifier la hauteur et la largeur de l'écran. Lors de l'exécution du scénario sur le banc de test, il est relativement facile d'exécuter toutes les séquences communes en prenant en compte les points de variabilité. Ainsi, cette action est exécutée uniquement sur les téléphones appropriés. Le métatype *VariabilityAlternative* (Fig. 5) peut être utilisé si un autre type d'action doit être effectuée sur d'autres téléphones mobiles.

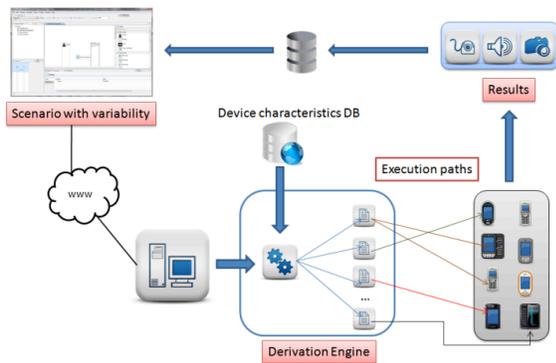


Fig. 7 Processus d'exécution des scénarios MATeL

Grâce aux fonctionnalités de MATeL, le testeur conçoit un scénario de test unique pour tous les mobiles ciblés ; le banc de test l'exécute correctement sur chaque appareil en suivant le flux d'exécution schématisé en Fig. 7. De plus, les scénarios peuvent être réutilisés pour des applications similaires.

5. Perspectives

Les travaux présentés ici sont bien adaptés pour les tests de type « boîte noire ». Une des limites de ce type de test est la difficulté d'évaluer précisément les caractéristiques des systèmes testés. Par exemple, il n'est pas possible de déterminer l'état de l'application dans lequel une erreur s'est produite. Nous avons identifié deux principaux problèmes à résoudre afin de permettre une meilleure interprétation de ces erreurs : (i) l'incapacité importante d'introspection des applications mobiles et (ii) une difficulté à concrétiser le lien entre les spécifications de l'application et son implémentation effective. Nous proposons d'améliorer cette plateforme en explorant une nouvelle direction. Nous sommes en train de travailler sur un modèleur d'applications mobiles dans lequel nous pouvons concevoir une application mobile comme une succession d'écrans. Ceci permettra un meilleur contrôle de tous les états/transitions lors de l'exécution de l'application. Pour chaque action, les pré-conditions et les post-conditions sont *a priori* connues. Ainsi, nous pouvons générer automatiquement le scénario de test MATeL.

6. Conclusion

Nous avons présenté dans cet article les défis et les grandes tendances liés au test d'applications mobiles en raison de la fragmentation. Quelques plateformes dédiées permettant l'accès à distance à des téléphones réels sont aujourd'hui disponibles. En raison de son ouverture et ses bases normatives (EMF, UML...), MATeL peut être facilement adapté et/ou étendu pour se connecter à n'importe quel banc de test industriel. Une fois la version industrielle du produit entièrement finie, nous avons l'intention d'expérimenter et d'évaluer une nouvelle approche pour la conception d'applications mobiles qui nous permettra d'enrichir notre proposition et ainsi obtenir une procédure plus sophistiquée pour tester des applications mobiles. En effet, nous travaillons sur la génération automatique de scénario MATeL. Notre approche est basée sur la conception de l'application au moyen de trois composantes principales : des écrans, des transitions entre ces écrans et des événements permettant de telles transitions. En rassemblant ces informations, nous pouvons facilement générer le scénario de test MATeL et le code source de

l'application. Ainsi, nous serons capables de surveiller le comportement de l'application pendant son exécution.

7. REFERENCES

- [1] Sun Microsystems. J2ME Building Blocks for Mobile Devices. White Paper on KVM and the Connected, Limited Device Configuration (CLDC), 2000
- [2] Menon, H.: Portability Analysis in Mobile Gaming using J2ME. Master's degree Thesis. Morgantown, West Virginia (2006)
- [3] Weiss, D., Zduniak, M.: Automated integration tests for mobile applications in java 2 micro edition. In: Proceedings of the 10th international conference on Business information systems (BIS'07), Witold Abramowicz (Ed.). Springer-Verlag, pp. 478--487 Berlin, Heidelberg. (2007)
- [4] Zduniak, M.: Automated GUI Testing of Mobile Applications. Master's thesis (2007)
- [5] Rajapakse, D. C.: Techniques for De-fragmenting Mobile Applications: Taxonomy. In: 20th Intl. Conf. on Software Engineering and Knowledge Engineering Conference (SEKE'08), San Francisco, USA (2008)
- [6] Agarwal, V., Goyal, S., Mittal, S., Mukherjee, S.: MobiVine - A Middleware Layer to Handle Fragmentation of Platform Interfaces for Mobile Applications. IBM Research Report 09009 (2009)
- [7] Neomades: <http://www.neomades.com>
- [8] Ridene, Y., Belloir, N., Barbier, F., Couture, N.: A DSML For Mobile Applications Testing. In: Proceedings of 10th Workshop on Domain-Specific Modeling in Splash, Reno, Nevada, USA, pp. 25--30 (2010)
- [9] Mob4Hire, <http://www.mob4hire.com>
- [10] Mobile complete, <http://www.deviceanywhere.com/>
- [11] OMG Unified Modeling Language (OMG UML), Superstructure Specification, V2.2, (2009)
- [12] Steinberg, D., Budinsky, F., Paternostro, M., Merks, E. EMF - Eclipse Modeling Framework, Second Edition, Addison-Wesley (2008)
- [13] Aofelaum, L., Doyle, J.: Model-Based Testing. In: Software Quality Week Conference (1997)
- [14] Kärnä, J., Tolvanen, J.-P., Kelly, S.: Evaluating the Use of Domain-Specific Modeling in Practice. In: Proceedings of 9th Workshop on Domain-Specific Modeling, Orlando (2009)
- [15] White, J., and Douglas, C., S.: Model-Driven Product-Line Architectures for Mobile Devices. In: Proceedings of the 17th Annual Conference of the International Federation of Automatic Control, Seoul, Korea (2008)
- [16] Myllymäki, T., Koskimies, K., Mikkonen, T.: On the structure of a software product-line for mobile software. In: Software Infrastructures for Component-Based Applications on Consumer Devices (in conjunction with EDOC 2002), Lausanne, Switzerland, pp. 85--91 (2002)
- [17] Bosch, J.: Software product lines: organizational alternatives. In: Proceedings of the 23rd International Conference on Software Engineering (ICSE '01). IEEE Computer Society, Washington, DC, USA, pp. 91--100. (2001)

Une approche de type optimisation pour l'auto-adaptation de services mobiles

Marcel Cremene
Université Technique
de Cluj-Napoca
Roumanie

cremene@com.utcluj.ro

Kuderna-Iulian Benta
Université Technique
de Cluj-Napoca
Roumanie

benta@com.utcluj.ro

Michel Riveill
Université de Nice
Sophia-Antipolis
France

riveill@unice.fr

ABSTRACT

Service adaptation control mechanisms are generally based on *if...then...* rules and this is considered somehow natural. However, this fact leads to several limitations: the situations that demand adaptation and the actions to apply must be established a priori by a human expert. In this paper we propose an alternative approach that is based on the idea of modelling an adaptation problem as an optimization one. The most important advantage is the fact that in our approach the solutions are searched automatically by the optimization algorithms. This leads to a more general and reusable control mechanism.

RESUME

L'adaptation de services est souvent imaginée comme étant naturellement fondé sur un mécanisme de contrôle à base de règles *if...then...*, particuliers à chaque service. Cela implique une anticipation des situations qui nécessitent une adaptation et des actions à exécuter. Dans ce papier nous proposons une approche différente, qui consiste dans la modélisation du problème de l'adaptation comme un problème d'optimisation. L'avantage principal est que, ce n'est plus à l'expert humain de chercher a priori les solutions mais aux algorithmes de recherche, d'où sa généralité et réutilisation.

Catégorie et description

D.2.2 [Design Tools and Techniques]

Mots clé

Service, Contexte, Adaptation, Optimisation, Heuristiques

1. INTRODUCTION

L'adaptation de services constitués par assemblage de composants (ou services) représente déjà un thème de recherche consacré. Le fait de considérer un service comme décrit par un assemblage de composants simplifie le problème d'adaptation. Les services, dans le sens du Service Oriented Computing – SOC, et les composants logiciels posent des problèmes similaires et c'est pour ça que nous traitons ensemble ces deux concepts.

Des nombreux solutions on été proposées pour l'adaptation, surtout au niveau intergiciel. Mais, malgré ce grand nombre de solutions existantes, on est encore loin d'avoir épuisé ce sujet très complexe et d'une nature fortement interdisciplinaire. Ce fait est démontré d'ailleurs même par ce nombre très large d'intergiciels et des plates-formes d'adaptation proposés mais assez peu réutilisés. D'ailleurs, le terme « adaptation » couvre un espace très

large des situations et des scénarios différents. Il n'existe pas aujourd'hui une solution générique qui est la meilleure pour tous les problèmes d'adaptation mais plutôt des solutions adaptées a priori à certaines conditions particulières.

Parce que le terme « adaptation » peut avoir des significations très variés, nous avons choisi d'utiliser plutôt le terme « auto-adaptation » dans le sens défini dans le domaine d'Autonomic Computing. L'auto-adaptation [1] est la capacité d'un système informatique de gérer de manière autonome son propre comportement et structure afin d'atteindre un certain objectif. La présence de l'auto-adaptation est indiquée par l'existence d'une des caractéristiques de type auto-* (*self-**): auto-réparation, auto-configuration, auto-optimisation, auto-protection, auto-personnalisation en fonction du contexte et des besoins de l'utilisateur.

Aujourd'hui, on ne connaît pas une solution générale, complète et facilement réutilisable au problème de l'auto-adaptation de services. Selon les études de Salehie et all. [1], peu des plates-formes peuvent satisfaire tous les propriétés de type auto-*, dont l'auto-protection est la moins présente. Le même auteur remarque le fait que le problème de l'auto-adaptation de services est lié à différents domaines : Systèmes Autonomes, Théorie de Contrôle, Intelligence Artificielle, Génie Logicielle, etc. et qu'on utilise pour l'instant assez peu les instruments de l'Intelligence Artificielle, notamment l'apprentissage.

Une des principales limitations des solutions existantes provient du fait que le contrôle de l'adaptation est construit souvent dans une approche de type logique symbolique, en utilisant des règles *if...then....* Ces règles, ainsi que les actions qu'elles déclenchent pour adapter le service sont très spécifiques à l'application/service visée et donc très peu réutilisables. Le système de règles, qui doit être défini a priori par un expert humain, nécessite l'anticipation de toutes les situations possibles qui demandent une adaptation et aussi des solutions possibles pour appliquer l'adaptation. Mais cela n'est pas toujours possible. Dans la pratique, on ne peut pas tout anticiper. Et même quand l'anticipation est possible, cela introduit un cout de développement important (qui inclue aussi la résolution des potentiels conflits entre les règles). Ces observations nous ont motivé à chercher une approche différente pour l'auto-adaptation non-anticipative de services.

Dans ce papier on propose une approche qui n'est pas basée sur des règles particulières au service. Cette approche essaie de garder un niveau très générique et réutilise la puissance des algorithmes de recherche et d'optimisation existants. L'idée de base consiste dans la modélisation du problème d'adaptation comme un

problème d'optimisation. Grace à cette approche, ce n'est plus le devoir de l'expert humain d'identifier a priori les solutions du problème d'adaptation pour toutes les situations envisageables mais c'est au système de trouver lui-même ces solutions. Evidemment, cette approche a ses limites.

Un exemple d'application de l'approche proposé est un service de transfère des données accessible sur un terminal mobile, qui s'adapte aux besoins de sécurité de l'utilisateur et du type d'application et qui essaie dans le même temps de minimiser les ressources utilisées et le temps de réponse. On propose une solution alternative et plus générale au problème présenté dans [13, 14], qui a été résolu dans une approche à base des règles. Selon [13, 14] les propriétés de sécurité : intégrité, authenticité, confidentialité, non-répudiation, contrôle de l'accès, peuvent être assurées de manière flexible en assemblant des composants de sécurité. Ces composants implémentent des opérations de base comme : hachage, concaténation, génération des clés, cryptage symétrique, cryptage asymétrique, etc. Pour chaque composant, plusieurs versions peuvent exister (ex. divers algorithmes d'encryptage). Un composant peut être déployé sur le terminal mobile ou utilisé comme un service distant. Les propriétés de sécurité à satisfaire ainsi que le niveau de sécurité nécessaire sont déterminés par le type des données à transférer et aussi par les préférences de l'utilisateur. Le problème à résoudre est le suivant : étant donné a) les propriétés de sécurité à satisfaire et leur niveau et b) un nombre des composants de sécurité existants (locales – sur le terminal mobile et distants – des services web par exemple), on cherche à déterminer l'assemblage de composants qui satisfait les demandes (propriétés et niveau de sécurité) mais aussi minimiser le temps de réponse et l'effort computationnel du terminal mobile.

Ce papier est organisé comme il suit : dans la section suivante on analyse quelques approches liées à notre thème de recherche, la section trois présente l'approche proposée, la section quatre présente quelques résultats de nos expérimentations et la dernière section contient les conclusions.

2. PROPOSITIONS RELEVANTES EXISTANTES

Un aspect important à-propos du sujet de ce papier est le caractère non-anticipatif de l'adaptation. Dans [5] nous avons identifié le problème de l'adaptation non-anticipative comme étant causé principalement par l'utilisation du mécanisme Événement-Condition-Action (ECA). Les événements, les conditions et les actions sont spécifiques à l'application (non réutilisables), ils doivent être prévus a priori par l'expert humain (par anticipation) et n'évoluent pas ultérieurement sans son intervention. Dans [5] nous avons proposé une architecture générale d'un système adaptatif ainsi que des opérateurs d'agrégation et des mesures de l'adéquation service-contexte. Dans sa thèse [15], Khan traite aussi le problème de l'adaptation dynamique non-anticipative des applications mobiles. L'auteur insiste surtout sur la capacité du système auto-adaptatif de prendre en compte l'apparition et la disparition dynamique des composants/services. Son travail est basé sur les projets IST MADAM [16] et MUSIC [17]. Dans son approche, l'auteur utilise des ontologies (qui permettent de raisonner sur le service et sur le contexte) et des fonctions d'utilité (qui permettent de mesurer l'adéquation d'un service à un contexte). Plusieurs variantes de services sont générées et la meilleure (celle qui a l'utilité maximale) est sélectionnée. L'auteur utilise un algorithme de recherche exhaustif, récursif, pour trouver

la meilleure solution parmi toutes les variantes possibles. Dans un espace large de recherche cela peut poser des problèmes et des heuristiques sont nécessaires.

Nous avons observé que ce problème d'adaptation non-anticipative est en fait assez proche de celui de l'optimisation car l'objectif est en général de maximiser l'utilité du service et de minimiser ses coûts. Ce fait qui nous a conduits à s'intéresser de l'optimisation. D'ailleurs, cette observation permet de faire un lien entre les domaines Ubiquitous/Pervasive/Context-Aware Computing et le domaine Search Based Software Engineering (SBSE).

Dans le domaine SBSE on cherche principalement à identifier des applications des algorithmes méta-heuristiques dans le domaine d'ingénierie logicielle. Un nombre important de travaux dans ce domaine s'intéressent à l'optimisation des paramètres de qualité de service (QoS). Dans [2] les auteurs proposent l'utilisation des algorithmes génétiques pour la composition optimale d'un service web décrit par un flux (workflow) contenant des services abstraits pour lesquels existent un certain nombre des services concrets alternatifs. La thèse de Loubery [18] propose la plate-forme d'adaptation appelée « Kalimucho », sensible au contexte, pour la gestion de la qualité de service. Cette proposition insiste notamment sur la définition des paramètres de QoS relatifs au contexte (pérennité, utilité). Cette approche est basée dans le même temps sur un modèle de contrôle de type événement-condition-action (ECA). Par rapport au modèle ECA classique, dans ce cas l'action d'adaptation suppose la génération des alternatives de reconfiguration possibles qui sont ensuite évaluées en utilisant les paramètres de QoS. Même dans ce cas, le fait d'utiliser un mécanisme de type ECA ne convient pas à l'objectif de ce papier qui était de proposer une alternative à ce mécanisme (qui est anticipatif et dépendant de l'application par sa nature). Aussi, même si un nombre important de papiers s'intéressent au QoS, à notre avis, la partie essentielle consiste en fait dans la manière de modéliser un problème d'adaptation comme un problème d'optimisation. Cette méthodologie de modélisation n'est pas traitée de manière systématique dans les propositions que nous avons étudiées.

L'objectif de ce papier est de proposer une méthode générale permettant de modéliser un problème d'adaptation comme un problème d'optimisation.

3. MODELISATION DU PROBLEME D'ADAPTATION COMME UN PROBLEME D'OPTIMISATION

En général, l'optimisation consiste dans l'identification des points de minimum ou de maximum d'une fonction donnée. Dans le cas d'un service, nous pouvons discuter d'optimisation si on définit une fonction associée au service, qu'on souhaite minimiser ou maximiser. Les paramètres de cette fonction représentent les paramètres du service qu'on peut modifier afin d'adapter ce service. Selon nos hypothèses de départ (approche à composants/SOA), ces paramètres concernent l'architecture du service et les paramètres des composants. Si cette fonction est une mesure de l'adéquation du service par rapport aux objectifs de l'adaptation alors le problème d'adaptation devient un problème d'optimisation et nous pouvons utiliser des algorithmes connus pour la résoudre.

Pour modéliser le problème d'adaptation comme un problème d'optimisation il est nécessaire de résoudre les sous-problèmes suivants :

- 1) *Représenter ce qui est une solution.* Dans une approche a composants (ou services) une solution est décrite par une architecture de composants (ou services) interconnectées. La représentation choisie doit être compatible avec les algorithmes d'optimisation existants qu'on souhaite utiliser.
- 2) *Evaluer une solution.* L'adaptation est faite pour atteindre certains objectifs. Une solution est supérieure à une autre parce qu'elle permette d'atteindre ces objectifs de manière plus efficace. L'évaluation introduit un ordre entre les solutions possibles.
- 3) *Définir l'espace de recherche des solutions.* Contrairement à l'approche à base de règles ou l'expert humain connaît a priori la solution, dans cette approche il faut chercher la solution parmi un nombre des possibilités existantes. En général, plus l'espace est large, plus le temps de recherche est élevé donc il est nécessaire de réduire cet espace le plus possible.
- 4) *Choisir une certaine euristique pour l'algorithme de recherche des solutions.* Dans un espace large un algorithme de recherche exhaustif est inutilisable. Plusieurs heuristiques générales existent : hill-climbing, recuit simulé, algorithmes génétiques, etc. Il n'y a pas une qui est supérieure aux autres quelque soit le problème. En plus, on peut définir notre propre heuristique particulière.

Appart les problèmes liés strictement a la modélisation du problème d'optimisation, pour construire un système complet il est nécessaire de résoudre encore un problème important :

- 5) *Intégrer dans l'architecture d'un système auto-adaptatif le mécanisme de contrôle* basé sur l'algorithme d'optimisation. Cette intégration concerne la gestion du cycle de vie de l'adaptation et les mécanismes nécessaires pour l'application des solutions.

Dans les sous-sections suivantes on discute avec un peu plus des détails ces problèmes ainsi que l'application de ces principes a l'exemple présenté dans l'introduction concernant le service de sécurité adaptatif.

3.1 Représentation d'une solution

Une solution correspond à un service (une certaine configuration d'un service). Un service est décrit par un assemblage de composants/services. Chaque composant/service présente des interfaces et des paramètres. Dans le modèle qu'on utilise pour décrire la solution on doit inclure tous les paramètres du service qu'on peut reconfigurer pour adapter ce service. Dans une approche de type adaptation architecturale les primitives de reconfiguration prises en compte sont :

- *Paramétrisation* des composants : changer la valeur d'un paramètre p_j d'un composant C_i ,
- *Ajout* d'un composant supplémentaire (ainsi que les connexions afférentes aux nouveaux ports),
- *Elimination* d'un composant (avec une 'réparation' des connexions libères),
- *Remplacement* d'un composant par un autre,

- *Migration* d'un composant entre deux machines physiques en gardant ses connexions logiques.

Dans une approche a services (de type SOC/SOA) des composants de liaison de type *if, for, split, join* peuvent exister aussi dans l'architecture (correspondant aux éléments du flux qui décrivent un service composite).

Un exemple de modèle de solution est donné par Canfora et all. dans [2]. Ce modèle se situe dans une approche de type SOC ou il y a une seule primitive de reconfiguration : le remplacement. Chaque service peut être remplacé par un nombre de N versions similaires du même service mais ayant des caractéristiques de Quality of Service (QoS) différents : temps de réponse, réputation, disponibilité, cout d'accès. Une solution est décrite dans ce cas simplement, par un vecteur de valeurs entiers $[i_1...i_k]$. Le service composite est constitué par k services dits abstraits, pour chaque service abstrait S_{a_j} on connaît un nombre de versions N_j de services dits concrets et la valeur i_j du vecteur qui décrit une solution représente l'index du service concret choisi. Cette représentation simple convient très bien aux algorithmes classiques d'optimisation de genre *hill-climbing* et *algorithmes génétiques* (GA) par exemple.

Le modèle de type vecteur est adéquat pour les situations quand l'adaptation suppose des modifications qui ne changent pas radicalement l'architecture du service.

Une limite de la solution de Canfora, de notre point de vue, c'est le fait que la seule primitive de reconfiguration supportée c'est le remplacement. Une solution pour dépasser cette limite est de trouver une équivalence entre le remplacement et les autres primitives. Ainsi que :

- La *paramétrisation* peut être vue comme un remplacement par le même composant mais ayant une autre valeur du paramètre visé. Il existe quand même un problème avec les paramètres ayant un domaine de variation continu car cette approche d'optimisation est de type discret/combinatoire.
- *L'ajout d'un nouveau composant* C_x est équivalent au remplacement d'un composant existant C_j par un nouveau composant C_y , ou C_y est le résultat de la composition entre C_j et C_x .
- *L'élimination* d'un composant est équivalente au remplacement du composant par un lien ou par un composant nul, qui ne fait rien.
- *La migration* d'un composant est le remplacement du composant par un autre composant identique mais qui se trouve sur une autre machine physique et en gardant les mêmes connexions logiques.

Une autre manière de représenter une solution est celle proposée par Aversano et all. dans [3]. Pour décrire un flux (workflow) correspondant à un service composite, les auteurs proposent un modèle de type arbre binaire, compatible avec la technique qui s'appelle *Genetic Programming* (GP). Dans cet arbre binaire, les nœuds terminaux représentent des services concrets et d'autres types de nœuds spéciaux sont utilisés pour les différentes types de composition : séquentielle ou parallèle et aussi pour les conditions (*switch*) et boucles (*for, while*). Ce modèle est utilisé dans [3] pour la composition automatique de services web. Cette approche peut être vue comme une alternative aux techniques de planification (AI planning) [4] qui sont d'ailleurs très utilisés dans le domaine de la composition de services.

Le modèle de type arbre binaire est plus adéquat dans les situations quand l'adaptation suppose une modification radicale de l'architecture du service parce qu'il permet de générer facilement des variantes architecturale très divers. En plus, ce modèle ne nécessite pas l'utilisation des équivalents des primitives de reconfiguration avec le remplacement comme c'était le cas du modèle vectoriel présenté avant. L'inconvénient est que c'est un modèle plus complexe.

Concernant l'exemple présenté dans l'introduction, on fait l'hypothèse que le type des données à transférer et les préférences de l'utilisateur permettent de déterminer (selon les politiques de sécurité décrites dans [13, 14] et en appliquant des patrons d'assemblage prédéfinis) une architecture abstraite contenant des types des composants. Pour chaque type de composant plusieurs versions de composants concrets existent. Ces versions ont la même fonction mais des caractéristiques différentes : niveau de sécurité, temps de réponse, effort computationnel, sont locales ou distantes. Pour ce genre de problème nous pouvons utiliser la représentation vectorielle - chaque élément du vecteur-solution représente l'indice d'un composant concret.

3.2 Evaluation d'une solution

L'évaluation d'une solution est un aspect clé pour notre approche car ça permet d'ordonner les solutions possibles. Par évaluation on comprend en général l'attachement d'un score à chaque solution afin de pouvoir les hiérarchiser. Ce score peut être représenté par :

- Une valeur réelle – directement utilisable par les algorithmes d'optimisation standard (avec un seul critère) ;
- Un vecteur de valeurs réelles – utilisables par les algorithmes d'optimisation multicritères. Ce modèle est plus général car le score unique est souvent calculé à partir des valeurs d'un vecteur. Ainsi que, dans notre cas, l'évaluation consiste dans l'association d'un vecteur d'attributs à chaque solution. Dans la suite, on va appeler ce modèle *vecteur d'évaluation*.

Afin d'évaluer une solution il est nécessaire de :

- Définir les objectifs de l'adaptation,
- Définir les attributs qui indiquent dans quelle mesure un service atteint les objectifs de l'adaptation définis,
- Identifier les instruments de mesure (capteurs, API de monitoring, etc.) des valeurs des attributs,
- Définir des opérateurs permettant l'agrégation des attributs dans l'architecture du service (calculer les attributs du service si on connaît les attributs de ses composants/services internes),
- Définir des formules d'agrégation des attributs du vecteur d'évaluation sous la forme d'un score unique ou sous la forme d'un autre vecteur d'une taille plus

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.
Conference '10, Month 1–2, 2010, City, State, Country.
 Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

réduite (car les algorithmes d'optimisation multicritères classiques ne peuvent pas gérer de manière efficace plus de 3-4 critères).

3.2.1 Définition des objectifs de l'adaptation

L'adaptation est faite toujours pour atteindre certains objectifs. Une grande partie des objectifs sont généraux, indépendants de l'application. Il peut y avoir aussi des objectifs particuliers, plus spécifiques à l'application.

Des objectifs généraux sont : minimiser le temps de réponse, minimiser la consommation de ressources, minimiser la probabilité d'interruption d'une connexion réseau, maximiser la durée de vie de la batterie, maximiser la qualité d'un flux multimédia, etc. En général, on essaie de minimiser les coûts, la consommation de ressources et de maximiser l'utilité, la qualité, la vitesse.

Dans une approche de type optimisation il est préférable de définir dès le départ les objectifs d'adaptation en terme de minimisation ou maximisation. Ainsi que, les propriétés comme : auto-réparation, auto-configuration, auto-optimisation, auto-protection, auto-personnalisation au contexte et besoins de l'utilisateur doivent être représentés comme des objectifs de type minimisation ou maximisation d'une certaine fonction. Cela n'est pas toujours simple et les chercheurs qui utilisent des fonctions d'utilité réclament une certaine difficulté pour exprimer ces fonctions.

3.2.2 Définition des attributs d'évaluation

Le but de l'évaluation est de nous offrir une mesure de l'adéquation du service aux objectifs de l'adaptation. Par exemple, les attributs de QoS : temps de réponse, disponibilité, réputation, coût d'accès, niveau de sécurité et d'autres peuvent être considérés comme des mesures de qualité. Ce sont des mesures de l'adéquation d'un service à des objectifs très généraux : temps de réponse minimal, coût d'accès minimal, réputation maximale, etc.

Les paramètres standards de QoS ne sont pas les seuls à prendre en compte pour évaluer un service (une solution). On peut envisager aussi des paramètres QoS étendus, relatifs au contexte. Par exemple, l'utilité de l'information fournie par un service peut dépendre de la localisation de l'utilisateur. Dans [5] nous avons introduit la notion d'*adéquation service-contexte*, qui est aussi une mesure pour évaluer un service de point de vue adaptation.

Le concept d'*utilité* a aussi gagné du terrain dans le domaine de l'adaptation. Dans [6] une utilité différente est associée à des versions alternatives de composants, selon le contexte. Par exemple, une interface auditive est plus utile qu'une interface visuelle quand on ne peut pas regarder l'écran ou pour des utilisateurs malvoyants. En général, tout attribut qui reflète la mesure dans laquelle un service atteint un objectif de l'adaptation peut être inclus dans le vecteur d'évaluation.

3.2.3 Identification des sources des valeurs des attributs

Certains attributs nécessitent une méthode de mesure. Par exemple, le temps de réponse d'un service web doit être mesuré en lui envoyant des paquets et en regardant la durée aller-retour ; la mémoire consommée, le temps d'occupation du CPU, ou le débit réseaux disponible peut être obtenu en interrogeant des APIs dédiés, etc. Il est nécessaire d'identifier les services d'observation pour chaque attribut de ce type.

D'autres attributs peuvent être fixés par un opérateur humain, par exemple le coût d'accès, le type de donnée fourni par un service, etc. Il est nécessaire de décrire les composants/services avec des métadonnées indiquant ces attributs.

3.2.4 L'agrégation architecturale des attributs

Après une reconfiguration du service, on doit pouvoir mesurer ou calculer la valeur de chaque attribut du vecteur d'évaluation associé à ce service/solution. Connaissant les valeurs des attributs pour chaque composant interne (par mesure ou par spécification explicite au niveau des métadonnées) le problème est de déterminer les valeurs des attributs pour l'ensemble - le service composite. Pour chaque attribut il est nécessaire de définir un opérateur d'agrégation, en tenant compte aussi de la topologie architecturale. L'agrégation des attributs QoS standard est discutée dans [2] et aussi dans d'autres papiers. Dans [5] nous avons présenté aussi quelques exemples d'opérateurs d'agrégation basés sur des opérateurs arithmétiques, logique ou même sur la logique floue.

Afin de pouvoir effectuer l'agrégation architecturale, pour chaque attribut a_i il faut spécifier :

- La formule/opérateur/algorithme d'agrégation pour une structure séquentielle de deux composants. Par exemple, l'attribut 'temps de réponse' est additif, la formule est une somme.
- La formule/opérateur/algorithme d'agrégation pour une structure parallèle de deux composants. Par exemple, pour l'attribut 'temps de réponse' ce sera le temps maximal, l'opérateur est la fonction 'max'.
- Les formules/opérateurs/algorithmes d'agrégation pour des structures spéciales de composition de type *if*, *while*, *join*, *split*, etc. Par exemple, pour l'attribut 'temps de réponse', un *if* introduit une probabilité p de suivre un chemin et $1-p$ l'autre chemin et donc le temps de réponse total dépendra de p [2].

3.2.5 L'agrégation des attributs du vecteur d'évaluation

Dans l'hypothèse que nous avons associé à chaque solution/service un vecteur d'évaluation contenant les valeurs des attributs, le problème qui se pose ensuite est de pouvoir comparer deux solutions/services différentes et décider quelle est la meilleure.

Une solution pour ce problème est le score unique - d'associer une seule valeur à chaque vecteur. Cette association est basée sur une fonction d'agrégation. L'expression de cette fonction dépend de la nature des attributs. Dans l'exemple présenté dans [2] le vecteur peut être considéré comme étant composé par les valeurs des quatre attributs de QoS : temps de réponse - t , coût d'accès - c , réputation - r , disponibilité - d . La formule d'agrégation qui permet de comparer les solutions proposées dans [2] a la forme décrite par l'équation (1):

$$eval = \frac{w_1 \cdot r + w_2 \cdot d}{w_3 \cdot t + w_4 \cdot c} \quad (1)$$

Une autre formule possible est décrite par l'équation (2) - une combinaison linéaire dans laquelle les poids à minimiser sont négatives et les poids à maximiser sont positives :

$$eval = w_1 \cdot r + w_2 \cdot d - w_3 \cdot t - w_4 \cdot c \quad (2)$$

Dans les équations (1) et (2) les valeurs w_i représentent des poids qui permettent de donner plus d'importance à un attribut ou à un autre. Par exemple, si le coût c est un attribut plus important que les autres attributs alors w_4 aura la plus grande valeur des poids.

Des formules plus complexes, qui impliquent des fonctions exponentielles ou même des systèmes d'inférences à base des règles de logique floue peuvent être utilisés aussi pour obtenir le score unique.

Il n'y a pas une méthode précise pour définir ces fonctions d'agrégation. Mais la forme de la fonction peut conduire à un espace de recherche plus ou moins facile à parcourir par les algorithmes d'optimisation (avec plus ou moins de minimum locaux), fait qui influence les performances.

Un autre problème est le choix des poids w_i car elles dépendent des préférences de l'utilisateur (préférence qualité versus coût) et du contexte (pour un terminal mobile est plus critique d'optimiser ses ressources que pour une machine fixe). Il peut être difficile pour l'utilisateur de spécifier a priori ses préférences afin de pouvoir déterminer précisément les valeurs de poids.

Une alternative qui nous permet d'éviter d'exprimer la fonction d'agrégation est l'utilisation des algorithmes d'optimisation multicritères. Dans ce cas, on utilise directement le vecteur d'évaluation. Mais, cette fois-ci, le résultat de l'exécution de l'algorithme d'optimisation ne sera plus une seule solution (la meilleure) mais un set de bonnes solutions - ce qu'on appelle des fois *le front de Pareto*. Ensuite, pour choisir une seule solution il va falloir introduire encore une étape de sélection. Par exemple, l'utilisateur pourra être saisi pour indiquer explicitement une solution parmi les moins cher, celles qui lui conserve le plus la batterie ou celle qui est la plus rapide. Même si il faut impliquer l'utilisateur, cela peut être plus naturel que de demander à l'utilisateur de spécifier a priori des poids.

Dans l'exemple présenté dans l'introduction, les attributs utilisés pour évaluer une solution sont : le temps de réponse, l'effort computationnel au niveau du terminal mobile (qui influence la consommation d'énergie) et le niveau de sécurité. Le temps de réponse s'agrège au niveau de l'architecture en utilisant des relations comme celles présentées dans [2]. L'effort computationnel est calculé seulement pour les composants déployés sur le terminal mobile. Cet attribut a un caractère additif : l'effort computationnel du service de sécurité est donné par la somme des efforts computationnels des composants de sécurité. Le niveau de sécurité est décrit pour chaque propriété de sécurité. De point de vue agrégation architecturale, pour le niveau de sécurité on utilise la fonction *min* - le niveau minimal de sécurité donne le niveau global de sécurité.

L'objectif de l'adaptation est de maximiser le niveau de sécurité et de minimiser le temps de réponse et l'effort computationnel. On peut utiliser une formule d'évaluation globale comme celles décrites par les équations (1) et (2) par exemple.

3.3 L'espace de recherche de solutions

Théoriquement, l'espace de recherche de solutions contient tous les solutions/services possibles. Dans le même temps, nous avons tout l'intérêt de réduire au maximum cet espace afin d'accélérer le processus de recherche (mais sans perdre des solutions importants, évidemment).

Dans l'exemple décrit dans [2], un service constitué par l'assemblage de N services abstraits ayant chacun M versions de services concrets qui peuvent être choisis il y a un nombre de M^N

solutions possibles (tous les combinaisons). D'ailleurs, il s'agit d'un problème NP-difficile.

Une idée qui peut nous aider de réduire la dimension de l'espace de recherche et d'utiliser des contraintes qui éliminent les solutions impossibles ou improbables. Par exemple, si un service est distribué sur un dispositif mobile et aussi sur d'autres machines fixes, afin d'effectuer une adaptation qui a comme objectif de maximiser la durée de vie de la batterie on va prendre en compte seulement les composants situés sur le terminal mobile. Si on détecte un problème de langue incompatible par exemple, on va essayer seulement des combinaisons entre les composants de l'architecture initiale du service et des composants qui modifient l'attribut langue. Ces contraintes peuvent être définies sous la forme d'un set de règles générales, indépendantes d'un certain service. Aussi, toutes les combinaisons qui ne respectent pas la compatibilité syntaxique peuvent être éliminées comme étant des solutions infaisables.

Dans le cas de l'exemple présenté dans l'introduction, l'espace de recherche des solutions est constitué par toutes les combinaisons possibles des composants concrets qui entrent dans l'architecture abstraite du service de sécurité. D'ailleurs, c'est un espace très similaire à celui décrit dans [2].

3.4 Algorithmes de recherche des solutions

Une fois que nous avons une méthode pour évaluer les solutions et un espace de recherche bien défini, nous pouvons utiliser un algorithme d'optimisation (ou de recherche) existant. Nous avons un premier choix entre un algorithme standard ou un algorithme multicritères.

Parmi les algorithmes standard, ceux qui ont été déjà utilisés pour des problèmes d'optimisation de services sont : les algorithmes de type 'integer programming', les algorithmes basés sur une recherche locale (local search) - hill-climbing, recuit simulée et les algorithmes génétiques. Dans la section 4 on présente quelques tests avec différents algorithmes.

Les algorithmes multicritères ont un problème : la grande majorité marche bien avec 2-3 critères mais pas plus de 4. Une solution est d'utiliser des fonctions d'agrégation afin de réunir tous les attributs de type utilité/qualité dans un score unique d'utilité et tous les attributs de type ressources/couts dans un autre score unique de couts. Ensuite, on essaie de maximiser l'utilité et minimiser les couts en utilisant des algorithmes avec deux critères. Le type d'algorithme dépende assez peu dans cette approche du type de problème à résoudre.

3.5 L'intégration du mécanisme de recherche/optimisation dans l'architecture d'un système auto-adaptatif

L'utilisation des algorithmes de recherche pour optimiser des services web n'est pas une nouvelle idée. Il existe tout un domaine concernant l'utilisation des algorithmes de recherche ou d'optimisation pour résoudre des problèmes de l'ingénierie logicielle : Search Based Software Engineering (SBSE). Dans [3] nous avons vu que la technique GP a été utilisée pour la composition de services, comme alternative à la méthode de planification. Par contre, dans ce domaine il existe beaucoup moins d'applications de ces algorithmes pour des problèmes d'adaptation dynamique de services.

Pour construire des systèmes auto-adaptatifs, il existe une architecture générale qui est largement acceptée comme architecture de référence dans le domaine 'Autonomic Computing' [7]. Cette architecture est basée sur un modèle de contrôle en boucle fermée et implémente les étapes illustrée dans la figure 1, présentée dans [8].

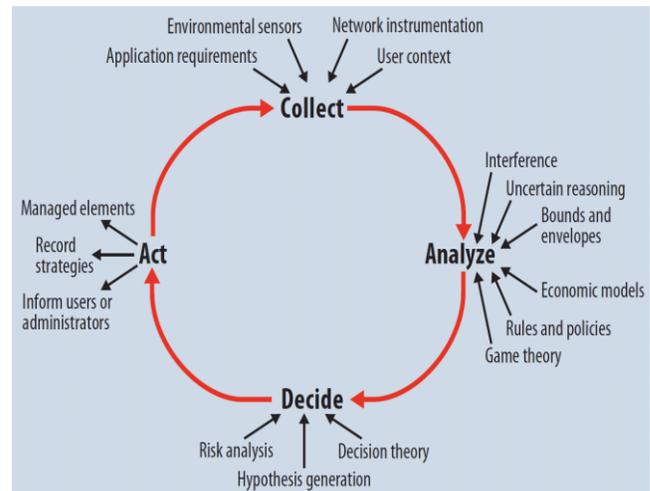


Figure 1. La boucle d'adaptation autonome, selon [8].

Comme nous pouvons observer dans la figure 8, les étapes à parcourir pour implémenter la boucle d'adaptation sont les suivantes :

- *La collection* des informations sur le service et sur son contexte est la première étape. Par rapport à notre approche, dans cette étape on utilise des différentes sources (voir la section 2.2.3.) afin d'obtenir les valeurs des attributs. Ensuite on applique les opérateurs d'agrégation (voir la section 2.2.4.) pour arriver finalement au vecteur d'évaluation correspondant à l'état actuel du service.
- *L'analyse* des informations collectées sert à fournir une base pour l'étape suivante de décision concernant l'action d'adaptation. Dans notre modèle, à ce niveau on applique les fonctions d'agrégation (voir la section 2.2.5.) sur les valeurs des attributs du vecteur d'évaluation.
- *La décision* est l'étape dans laquelle on détermine si une action d'adaptation est nécessaire et laquelle. Dans notre modèle, à ce niveau on peut introduire un seuil et comparer le score de l'état actuel. Si le score n'est pas satisfaisant alors on déclenche une action d'adaptation. Une autre possibilité, plus coûteuse, est de faire des simulations dans un processus parallèle, avec des services, et retenir le meilleur (ou les meilleurs). Si le meilleur dépasse la solution actuelle d'un certain seuil alors on déclenche l'action d'adaptation.
- *L'action* concerne la mise en œuvre de l'adaptation est réalisée dans notre modèle en deux étapes : a) effectuer une recherche pour trouver la meilleure solution dans le nouveau contexte et b) réaliser le passage effectif entre la solution actuelle et la nouvelle solution. Si on a déjà trouvé une meilleure solution (dans un processus parallèle) on applique directement la phase b). Pour cela, on s'appuie sur un des nombreux intergiciels qui permet d'effectuer des opérations au niveau du service en

exécution. Un exemple d'intergiciel permettant une reconfiguration dynamique ainsi qu'une gestion dynamique de l'apparition/disparition des composants est WComp [9] par exemple.

4. EXPERIMENTATION

Afin de pouvoir tester la validité de nos hypothèses, nous avons implémenté une plate-forme de test en java. Pour les algorithmes évolutifs nous avons utilisé la plate-forme libre ECJ [11] version 2.0.

Nous décrivons un service composite sous la forme d'un processus BPEL et nous expérimentons la reconfiguration architecturale par remplacement des composants. Comme nous avons discuté dans la section 2, pour les autres primitives de reconfiguration (paramétrisation, ajout, élimination, migration) il existent des équivalences avec le remplacement.

Dans nos tests, nous avons fait l'hypothèse que les versions alternatifs sont déjà connus. En pratique, ce problème implique la recherche des services/composants disponibles qui peuvent remplacer un composant existant dans l'architecture du service. Nous avons étudié pour l'instant la convergence des algorithmes et le temps nécessaire pour trouver une solution. Plusieurs algorithmes de recherche ont été évalués :

- Deux types d'algorithmes génétiques : *IntGA* est un algorithme inspiré de celui proposée dans [2] et *LongGA* est un algorithme génétique basé sur un génome différent, plus long.
- Algorithmes de type évolution différentielle [10] : *TruncDE*, *XueDE*, *LongDE*.
- Algorithmes de type recherche locale : hill-climbing et recuit simulé.

Dans ce papier, nous ne donnons pas des détails sur ces algorithmes car ils sont présentés en détail dans notre papier [12]. Ce qui nous intéresse ici c'est juste leurs performances, pour voir si l'approche est adéquate à une adaptation dynamique.

La figure 2 présente la convergence des divers algorithmes testés. L'axe vertical représente la valeur de la fonction d'évaluation et sur l'axe horizontal il y a les générations.

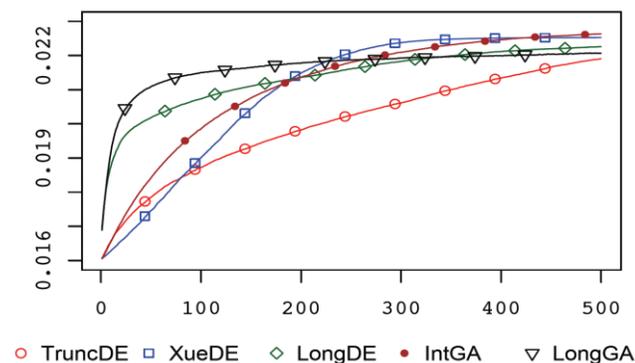


Figure 2. Convergence des algorithmes.

La figure 3 présente le temps nécessaire pour trouver une solution. Les tests sont effectués sur une machine personnelle assez typique (CPU 2.4GHz, 2G RAM). L'axe vertical indique le temps en millisecondes et sur l'axe horizontal nous avons les scénarios de simulation. Un scénario est décrit par un rapport N/M ou N est le

nombre des composants abstraits et M est le nombre des versions par composant abstrait.

On observe que le temps nécessaire pour trouver une bonne solution (l'optimal n'est pas garanti par ces algorithmes heuristiques) est d'ordre à 100 millisecondes même pour des scénarios très complexes.

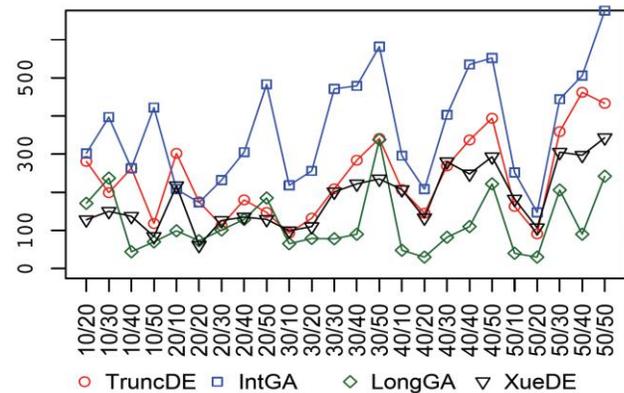


Figure 3. Temps de calcul des algorithmes (millisecondes).

Nous avons évalué plusieurs types d'algorithmes. Une solution générale représente les algorithmes évolutifs mais, pour des problèmes comme celui présenté dans [2] les algorithmes de type « hill-climbing » ont prouvé aussi des performances très bonnes et ils sont plus rapides. Le problème avec les algorithmes de recherche locaux (ex. hill-climbing) est le fait qu'ils peuvent rester dans des minimums locaux.

De point de vue implémentation, le scénario avec le service de sécurité décrit dans l'introduction est encore en cours de développement.

5. CONCLUSIONS

Dans ce papier nous avons présenté les étapes nécessaires à parcourir pour modéliser un problème d'adaptation sous la forme d'un problème d'optimisation. Le modèle proposé peut être intégré facilement avec l'architecture générale d'un système auto-adaptatif. Nous avons discuté brièvement comment cette approche peut être appliquée pour construire un service de sécurité adaptatif. L'implémentation de ce service de sécurité est encore en cours. Nous avons réalisé des tests avec plusieurs algorithmes d'optimisation qui montrent qu'il est possible de trouver une solution en moins d'une seconde même dans un très grand espace de recherche en utilisant des algorithmes heuristiques. Notre objectif futur est d'implémenter la solution complète basée sur l'approche proposée et tester plusieurs scénarios.

6. REMERCIEMENTS

Ce travail de recherche a été soutenu par la Ministère Roumain de l'Éducation et Recherche, l'unité CNCS-UEFISCDI, projet TE 252, contrat 33/2010.

7. REFERENCES

- [1] Mazeiar Salehie and Ladan Tahvildari. 2009. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.* 4, 2, Article 14 (May 2009), 42 pages. DOI=10.1145/1516533.1516538 <http://doi.acm.org/10.1145/1516533.1516538>

- [2] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. 2005. An approach for QoS-aware service composition based on genetic algorithms. In *Proceedings of the 2005 conference on Genetic and evolutionary computation (GECCO '05)*, Hans-Georg Beyer (Ed.). ACM, New York, NY, USA, 1069-1075. DOI=10.1145/1068009.1068189 <http://doi.acm.org/10.1145/1068009.1068189>
- [3] Aversano, L., Di Penta, M., and Taneja, K. (2006). A Genetic Programming Approach to Support the Design of Service Compositions. *Computer Systems Science & Engineering*, 21(4), 247-254.
- [4] Carman, M., Serafini, L., and Traverso P. Web Service Composition as Planning, *ICAPS'03 Workshop on Planning for Web Services*, Trento, Italy, June 2003.
- [5] Marcel Cremene, Michel Riveill, and Christian Martel. 2007. Autonomic Adaptation solution based on Service-Context Adequacy Determination. *Electron. Notes Theor. Comput. Sci.* 189 (July 2007), 35-50. DOI=10.1016/j.entcs.2007.05.046 <http://dx.doi.org/10.1016/j.entcs.2007.05.046>
- [6] Jacqueline Floch, Svein Hallsteinsen, Erlend Stav, Frank Eliassen, Ketil Lund, and Eli Gjørven. 2006. Using Architecture Models for Runtime Adaptability. *IEEE Softw.* 23, 2 (March 2006), 62-70. DOI=10.1109/MS.2006.61 <http://dx.doi.org/10.1109/MS.2006.61>
- [7] Jeffrey O. Kephart and David M. Chess. 2003. The Vision of Autonomic Computing. *Computer* 36, 1 (January 2003), 41-50. DOI=10.1109/MC.2003.1160055 <http://dx.doi.org/10.1109/MC.2003.1160055>
- [8] Simon Dobson, Roy Sterritt, Paddy Nixon, and Mike Hinchey. 2010. Fulfilling the Vision of Autonomic Computing. *Computer* 43, 1 (January 2010), 35-41. DOI=10.1109/MC.2010.14 <http://dx.doi.org/10.1109/MC.2010.14>
- [9] Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Vincent Hourdin, Daniel Cheung-Foo-Wo, Eric Callegari, Michel Riveill. 2009. WComp middleware for ubiquitous computing: Aspects and composite event-based Web services, *Annals of Telecommunications* 43, 3 (April 2010) Springer Paris, 197-214 Doi: 10.1007/s12243-008-0081-y <http://dx.doi.org/10.1007/s12243-008-0081-y>
- [10] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. of Global Optimization* 11, 4 (December 1997), 341-359. DOI=10.1023/A:1008202821328 <http://dx.doi.org/10.1023/A:1008202821328>
- [11] S. Luke. Ecj - a java-based evolutionary computation research system, 2010. <http://www.cs.gmu.edu/~eclab/projects/ecj/>
- [12] Florin Claudiu Pop, Denis Pallez, Marcel Cremene, Andrea G.B. Tettamanzi, Mihai Suciuc, Mircea Vaida. 2011. QoS-Based Service Optimization using Differential Evolution. *Proceedings of GECCO'2011*. Publication en cours. <http://www.sigev.org/gecco-2011/>
- [13] Kamel, M., Boudaoud, K., and Riveill, M. 2010. A Low-energy Consuming and Component-based Security Management Architecture for Mobile Devices. *In Proceedings of Security and Management*, 167-173.
- [14] N. Nobelis, K. Boudaoud, M. Kamel, M. Riveill. 2010. A user-centric approach for secure communication protocols. *In ASIGE10 – Architects of Secure Networks*, Genova, Italy, May 2010, invited paper.
- [15] Khan, M.U. 2010. *Unanticipated Dynamic Adaptation of Mobile Applications*. PhD Thesis. University of Kassel, Kassel University Press.
- [16] MADAM, IST Project, <http://www.intermedia.uio.no/display/madam/Home>
- [17] MUSIC, IST Project, <http://ist-music.berlios.de/site/>
- [18] Louberry, C. 2010. *KALIMUCHO: Adaptation au Contexte pour la Gestion de la Qualité de Service*. Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour. Université de Pau et des Pays de l'Adour.

Requirement Language tooling with Xtext

Manzoor Ahmad
University of Toulouse
CNRS/IRIT

F-31062 Toulouse Université Cedex, France
manzoor.ahmad@irit.fr

Jean-Michel Bruel
University of Toulouse
CNRS/IRIT

F-31062 Toulouse Université Cedex, France
bruel@irit.fr

ABSTRACT

Self Adaptive systems autonomously modify their behavior at run-time in response to changing environmental conditions. We need to incorporate uncertainty into the specification of these systems. The identification of the adaptive features of a self adaptive system is the most important step in the development of these systems. In this paper an approach for the development of self adaptive systems is shown using RELAX which is a new requirements engineering language for Dynamic Adaptive Systems, where explicit constructs are included to handle uncertainty and a SysML profile conceived for the self adaptive systems. Our contribution helps in the identification and precise description of the adaptive features of a self adaptive system by proposing an environment based on RELAX language for which an Xtext tool using eclipse is developed. At this stage, the environment gives a description of functional and non functional properties of the system and transform these into SysML requirements diagram.

Keywords

Requirements, Specification, Metamodeling, Domain Specific Languages, RELAX, SysML

1. INTRODUCTION

An adaptive system is a set of interacting or interdependent entities, real or abstract, forming an integrated whole that together are able to respond to environmental changes or changes in the interacting parts. A system has goals that must be satisfied and, whether these goals are explicitly identified or not, system requirements should be formulated to guarantee goal satisfaction. This fundamental principle has served systems development well for several decades but is founded on an assumption that goals are fixed. In general, goals can remain fixed if the environment in which the system operates is stable [1]. With the increase in size, heterogeneity and complexity of systems, it is difficult to anticipate all the states of the system and dynamically self-adapt

to changing environmental conditions [2].

Figure 1 summarizes our work: first of all requirements are divided into functional (core functionality of the system) and non functional (security, persistence etc.), then we apply a process to get the requirements directly linked to adaptive feature of the system, these requirements are modeled using a specific profile SPAS (SysML Profile for Adaptive Systems) [3], and then integrated with the classical application. For the last part, we will use COTS (Components Off The Shelf) connectors/associations which is a work in progress [4].

The focus of this paper is on a DSL (Domain Specific Language) for adaptive systems based on RELAX which is a requirements engineering language for DASs, where explicit constructs are included to handle uncertainty. The need for DASs is typically due to two key sources of uncertainty: First is the uncertainty due to changing environmental conditions, such as sensor failures, noisy networks, malicious threats, and unexpected (human) input; the term environmental uncertainty is used to capture this class of uncertainty. A second form of uncertainty is behavioral uncertainty, whereas environmental uncertainty refers to maintaining the same requirements in unknown contexts, behavioral uncertainty refers to situations where the requirements themselves need to change. It is difficult to know all requirements changes at design time and, in particular, it may not be possible to enumerate all possible alternatives [2].

The vocabulary of RELAX is designed to enable the analysts to identify the requirements that may be relaxed when the environment changes. RELAX addresses both of the uncertainties mentioned above. For example, the system might wish to temporarily relax a noncritical requirement in order to ensure that critical requirements can still be met. RELAX outlines a process for translating traditional requirements into RELAX requirements. The only focal point is for the requirement engineers to identify the point of flexibility in their requirements. For this purpose requirements are divided into invariant requirements that should never change and Non-invariant or relaxed requirements where the requirements can be temporarily relaxed under certain conditions.

For the generation of DSL, Xtext is used as an editor generator as well as a code generator framework. Xtext is a framework for the development of domain-specific languages and other textual programming languages. It is tightly in-

egrated with the EMF (Eclipse Modeling Framework) and leverages the Eclipse Platform in order to provide a language-specific IDE (Integrated Development Environment) [5].

SysML is a best language for the development of a tool based on RELAX. In order to link SysML and RELAX, in first place, we have developed a DSL and then this work will be used to develop a profile called SPAS (SysML Profile for Adaptive Systems).

The notion of variability in requirements engineering is discussed in several research works: as in [6], the authors adopt a goal oriented approach for the requirements engineering. While eliciting and specifying functional and non-functional requirements, analysts attempts to characterize stakeholders needs, possibly asking two questions that: what the system should do and how well the system should do. The authors in [6] are of the view that the variability in what and how may be further enhanced by the where and when due to the openness of the operational environment. The resulting requirements also includes monitoring specification that take into account the variability in the operational context, evaluation criteria and alternative software behaviors to be adopted at run-time by the software system, to ensure the achievement of the intended user goals.

The work in [7] deals with the elicitation and reasoning about monitoring and switching concerns induced by varying context - after detecting requirements violations - the software switches from one behavior to another behavior. A problem frame notation is adopted to capture and reason variant problems in context, from which a monitoring problem is derived, on the basis of which a switching problem is derived.

In [8] a Domain Specific Language for WSN (Wireless Sensor Networks) is developed. The author is of the view that software development for WSN, if done with low-level programming languages is very difficult which requires detailed knowledge of the very limited hardware; testing and debugging programs on the sensor nodes is very expensive and difficult. A domain specific language provides the domain expert with terms and notations that match his/her cognitive space and intuition.

The rest of the paper is organized as follows: in section 2, background of the concepts mentioned above is elaborated, section 3 shows the steps required for the generation of a code editor for RELAX using Xtext and the steps required to transform textual requirements to graphical format, section 4 concludes the paper and section 5 gives an idea about the future work.

2. BACKGROUND

2.1 RELAX overview

RELAX takes the form of a structured natural language, including operators designed specifically to capture uncertainty [9], their semantics is also defined. Typically textual requirements prescribe behavior using a modal verb such as SHALL that defines functionality that a software system must always provide. For self-adaptive systems however, environmental uncertainty may mean that it is not always possible to achieve all of those SHALL statements; or behav-

ioral uncertainty may allow for trade-offs between SHALL statements to relax noncritical statements in favor of other, more critical ones. RELAX therefor identifies two types of requirements: one that can be relaxed in favor of other ones called variant or relaxed and other that should never change called invariant. Table 1 gives the set of RELAX operators, organized into modal, temporal and ordinal operators and uncertainty factors.

The conventional modal verb SHALL is retained for expressing a requirement with RELAX operators providing more flexibility in how and when that functionality may be delivered. More specifically, for a requirement that contributes to the satisfaction of goals that may be temporarily left unsatisfied, the inclusion of an alternative, temporal or ordinal RELAX-ation modifier will define the requirement as RELAXable. Each of the relaxation operators define constraints on how a requirement may be relaxed at run time. In addition, it is important to indicate what uncertainty factors warrant a relaxation of these requirements, thereby requiring adaptive behavior. This information is specified using the MON (monitor), ENV (environment), REL (relationship), and DEP (dependency) keywords.

3. WHY DEFINE A SPECIFIC PROFILE

SysML incorporates expressing the requirements in an efficient manner through requirements diagram so a link between SysML and RELAX would help in modeling the requirements efficiently. SPAS was conceived keeping in mind this link, it will solve a number of limitations of RELAX language:

- It will provide a development environment and a graphical support for expressing all the variables of RELAX.
- It will allow to use SysML which is becoming an industry standard day by day.

Our first endeavor in achieving the desired SPAS profile is the development of a DSL based on RELAX and as a first step we are going to transform textual requirements into SysML requirements diagram. The grammar of RELAX is used as a meta model and based on this meta model, we will be able to model our requirements.

3.1 Domain Specific Language

A domain specific language is a dedicated language, which focuses on a particular domain. A domain specific language for RELAX would bridge the gap between requirements even in a formatted version such as RELAX statements and the overall system model. Xtext is a framework for the development of domain specific languages and other textual programming languages and helps in the development of an IDE (Integrated Development Environment) for the DSL. Some of the IDE features that are either derived from the grammar or easily implementable are: syntax coloring, model navigation, code completion, outline view, and code templates. For this purpose, Xtext is used as generator for the code editor of RELAX.

3.2 Tooling RELAX

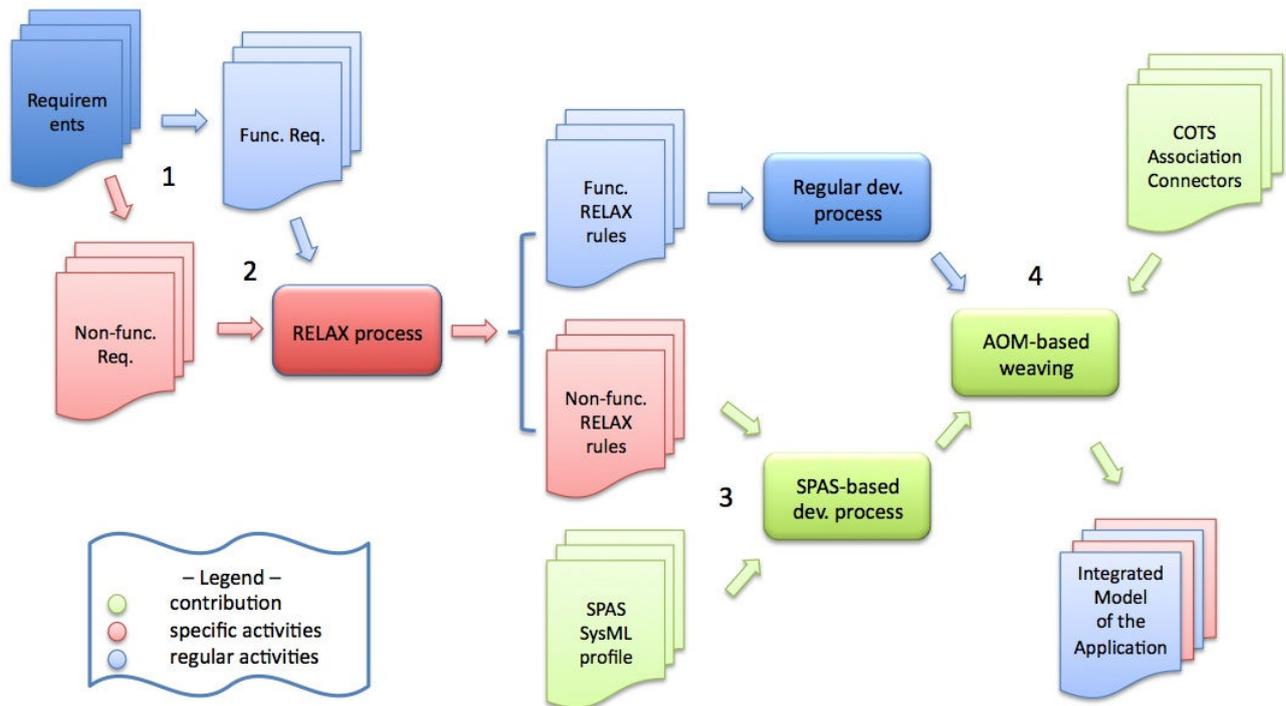


Figure 1: Integrated Work plan.

RELAX operator	Description
Modal Operators	
<i>SHALL</i>	a requirement must hold
<i>MAY ... OR</i>	a requirement specifies one or more alternatives
Temporal Operators	
<i>EVENTUALLY</i>	a requirement must hold eventually
<i>UNTIL</i>	a requirement must hold until a future position
<i>BEFORE, AFTER</i>	a requirement must hold before or after a particular event
<i>IN</i>	a requirement must hold during a particular time interval
<i>AS EARLY, LATE AS POSSIBLE</i>	a requirement specifies something that should hold as soon as possible or should be delayed as long as possible
<i>AS CLOSE AS POSSIBLE TO [frequency]</i>	a requirement specifies something that happens repeatedly but the frequency may be relaxed
Ordinal Operators	
<i>AS CLOSE AS POSSIBLE TO [quantity]</i>	a requirement specifies a countable quantity but the exact count may be relaxed
<i>AS MANY, FEW AS POSSIBLE</i>	a requirement specifies a countable quantity but the exact count may be relaxed
Uncertainty Factors	
ENV	defines a set of properties that define the system's environment
MON	defines a set of properties that can be monitored by the system
REL	defines the relationship between the ENV and MON properties
DEP	identifies the dependencies between the (relaxed and invariant) requirements

Table 1: RELAX Operators.

To show the DSL for RELAX; we start by taking some non-functional requirements. These requirements are taken from a case study for a car crash crisis management system [10].

- Non Functional Requirement 1: The system shall be available and operational 24 hours a day.
- Non Functional Requirement 1.1: The system shall be operational throughout the year except for a maximum downtime of 2 hours every 30 days for maintenance.
- Non Functional Requirement 1.2: The system shall recover in a maximum of 30 seconds upon failure.

These requirements are shown in text format and we are interested in transforming these textual requirements into graphical form so that the gap between the textual requirements and the overall system model should be bridged.

3.3 A Textual Editor

Parameters of RELAX operators are typed as follows: p is an atomic proposition, e is an event, t is a time interval, f is a frequency and q is a quantity. An event is a notable occurrence that takes place at a particular instant in time. A time interval is any length of time bounded by two time instants. A frequency defines the number of occurrences of an event within a given time interval. If the number of occurrences is unspecified, then it is assumed to be one. A quantity is something measurable, that is, it can be enumerated. In particular, a RELAX expression φ is said to be quantifiable if and only if there exists a function Δ such that $\Delta(\varphi)$ is a quantity. A valid RELAX expression is any conjunction of statements $s_1 \dots s_m$ where each s_i is generated by the following grammar:

$$\begin{aligned} \varphi := & \text{true} \mid \text{false} \mid p \mid \text{SHALL } \varphi \\ & \mid \text{MAY } \varphi_1 \text{OR MAY } \varphi_2 \\ & \mid \text{EVENTUALLY } \varphi \mid \varphi_1 \text{UNTIL } \varphi_2 \\ & \mid \text{BEFORE } e \varphi \mid \text{AFTER } e \varphi \mid \text{IN } t \varphi \\ & \mid \text{AS CLOSE AS POSSIBLE TO } f \varphi \\ & \mid \text{AS CLOSE AS POSSIBLE TO } q \varphi \\ & \mid \text{AS } \{\text{EARLY, LATE, MANY, FEW}\} \\ & \text{AS POSSIBLE } \varphi \end{aligned}$$

The semantics of RELAX expressions is defined in terms of FBTL (Fuzzy Branching Temporal Logic) [2]. FBTL can describe a branching temporal model with uncertain temporal and logical information. It is the representation of uncertainty in FBTL that makes it suitable as a formalism for RELAX. Figure 2 shows a snapshot of the grammar written in Xtext.

Xtext comes up with a handful of wizards, with the help of which one can define their DSL. After defining the DSL, the next step to follow is to define the grammar. The grammar specifies the meta model and the concrete syntax for DSL. To specify the grammar, Xtext grammar language is used.

3.4 Code Generator

```

grammar org.jmb.Relax with org.eclipse.xtext.common.Terminals

generate relax "http://www.jmb.org/Relax"

Model :
  (imports+=Import)*
  (elements+=RelaxExpression)*;

Import :
  'import' importURI=STRING;

RelaxExpression :
  SimpleType | AtomicExpression;

SimpleType :
  'type' name=ID;

AtomicExpression :
  'statement' name=ID ('extends' extends=[RelaxExpression])? '
  properties+=Property*
  types+=Type*
  ';

Property :
  'property' name=ID '=' text=STRING;

Type :
  'type' name=ID '=' text=STRING;

```

Figure 2: RELAX Grammar.

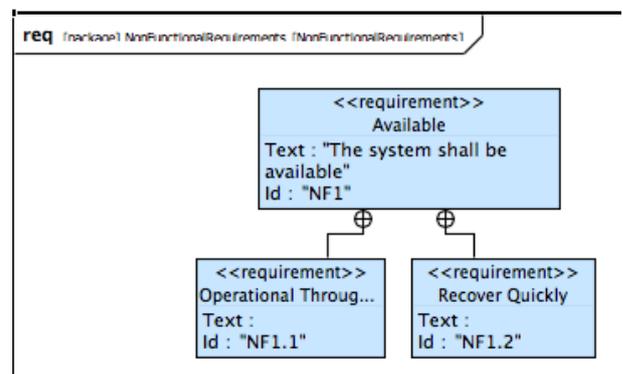


Figure 3: Generated SysML Requirements Diagram.

Among the benefits of Xtext, we have benefited from the code generation framework that is automatically generated from the grammar. A code generator has been written that is capable of processing the models created with the DSL editor. Figure 4 shows the generated code. On the right side of the screenshot, the generated SysML requirements are shown. We have not fully explored the possibilities of the language for several reasons. The first one is the fact that the grammar is not yet completed. The second one is that we have found a bit disappointing that Xtext has defined its own transformation language. We might want to use another, more widely used transformation language such as Atlas Transformation Language (ATL) [11] or Kermeta [12].

3.5 Transformed Requirements

Non-functional requirements in textual format are being transformed into graphical format with the help of RELAX grammar. Figure 3 shows a snapshot of the generated requirements diagram.

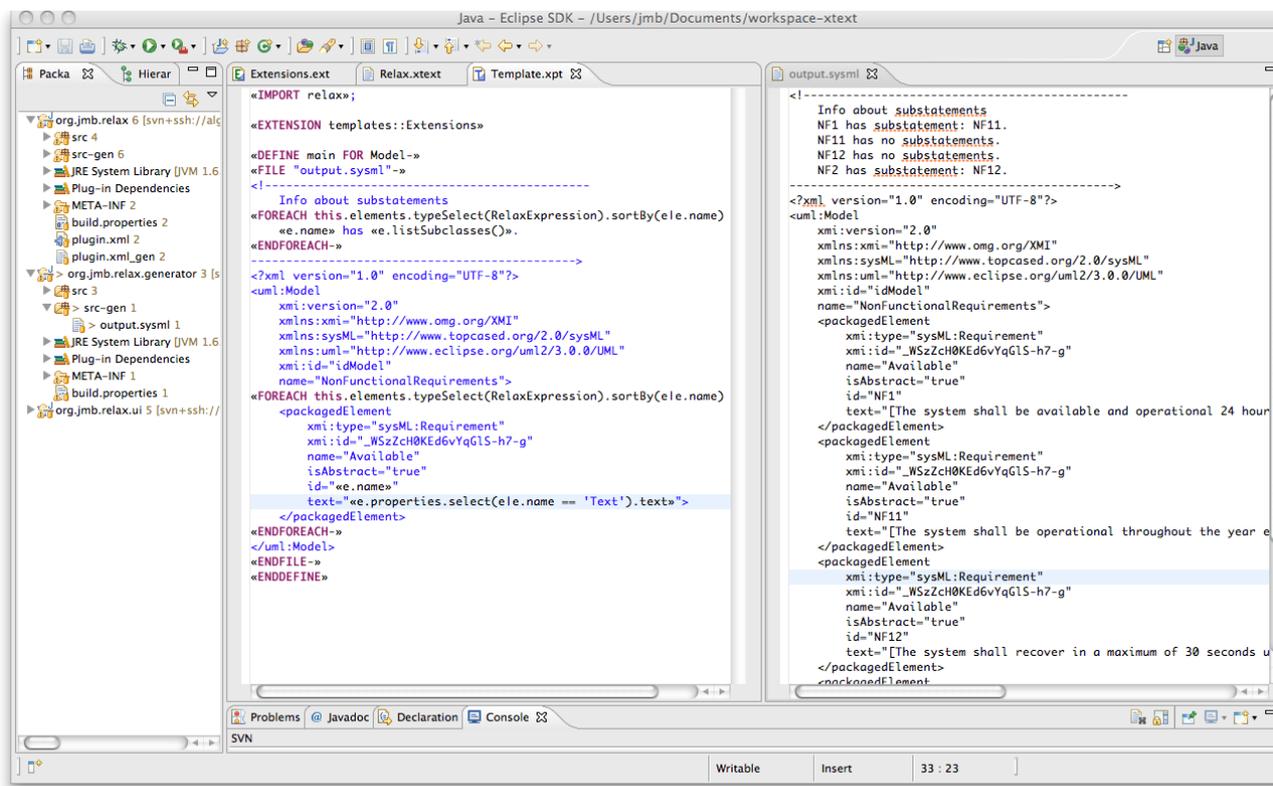


Figure 4: Generated Code.

4. CONCLUSION

Self-adaptive systems are capable of autonomously modifying their behavior at run-time in response to changing environmental conditions. Requirements play an important role, as they tend to change for these systems. For this, we need to identify those requirements that are concerned with the adaptability features of the self-adaptive system. In this paper an approach for the development of self adaptive systems is shown using RELAX which is a new requirements engineering language for Dynamic Adaptive Systems, where explicit constructs are included to handle uncertainty and a SysML profile conceived for the self adaptive systems. As a first step, a domain specific language is developed based on RELAX that would bridge the gap between requirements even in a formatted version such as RELAX statements and the overall system model.

5. FUTURE WORK

In this paper, a code editor is generated for RELAX; which is a requirement engineering language for self-adaptive systems. Future extensions of this work includes the development of all RELAX operators. We have started the RELAX editor with a small subset of the RELAX grammar. A DSL for RELAX would incorporate all the functionalities defined in RELAX and an advanced form of this work would be a graphical editor with all the variables embedded in it. This work is linked to the development of SysML profile for adaptive systems which is in the process of conception. The presented work is a part of an integrated work plan as depicted in Figure 1 so future work is centered around the

development of SPAS.

6. REFERENCES

- [1] Jon Whittle, Pete Sawyer, Nelly Bencomo and Betty H.C. Cheng. A language for self adaptive systems. Proceedings of the 2008 International Workshop on Service-Oriented Computing Consequences for Engineering Requirements Pages: 24-29 ISBN: 978-0-7695-3631-6.
- [2] Jon Whittle, Pete Sawyer, Nelly Bencomo, Betty H.C. Cheng and Jean-Michel Bruel. RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive systems. Proceedings of the 2009, 17th IEEE International Requirements Engineering Conference, Pages: 79-88 ISSN:1090-705X, 978-0-7695-3761-0.
- [3] Jean-Michel Bruel, Nicolas Belloir and Manzoor Ahmad. SPAS: un profil SysML pour les systèmes auto-adaptatifs. In 15ème Colloque National de la Recherche en IUT (CN-RIUT), Lille, June 8th to June 9th, 2009.
- [4] Bruel J.-M., Ahmad M., Beugnard A. From Composition to Connectors. 4th International School on Model Driven Development for Distributed Realtime Embedded Systems (MDD4DRES) April 20-24, 2009 Aussois France.
- [5] <http://www.eclipse.org/Xtext/documentation/>
- [6] Nauman A. Qureshi, Anna Perin. Engineering adaptive requirements. ICSE Workshop on Software Engineering for

Adaptive and Self-Managing Systems 2009 Vancouver, BC, Canada May 18-May 19 ISBN: 978-1-4244-3724-5.

[7] Mohammed Salifu, Yijun Yu and Bashar Nuseibeh. Specifying Monitoring and Switching Problems in Context. 15th IEEE International Requirements Engineering Conference (RE 2007) Delhi, India October 15-19 October ISBN: 0-7695-2935-6.

[8] Daniel A. Sadilek: Domain-Specific Languages for Wireless Sensor Networks. Modellierung 2008, 12.-14. März 2008, Berlin.

[9] J. Whittle, P. Sawyer, N. Bencomo, and B. H. C. Cheng. Reassessing languages for requirements engineering of self-adaptive systems. In RE Workshop for SOCCER08. Technical Report, 2008.

[10] Jorg Kienzle, Nicolas Guelfi and Sadaf Mustafiz. Crisis Management Systems: A Case Study for Aspect-Oriented Modeling. SOCS-TR-2009.3 Version 1.0.1.

[11] Frédéric Jouault, Freddy Allilaire, Jean Bézivin, Ivan Kurtev and Patrick Valduriez. ATL: a QVT-like transformation language. OOPSLA Companion, 2006, pp. 719-720.

[12] Zoé Drey, Cyril Faucher, Franck Fleurey, and Didier Vojtisek. Kermeta language reference manual, 2006.

Scénarios pour un Micro Smart Grid Autonominique

Sylvain Frey^{1 2}
Ada Diaconescu¹

François Huguet¹
David Menga²

Isabelle Demeure¹
Cédric Mivielle¹

¹ Télécom ParisTech – CNRS LTCI
46, rue Barrault
75634 Paris Cedex 13 FRANCE

{prénom}.{nom}@telecom-paristech.fr

² EDF R&D

1, Avenue du Général de Gaulle
92140 Clamart FRANCE

{prénom}.{nom}@edf.fr

RÉSUMÉ

L'informatique autonome est une vision inspirée par la biologie, qui tente de répondre aux problématiques croissantes de complexité dans les systèmes informatiques modernes, hétérogènes, dynamiques et à grande échelle. Ce domaine récent recherche aujourd'hui la maturité en se confrontant à des applications concrètes qui mettraient en avant ses avancées dans la gestion de la complexité.

Nous pensons que le cas des *micro smart grids* - réseaux électriques intelligents à l'échelle d'un quartier - constitue un défi à la mesure des ambitions autonomiques, ce que nous nous proposons d'illustrer à travers divers scénarios. Ce travail de mise en scène des capacités autonomiques, inédit dans le cadre *smart grid*, révèle les verrous - mais aussi les perspectives nouvelles - inhérents à la mise en œuvre de ces nouveaux types d'écosystèmes.

Categories and Subject Descriptors

J.7 [Computer Application]: Computers in Other Systems – *command and control, consumer products.*

General Terms

Design, Experimentation, Human Factors.

Keywords

informatique autonome, scénario, vision, systèmes décentralisés, micro smart grids.

1. INTRODUCTION

Avec la démocratisation de l'informatique, l'explosion d'Internet et aujourd'hui la diffusion à grande échelle d'équipements mobiles communicants - *smart phones*, tablettes, etc. - la complexité des systèmes et environnements informatiques ne cesse de s'accroître. Les composantes de cette complexité sont multiples : hétérogénéité, dynamisme, grand nombre des équipements, mais aussi présence d'utilisateurs humains, eux-mêmes complexes et imprévisibles, au centre des systèmes informatiques. Il est donc nécessaire de proposer des solutions d'une part à la complexité

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'11, June 6–8, 2011, Toulouse, France.

intrinsèque des systèmes informatiques (du point de vue du concepteur, de l'administrateur, en un mot de l'expert) et d'autre part à la complexité apparente, perçue par l'utilisateur humain (non-expert).

L'informatique autonome est une vision [1,2] qui propose des solutions conceptuelles et pratiques à la gestion de la complexité. L'essor de ce domaine a apporté de nombreuses contributions [3] à la compréhension et à la conception de tels systèmes complexes. Cependant, faute de *killer application*, la révolution annoncée par les fondateurs a fait long feu [4]. Le vision autonome peine aujourd'hui à se concrétiser au-delà de preuves de concept, et la diffusion du domaine dans le monde industriel piétine.

Il est donc crucial pour la communauté autonome de progresser dans la direction de l'applicabilité de ses solutions. Notamment, l'ingénierie sera au cœur des apports qui permettront au domaine autonome d'accomplir la destinée esquissée par ses fondateurs [4]. Dans cette optique, nous pensons que l'élaboration de scénarios, réalistes et complexes, représente une étape préliminaire importante dans la compréhension des systèmes en jeu. A partir de ces scénarios, une analyse de la complexité et des enjeux de conception ouvrira la voie à une démarche d'ingénierie, en permettant le prototypage de ces systèmes.

Le *smart grid* [5] est le réseau électrique du futur, issu de l'augmentation des réseaux actuels par des systèmes d'information sophistiqués. Sa conception et sa réalisation posent de nombreux problèmes, entre autres de complexité, aux spécialistes du domaine. De nombreuses évolutions seront nécessaires avant de parvenir à un réseau efficace, robuste, et à grande échelle. Dans le même temps, le domaine manque de visibilité quant aux conséquences techniques, sociales et économiques qui accompagneront l'émergence de ces nouveaux réseaux.

Nous proposons donc le cas des *micro smart grids* – réseaux électriques intelligents à l'échelle d'un quartier – comme cadre global pour une série de scénarios. Chacun de ces scénarios sera accompagné d'une grille de lecture analysant la complexité mise en jeu. Enfin, des solutions à cette complexité, inspirées par l'informatique autonome, seront proposées et discutées. La dimension socioculturelle de ces scénarios, soulevant de nombreuses questions fondamentales pour la réalisation des *smart grids*, ne sera pas traitée ici.

L'organisation de ce papier sera la suivante : après une présentation des domaines de l'informatique autonome et des *smart grids*, chaque scénario sera exposé, accompagné d'une analyse et d'une proposition de solution. En conclusion, les implications futures de ces travaux de scénarisation et d'analyse seront discutées.

2. INFORMATIQUE AUTONOMIQUE

La vision autonome, au carrefour d'influences multiples telles que la biologie, le génie logiciel et l'intelligence artificielle, vise à améliorer la gestion de la complexité dans les systèmes informatiques modernes, hétérogènes, dynamiques et à grande échelle. Après un court résumé des principes fondateurs, certains verrous seront identifiés, qui justifieront l'utilité des scénarios présentés plus loin.

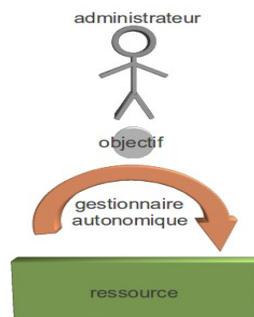
2.1 Principes fondateurs

Le système nerveux autonome, situé dans la colonne vertébrale, est un sous-ensemble du système nerveux central chez les vertébrés. Il régule en permanence, de manière inconsciente, des fonctions vitales de l'organisme telles que le rythme cardiaque, la digestion ou l'équilibre. Ainsi, le cerveau libéré de ces préoccupations de bas niveau peut se focaliser sur des tâches de haut niveau d'abstraction. Par exemple, la marche ne fait pas intervenir la gestion de l'équilibre au niveau conscient : le marcheur est concentré sur son objectif (où aller) et non sur les moyens de l'atteindre (comment y aller).

Inspiré directement par cette capacité à gérer de manière inconsciente et transparente des aspects de bas niveau d'un système complexe – l'organisme – un système informatique autonome est défini par un certain nombre de propriétés qu'il est capable de maintenir seul, sans intervention extérieure d'un administrateur humain. Ces propriétés, parfois appelées *auto-** ou *self-**, sont historiquement les suivantes [2] :

- **auto-configuration** : un système autonome peut intervenir sur lui-même et s'adapter à son contexte de manière proactive, sans intervention extérieure ;
- **auto-protection** : un système autonome se protège contre les menaces extérieures et anticipe les variations de son environnement ;
- **auto-réparation** : un système autonome détecte et répare ses erreurs internes ;
- **auto-optimisation** : un système autonome cherche en permanence à accomplir au mieux les tâches pour lesquelles il a été conçu.

Illustration 1 : Architecture de référence d'un élément autonome



L'architecture de référence d'un élément autonome est celle d'une boucle de rétroaction, appelée gestionnaire autonome. Celui-ci fournit une interface de haut niveau à l'administrateur, qui peut exprimer des objectifs de haut niveau pour la ressource. Étant donné ces objectifs, éventuellement contradictoires et

dynamiques, le gestionnaire supervise et adapte la ressource en conséquence. La gestion autonome se doit d'être transparente : si idéalement l'humain n'a jamais à intervenir pour des aspects de bas niveau, il peut tout de même reprendre la main à tout instant s'il le désire.

2.2 Évolutions, verrous

Un élément autonome, dans sa forme de référence, est donc constitué d'un couple ressource-gestionnaire autonome suivant un objectif de haut niveau. Cependant, les systèmes actuels – hétérogènes, à grande échelle, dynamiques, ouverts – amènent à considérer une complexité grandissante :

- de multiples ressources, avec des modes de fonctionnement et des usages différents, voire incompatibles, avec des nécessités de collaboration pour un résultat global ;
- de multiples boucles autonomiques, adressant des aspects de gestion particuliers (performance, énergie, sécurité, etc.) à de multiples niveaux de granularité (ressource, organisation de ressources, zone géographique, etc.) ;
- de multiples objectifs, potentiellement incompatibles, avec diverses priorités, et concernant divers ensembles d'éléments autonomiques ;
- de multiples humains non-experts, à la fois utilisateurs et administrateurs.

Les interactions parfois ambiguës entre ces entités se multipliant de manière incontrôlable, il devient difficile, voire impossible, d'appréhender le fonctionnement de tels systèmes – a fortiori, de les concevoir de manière traditionnelle. L'architecture de référence ne suffisant pas à guider la conception de multiples éléments autonomiques en interaction, de nouvelles propositions architecturales seront nécessaires dans le domaine.

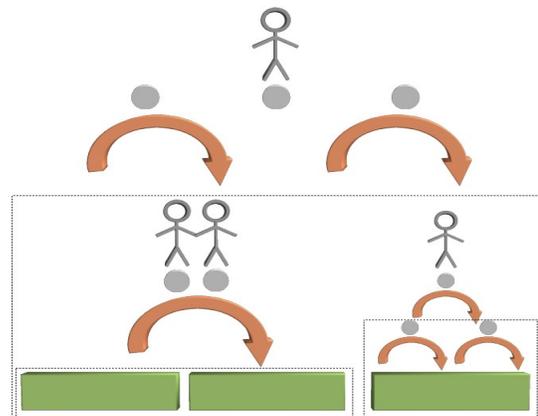


Illustration 2 : Système autonome complexe, composé de multiples éléments autonomiques.

Nous proposons donc d'établir des scénarios accompagnés de grilles de lecture et de propositions architecturales adaptées, avec pour but d'avancer dans la compréhension de la complexité et de sa gestion dans un cadre autonome.

3. MICRO SMART GRIDS

Cette partie présente une vision simplifiée des réseaux électriques et de la problématique de la gestion active de la demande.

3.1 Gestion active de la demande

Le réseau électrique peut s'assimiler à un graphe dont les arêtes (lignes haute tension, câbles domestiques, etc.) transportent l'énergie électrique depuis les nœuds producteurs (centrales, éoliennes, panneaux photovoltaïques, etc.) jusqu'aux nœuds consommateurs : appareils ménagers, éclairages, chaudières, etc. Certains nœuds sont à la fois consommateurs et producteurs, on parle alors de « prosommateur » ou « *prosumer* » ; c'est notamment le cas des stocks (batteries, centrale hydroélectrique, ballons d'eau chaude, etc.) qui peuvent consommer une certaine quantité d'énergie, puis la restituer ultérieurement.

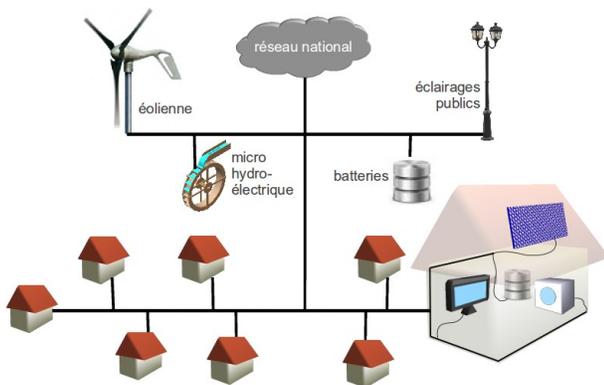


Illustration 3 : Exemple de réseau électrique de quartier, avec le sous-réseau d'un foyer représenté.

Ici, la taille du réseau considéré sera limitée à l'étendue d'un quartier, éventuellement d'une résidence ou d'un campus. Ainsi, on peut considérer différentes granularités :

- les réseaux particuliers, à l'intérieur d'une habitation ou d'un bâtiment ; ces réseaux sont délimités par un compteur, qui identifie et mesure les échanges électriques avec l'extérieur ;
- le réseau du quartier, qui comprend et relie entre eux les réseaux particuliers ;
- le réseau national, englobant le réseau de quartier.

Outre l'aménagement et l'entretien, il est nécessaire de contrôler à tout instant la *charge* des réseaux, en particulier, que production et consommation s'équilibrent. Un écart trop important entre ces deux valeurs peut entraîner la détérioration des équipements, voire des arrêts en cascade (*black-out*). Concrètement :

- quand la charge est haute (consommation supérieure à la production), la puissance fournie par les producteurs est augmentée, de nouveaux moyens de production sont démarrés, les consommations en cours sont réduites, arrêtées ou reportées.
- quand la charge est basse (production supérieure à la consommation), la puissance fournie par les producteurs

est réduite, certains moyens de production sont éteints, les consommateurs qui le peuvent se déclenchent prématurément.

Les moyens de stockage jouent ici un rôle de tampon, se comportant comme consommateur ou producteur en fonction de la situation, dans la limite de leur capacité de charge maximale. La gestion d'un réseau électrique ne consiste donc pas uniquement à limiter les consommations et maximiser les productions, mais bien à gérer leur répartition dans le temps.

3.2 Micro smart grid

L'augmentation incessante des volumes consommés rend la gestion globale du réseau électrique de plus en plus coûteuse et complexe. L'écart entre les creux et les pics de consommation (e.g. lors des soirées d'hiver) oblige les fournisseurs à surdimensionner leurs moyens de production, pour pouvoir faire face à une demande maximale rare mais toujours plus élevée.

Dans le même temps, des équipements de production et de stockage (panneaux solaires, éoliennes, batteries, etc.) font ou vont faire leur apparition sur le marché. Les sous-réseaux particuliers, jusqu'alors exclusivement consommateurs, commencent à acquérir la capacité de participer à la gestion active de la charge. Se pose alors le problème de la gestion de millions de moyens de production et de stockage additionnels, détenus par des particuliers donc moins contrôlables que les moyens traditionnels. En effet, pour des raisons, entre autres, de respect de la propriété et de la vie privée, il ne sera pas toujours possible d'intervenir directement sur les équipements en jeu.

Pour faire face aux défis de gestion que représentent les réseaux électriques dans un futur proche, la vision *smart grid* propose le doublage du réseau électrique traditionnel par un réseau informatique via lequel s'effectue une gestion intelligente de la charge électrique. L'appellation *micro smart grid* s'applique aux réseaux de la granularité considérée ici (quartier, résidence) [5].

Dans un *micro smart grid*, chaque foyer est acteur du réseau en revendant sa production ou en adaptant sa consommation en fonction de la charge. Le système qui gère le réseau du quartier rend compte en direct de la charge qui détermine les tarifs de vente et d'achat en cours, en fonction des abonnements des usagers. Les productions et consommations des foyers sont mises en relation, permettant ainsi une gestion fine de l'offre et de la demande, pratiquement usage par usage.

La nature de la gestion du quartier et les règles qui y régissent les échanges énergétiques restent encore à déterminer. Ces questions sont d'ordre économique, politique, sociétal et commercial, nous ne les aborderons pas ici. Les scénarios présentés plus loin seront donc agnostiques à ce sujet, et ouverts à des analyses futures de ces aspects.

Au final, une gestion efficace de la charge est bénéficiaire pour tous les acteurs du *micro smart grid* :

- les particuliers, par une meilleure gestion de leurs productions et consommations, réduisent leur facture énergétique tout en prenant part activement à l'établissement d'un réseau électrique plus efficace pour tous ;

- le réseau de quartier permet l'inter-médiation entre les participants au *micro smart grid*, valorise les équipements publics locaux et réduit l'empreinte énergétique du quartier vis à vis du réseau national ;
- le réseau national est soulagé de la gestion électrique du quartier, qui est capable de s'adapter au mieux à la charge nationale : le *micro smart grid* contribue donc aussi directement au *smart grid* global.

3.3 Autonomique pour le micro smart grid

Les scénarios présentés ici visent à mettre en scène la vision *micro smart grid* en tant qu'exemple canonique de système complexe, tout en proposant des solutions autonomiques à la gestion de cette complexité. Notamment, les idées autonomiques et *micro smart grid* se rejoindront sur les points suivants :

- les équipements doivent prendre des décisions, en fonction d'objectifs de haut niveau (consommation, production, qualité de service, sécurité, etc.) et d'un contexte local imprévisible ; ils doivent être capable de collaborer pour réaliser une gestion efficace de la charge ; le caractère intrinsèquement ouvert du réseau électrique doit être prise en compte ;
- il est nécessaire de mettre en jeu des gestionnaires de haut niveau, à portée étendue (pièce, domicile, quartier, etc.) dans divers domaines (gestion de la température, de la sécurité, etc.) ;
- la gestion doit prendre en compte la présence d'humains : administrateurs et utilisateurs, éventuellement non-experts des domaines en jeu.

4. SCENARIOS & ANALYSES

Plusieurs scénarios sont présentés ici suivant une logique de complexité croissante, en considérant successivement un équipement unique, un foyer, puis le quartier tout entier. Certains des éléments utilisés, notamment les équipements intelligents ou la tarification dynamique, n'existent encore qu'à l'état de projet. Cependant, ils font partie de la vision *smart grid* et feront leur apparition au cours des années à venir [5,7].

Les scénarios sont accompagnés d'analyses des difficultés clefs, en termes de fonctionnalités nécessaires à la réalisation des systèmes mis en scène. Ces analyses sont suivies de propositions de solutions architecturales : nous ne traiterons pas de la réalisation des fonctionnalités requises, mais d'une approche générale quant à leur agencement dans un ou plusieurs systèmes autonomiques.

4.1 Scénario « chauffage »

A l'occasion de travaux de rénovation, Mark a investi dans un chauffage au sol dit « intelligent ». Grâce à un ensemble de capteurs, le système peut recevoir une consigne de température qu'il maintient au plus près. Un ballon d'eau chaude peut

accumuler de la chaleur qui est ensuite restituée, en fonction du coût de l'électricité, des consignes de Mark, de la présence des occupants dans l'appartement, de la température effective, de la météo. Mark a également activé une fonctionnalité de géolocalisation ; connaissant sa position, le système peut réduire la température pendant ses absences, et s'assure d'un retour à la normale pour son arrivée.

Mark peut suivre en direct l'évolution de son système : consommation électrique, température souhaitée et température réelle, économies réalisées grâce au stockage. L'investissement de départ a été conséquent, mais Mark perçoit vite la différence sur sa facture. Des prévisions sur plusieurs mois lui permettent d'anticiper l'amortissement de son chauffage. Outre l'aspect écologique des réductions de consommation, le régulateur s'adapte aux besoins des occupants à partir d'instructions simples. En consultant l'historique, Mark s'est rendu compte que certains réglages ponctuels qu'il effectuait manuellement – par exemple, dans sa salle de bains – ont été intégrés par le système, qui anticipe désormais sa douche matinale.

Curieux, Mark s'est aventuré à jouer avec les fonctionnalités les plus avancées du régulateur. Celui-ci lui a proposé divers profils de gestion, privilégiant le confort, les économies d'énergie ou les réductions de facture. Après plusieurs expérimentations, Mark s'est décidé pour un mode très économe qui a encore plus réduit ses dépenses énergétiques. Et lorsqu'il reçoit de la visite, le système restaure automatiquement des conditions de confort plus convenables.

Analyse : Ce scénario soulève les difficultés suivantes :

- la gestion des **conflits** entre les **objectifs** et le **contexte** dynamique et imprévisible de l'équipement considéré ;
- la nécessité d'**anticiper** ce contexte, et notamment de développer des capacités d'**apprentissage**.

L'équipement mis en scène est soumis à deux objectifs variables et contradictoires, « maintien de la température » contre « limitation de la consommation électrique », dans un contexte dynamique et imprévisible. Ce contexte doit être mesurable par le système : capteurs de température, mesure de la consommation électrique, de la charge, du stock courant, éventuellement capteurs de présence et géolocalisation. La mesure de ce contexte est par nature imprécise et doit prendre en compte de possibles erreurs (mouvements non identifiés dans une pièce, panne de GPS, etc.).

Le système doit non seulement s'adapter au contexte (température, charge du foyer) mais il doit être aussi capable de le prévoir pour parvenir à une gestion efficace de son stock. Dans le scénario, le système doit anticiper le retour de Mark du travail ; il recharge son stock pendant la journée en fonction d'une prévision de la consommation pour la soirée. Cette prédiction peut nécessiter certaines capacités d'apprentissage à plus ou moins long terme, pour évaluer les besoins d'une soirée standard, les variations possibles de la demande, etc.

Des événements imprévus (pic de charge inattendu, consigne impromptue de l'utilisateur, vague de froid ou de chaleur, etc.) peuvent venir remettre en cause les prévisions et le fonctionnement normal du système. Cependant, il doit être capable d'anticiper de tels écarts, de prévoir une marge de manœuvre pour faire face aux imprévus et de réagir de manière adéquate aux perturbations extérieures. Des objectifs sophistiqués peuvent aider le système à trancher le dilemme « marge de manœuvre contre optimisation ».

Solution proposée : La gestion du système de chauffage fait intervenir deux fonctionnalités principales : le maintien de la température et la gestion du stock énergétique. Chacune fait intervenir des logiques spécifiques à son domaine : sensibilité au contexte, prévision, décision, apprentissage. Suivant le principe de la séparation des préoccupations, nous proposons donc de les isoler chacune dans une boucle autonome dépendant d'un objectif : maintien de la température ou limitation de la consommation en énergie (cf. Illustration 4).

Le découplage entre les deux boucles apporte une bonne modularité et une flexibilité qui permet le passage à l'échelle, comme nous le verrons dans les scénarios suivants. Une telle architecture décentralisée a déjà été testée dans un scénario de ce type [8]. Dans des situations plus critiques ou moins ouvertes, un gestionnaire unique – offrant un meilleur contrôle dans une logique de décision centralisée – pourrait être préférable.

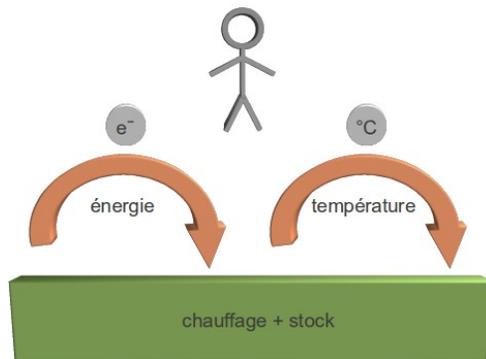


Illustration 4 : Proposition d'architecture pour la gestion autonome d'un chauffage avec stock.

Par rapport à une solution avec une boucle de gestion unique, il est nécessaire de mettre en place une logique de coordination entre les deux gestionnaires (partage des prévisions, adaptation de la gestion du stock en fonction des objectifs de confort ou vice-versa). Cette logique, résolvant le conflit intrinsèque entre les deux objectifs, peut être intégrée à chaque gestionnaire (exemple : [9]) ou être assemblée en un troisième gestionnaire contrôlant les deux premiers (exemple : [10]). Les modalités de la décision, privilégiant l'un ou l'autre des objectifs, dépendent des objectifs de haut niveau exprimés par l'utilisateur. Les avantages d'une architecture à gestionnaires multiples se révéleront plus en détails dans la suite des scénarios, où nous supposerons que la logique de résolution de conflit est distribuée entre les gestionnaires impliqués.

4.2 Scénario « chambre »

Poursuivant ses expérimentations de nouveaux équipements pour son appartement, Mark a installé dans sa chambre une fenêtre de toit (de type VELUX) sophistiquée. Pourvue d'un actionneur électrique pilotable, la fenêtre possède un contrôleur « intelligent », qui récupère les données des capteurs de température et de présence existant dans la pièce, les consignes du système de chauffage, mais aussi la météo de la région. Outre

des options de contrôle à distance, le contrôleur propose divers programmes, destinés à renouveler l'air de la pièce tout en entretenant sa température.

Le fonctionnement de la fenêtre ne posant pas de problème de sécurité, Mark lui donne l'objectif de s'ouvrir 10 minutes par jour, à condition que personne ne se trouve dans la pièce et que les conditions extérieures (température, pluie, vent) le permettent, notamment vis à vis de l'objectif de température existant. En cas de conditions continuellement défavorables, la fenêtre ne s'ouvre pas et rapporte l'événement.

En consultant l'historique de ses systèmes, Mark a pu constater que loin de mettre en péril ses objectifs d'économie d'énergie, sa nouvelle fenêtre participe au contraire à la régulation thermique ; en s'ouvrant aux moments les plus opportuns, elle soulage parfois la tâche du système de chauffage.

Analyse : Ce scénario pose les problématiques suivantes :

- l'**intégration** d'équipements divers, à petite échelle.
- la **collaboration** entre équipements pour la gestion d'un objectif commun (ici, la température).

Pour simplifier le scénario, nous considérerons que la consommation électrique de la fenêtre est négligeable et qu'elle ne participe donc pas à l'objectif de limitation des dépenses énergétiques.

Dans sa forme la plus simple, le degré d'intégration entre le chauffage et la fenêtre peut être minimal : ils n'ont aucune relation directe et interagissent au travers de leur environnement physique par leur participations respectives au maintien de la température dans la chambre. Ce type d'intégration, bien que limité, est tout à fait viable [8].

Dans une forme plus avancée, les équipements doivent partager des prédictions de leur comportement propre et se prendre en compte l'un l'autre dans leur logique de gestion. Par exemple, en période de grand froid, la fenêtre prévient le radiateur de la nécessité d'une baisse de température au cours de la journée. Le radiateur indique en retour le moment où il estime qu'il sera au mieux capable de faire face à cette baisse (car son stock est plein, la charge est basse, le retour des utilisateurs n'est pas imminent, etc.). L'accord entre les deux permet d'effectuer l'ouverture de la fenêtre au moment le plus opportun, avec une perturbation minimale, et à moindre coût, de la température de la pièce.

Cette « sociabilité » des équipements influence leur conception à la source, par la prise en compte d'interactions avec d'autres systèmes potentiellement imprévisibles et facteurs d'erreurs. Ainsi, l'intégration d'un équipement convenablement conçu dans un système plus vaste ou différent ne change pas son implémentation de manière fondamentale. Le caractère « situé » des systèmes doit donc être au centre des préoccupations des concepteurs.

Solution proposée : En reprenant la proposition du scénario précédent, la situation est la suivante : l'utilisateur spécifie trois objectifs (consommation, température, ventilation) qui sont traités par quatre gestionnaires : deux pour le chauffage et deux pour la fenêtre. Dans le cas où les deux équipements communiquent, une fonctionnalité de collaboration doit être ajoutée entre les boucles responsables de la gestion de la température (voir Illustration 5).

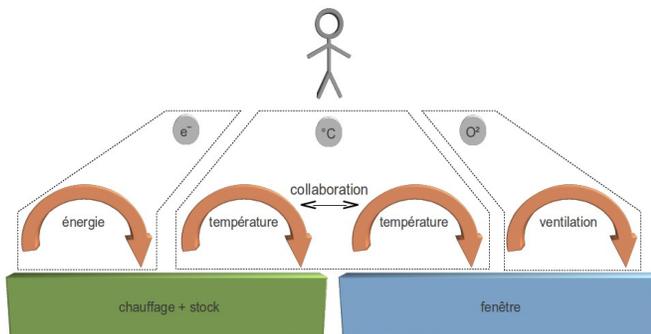


Illustration 5 : Proposition d'architecture pour la chambre autonome.

Si la fonctionnalité de collaboration demande une extension des deux gestionnaires de température, elle n'affecte en rien les deux autres boucles. Ainsi, les gestionnaires « énergie » et « ventilation » peuvent être conçus indépendamment et réutilisés, que l'équipement en question soit solitaire ou intégré avec d'autres. Les gestionnaires de température peuvent également réutiliser des fonctionnalités communes (gestion des erreurs de mesures, profils thermiques, prévisions météorologiques, etc.) voire les mutualiser.

L'architecture proposée vise à rendre le système aussi ouvert et extensible que possible. L'introduction d'un gestionnaire supplémentaire (par exemple, gestion de la sécurité pour la fenêtre) ne vient pas remettre en cause le cœur des fonctionnalités des autres gestionnaires, mais simplement la logique d'intégration entre les boucles directement en contact. L'ajout d'autres équipements concernés par un objectif donné (par exemple, pour la température : un chauffage d'appoint, une climatisation) n'affecte pas non plus les objectifs ou les gestionnaires dépendant des autres aspects (énergie, ventilation, sécurité, etc.).

4.3 Scénario « réseau de quartier »

Sal, la voisine de Mark, vient de lancer son lave-linge. Son appartement n'étant pourvu ni de moyens de production, ni de moyens de stockage, il rejoint la liste des foyers consommateurs du quartier.

Le foyer de Mark, à présent pourvu d'une batterie en complément des équipements existants, a acquis la capacité de stocker et revendre de l'électricité. Détectant une hausse de la charge du réseau du quartier, le système effectue une analyse interne pour décider des suites à donner. Il parvient aux conclusions suivantes : la batterie est chargée, la ventilation des pièces de la maison a déjà été effectuée, aucune consommation notable n'est envisagée dans l'immédiat et le stock du ballon d'eau chaude est a priori suffisant pour faire face seul à d'éventuels aléas thermiques.

Le système de Mark décide donc de consacrer une partie de la charge de la batterie à la demande extérieure. Le tarif de rachat est avantageux pour Mark ; de son côté, le foyer de Sal bénéficie d'une électricité locale, donc moins onéreuse.

Un peu plus tard, l'arrivée imprévue de Mark vient augmenter la consommation de son domicile. Entreprenant une séance de

travaux ménagers d'ordinaire réservée aux week-ends, Mark a ouvert ses fenêtres en grand. Les radiateurs, pris de court, se voient contraints de puiser dans leur stock et font même appel à la batterie pour rétablir une température confortable.

Le système de Mark n'étant plus en mesure d'honorer la demande sans risquer de préjudice à ses propres intérêts (thermiques et économiques) il cesse donc d'alimenter le quartier, qui perd un fournisseur pour les consommations de Sal.

Analyse : L'exemple du réseau de quartier est typique des organisations rassemblant des entités concurrentes, avec des objectifs propres. Chaque foyer tente de minimiser sa facture et maximiser ses profits, même si certains critères moins égoïstes peuvent entrer en ligne de compte. Dans l'absolu, le réseau de quartier n'a pas de contrôle direct sur les différents foyers ; il peut simplement inciter à la collaboration par des offres tarifaires avantageuses. Les décisions d'achat et de vente ne sont prises qu'au niveau des foyers, soumis à des contraintes internes impondérables, comme montré ici.

Le scénario introduit plusieurs difficultés importantes :

- le **passage à l'échelle** ;
- la nécessité d'une **gestion globale** ;
- l'**hétérogénéité** des domaines en interaction.

La première de ces difficultés exclut toute solution entièrement centralisée, qui ne supporterait pas, ou difficilement, la gestion de centaines de foyers prenant leurs propres décisions. De plus, sur le plan du respect de la vie privée, il n'est pas envisageable que des utilisateurs confient la gestion des équipements de leur vie quotidienne à une entité extérieure. Cette nature « bottom-up » de la gestion de la charge est un changement de paradigme profond, qui fait écho à la décentralisation des premiers scénarios.

La gestion globale est liée à l'équilibre des intérêts des différents partis (foyers, quartier, réseau national). Étant données les variations inéluctables sur le long terme (moyens de production, offre moyenne, demande moyenne, coûts de l'énergie) il est indispensable pour le réseau du quartier de se doter de moyens de mesure et de régulation globaux – offres tarifaires, moyens de production et de stockage. Les politiques de cette régulation dépendront des règles qui régissent le réseau de quartier, lesquelles sont ici volontairement laissées dans le flou.

Enfin, l'hétérogénéité des domaines en interaction apparaît au travers de la succession de perturbations en fin de scénario : le retour imprévisible de Mark, entraînant une perturbation de la température, d'où une hausse de la consommation, d'où une baisse de la production au niveau du quartier, d'où éventuellement une hausse du tarif pour le système de Sal. Cet enchaînement de conséquences illustre l'émergence de comportements imprévisibles dans les systèmes hétérogènes répartis, et plaide à nouveau pour une décentralisation des prises de décision, assortie d'une séparation des préoccupations (localisation de l'utilisateur, gestion de la température, gestion de la consommation énergétique du foyer, gestion du réseau de quartier).

Solution proposée : La solution proposée doit être compatible avec les précédentes. En effet, l'existence d'une gestion de la charge au niveau du quartier et la présence d'équipements « intelligents » dans un foyer sont des faits indépendants. Ainsi, les premiers scénarios décrivaient l'intérieur d'un foyer sans dépendre de la nature du réseau du quartier. De la même manière,

un réseau de quartier « intelligent » peut être installé sans que pour autant tous les foyers concernés ne possèdent d'équipements intelligents. Ce problème de la gestion de l'existant est une contrainte forte pour l'établissement des *smart grids*.

Nous proposons d'équiper chaque foyer d'une passerelle vers le réseau de quartier. Cette passerelle identifie le foyer et sert d'intermédiaire entre l'intérieur (privé, relevant de la gestion de l'utilisateur) et l'extérieur (relevant de la gestion du réseau du quartier). Elle pourrait se situer d'un point de vue logique au niveau du compteur électrique, qui sépare déjà réseaux publics et privés (respectivement en « amont » et en « aval » du compteur), et que certains travaux visent déjà à rendre « intelligent » [6].

Le rôle de la passerelle s'apparente à celui d'une boucle autonome : obéissant aux objectifs de l'utilisateur en termes de consommation d'énergie, elle supervise en permanence l'état du réseau du quartier, du réseau du foyer, et coordonne les productions et consommations locales avec l'état du « marché énergétique » du quartier au travers de communications de haut niveau (négociations, prévisions, transactions, etc.).

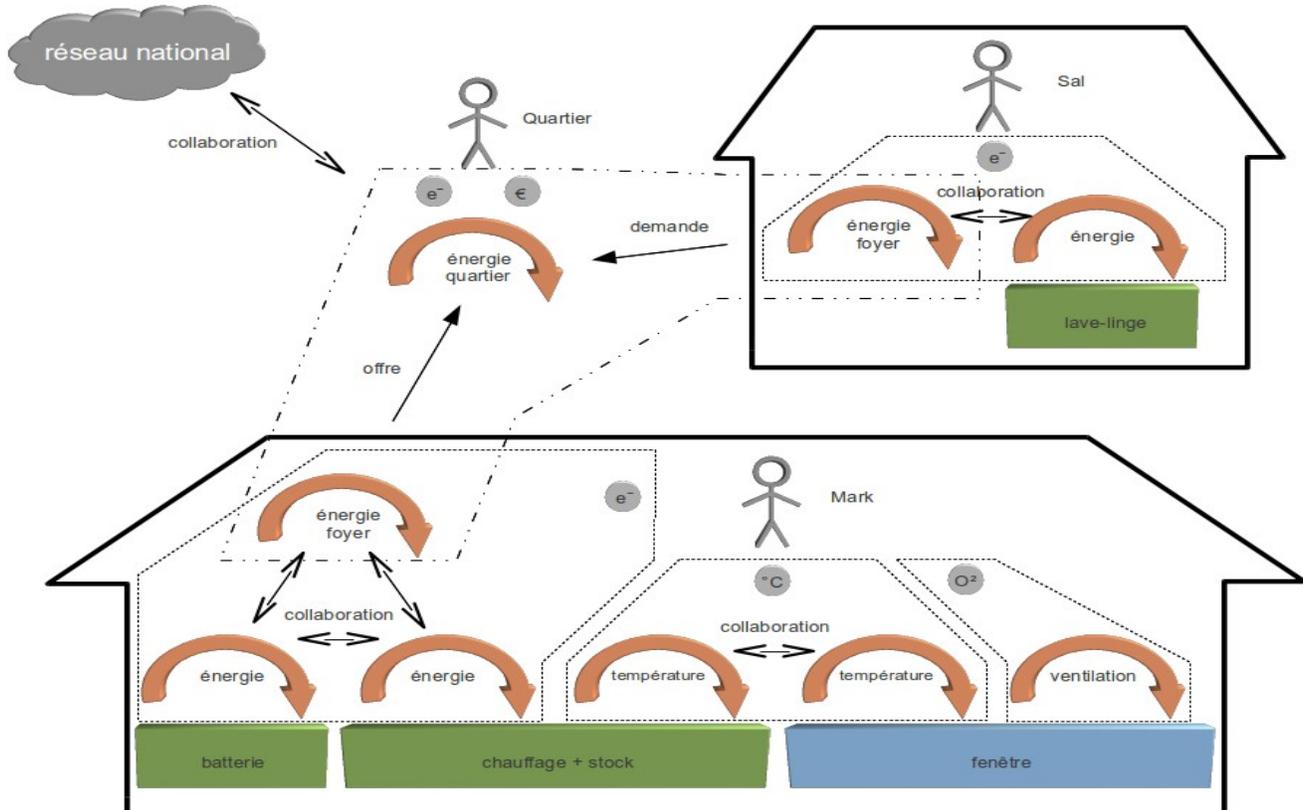
Au niveau du quartier, un gestionnaire autonome tient le rôle d'intermédiaire entre les différents foyers, en fonction de ses objectifs propres en matière de gestion énergétique, de rentabilité. Ce gestionnaire pourrait logiquement se situer au niveau du poste électrique, qui définit déjà les réseaux de taille « quartier ». C'est aussi à ce niveau que s'effectue le lien avec le réseau national pour ce qui est de la gestion de la charge à grande échelle. Cette gestion spécifique, posant notamment des problèmes de très grandes échelles, est au-delà du cadre des scénarios présentés ici, elle ne sera pas traitée.

Les relations entre les systèmes présentés dans le scénario sont complexes. À l'intérieur du foyer, la passerelle ne peut forcer directement les équipements à produire ou consommer. Son rôle se cantonne à informer les équipements de la charge du réseau du foyer, lesquels équipements prennent la décision, en fonction de cette charge et leurs propres contexte et objectifs, de produire, de consommer, etc. De même, le gestionnaire du quartier remplit un rôle similaire, en incitant les foyers à alléger ou alourdir leur charge, en fonction de la charge du quartier et de la charge nationale.

La solution proposée est structurée par les objectifs de gestion qui définissent des « domaines » (zones pointillées sur l'illustration 6). Ces domaines rassemblent tous les gestionnaires ayant une influence sur l'objectif en question, permettant ainsi l'identification des ensembles d'éléments autonomiques pertinents vis à vis d'une préoccupation donnée. Chaque domaine, avec ses règles d'administration et ses gestionnaires, peut ainsi se concevoir en séparation des autres. Les domaines interagissent entre eux au travers d'éléments autonomiques pourvus de plusieurs boucles autonomiques et appartenant donc à plusieurs domaines. Par exemple, le chauffage appartient aux domaines « température » et « électricité », avec une boucle autonome dans chacun d'eux. Les interactions entre domaines dépendent alors de la logique d'intégration entre les différentes boucles, propre aux éléments autonomiques considérés (cf. scénario « chauffage »).

A noter également que certains éléments peuvent influencer les autres sans appartenir à un domaine : par exemple, une climatisation sans gestionnaire autonome peut affecter la température d'une pièce. Un tel équipement ne faisant pas partie du système formé par l'ensemble des gestionnaires autonomiques (la partie *smart* du *smart grid*) il doit donc être considéré comme un élément de l'environnement, donc extérieur à ce système, puisque aucune gestion « intelligente » ne peut lui être appliquée.

Illustration 6 : Proposition de solution pour le scénario « réseau de quartier ».



5. CONCLUSION

Nous venons de présenter trois scénarios successifs, décrivant ce que pourrait être la gestion active de la demande énergétique dans un futur proche, à l'échelle d'un équipement, d'un foyer, d'un quartier. Ces scénarios, entre autres aspects, ont révélé la complexité intrinsèque de ce genre de système ouvert, dynamique, à grande échelle ; notamment :

- le dynamisme des équipements et des contextes dans lesquels ils évoluent ;
- les problèmes de passage à l'échelle, de la multiplication arbitraire du nombre d'équipements en jeu ;
- les problèmes posés par l'hétérogénéité des équipements, l'ouverture des systèmes, la prise en compte de l'existant ;
- la place des êtres humains, imprévisibles et exigeants, au centre des systèmes ;
- les conflits inévitables entre objectifs et contexte, entre objectifs eux-mêmes, entre autorités administratives, entre décisions prises par des entités différentes.

Pour faire face à cette complexité, nous avons effectué diverses propositions architecturales, destinées à guider l'organisation des fonctionnalités suivantes :

- la mesure, la prévision et l'adaptation au contexte local ; l'évaluation et l'adaptation à long terme des systèmes et de leurs gestionnaires ;
- la prise de décision décentralisée, en fonction du contexte local et par rapport à des objectifs, locaux et globaux ;
- la collaboration entre systèmes, prenant en compte intrinsèquement le caractère distribué et imprévisible des décisions ;

Les solutions proposées étendent l'architecture canonique des systèmes autonomiques, en multipliant objectifs et boucles de contrôle, lesquelles doivent présenter des fonctionnalités essentielles d'intégration, voire de collaboration. Les objectifs, en rassemblant les boucles autonomiques spécifiques concernées, dessinent alors les sous-structures locales appelées ici « domaines » dont la combinaison forme le système global. Cette séparation des préoccupations ouvre la voie à des supervisions et des analyses ciblées du système global et de sa complexité.

Les scénarios présentés sont destinés à être étendus et enrichis ; des prototypes successifs permettront d'en vérifier la validité et la profondeur. Nous espérons toutefois que dès à présent, leur élaboration et leur exposé auront contribué à la compréhension des *micro smart grids*, des systèmes autonomiques, et des défis conceptuels et techniques que la complexité posera aux ingénieurs et aux chercheurs de demain.

6. RÉFÉRENCES

- [1] IBM, "Autonomic Computing: IBM's perspective on the State of Information Technology", available at: <http://www.research.ibm.com/autonomic/manifesto/>
- [2] Jeffrey O. Kephart and David M. Chess. 2003. The Vision of Autonomic Computing. *Computer* 36, 1 (January 2003), 41-50. DOI=10.1109/MC.2003.1160055
- [3] Markus C. Huebscher and Julie A. McCann. 2008. A survey of autonomic computing – degrees, models, and applications. *ACM Comput. Surv.* 40, 3, Article 7 (August 2008), 28 pages. DOI=10.1145/1380584.1380585
- [4] Simon Dobson, Roy Sterritt, Paddy Nixon, and Mike Hinchey. 2010. Fulfilling the Vision of Autonomic Computing. *Computer* 43, 1 (January 2010), 35-41. DOI=10.1109/MC.2010.14
- [5] European Commission. 2006. Vision and Strategy for Europe's Electricity Networks of the Future. <http://www.smartgrids.eu/documents/vision.pdf>.
- [6] <http://linky.erdfdistribution.fr/>
- [7] Birger Becker, Florian Allerdig, Ulrich Reiner, Mattias Kahl, Urban Richter, Daniel Pathmaperuma, Hartmut Schmeck and Thomas Leibfried. 2010. Decentralized Energy-Management to Control Smart-Home Architectures. *Lecture Notes in Computer Science*, 2010, Volume 5974/2010, 150-161, DOI: 10.1007/978-3-642-11950-7_14
- [8] Sylvain Frey, Philippe Lalanda, and Ada Diaconescu. 2010. A Decentralised Architecture for Multi-objective Autonomic Management. In *Proceedings of the 2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '10)*. IEEE Computer Society, Washington, DC, USA, 267-268. DOI=10.1109/SASO.2010.34
- [9] Kephart, J.O.; Hoi Chan; Das, R.; Levine, D.W.; Tesauro, G.; Rawson, F.; Lefurgy, C.; , "Coordinating Multiple Autonomic Managers to Achieve Specified Power-Performance Tradeoffs," *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*, vol., no., pp.24, 11-15 June 2007 DOI: 10.1109/ICAC.2007.12
- [10] Rajarshi Das, Jeffrey O. Kephart, Charles Lefurgy, Gerald Tesauro, David W. Levine, and Hoi Chan. 2008. Autonomic multi-agent management of power and performance in data centers. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track (AAMAS '08)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 107-114.
- [11] Mark Weiser. 1999. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (July 1999), 3-11. DOI=10.1145/329124.329126

Adaptabilité des flux vidéos dans les logiciels collaboratifs: Test de qualité par référence prédéfinie

Jean-Baptiste Aupet

Jean-Christophe Lapayre

Yanbo Shou
LIFC - EA 4269,
Université de Franche-Comté,
16, Route de Gray
25030 BESANCON Cedex, France.
[jbaupet, jclapayre, yshou]@lifc.univ-fcomte.fr

Keywords

Adaptation, qualité, métriques, multimédia, temps-réel, streaming, synchrone, vidéo

1. INTRODUCTION

Adapter les données qui transitent au sein d'un environnement n'atteste pas que les contenus sont effectivement adaptés ou en tout cas de manière optimale. L'adaptabilité permet de mettre en œuvre des techniques lorsque l'utilisateur n'est pas satisfait du résultat obtenu. Encore faut-il que celui-ci sache comment utiliser ces techniques et surtout les utiliser au moment adéquat. Il apparaît donc la nécessité d'effectuer ces actions de manière automatisée pour éviter à l'utilisateur d'avoir à faire des manipulations pour lesquelles il n'est pas compétent ou pour lesquelles il n'a pas envie de perdre son temps dans des menus de configurations. Le fait de mettre en place une politique d'adaptabilité permet à l'utilisateur d'oublier totalement la partie configuration afin que l'adaptation de son environnement au travail soit transparent. Pour que ce système puisse totalement rendre transparente cette phase d'auto adaptation du contexte et des données, le système doit prendre en compte l'ensemble des paramètres utiles et nécessaires au fonctionnement des applications.

Il y a un paradigme en informatique qui consiste à toujours faire des compromis entre ce que l'on peut faire et ce que l'on veut faire. L'objectif principal est d'arriver à trouver le juste équilibre entre les deux afin d'arriver à un état d'auto adaptabilité correspondant à la situation et permettant de se rapprocher le plus possible d'un fonctionnement idéal.

Dans le cas de la vidéo, les paramètres à prendre en compte sont les différentes dégradations que peut subir une vidéo lorsque l'on souhaite la diffuser en temps réel à plusieurs clients d'un système collaboratif disposant tous de capacités de calcul et d'affichage différentes.

2. LE CONTENTEMENT UTILISATEUR

Le principal défi de l'adaptation est de permettre à tous les utilisateurs quelque soit leur environnement, leurs besoins et leurs préférences d'accéder aux informations le plus simplement possible. Le but de l'adaptation est d'assurer à l'utilisateur qu'il pourra profiter des outils qui lui sont proposés dans les meilleures conditions. L'adaptation peut aussi gérer le fait que les outils qui ne sont pas utilisables ou qui peuvent poser des soucis du fait des contraintes du système ne soient pas proposés ou alors dans une version dégradée ne mettant pas en péril le fonctionnement général du système. Un utilisateur ne connaissant pas suffisamment les contraintes et les besoins d'un système ne doit pas pouvoir choisir des options ou des outils qui ne fonctionneraient pas ou pire mettraient le système en défaut. Dans ce cas, le contentement de l'utilisateur serait mis à mal du fait du non fonctionnement de l'application.

L'environnement doit donc prendre en compte ces contraintes afin de ne pas offrir la possibilité à l'utilisateur de changer des paramètres qui ne seraient pas adaptés d'une part à ses capacités de prise de décisions et d'autre part aux contraintes du système.

2.1 Prise en compte du contentement

Il est possible de déterminer le contentement des utilisateurs grâce à des études sur l'utilisation des outils et des évaluations faites à posteriori pour améliorer le système selon la perception des utilisateurs. Mais cela ne permet pas de faire évoluer l'expérience des utilisateurs pendant l'utilisation du logiciel.

Pour fournir une adaptabilité en temps réel il est nécessaire d'avoir un retour immédiat¹ des utilisateurs sur leur expérience et leur utilisation de l'environnement et de ses outils afin de les adapter en temps réel suivant les désirs des utilisateurs. Cependant l'expérience peut varier selon les personnes, les environnements, et les perceptions de chacun : le caractère subjectif de cette perception est très difficile à prendre en compte. Un utilisateur connaissant le fonctionnement général de l'application saura si un problème vient de la connexion ou d'un manque de capacité de calcul et ainsi pourra effectuer les ajustements nécessaires alors qu'un utilisateur lambda qui ne connaît pas forcément les contraintes du système aura une perception négative de l'application ne retenant que les aspects problématiques.

1. feedback

2.2 La perception des utilisateurs

Chaque individu possède sa propre vision de son environnement, que ce soit dans la vie réelle ou dans son environnement de travail. Il est très difficile, parfois même impossible, d'établir des règles générales qui permettraient d'attester de manière sûre qu'un utilisateur possède un environnement de travail totalement adapté à ses besoins. C'est pour cela qu'il faut toujours fournir la possibilité à l'utilisateur d'agir sur son environnement pour qu'il le configure selon ses besoins mais aussi ses envies. En effet, un utilisateur aura besoin de certaines fonctionnalités pour effectuer un travail un jour et le lendemain voudra utiliser des outils différents, ou accédera à son espace de travail via un réseau mobile. Tous ces points doivent être pris en compte lorsque l'on doit fournir des services d'adaptabilité dans un environnement de travail.

2.3 Prise de conscience de l'adaptabilité : Awareness

Dès lors que des utilisateurs travaillent en collaboration présente, c'est-à-dire en même temps et dans le même lieu, des informations sont échangées de manière implicite ou explicite. Ces informations créent chez chaque participant une conscience de groupe, qui leur permettent :

- de comprendre et mesurer l'activité et la dynamique des autres utilisateurs et donc du groupe
- de situer sa propre action au sein du groupe, et ainsi de coordonner les activités entre elles afin d'interagir avec les autres participants.

Cette notion de conscience de groupe (awareness) est originellement définie comme étant la compréhension des activités des autres, qui permet de donner un contexte à sa propre activité. Dans une équipe virtuelle, la disparition de la co-présence entraîne la disparition de certaines informations, particulièrement les informations implicites, aboutissant à la disparition de la conscience de groupe. Le but, dans le contexte des applications de collaboration/coopération, est donc de reconstruire la conscience de groupe grâce à des modèles et par le biais de mécanismes adaptés [4].

Modéliser cette conscience de groupe nécessite :

- l'identification des informations utilisées en co-présence pour construire la conscience de groupe.
- une synthèse claire et utilisable des nombreuses informations que l'on peut obtenir des systèmes informatiques.
- une représentation à l'utilisateur d'une information pertinente mais non obstructive.

On distingue dans la littérature [2] [16] [23] trois formes de conscience de groupe :

- *group awareness* qui a pour but de rendre tangible le travail en groupe et d'être conscient de l'état et des activités des partenaires.
- *Workspace awareness* (conscience de l'espace de travail commun) qui se situe à un plus grand niveau de détail et permet aux utilisateurs travaillant sur les mêmes documents d'être au fait des modifications apportées par les partenaires situés dans le même espace de travail.
- *process awareness* (conscience du procédé commun) qui permet aux participants de situer leurs actions dans le contexte plus large d'un projet global et coordonné.

Parmi les différents modèles proposés pour résoudre le pro-

blème de la conscience de groupe dans les outils de collaboration/coopération nous pouvons citer les modèles basés sur la publication/souscription et les modèles spatiaux [16]. Le modèle publication/souscription peut être décrit par les producteurs d'information qui publient des événements et des messages vers le système et les consommateurs d'information qui souscrivent des abonnements pour certaines catégories d'événements déterminées soit par leur sujet, soit par leur contenu. Le modèle spatial, quant à lui, est fondé sur une métaphore spatiale et définit la conscience de groupe en terme de positions, proximité et distance à l'intérieur d'un espace virtuel commun [6].

2.4 Spécificités des flux vidéo : métriques d'évaluation de la qualité

Les métriques de qualité d'une vidéo sont un ensemble de techniques mathématiques qui permettent d'évaluer une vidéo selon des paramètres de dégradations qui sont perçues par l'œil humain au moment de la visualisation de la vidéo. Le principe est d'extraire et d'analyser les caractéristiques spatio-temporelles d'une vidéo (i.e. la résolution, le nombre d'images par secondes, la couleur etc.) pour obtenir une valeur finale qui peut représenter de manière objective sa qualité. Sachant qu'une vidéo est constituée d'un ensemble d'images successives, les techniques existantes sont généralement basées sur le test de qualité des images qui composent la vidéo, mais la notion de temporalité doit aussi entrer dans l'évaluation de la qualité.

Les tests de qualité sont très importants dans les applications de traitement de vidéo et d'image pour tester et optimiser la performance des algorithmes, comme la compression, la restauration, l'impression...

Désormais avec le développement d'internet et des technologies sans fil, on peut accéder aux services multimédias depuis différents types de terminaux (PDA, tablettes, téléphones mobiles etc.) dont la puissance de calcul est souvent limitée par leurs configurations matérielles car certains d'entre eux ne peuvent pas décoder des vidéos ayant une résolution trop haute ou utilisant des techniques de compression trop gourmandes en puissance de calcul [3]. Pendant la transmission, des paquets peuvent aussi être perdus ou arriver en retard lorsque le débit du réseau est trop faible. Il est donc impératif de trouver des métriques permettant d'évaluer la qualité des vidéos tout au long de la transmission, pour que le fournisseur puisse réadapter les flux en fonction des retours² provenant des terminaux [1].

Ces recherches ont abouties à l'étude des techniques de test qui existent dans la littérature permettant de décrire de manière sûre et objective que le test de qualité se rapproche de la perception humaine. Dans un premier temps, il est nécessaire de comprendre comment un humain perçoit une vidéo, quel phénomène physiologique fait dire à un utilisateur que la vidéo est de bonne qualité et comment l'œil humain perçoit les dégradations que subit une vidéo.

La comparaison des différents types de test et des solutions proposées par d'autres équipes de recherche avec leurs avantages et inconvénients sont présentées dans les sections suivantes en abordant les caractéristiques du système visuel humain.

-
2. feedbacks

2.4.1 Le Système Visuel Humain

Afin de comprendre comment peut être évaluée la qualité d'une vidéo, il faut comprendre comment l'œil humain perçoit les signaux et à partir de quel moment un humain considère qu'une vidéo est de mauvaise qualité. Il convient donc d'étudier le SVH³ et de le comparer aux métriques afin de déterminer quel métrique permet de se rapprocher le plus possible de la réalité.

Une modélisation complète du système visuel humain est impossible pour le moment, car certains mécanismes du SVH sont encore en cours de recherche. Il existe plusieurs approches de modélisation, celle la plus connue est la décomposition perceptive de signaux visuels, qui considère le SVH comme un système de filtrage de signaux de luminance. Cette modélisation du SVH comprend principalement trois étapes : la décomposition de signaux de luminance, la fonction de sensibilité de contraste et le masquage de signaux [17].

Décomposition de signaux. Les images que l'on perçoit sont en réalité des signaux de luminance qui sont capturés par les yeux humains. A la suite d'expérimentations physiologiques, il a été découvert que le SVH décompose les signaux en plusieurs canaux selon leurs fréquences. Généralement il existe deux types de décompositions, la décomposition temporelle [12] et la décomposition spatiale [21] [17].

La vision temporelle contient deux ou trois canaux, un ou deux canaux transitoires et un canal soutenu. Le premier est sensible aux objets mobiles et le deuxième est plutôt chargé de la perception d'objets fixes. La vision spatiale peut être divisée en cinq sous-bandes de fréquence différente et huit orientations (figure 1).

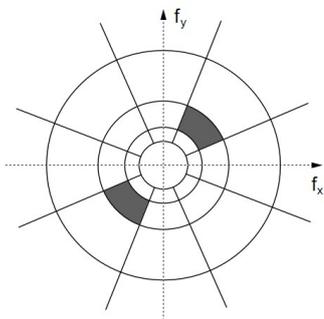


Figure 1: Décomposition spatiale de signaux de luminance

Fonction de sensibilité de contraste. La perception visuelle humaine varie en fonction de la fréquence du signal. C'est-à-dire qu'un signal peut être détecté par l'œil humain lorsque son contraste atteint une certaine valeur. Cette valeur est nommée seuil de perception. La détection du seuil dépend de la fréquence du signal. Pour l'instant, il n'existe aucune représentation mathématique.

Masquage de signaux. La sensibilité de contraste est responsable de la perception d'un seul signal. Quand plusieurs signaux sont présents, et qu'ils se situent dans le même canal de fréquence, il y a des interférences entre eux, et cela modifiera leurs seuils de perception. Ce phénomène est nommé le masquage de signaux. Par exemple, le fait qu'un pixel noir soit placé sur un fond blanc fera qu'il sera plus visible que si il est placé sur un fond bleu foncé car il est masqué par la couleur du fond.

Lorsque l'on considère deux signaux, un signal cible dont le seuil de perception est noté C_{T0} et un autre signal de masque ayant un seuil de C_M , le véritable seuil C_T du signal de cible est donné par la formule ci-dessous donnée dans [17], on peut constater que quand $C_M > C_{T0}$, la valeur de C_T est augmentée, cela signifie que le signal cible devient plus difficile à détecter (Equation 1).

$$C_T = \begin{cases} C_{T0} & \text{Si } C_M < C_{T0} \\ C_{T0} \left(\frac{C_M}{C_{T0}}\right)^\epsilon & \text{Sinon} \end{cases} \quad (1)$$

D'autres approches sont possibles : quel que soit le type de modélisation, ce ne sont que des descriptions approximatives des fonctionnalités du SVH. Ces informations ne sont pas suffisamment complètes pour développer une métrique capable d'évaluer précisément la qualité d'une vidéo sans passer par une phase de test subjectif. C'est la raison pour laquelle les recherche actuelles tentent de simuler le fonctionnement du SVH par des méthodes qui seront présentées dans la section suivante.

2.4.2 Les Dégradations dans les vidéos

Le test de qualité de vidéo est une détection des dégradations qui peuvent influencer la perception visuelle. Avant d'effectuer des tests il faut connaître précisément les différents types de dégradations qui peuvent survenir durant la transmission et la lecture d'une vidéo. Cette partie expose la liste des dégradations possibles et leurs facteurs. Les exemples de dégradations sont illustrés dans la figure 2.

- Pixelisation : Généralement les applications de compression de vidéo ou d'image utilisent l'algorithme DCT⁴ qui peut générer des agglomérations de pixels de la même couleur, ce qui engendre un affichage de carrés dans l'image. Il s'agit d'une compression avec perte qui est effectuée pendant l'encodage de la vidéo, pour réduire la taille des données à envoyer.
- Flou : C'est une dégradation qui engendre une perte de contour ou une diminution du contraste dans les images. Cette dégradation est principalement rencontrée dans les applications de compression pour diminuer la résolution d'une image ou supprimer des mouvements dans une vidéo.
- Bruit : La dégradation de génération de bruit visuel correspond à une variation aléatoire de la luminosité ou des couleurs. Elle se caractérise par l'apparition de pixels de couleurs non attendues car les valeurs RGB⁵ ont été mo-

3. Système Visuel Humain

4. Discrete Cosine Transform

5. Codage des couleurs en Rouge Vert Bleu

difiées ou perdues pendant la transmission ou la compression à cause d'une perte de paquets ou de précision.

- Saccade : Une vidéo est constituée d'une suite d'images successives perçues comme fluide lorsque le décodeur affiche minimum 24 images par seconde⁶. Si certains paquets sont perdus durant la transmission ou arrivent trop tard pour être affichés de manière cohérente à cause d'un problème de connexion réseau, il y aura des images qui ne pourront pas être décodées correctement. Si le nombre d'images perdues est trop important, le spectateur ne verra pas de mouvement pendant quelques secondes (image figée).



Figure 2: Les dégradations d'Image

2.4.3 Difficultés des tests de qualité d'une vidéo

Depuis une dizaine d'années, de nombreuses recherches ont été faites dans le domaine du test de vidéo, mais très peu de progrès ont été accomplis [13]. Le test de qualité de vidéo est un sujet de recherche très vaste mettant en jeu des techniques physiologiques et physiques comme nous l'avons vu précédemment au sujet du SVH. Les algorithmes existant dans la littérature fournissent des résultats corrects dans des cas particuliers d'utilisation en environnement connu [9] [12] [11] [10].

La meilleure façon d'évaluer la qualité d'une vidéo est le test subjectif. Certains résultats montrent une différence d'appréciation très importante selon les personnes testées[8]. Pour représenter mathématiquement la perception visuelle humaine, il faut effectuer une modélisation complète et précise du SVH. Une perception visuelle est en effet une projection, sur le cortex cérébral, des signaux de luminance qui sont capturés par les yeux humains mais dans l'état actuel des recherches, nul ne sait comment ces signaux sont transmis vers le cerveau et comment celui-ci les évalue. Le SVH est un système très complexe, non linéaire [19]. La connaissance du système est très limitée, la recherche dans le domaine de cette science de la vision n'a pas atteint un niveau suffisamment approfondi pour définir une métrique adaptée. Les solutions dans la littérature n'offrent qu'une simulation simple du SVH ne prenant en compte que quelques caractéristiques du SVH c'est pourquoi il faut les utiliser très prudemment, car la perception visuelle est une procédure complexe dans laquelle il faut prendre en considération d'autres

6. Selon le principe de la persistance rétinienne

arguments tels que la lumière ambiante, la distance entre l'écran et l'œil, la taille de l'écran. . . . Un test de performance des métriques réalisés par le VQEG⁷ [13] montre que même des métriques complexes n'apportent aucun avantage par rapport aux métriques qui n'utilisent qu'un modèle mathématique simple.

Beaucoup de métriques ont été proposées dans la littérature mais les recherches en science de la visionne sont pas assez avancées pour permettre de créer des modèles utilisables en informatique. Dans la section suivante sont présentées les métriques et caractéristiques du SVH pouvant être implémentées en informatique.

2.4.4 Les différents types de test

Selon leurs modes de fonctionnement, on peut distinguer deux types de test : le test subjectif et le test objectif.

Test subjectif. Un test est subjectif lorsque la qualité d'une vidéo est évaluée par des humains. La vidéo à est évaluée par un panel de personnes auxquelles il est demandé de noter la qualité sur une échelle de très mauvaise à excellente. A la fin du test, les notes sont récupérées et la moyenne est calculée⁸ [8]. Pour un test subjectif il existe deux méthodes, qui sont SSCQE⁹ et DSCQS¹⁰ [19]. En SSCQE, le spectateur regarde une seule vidéo et note sa qualité, et en DSCQS plusieurs vidéos de différentes qualités sont montrées pendant le test. Dans certains tests subjectifs, on évalue aussi la qualité du son [22]. Dans des vidéos mal compressées, il y a des moments où le son n'est pas synchronisé avec l'image, et cela rend la lecture de vidéo inconfortable.

Le test subjectif reste une solution plus fiable car il renvoie des résultats représentant la qualité perçue d'une vidéo. Mais il est lent et coûteux. En pratique, ce genre de test est utilisé pour valider les méthodes de test objectif.

Test objectif. Pour un test objectif, des algorithmes sont utilisés pour évaluer automatiquement la qualité des vidéos sans avoir à faire passer de test subjectif. L'idée est de développer un système qui peut lire et analyser les informations d'une vidéo dégradée en prenant en compte les caractéristiques principales du SVH pour au final donner une note qui représente la qualité de la vidéo. Il existe trois sous-types de test objectif selon Wang et Bovik [19] :

- Test avec référence complète¹¹ Pendant le test, la vidéo originale et la vidéo dégradée sont disponibles. La vidéo originale est considérée comme une vidéo parfaite, sans dégradation et sert de référentiel pour la comparaison des données entre les deux vidéos. La plupart de métriques existantes sont basées sur cette technique. Cependant, dans la plupart des cas, la vidéo originale n'est pas accessible dans un système distribué, notamment en mode temps réel.
- Test avec référence réduite¹² La vidéo originale n'étant pas disponible, on extrait des informations descriptives de la vidéo dégradée pour les envoyer au serveur vidéo.

7. Video Quality Experts Group

8. Mean Option Score

9. Single Stimulus Continuous Quality Evaluation

10. Double Stimulus Continuous Quality Scale

11. Full-Reference Test

12. Reduced-Reference Test

Celui-ci va alors évaluer la qualité de la vidéo dégradée grâce à ces informations au lieu de faire une comparaison exhaustive entre les deux vidéos. Certains modèles ont déjà été réalisés, par exemple celui de NTIA¹³ [15], initialement conçu pour tester la qualité des vidéos pour la télévision analogique. Ce modèle est capable de collecter les informations de la vidéo reçue durant la transmission et la lecture, et les retourner au serveur via un canal spécial et sécurisé.

- Test sans référence¹⁴ Le test sans référence reste un modèle idéal mais déjà utilisé pour les images. Il n'existe aucune implémentation capable d'évaluer une vidéo sans accès à la vidéo originale. Afin de pouvoir retourner un résultat qui correspond à l'impression visuelle humaine, il faut faire une modélisation complète du SVH, et ce n'est pas encore possible. Le test avec référence complet est souvent appelé test Hors-Service alors que les tests avec référence réduite et sans référence sont considérés comme des tests En-Service .

2.5 Métriques de contrôle de qualité

Il est très difficile de modéliser mathématiquement la perception sensorielle humaine c'est pourquoi il est nécessaire de trouver des métriques implémentables et calculables comme dans [5] et [7] permettant de représenter cette perception le plus fidèlement possible.

2.5.1 Métriques de test avec référence complète

Les métriques avec référence complète sont relativement plus facile à implémenter, car lorsque l'image originale est accessible il suffit de parcourir et de comparer les valeurs de chaque pixel de l'image originale et de l'image dégradée. Généralement ces métriques sont utilisées pour tester la qualité des images. Pour tester la qualité d'une vidéo, qui peut contenir jusqu'à une trentaine d'images par seconde, une comparaison de toutes les images est très lourde. Notamment dans une application distribuée en temps réel où parfois la vidéo originale n'existe même pas (vidéo-conférence). Cependant ces métriques restent très performantes, c'est pourquoi il est possible de s'en inspirer.

Détection d'erreurs (PSNR). Il s'agit d'un test avec référence complète qui est basé sur la détection d'erreurs basé sur le calcul de deux arguments, le MSE¹⁵ et le PSNR¹⁶. Ces deux paramètres sont initialement utilisés en statistique pour estimer la différence entre une valeur biaisée et la valeur originale. Celles ci sont principalement utilisées en traitement d'image ainsi que lors de la compression, pour mesurer la distorsion dans les images numériques.

Deux informations sont utilisées en paramètres d'entrée du programme, l'image dégradée et l'image originale. Cette dernière est considérée comme une image ayant une qualité parfaite, sans dégradation. Ensuite les pixels des deux images sont parcourus et leurs valeurs sont comparées successivement. Un pixel est dégradé lorsque les deux valeurs sont différentes. Les formules de calcul sont montrées ci-dessous.

13. National Telecommunications and Information Administration

14. No-Reference Test

15. Mean Squared Error

16. Peak Signal to Noise Ratio

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (2)$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad (3)$$

où N est le nombre de pixels et L est la valeur maximale d'un pixel. Pour des raisons de simplicité la plupart des tests utilisent des images monochromes, les pixels sont codés sur un seul octet (8 bits), la valeur de L est donc 255. Plus la valeur PSNR est élevée, meilleure est la qualité.

L'avantage de cette métrique est qu'elle est relativement simple à implémenter pour les tests de qualité d'une vidéo. Mais celle-ci qui peut contenir jusqu'à trente images par seconde. L'analyse complète de l'ensemble des images n'est pas possible en temps réel. L'autre inconvénient de PSNR/MSE est que dans certains cas d'images complexes, le résultat ne correspond pas totalement à l'impression visuelle. Dans l'article de wang et Bovik [19], l'image classique de Lenna permet de voir les valeurs MSE 3 pour chaque dégradation effectuée.



Figure 3: évaluation de l'image Lenna avec la détection d'erreur

Utilisation de la loi de Weber (rPSNR). Le PSNR a été beaucoup critiqué dans la littérature, car la comparaison des valeurs des pixels n'est pas suffisante pour évaluer la qualité des images. Certaines dégradations ne sont pas visibles pour l'œil humain et peuvent être ignorées pendant le test. Par exemple, un pixel noir sur un fond bleu foncé. Les chercheurs tunisiens ont introduit une nouvelle métrique [7], nommée PSNR relatif ou rPSNR, qui intègre la loi de Weber dans la détection d'erreurs pour masquer les dégradations moins visibles et rendre le résultat des tests plus proche de la perception visuelle.

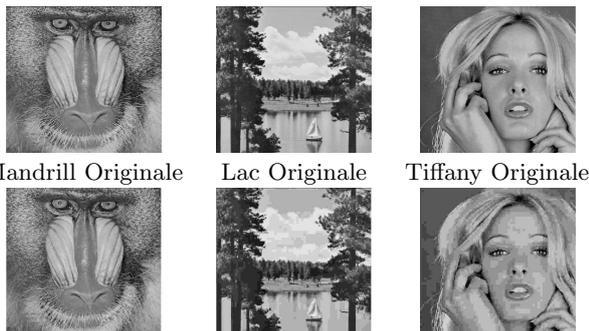
E.H.Weber, physiologiste allemand, avait conclu que la quantité d'excitation nécessaire, pour que l'œil puisse détecter une sensation, est en rapport avec la sensation initiale. Pour qu'une dégradation soit visible, il faut que le changement d'intensité d'excitation soit assez important. Ensuite, le philosophe, Gustave Fechner, a formulé mathématiquement la loi Weber-Fechner selon laquelle la sensation perçue varie proportionnellement au logarithme de l'intensité d'excitation.

Les formules redéfinies de rPSNR et rMSE sont données ci-dessous. La valeur de rPSNR varie de 0 à ∞ : plus la valeur est faible, plus la différence entre les deux images X et Y est importante, et est égale à ∞ lorsque les deux images sont identiques.

$$rMSE = \frac{1}{N} \sum_{i=1}^N \left(2 \times \frac{X_i - Y_i}{X_i + Y_i} \right) \quad (4)$$

$$rPSNR = 10 \log_{10} \frac{L^2}{rMSE} \quad (5)$$

Un exemple est fourni dans la figure 4 pour montrer la performance de rPSNR. Les trois images sont compressées en JPEG. Selon les résultats de PSNR, l'image Lac compressée possède la meilleure qualité, et l'image Mandrill compressée et Tiffany compressée ont presque la même qualité. Ce résultat est erroné car il a été démontré que parmi les trois images compressées, celle de Mandrill possède la meilleure qualité par rapport aux deux autres. Le résultat de rPSNR est donc plus proche de la perception visuelle humaine.



Mandrill Compressée	Lac Compressée	Tiffany Compressée
PSNR = 25.14	PSNR = 25.4	PSNR = 23.89
rPSNR = 65.13	rPSNR = 62.33	rPSNR = 58.11

Figure 4: évaluation de la qualité des images avec le PSNR relatif

Dégradations structurelles. PSNR et rPSNR sont des métriques basées sur la détection d'erreurs dont le principe est de comparer les valeurs des pixels. Ces métriques sont largement utilisées, notamment le PSNR car elles sont très faciles à mettre en œuvre. Elles ne font que détecter des différences de pixels sans chercher à estimer leur influence visuelle. De plus, la perception visuelle ne peut pas toujours être représentée par des différences de valeurs des pixels.

L'équipe de recherche de l'Université de Texas à Austin [18], [20] a proposé une nouvelle métrique basée sur la détection des dégradations structurelles. Au lieu de chercher des pixels

erronés, l'accent est mis sur les dégradations structurelles dans l'image. Une des fonctionnalités principales du SVH est d'extraire les structures de la zone visionnée car les images sont structurées par des contours et des couleurs. La mesure de ces structures dégradées permet de renvoyer des résultats proches de la perception visuelle.

Cette métrique est un test avec référence complète. On suppose que $x = \{x_i | i = 1, 2 \dots N\}$ et $y = \{y_i | i = 1, 2 \dots N\}$ sont respectivement les signaux de l'image originale et de l'image dégradée. L'indice de qualité est défini par la formule suivante :

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \quad (6)$$

d'où

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (7)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad \sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (8)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (9)$$

La valeur de Q est comprise dans l'intervalle $[-1, 1]$, la meilleure valeur est 1, si et seulement si $x_i = y_i$ pour tout $i = 1, 2 \dots N$. La métrique modélise les dégradations comme une combinaison de trois facteurs : la perte de corrélation, la dégradation moyenne et la dégradation de contraste. La formule peut être réécrite comme un produit des trois composants :

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \times \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \times \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (10)$$

Le premier composant représente la corrélation linéaire entre les signaux x et y, le deuxième est la similarité des valeurs moyennes de x et de y, et le troisième est la similarité de contraste entre deux images.

Pour tester la performance de cette métrique, la qualité de l'image Lenna est évaluée et comparée aux résultats de PSNR / MSE (figure 5).

Par rapport aux résultats de MSE, cette métrique renvoie des résultats qui représentent mieux l'impression visuelle humaine. Les auteurs ont conclu que si les images à comparer sont créées depuis la même image originale et qu'elles subissent le même type de dégradation, il suffit d'utiliser PSNR / MSE pour évaluer leur qualité. Si les types de dégradation sont différents, il est plus avantageux d'utiliser cette nouvelle métrique.

2.5.2 Métriques de tests avec référence réduite

Par rapport au test avec référence complète, les tests avec référence réduite sont plus adaptés au test de la qualité des vidéos. Au lieu de comparer exhaustivement toutes les images de la vidéo originale et de la vidéo dégradée, le test avec référence réduite ne mesure que certaines caractéristiques de la vidéo dégradée telles que la résolution, le nombre d'images par seconde¹⁷, le format etc... Ces informations sont extraites de la vidéo dégradée et comparées avec la vidéo originale pour estimer la qualité de la vidéo. Le problème est

17. FPS=Frames Per second

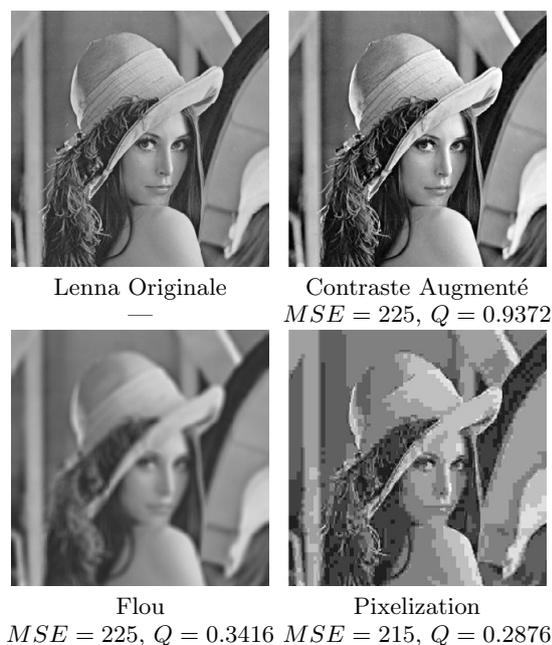


Figure 5: Evaluation de la qualité des images avec la détection de dégradations structurales

que les paramètres choisis ne représentent pas toujours correctement l'impression visuelle humaine.

Reconstruction de vidéo dégradée. Le problème le plus courant pour évaluer la qualité des vidéos dans un système distribué est l'accès à la vidéo originale. Certains auteurs posent le postulat que lorsque la qualité de la n^{e} image dans la vidéo dégradée est testée, il est toujours possible de retrouver la n^{e} image dans la vidéo originale, mais en pratique cela n'est pas toujours le cas. Les flux vidéo qui circulent sur internet utilisent souvent le protocole UDP qui est plus léger, mais beaucoup moins fiable par rapport au protocole TCP. Il y a fréquemment des paquets qui sont perdus pendant la transmission. Une vidéo est constituée d'une suite d'images, si les paquets perdus contiennent des données d'en-tête d'une image et que le décodeur n'arrive pas à la décoder correctement, elle sera ignorée donc perdue.

L'équipe TKN de l'Université Technologique de Berlin a implémenté une solution dans leur serveur de vidéo, nommé EvalVid [11]. Ce serveur est capable de générer, tout au long de la diffusion vidéo, des fichiers de traces, avec lesquels celui-ci peut reconstruire la vidéo dégradée sur le serveur, et estimer la qualité de la vidéo à distance. A la place d'une image perdue dans la vidéo dégradée, EvalVid insère la dernière image affichée. Cette stratégie nous permet de reconstruire une vidéo dégradée ayant le même nombre d'images que la vidéo originale. EvalVid est implémenté en C. Grâce à la puissance du langage, il peut lire et envoyer des flux de vidéo, mais peut aussi gérer la construction des traces pour la reconstruction.

Les fichiers de traces stockent les informations décrivant la

situation de la diffusion de la vidéo :

- Dans le fichier de traces des images, il enregistre les numéros et les types¹⁸ des images envoyées. Il stocke aussi l'estampille¹⁹ de chaque image et le nombre de paquets qui sont créés pour contenir ses données.
- Dans le fichier de traces des paquets, EvalVid enregistre l'estampille, l'identifiant et la taille de chaque paquet qu'il a diffusé.

Le client génère aussi deux fichiers de traces qu'il retourne au serveur au fur et à mesure de la réception de la vidéo. En comparant les fichiers, on peut facilement retrouver tous les paquets et les images qui sont perdus, c'est-à-dire les paquets et les images qui n'existent que dans le fichier de traces du serveur. Les saccades au sein de la vidéo dégradée peuvent ainsi être calculées à l'aide des estampilles. Avec ces informations, le serveur peut reconstruire la vidéo dégradée et évaluer sa qualité avec la métrique PSNR. La vidéo subit toujours le même type de dégradation, donc la métrique PSNR est suffisante pour tester sa qualité.

EvalVid n'utilise pas de nouvelle métrique mais utilise une idée intéressante : L'image à tester n'est pas forcément présente dans la vidéo dégradée. Au lieu d'envoyer des données très volumineuses pour tester la qualité de toutes les images de la vidéo dégradée, la qualité est évaluée avec les traces de la transmission des paquets pour la reconstruction de la vidéo dégradée sur le serveur. Par conséquent, le serveur risque d'être surchargé si les connexions et les demandes de test deviennent trop nombreuses. De plus le test est uniquement effectué par rapport aux paquets qui sont perdus, la distorsion des valeurs n'est pas prise en compte.

2.5.3 Métrique de test sans référence

Le test sans référence est le cas le plus difficile à traiter, car il n'y a pas de référence pour comparer les valeurs. Dans la littérature, certains chercheurs soulignent qu'une comparaison des valeurs des pixels n'est pas vraiment un test de qualité, mais plutôt un test de fidélité, car le SVH n'a pas besoin de la référence pour juger la qualité d'une vidéo. Théoriquement la qualité d'une vidéo devrait pouvoir être testée uniquement avec les informations (contraste, contours, etc.) qui sont directement extraites de la vidéo dégradée. Certaines équipes ont proposé des solutions basées sur les algorithmes de compression, qui consistent à chercher des traces qui sont laissées pendant cette phase [9]. Cependant, elles ne fonctionnent que pour des formats spéciaux car elle ne détectent pas tous les types de dégradation. Dans cette section est présenté l'ensemble de solutions de test sans référence décrit dans la littérature.

Détection de pixelisation. Les formats JPEG et MPEG utilisent le DCT pendant la compression des données pour convertir les signaux de l'image en signaux fréquentiels. Il s'agit d'une compression avec perte, qui supprime des bits ayant une haute fréquence spatiale, car le SVH est moins sensible aux variations de luminance à haute fréquence. Une suppression de 50% de ces bits n'engendrera qu'une perte de 5% des données originales.

La compression commence par une segmentation qui découpe l'image en plusieurs petites zones de taille 8×8 pixels

18. En format MPEG-4, il existe 4 types d'image : I, P, B, S.

19. timestamp en anglais

(figure 6), car celle-ci correspond à la taille maximale que les circuits intégrés pouvaient traiter lorsque l'algorithme a été développé. Ensuite les zones sont transformées par l'opérateur DCT, qui est défini par la formule ci-dessous :

$$f(i, j) = \frac{2}{N} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right] \quad (11)$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{Si } x = 0 \\ 1 & \text{Si } x > 0 \end{cases} \quad (12)$$

où N est la taille de découpage. Après l'opérateur DCT, une matrice de $N \times N$ pixels est convertie en une matrice de $N \times N$ fréquences. Ensuite cette matrice doit encore passer à la quantification pour atténuer et supprimer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. C'est à cette étape que les informations sont perdues.

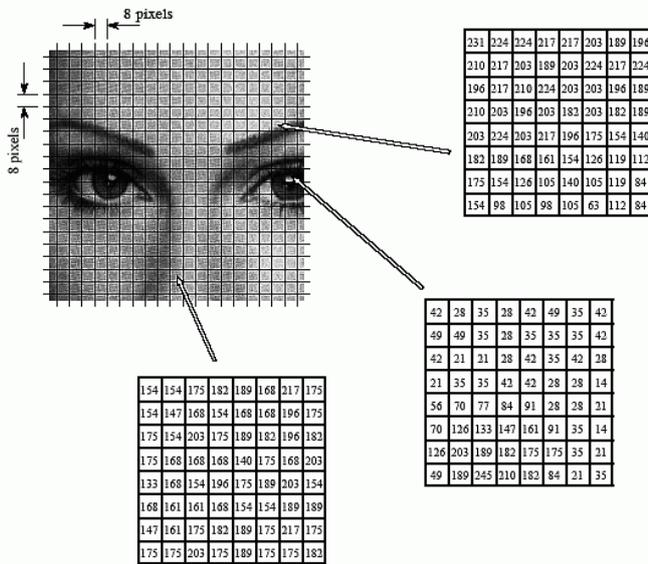


Figure 6: Découpage de l'image en JPEG

Cette stratégie de compression génère des dégradations de pixelisation et de flou dans les images ou les vidéos qui sont compressées avec la transformation DCT (figure 7). Les zones sont traitées indépendamment ce qui provoque une discontinuité entre 2 zones adjacentes engendrant une image moins nette et floue, car les signaux ayant une haute fréquence sont perdus.

L'équipe de l'Université de Texas à Austin a proposé une autre technique de test qui est basée sur la détection des discontinuités des signaux [19], [18]. La technique la plus efficace pour détecter des dégradations de pixelisation et de flou est de convertir les signaux en domaine de fréquence en utilisant la Transformée de Fourier rapide (FFT). Les tests



Image originale

Image compressée

Figure 7: Exemple de la compression JPEG

doivent s'effectuer verticalement et horizontalement. Pour une raison de simplicité, pour une raison de simplicité la taille de l'image est de $M \times M$ où M est un multiple de 8. Les signaux de l'image sont stockés dans un tableau :

$$f = \{f[i, j]; 0 < i, j < M - 1\} \quad (13)$$

Les différences entre les pixels de chaque ligne sont calculées et stockées dans un nouveau tableau :

$$g = \{g[i, j]; 0 < i, j < M - 1\} \quad (14)$$

$$g[i, j] = \begin{cases} 0 & 0 \leq i \leq M - 1, j = 0 \\ |f[i, j] - f[i, j - 1]| & 0 \leq i \leq M - 1, 1 \leq j \leq M - 1 \end{cases} \quad (15)$$

Ensuite ces valeurs sont copiées dans un autre tableau de taille $N \times L$ où N est une puissance de 2, et l'opérateur FFT est appliqué à chaque élément puis les résultats obtenus sont stockés.

$$X = \{X[l]; 0 \leq l \leq N - 1\} \quad (16)$$

et la densité spectrale de puissance de l'image est égale à

$$P = \{P[l]; 0 \leq l \leq \frac{N}{2}\} \quad (17)$$

$$P = \begin{cases} 2|X[l]|^2 & 1 \leq l \leq \frac{N}{2} - 1 \\ |X[l]|^2 & l = 0, \frac{N}{2} \end{cases} \quad (18)$$

L segments sont traités et la densité spectrale de puissance globale est la moyenne de $\{P[l]; 0 \leq l \leq \frac{N}{2}\}$:

$$P[l] = \frac{1}{L} \sum_{k=1}^L P^{(k)}[l], 0 \leq l \leq \frac{N}{2} \quad (19)$$

Pour tester la performance de cette technique, les densités spectrales de puissance de l'image Lenna compressée en

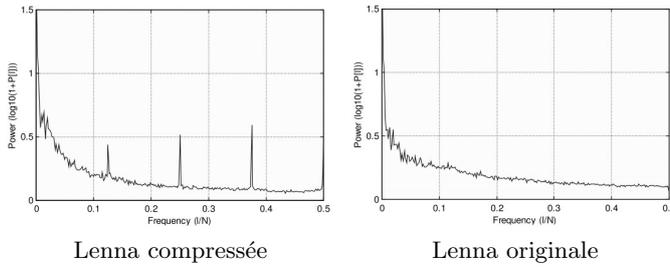


Figure 8: Densités spectrales de puissance de Lenna

JPEG sont calculées et comparées avec celles de l'image originale.

Des pics sont détectés aux positions $\frac{N}{8}$, $\frac{2N}{8}$, $\frac{3N}{8}$ et $\frac{4N}{8}$ qui indiquent les discontinuités entre les zones, et la dégradation de flou est représentée par le changement de valeur des hautes fréquences aux basses fréquences. Maintenant pour évaluer la qualité de l'image, il suffit de trouver un seuil et de compter le nombre de pics dont les valeurs dépassent le seuil de visibilité.

L'équipe de l'Université de Toyama a indiqué que le test avec la transformation FFT est encore très lourd, car il faut parcourir plusieurs fois l'image pour calculer les différences de signaux et les convertir en domaine de fréquence. Ils ont donc repris la même idée avec des calculs simplifiés.

L'image est compressée avec une taille de découpage de 8×8 pixels (il existe aussi des découpages par 16×8 et 16×16 pixels). Pour détecter des dégradations de pixelisation, il suffit de calculer les différences des signaux sur les bordures des zones, entre deux pixels adjacents. La formule ci-dessous montre une détection des discontinuités dans le sens horizontal.

$$B_h = \frac{1}{M(\frac{N}{8} - 1)} \sum_{i=1}^M \sum_{j=1}^{\frac{N}{8}-1} |d_h(i, 8j)| \quad (20)$$

Les signaux de l'image sont stockés dans un tableau $x(m, n)$ pour $m \in [1, M]$ et $n \in [1, N]$, et la différence entre deux signaux peut être calculée de la façon suivante :

$$d_h = x(m, n + 1) - x(m, n), n \in [1, N - 1] \quad (21)$$

La détection du flou dans l'image est plus complexe pour mesurer la perte de l'activité des signaux. Le flou est principalement engendré par la suppression des hautes fréquences, cette détection comprend deux facteurs :

Le premier est la moyenne des différences des signaux dans la même zone (8×8) :

$$A_h = \frac{1}{7} \left[\frac{8}{M(N-1)} \sum_{i=1}^M \sum_{j=1}^{N-1} |d_h(i, j)| - B_h \right] \quad (22)$$

Le deuxième est le taux de passage à zéro. C'est-à-dire le point où la courbe de fréquence change de signe :

$$Z_h = \begin{cases} 1 & \text{si Passage à Zéro} \\ 0 & \text{sinon} \end{cases} \quad (23)$$

$$Z_h = \frac{1}{M(N-2)} \sum_{i=1}^M \sum_{j=1}^{N-2} z_h(i, j) \quad (24)$$

Au final, les résultats des calculs horizontaux et verticaux sont combinés pour obtenir le résultat final S .

$$B = \frac{B_h + B_v}{2}, A = \frac{A_h + A_v}{2}, Z = \frac{Z_h + Z_v}{2} \quad (25)$$

$$S = \alpha + \beta B^{\gamma_1} A^{\gamma_2} Z^{\gamma_3} \quad (26)$$

où α , β , γ_1 , γ_2 et γ_3 sont des paramètres que l'on doit distinguer lors des tests subjectifs.

Par rapport aux autres métriques étudiées précédemment, cette nouvelle solution sans référence est beaucoup plus complexe, car pendant le test, il n'existe pas de référence pour réaliser des comparaisons de valeurs. Cependant grâce aux particularités de l'algorithme de compression de JPEG, des traces sont laissées parmi les signaux de l'image qui permettent d'estimer la qualité de l'image compressée.

L'avantage de cette solution est de ne pas avoir besoin de référence pour estimer la qualité des images. Mais cette solution ne fonctionne que pour les images qui sont compressées avec la transformation DCT, et reste encore lourde pour tester la qualité d'une vidéo.

2.5.4 Amélioration des performances par détection des zones d'intérêts

Généralement le principe des métriques de test avec référence complète est de parcourir l'ensemble des pixels d'une image afin d'extraire et de comparer leurs valeurs, mais ne tient pas compte des cas où les dégradations ne sont pas détectables ou si elles se situent à des endroits qui sont moins intéressants pour l'observateur.

L'œil humain possède un angle de vision de 200×135 degrés, mais l'acuité visuelle maximale se trouve seulement dans la fovéa, qui ne fait que 2 degrés de l'angle de vision. La fovéa est repositionnée rapidement toutes les 100 à 500 millisecondes par le SVH, qui cherche simultanément des endroits nécessitant une observation plus soignée [14]. Les expérimentations physiologiques ont montré que même si le temps d'observation n'est pas limité, l'observateur regarde toujours aux endroits apparaissant les plus importants. Pour que le résultat des tests soit le plus précis possible, les dégradations doivent être détectées en prenant en compte leur importance dans les zones importantes pour l'impression visuelle.

L'équipe de l'Université de Queensland [14] a proposé une technique de test basée sur la détection des zones d'intérêts. Le test est composé de deux parties, la détection des dégradations, et la détection des zones d'intérêts qui permet

de trouver les zones les plus importantes. Avant de lancer l'évaluation de la qualité, les zones d'intérêts sont extraites de cette image.

Le principe est de faire une segmentation qui découpe l'image en plusieurs zones. Ensuite chaque zone obtient une note d'importance en fonction de ses caractéristiques lumineuses et chromatiques. Les résultats sont stockés dans un tableau associant une zone à une note qui représente son niveau d'importance dans l'image. Au final les dégradations sont sélectionnées par les zones d'intérêts. Un ensemble de pixels dégradés dans une zone importante de l'image influencera de manière plus importante l'impression visuelle humaine.

Théoriquement le niveau d'importance d'une zone est déterminé par neuf facteurs :

1. **Contraste** : Une zone ayant un contraste plus élevé que ses voisines attire plus facilement l'attention de l'observateur.
2. **Taille** : Les grandes zones sont plus attractives que les petites.
3. **Forme** : Les zones longues et fines sont plus attractives.
4. **Couleur** : Une zone est plus visible si sa couleur est distinctive par rapport à celles de ses voisines.
5. **Mouvement** : Le SVH est très sensible aux mouvements, une zone est attractive si elle contient des mouvements.
6. **Position** : Les expérimentations ont montré que dans la plupart de cas, l'observateur regarde directement au milieu de l'écran (25% de la largeur).
7. **Premier-plan / Arrière-plan** : L'observateur focalise souvent sur les objets qui se situent au premier plan.
8. **Personne et Contexte** : La recherche de la science de la vision a montré que dans une scène, l'observateur regarde d'abord les personnages, notamment leurs visages, leurs yeux etc. La focalisation peut être aussi affectée par le contexte de la scène. Mais ces deux facteurs ne sont pas modélisés dans le modèle.

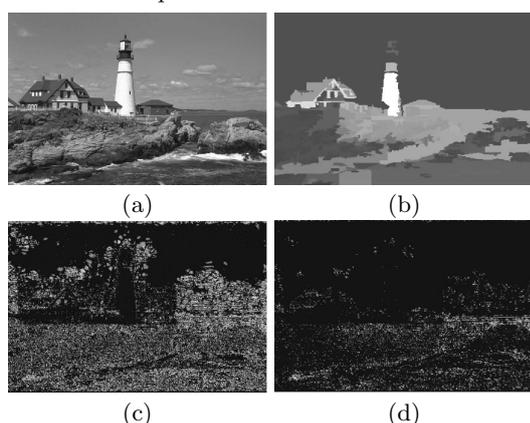


Figure 9: Test basé sur la détection des zones d'intérêts

Les résultats de la détection des zones d'intérêts sont illustrés dans la figure 9 où (a) est l'image originale et (b) est l'ensemble de zones d'intérêts détectées. Plus la couleur est

claire, plus la zone est importante. Des pixels dégradés (pixels blancs) sont détectés dans l'image (c) et ces dégradations sont ensuite sélectionnées par zones d'intérêts. Dans l'image (d), certains pixels dégradés ont été masqués et ignorés par le système (ciel, nuage etc.), sauf ceux qui sont dans les zones importantes (balise lumineuse, roches).

La détection des zones d'intérêts est une technique intéressante qui permet de mettre en évidence les zones les plus attractives dans une image. Le test des performances a aussi montré que cette technique renvoie un résultat assez proche de la perception visuelle. Cette métrique peut être combinée avec d'autres pour essayer de développer une nouvelle solution de test. Le seul problème est que le fait de chercher les zones d'intérêts de chacune des images dans une vidéo engendre un ralentissement de la lecture.

3. IMPLÉMENTATION D'UNE SOLUTION DE TEST PAR RÉFÉRENCE PRÉDÉFINIE

La plupart de métriques qui ont été présentées dans l'état de l'art sont utilisées pour tester la qualité des images car celles-ci ne nécessitent pas de calculs instantanés. Le test de la qualité des vidéos est plus complexe à gérer. Il faut que la métrique soit légère, efficace et facile à mettre en œuvre pour que les calculs ne ralentissent pas la lecture de la vidéo et surchargent le système. Nous avons développé un outil de test, qui consiste à mesurer la qualité des vidéos avec une référence prédéfinie en temps réel. Au lieu de parcourir et d'analyser tous les pixels d'une image pour évaluer sa qualité, nous intégrons une référence dans chaque image de la vidéo, qui sera encodée, diffusée et décodée avec les autres données de la vidéo. Le récepteur de la vidéo possède un exemplaire de la référence non dégradée. Au moment du décodage, la référence dégradée est extraite de chaque image reçue et comparée avec la version locale pour estimer la qualité de toute la vidéo.

3.1 Référence prédéfinie

Lorsqu'une vidéo est dégradée pendant la transmission ou la lecture à cause d'une perte de paquets ou d'une puissance de calcul insuffisante, et que le décodeur ne peut pas décoder les images correctement, il y a deux possibilités : soit l'image dégradée est affichée mais avec des données incomplètes ce qui entraîne des dégradations sur l'image, soit l'image est ignorée et le système passe au décodage de l'image suivante. Si le nombre d'images ignorées est trop important, la vidéo devient saccadée. Une évaluation précise doit prendre en considération toutes les caractéristiques du SVH : extraire les informations spatio-temporelles, calculer le changement de contraste, détecter des zones d'intérêts etc. .La plupart des métriques et techniques présentées dans la section précédente sont utilisées pour le test de la qualité des *images*, qui ne demande pas de calcul instantané. Pendant la lecture d'une vidéo, une image ne dure que quelques millisecondes, temps maximum pour évaluer.

Dans un environnement temps réel (i.e. applications de vidéo conférence, de télémédecine, de e-learning), les flux vidéo ne sont pas stockés, une fois décodés et affichés, ces ressources sont immédiatement libérées. Donc nous ne pouvons pas reconstruire la vidéo dégradée à partir de la vidéo originale. De plus cette solution n'est pas très adaptée car s'il y a trop de connexions, les calculs des tests peuvent surcharger le ser-

veur.

Pour tester la qualité des vidéos, des techniques efficaces et moins gourmandes en ressources doivent être employées. La qualité de chaque image reçue n'est pas testée en intégralité (trop coûteux), mais seulement une petite partie de chaque image permettant d'estimer la qualité globale de toute la vidéo.



Figure 10: Les logos de Covotem™

Avant la diffusion d'une image, un logo de taille 60×60 pixels est intégré. A la réception la partie de l'image à évaluer est extraite et sa qualité est déterminée en calculant le rPSNR. Dans notre outil de test, pour que l'observateur soit conscient des saccades subies par la vidéo pendant la transmission, un logo rotatif de Covotem™ composé de 12 images (figure 10) numérotées de 0 à 11 a été choisi. Avant la diffusion d'une image ces logos sont intégrés de manière itérative, ce qui permet d'avoir un effet de rotation à la réception de la vidéo. D'autres logos peuvent bien entendu être utilisés (i.e. logo de laboratoire, logo de chaîne de télévision etc. . .).

Selon les contraintes du système, les dégradations se répandent aléatoirement dans l'image. Certaines images peuvent contenir le logo qui n'est pas couvert par des dégradations, mais l'objectif de cette technique n'est pas de tester la qualité de chaque image. Si la condition de réseau est très mauvaise, le logo sera forcément dégradé pendant quelques images. Ces dégradations peuvent être immédiatement capturées par notre programme et signalées à l'utilisateur.

Notre outil teste aussi la fluidité en comptant le nombre d'images par seconde (FPS), un facteur important qui peut aussi influencer le confort d'observation et l'impression visuelle de l'observateur.

3.2 Architecture de l'application

L'application est constituée de 3 parties (figure 11) : (1) Le diffuseur de vidéo, qui s'occupe de la capture des flux, de l'insertion du logo et de l'encodage H264, (2) la transmission utilisant le protocole RTP, et (3) le récepteur du flux, qui est chargé du décodage de l'image et de l'extraction du logo pour l'évaluation de sa qualité.

Une source de données peut être un fichier multimédia, une périphérique de capture (i.e. webcam, Camera IP etc.) ou une adresse URI indiquant la localisation des données. Notre outil est conçu pour tester la qualité des vidéos dans un environnement collaboratif où la source de données est souvent une webcam ou une caméra IP.

En mode collaboratif, tous les utilisateurs auront une vision de la vidéo ainsi que l'utilisateur source en monitoring local. Les données qui sont visualisées en local sont encodées et décodées de la même manière que sur les clients distants afin que l'utilisateur puisse avoir conscience de la qualité de sa vidéo. Celle-ci ne subit pas les dégradations dues à l'ache-

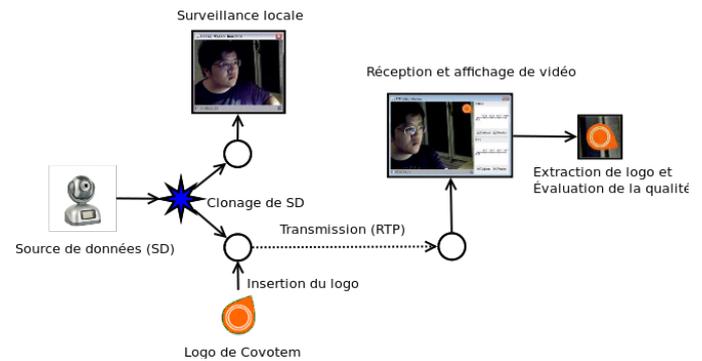


Figure 11: Architecture de l'application

minement des données et sera dans la plupart des cas moins dégradées que la vidéo transmise aux autres utilisateurs.

3.3 Choix techniques

Notre prototype est réalisé avec le langage de programmation java avec la librairie Xuggler permettant d'encoder en temps réel au format H264, permettant ainsi de capturer les flux vidéo depuis une webcam et de les encoder afin qu'ils soient transmis sur un canal de communication.

Xuggler fournit une API de compression/décompression puisant prenant en charge une multitude de paramètres et de profils H264. Il est ainsi très facile de mettre à jour l'encodage avec de nouvelles valeurs selon les résultats des tests et les contraintes des utilisateurs. La difficulté consiste à extraire le bon logo à la réception. Lorsque qu'une image est reçue, l'indice du logo courant n'est pas connu, il est impossible de le déduire de l'image précédente du fait de la probabilité de perte d'image. Cette étape est nécessaire afin de détecter l'indice du logo original correspondant pour évaluer sa qualité (figure 12).

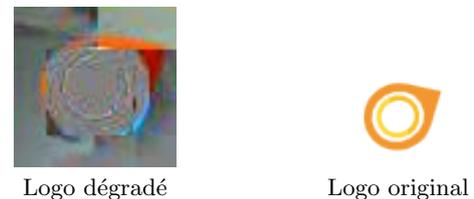


Figure 12: Exemple de logo dégradé et logo original

Pour comparer le logo dégradé correspondant à la version locale du logo, il est nécessaire de détecter la direction du logo. Après avoir été encodées, transmises et décodées, les valeurs RGB des pixels du logo sont modifiées. Afin de trouver le positionnement de la pointe du logo, une technique de détection basée sur le calcul de la similarité des couleurs RGB a été utilisée. Seize zones d'intérêts autour du logo extrait ont été définies puis la similarité entre la couleur de chaque zone et la couleur originale du logo ont été calculées (orange dans notre cas). La similarité entre deux couleurs peut être représentée par leur distance dans l'espace de couleur RGB, dont la formule de calcul est :

$$\Delta RGB = \sqrt{\Delta R^2 + \Delta G^2 + \Delta B^2} \quad (27)$$

où ΔR , ΔG et ΔB sont respectivement les différences des couleurs rouge, verte et bleue. Si la valeur ΔRGB est assez faible, les 2 couleurs sont considérées comme identiques. Par exemple, dans la figure 12, les couleurs détectées dans les zones 4 et 5 sont les plus proches de la couleur originale du logo, cela signifie que le pointe du logo se situe dans les zones 4 et 5 le logo original peut être chargé et comparé au logo dégradé pour évaluer la qualité. Suite aux tests expérimentaux, un seuil de 60.0 permet de détecter correctement la direction du logo.

Les métriques qui existent dans la littérature mesurent généralement plusieurs facteurs et les combinent pour obtenir une valeur finale. Dans notre outil de test, nous testons la netteté et la fluidité séparément. Car lorsque la vidéo est dégradée, il y a deux cas possibles, soit l'image est dégradée mais la vidéo reste fluide, soit l'image est nette, mais la vidéo est saccadée. Si nous voulons retourner ces informations à l'émetteur et faire de l'adaptation des flux, il est préférable de savoir la situation réelle à distance.

3.4 Performances

Pour tester la performance de notre outil, nous utilisons une plateforme collaborative de télémédecine (fig 13) avec plusieurs terminaux. Une dégradation artificielle du réseau est effectuée grâce à un outil de limitation de la bande passante. La connexion devient donc moins rapide et moins stable. L'interface graphique de notre outil contient deux panneaux sur lesquels le programme peut tracer les courbes avec les valeurs de PSNR et de FPS.

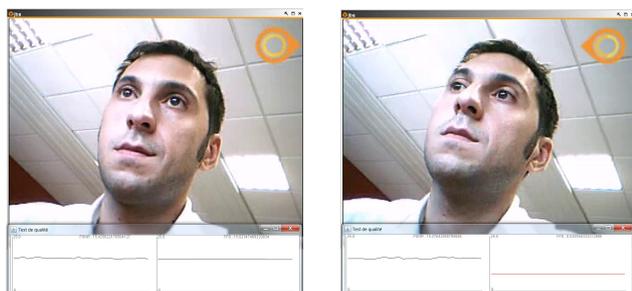


Figure 13: Captures d'écran de l'application

En comparant les deux captures d'écran (figure 13), dans l'image (a), la vidéo est nette et le logo n'est pas dégradé, les deux courbes restent stables ($PSNR = 20.38$, $FPS = 10.50$), dans l'image (b), quand la vidéo est dégradée, les deux courbes montrent une qualité dégradée ($PSNR = 12.51$, $FPS = 2.50$). Le changement des valeurs PSNR et FPS représente bien l'impression visuelle de la vidéo.

Pour garder et étudier les historiques, le programme génère deux fichiers de traces, avec lesquels nous avons tracé deux courbes (figure 14), une représente l'évolution des valeurs PSNR et l'autre FPS. Nous pouvons constater une forte corrélation entre les 2 courbes. Au début de la transmission, lorsque le système n'a pas encore alloué assez de ressources pour visualiser les flux de vidéo, la qualité de la vidéo est très mauvaise ($PSNR \leq 9$). Après de quatre secondes, la qualité devient acceptable ($PSNR \in [18; 20]$). A partir de la 45^{ème} secondes, lorsque le débit de réseau devient instable,

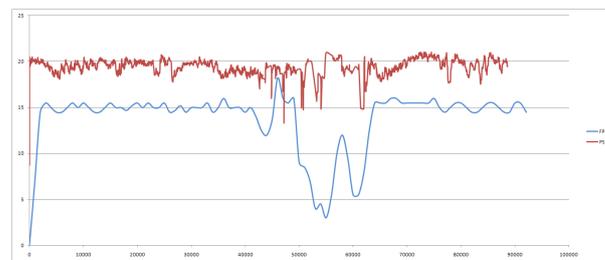


Figure 14: Test des performances

il y a une chute de qualité discernable dans les deux courbes.

4. CONCLUSION ET PERSPECTIVES

Les tests expérimentaux ont montré que notre technique de test est plus avantageuse dans un environnement temps réel, car elle est légère et n'affecte pas la lecture de la vidéo. Grâce au test de qualité d'une référence prédéfinie, il est possible de faire une estimation globale de la qualité de toute la vidéo. Cette estimation ne permet pas d'attester formellement qu'une vidéo est de qualité parfaite pendant toute la durée de la session mais que la perception générale est bonne. Ce test permet surtout de détecter les problèmes qui peuvent subvenir la diffusion de la vidéo et ainsi adapter le flux très rapidement aux contraintes du système.

Par la suite, il est possible d'ajouter une dimension de prise en compte des désirs des utilisateurs et ainsi privilégier un aspect plutôt qu'un autre par exemple la fluidité par rapport à la qualité ou inversement selon les besoins.

5. REFERENCES

- [1] Jean-Baptiste Aupet, Rami Kassab, and Jean-Christophe Lapayre. WAVA : a new web service for automatic video data flow adaptation in heterogeneous collaborative environments. In *CDVE'09 : 6th Int. Conf. on Cooperative Design, Visualization and Engineering*, volume 5738 of *LNCS*, pages 280–288, Luxembourg City, Luxembourg, September 2009.
- [2] Khalid Benali, Grégory Bourguin, Bertrand David, Alain Derycke, and Christine Ferraris. Collaboration / coopération. *Actes des deuxièmes assises nationales du GdR I3*, 2002.
- [3] Cyril Cassagnes, Philippe Roose, Marc Dalmau, and Christine Louberry. Kalimucho : software architecture for limited mobile devices. *ACM SIGBED Review, Special Issue on the - 2nd Workshop on Adaptive and Reconfigurable Embedded System (IST-004527 ARTIST2 Network of Excellence on Embedded Systems Design)*, 6(3), October 2009.
- [4] Marc Dalmau, Philippe Roose, and Sophie Laplace. Context aware adaptable applications - a global approach. *International Journal of Computer Science - IJCSI - Special Issue on Pervasive Computing Systems and Technologies*, 1(1), August 2009.
- [5] Shih-Fu Chang Dept and Shih fu Chang. Optimal video adaptation and skimming using a Utility-Based

- framework. In *Proc. Tyrrhenian Intl Workshop on Digital Communications, Capri Island*, 2002.
- [6] N. Elmarzouqi, E. Garcia, and J.-C. Lapayre. Accm : a new architecture model for cscw. *Procs. of the 10th IEEE and LNCS Int. Conf. on CSCW in Design*, 1 :84–91, April 2007.
- [7] Kacem HLH and Bouhleb MS. Mesure de la qualité des images par l'utilisation de la loi de weber. In *SETIT 2005, 3th Int. Conf. on Sciences of Electronic, Technologies of Information and Telecommunications*, Tunisia, March 2005.
- [8] Geneva ITU. Methodology for the subjective assessment of the quality of television pictures, 2002.
- [9] Chulho Jung, Sanghee Lee, and Eunseok Lee. Dynamic adaptive architecture for self-adaptation in videoconferencing system. *School of Information and Communication Engineering*, pages 36–47, 2005.
- [10] Zakia Kazi-Aoul, Isabelle Demeure, and Jean-Claude Moissinac. Une architecture générique pour la fourniture de services multimédia adaptables - illustration par un scénario. *Mobilité et Ubiquité*, 2004.
- [11] Jirka Klaue, Berthold Rathke, and Adam Wolisz. EvalVid - a framework for video transmission and quality evaluation. In *proc. of the 13th international conference on modelling techniques and tools for computer performance evaluation*, pages 255–272, 2003.
- [12] Hermann Hellwagner Klaus Leopold. Quality controlled temporal video adaptation. In *SPIE proceedings series Congress Internet multimedia management systems IV, Vol. 5242*, pages 163–174, 2003.
- [13] Ligang Lu, Zhou Wang, and Alan C Bovik. Why is image quality assessment so difficult? IN *PROC. IEEE INT. CONF. ACOUST., SPEECH, AND SIGNAL PROCESSING*, 4 :3313–3316, 2002.
- [14] Wilfried Osberger, Neil Bergmann, and Anthony Maeder. An automatic image quality assessment technique incorporating higher level perceptual factors. IN *PROC. IEEE INT. CONF. IMAGE PROC*, 3 :414–418, 1998.
- [15] M. H. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on broadcasting*, 50(3) :322, 312, 2004.
- [16] Franck Tarpin-Bernard. Travail coopératif synchrone assisté par ordinateur : Approche AMF-C = synchronous computer supported cooperative work : AMF-C approach. <http://cat.inist.fr/?aModele=afficheN&cpsidt=185256>, 1997.
- [17] Christian J van den Branden Lambrecht. A working spatio-temporal model of the human visual system for image restoration and quality assessment applications. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1, 1996.
- [18] Zhou Wang, Ligang Lu, and Alan C. Bovik. Image communication : Video quality assessment based on structural distortion measurement. *Signal Processing*, 19(2) :121–132, February 2004.
- [19] Zhou Wang, Hamid R Sheikh, and Alan C Bovik. Bovik, Objective video quality assessment. In *the Handbook of video databases : design and applications*, pages 1041–1078, 2003.
- [20] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multi-Scale structural similarity for image quality assessment. In *proc. IEEE ASILOMAR conf. on signals, systems and computers*, 2 :1398–1402, 2003.
- [21] Stefan Winkler. A perceptual distortion metric for digital color video. IN *PROC. SPIE*, 3644 :175–184, 1999.
- [22] Stefan Winkler, Christof Faller, Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly. Audiovisual quality evaluation of low-bitrate video. In *Human Vision and Electronic Imaging X*, volume 5666, pages 139–148, San Jose, CA, USA, March 2005. SPIE.
- [23] Jürgen Ziegler, Jörg M. Haake, Stephan Lukosch, and Volkmar Pipek. Context-adaptive interaction for collaborative work. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 4461–4464, New York, NY, USA, 2010. ACM.

Technique d'Interaction Tactilo-Tangible en Environnement Virtuel 3D

Louis-Pierre Bergé Emmanuel Dubois Minica Houry-Panchetti Mustapha Mojahid Cédric Sanza

Université Paul Sabatier, IRIT
118 Route de Narbonne
31062 Toulouse Cedex 9, France
+33 5 61 55 67 65

{prenom}.{nom}@irit.fr

RÉSUMÉ

Compte tenu du vieillissement de la population, le maintien à domicile se déploie. Pour contribuer à son développement, il apparaît un besoin de simulation en 3D pour faciliter la conception de ces espaces mais aussi pour présenter les équipements aux patients. Face à ces utilisateurs occasionnels, peu familiers avec des environnements virtuels 3D, des techniques d'interaction ubiquitaires, telles que des TUI (Tangible User Interface), offrent des facilités en termes de manipulation et de prise en main. Toutefois leur conception reste peu structurée. Dans un contexte pluri-métier tel que le maintien à domicile, les considérations de conception à intégrer sont multiples. Nos travaux ont donc conduit à la mise en place d'un processus de conception et son utilisation pour concevoir et implémenter un système interactif mixte de simulation d'un environnement de maintien à domicile.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – *artificial, augmented, and virtual realities*.

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *theory and methods*.

I.3.8 [Computer Graphics]: Applications.

General Terms

Design.

Mots clés

Environnement virtuel 3D, techniques d'interaction, conception et simulation de système de maintien à domicile, interaction tactilo-tangible.

1. INTRODUCTION

L'informatique ambiante à domicile se développe considérablement dans notre société contemporaine. Par exemple pour la sécurité des habitations, l'accès à distance à des services du domicile (domotique), mais aussi dans le domaine de la santé tel que le maintien à domicile. Partant du constat que la population française vieillit [27], l'hospitalisation coûtant cher à la société,

on observe un développement nouveau des services à la personne et des systèmes de maintien à domicile. Ces derniers permettent la prise en charge, chez eux, des patients dépendants grâce à une instrumentalisation du domicile par un réseau de capteurs qui réalise une télésurveillance du patient. Cette télésurveillance peut permettre le déclenchement d'alarmes en cas d'événements ponctuels indésirables (chute, feu, ...) ou bien servir à la collecte et à l'analyse d'informations permettant de suivre l'évolution de l'état du patient dépendant, pour prévoir et prévenir des évolutions défavorables. Le coût de la prise en charge des patients par un système de maintien à domicile est moindre par rapport à celui dans les établissements spécialisés.

L'équipement d'un domicile consiste à ajouter et positionner des capteurs dans l'environnement. Il est ensuite possible d'accéder aux données générées et de mettre en place des scénarios de comportement du système. Il existe des domiciles tout équipés mais ce sont des systèmes ad-hoc, impersonnels, où l'évolution est difficile et impose des coûts élevés. Il apparaît donc nécessaire de réaliser un outil d'aide à la conception et au déploiement des systèmes de maintien à domicile basés sur une approche par simulation 3D. Au-delà de la conception de ces environnements, cet outil doit permettre aux patients ou à leurs accompagnants de découvrir et de s'informer sur l'environnement équipé. Ces utilisateurs sont peu familiers avec des environnements virtuels 3D, c'est pourquoi nos solutions seront basées sur des techniques d'interaction ubiquitaires offrant des facilités en termes de manipulation et de prise en main par les utilisateurs. Ces techniques combinent des artefacts d'interaction physiques et numériques par exemple la réalité virtuelle et augmentée ou les interfaces tangibles (TUI : Tangible User Interface) : de manière générique nous appelons les systèmes utilisant ces techniques les Systèmes Interactifs Mixtes (SIM) [11]. En outre, les SIM offrent de nouveaux espaces à l'interaction puisque l'interaction est simplement basée sur la présence d'artefacts physiques. Selon la définition de Weiser, les SIM sont bien une forme de systèmes interactifs ubiquitaires car les mécanismes d'interaction sont issus d'objet de la vie quotidienne [35].

Après une revue des plateformes interactives pour l'assistance à la personne et des moyens d'interactions avec un environnement 3D, nous présenterons le prototype réalisé en détaillant le processus de conception utilisé et les techniques d'interaction proposées. Nous concluons sur des discussions sur la continuité à donner à ces travaux.

2. ÉTAT DE L'ART

Différentes approches ont été proposées pour constituer des supports à l'assistance à la personne. Nous nous appuyons ci-après sur quelques exemples pour mettre en évidence les diffé-

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'11, June 6–7, 2011, Toulouse, France.

Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

rents apports de ces plateformes. Certaines offrent des environnements 3D, nous analysons ensuite brièvement les moyens d'interaction existants pour manipuler ces environnements 3D.

2.1 Plateforme interactive pour l'assistance à la personne

2.1.1 AccesSim

Le projet AccesSim [1] vise au développement d'un système de Réalité Virtuelle constituant un outil innovant d'aide à la conception d'environnements accessibles. Les objectifs du projet sont multiples. Il doit permettre de former et sensibiliser les architectes, décideurs urbains et le grand public à la circulation en fauteuil roulant, de vérifier l'accessibilité des nouveaux projets urbains et des bâtiments publics, de mettre en situation simulée des utilisateurs dans un fauteuil roulant, d'aider à entraîner sans risque les personnes récemment handicapées à l'utilisation d'un fauteuil roulant et de partager entre les laboratoires un simulateur de fauteuil pour étudier et tester de nouvelles aides techniques aux personnes à mobilité réduite. Les utilisateurs interagissent dans l'environnement virtuel 3D en utilisant notamment un dispositif à retour d'effort traduisant par exemple l'inclinaison correspondant à la situation simulée et un système de tracking des mains et de la tête pour gérer les postures de l'utilisateur.

Ce projet n'offre pas de service, il **propose une simulation** du déplacement dans un environnement virtuel pour se familiariser avec un environnement physique et réel. Le concept est proche de nos études, mais le domaine est différent du maintien à domicile. De plus la sélection et la manipulation des objets et des menus présents dans l'environnement virtuel se réalisent avec une interaction clavier/souris.

2.1.2 Orange Healthcare

Orange Healthcare [20] est un projet proposant des solutions d'e-santé visant à améliorer le suivi des malades et la qualité des soins : le partage d'informations mises en réseau entre plusieurs médecins pour éclairer les diagnostics, le dossier médical électronique pour les patients, une infrastructure d'information et de communication pour le maintien à domicile.

Orange Healthcare est au cœur de la problématique des systèmes informatiques ambiants. Il **offre des services à la personne**, celle-ci est observée mais peu impliquée. Pour le maintien à domicile, ce projet ne propose pas de simulation de fonctionnement avec un environnement 3D par exemple. Pour finir l'interaction utilise des périphériques spécifiques au domaine et non informatique : téléphone, boîtier avec un bouton d'appel ...

2.1.3 Ailisa

Ailisa [19] a pour objectif de mettre en place, dans des environnements contrôlés, des plateformes d'évaluations de technologies de télésurveillance médicale et d'assistance en gériatrie. Une réflexion a lieu pour prendre en compte très tôt la dimension éthique de la prise en charge de la santé par des moyens technologiques. Dans ce projet plusieurs chambres en milieu hospitalier ont été équipées de différents capteurs : détecteurs volumétriques infrarouges, pèse-personne, tensiomètre, oxymètre de pouls, capteurs de chute. Un logiciel permet de surveiller en continu les données délivrées par les capteurs sous formes textuelles ou de courbes.

Ailisa est un exemple de système de maintien à domicile ad-hoc où **le patient est observé**. Il ne fournit pas d'outil de simulation pour prendre connaissance de l'environnement et informer les patients et leurs accompagnants. De plus le logiciel d'accès aux données est orienté pour des experts du domaine et présente une interaction « classique » (clavier/souris).

2.1.4 Bilan

Ces projets mettent en évidence trois catégories de systèmes liés à l'équipement de domicile et relevant de l'informatique ubiquitaire (capteurs) : simulation, offre de service, recueil de données. Seul le premier exemple constitue une aide à la conception mais selon une seule dimension particulière. En effet il permet la conception de l'environnement physique mais pas des équipements que l'on ajoute.

On constate également que les techniques d'interaction sont principalement des techniques conventionnelles : à notre connaissance peu d'efforts spécifiques dans la conception des systèmes ont été réalisés sur l'usage de technique d'interaction plus avancées.

La plateforme que nous proposons retient le principe de la simulation pour affiner le développement de tel système avant le déploiement réel. De plus la simulation facilite les itérations et la possibilité d'impliquer les utilisateurs du système via une prévisualisation. Celle-ci pourra se faire par l'intermédiaire d'un environnement virtuel 3D qui apporte une vue globale de l'environnement physique que l'on veut équiper. Il apparaît nécessaire de dépasser les techniques d'interaction conventionnelles car la manipulation d'environnement 3D avec le clavier et la souris est peu propice à une prise en main efficace par le grand public. Voyons donc brièvement les besoins en interaction dans un environnement virtuel 3D.

2.2 Interaction en Environnement Virtuel 3D

Les techniques d'interaction permettent l'accomplissement de certaines tâches dans les environnements virtuels 3D, elles sont un langage de communication entre l'homme et la machine. D'après Bowman [6] les différents tâches dans les environnements virtuels 3D sont : la navigation, la sélection et la manipulation d'objets, le contrôle d'application, la mise à l'échelle ou encore l'entrée de symboles. Dans notre cas, l'utilisateur de notre simulateur devra pouvoir naviguer dans l'environnement virtuel 3D, sélectionner des objets pour afficher leurs caractéristiques et les manipuler. Nous avons donc choisi de nous focaliser sur les tâches de navigation, de sélection et manipulation, et de contrôle d'application.

2.2.1 Navigation

La tâche de navigation est une activité qui permet à l'image de l'utilisateur (l'avatar) de se déplacer à l'intérieur de l'environnement virtuel 3D. Igarashi [14] propose une classification des techniques d'interaction pour la navigation : les techniques de déplacement basées sur la conduite d'avatar (driving), et sur une désignation de cible puis une téléportation (flying).

Conduite d'avatar

Les techniques de conduite d'avatar permettent à l'utilisateur d'agir sur son avatar de manière continue pour le faire avancer dans la bonne direction. Il existe des techniques physiques basées sur la métaphore de la marche réelle de l'utilisateur et des techniques dites virtuelles.

Au niveau des techniques physiques, il en existe trois différentes. La première consiste à détecter et interpréter une marche réelle de l'utilisateur et à effectuer un mapping en termes de déplacement induits dans l'espace numérique. Ce mapping peut être direct [6] ou inclure une mise à l'échelle [17]. La seconde consiste à détecter des attributs de la marche tels que le lever des pieds [28, 29], les pressions exercées dans les chaussures [16], des rotations [4] ; la marche est simulée et l'utilisateur reste physiquement toujours à la même place. Enfin la troisième solution s'appuie sur des dispositifs tels que des vélos immobilisés [6] ou des tapis roulants [9].

Au niveau des techniques virtuelles on peut citer quatre ensembles de techniques. Dans le premier l'utilisateur doit indiquer la direction qu'il veut suivre en fonction de la direction de son regard, en pointant la direction avec une ou deux mains [18] ou encore en tenant une caméra vidéo virtuelle dans la main [34]. Les instructions de départ ou d'arrêt se font par exemple avec des boutons. Dans le second ensemble, l'utilisateur se sert de son corps pour lancer, arrêter et donner la direction du mouvement [36] ou seulement sa colonne vertébrale et son torse [10] ou encore ses pieds [6] pour une meilleure précision sur la direction de l'avatar. Dans le troisième ensemble de techniques, le déplacement de l'avatar se fait « pas à pas », par exemple en écrivant des lettres avec un stylo sur une tablette tactile [2]. Enfin dans le quatrième ensemble l'utilisateur indique la route que l'avatar doit suivre en traçant un chemin sur le sol de l'environnement virtuel 3D [14] ou en faisant du drag'n drop¹ sur une carte [31] ou encore en indiquant des points de passage [6].

Désignation de cibles et téléportation

Dans les techniques de désignation de cible ou de téléportation dites « flying », l'utilisateur indique une destination cible dans l'environnement ou sur une représentation de l'environnement. L'avatar se déplace alors automatiquement jusqu'à la cible. On trouve des systèmes où l'utilisateur désigne la cible directement dans l'environnement virtuel 3D (cf. 2.2.2), sur une représentation 2D de l'environnement [31], ou avec la voix comme dans le système militaire Quickset [7].

Ainsi pour la navigation, les techniques physiques basées sur la marche réelle sont naturelles, intuitives pour l'utilisateur et procurent un fort sentiment d'immersion dans l'environnement virtuel. Cependant ces techniques sont fatigantes et donc inadaptées à une activité professionnelle quotidienne. Elles nécessitent des dispositifs spécifiques, plus complexes que ceux utilisés pour les techniques virtuelles.

Les techniques de pointage de direction sont les techniques virtuelles les plus communes. Elles apportent un bon niveau de contrôle à l'utilisateur. Lors de la planification de route, le déplacement est automatisé, cela permet à l'utilisateur d'effectuer d'autres tâches durant le voyage.

Enfin la téléportation ne permet pas l'exploration de l'environnement, l'utilisateur s'en sert seulement quand il connaît bien l'environnement. Ajouter une carte 2D du monde offre une vision globale de l'espace virtuel à l'utilisateur et constitue une aide à la familiarisation avec celui-ci.

2.2.2 Sélection et manipulation

La tâche de sélection permet d'identifier un objet particulier. Dans le monde réel, l'utilisateur prendrait un objet avec la main. La tâche de manipulation en trois dimensions se décompose en une tâche de positionnement (changer la position) et de rotation (changer l'orientation). Deux formes principales de techniques d'interaction liée aux tâches de sélection et de manipulation émergent : les techniques égocentriques, avec lesquelles l'utilisateur agit directement depuis l'intérieur de l'environnement virtuel, et les techniques exocentriques, avec lesquelles le monde virtuel est contrôlé depuis l'extérieur. Dans le premier cas, l'utilisateur est totalement immergé dans l'environnement, mais n'en fait pas partie dans le second cas.

Techniques égocentriques

Il existe deux métaphores pour les techniques égocentriques : la main virtuelle et le pointage. Dans la première, une main virtuelle permet de sélectionner et manipuler les objets dans l'environnement virtuel 3D. Le lien avec cette main virtuelle peut être direct avec la main de l'utilisateur [32], utiliser le principe de zone pour allonger le bras de l'utilisateur [5] ou la précision de la manipulation [22]. Dans la seconde métaphore, un rayon virtuel part de l'utilisateur ou de son avatar pour intercepter un objet virtuel qui sera alors sélectionné [Bolt, 1980]. Diverses améliorations ont été apportées pour diminuer les effets du tremblement de la main [13], pour avoir une manipulation plus précise des objets éloignés [30], l'ajout d'un curseur semi-transparent ou d'un cône [12] au bout du rayon pour avoir une surface plus grande de sélection, l'ajout de la voix [33] pour lever l'ambiguïté si plusieurs objets se retrouvent dans le curseur ou encore une projection tangible 2D de l'objet virtuel [24].

Techniques exocentriques

Stoakley [31] propose d'interagir à l'extérieur de l'environnement virtuel 3D à l'aide d'une représentation miniature de la scène virtuelle. Pierce [25], propose la technique des « Voodoo Dolls » pour faciliter la manipulation des objets de petites tailles : l'utilisateur a dans ses mains une maquette de l'environnement proche de l'objet et l'objet lui-même : la maquette sert alors de référence pour positionner et orienter précisément l'objet cible.

Ainsi pour la sélection et la manipulation, il ressort que les techniques basées sur la métaphore de la main virtuelle sont les plus naturelles et l'objet peut être manipulé avec 6 degrés de liberté en un seul mouvement. Mais elles ne permettent le déplacement de l'objet que dans une zone limitée. En limitant le contrôle à deux des six degrés de libertés seulement, les techniques de pointage sont très efficaces pour la sélection d'objets. Par contre ces techniques sont bien moins efficaces et précises pour déplacer les objets.

2.2.3 Contrôle de l'application

Le contrôle d'application est une tâche utilisateur dans laquelle une commande permet de demander au système d'effectuer une fonction particulière, de changer le mode d'interaction ou de changer l'état du système. Des techniques d'interaction différentes sont requises pour afficher les opérations de contrôle sur les objets et afficher des menus.

Les opérations de contrôle sur les objets sont plus précisément le redimensionnement, le changement de couleur ou encore la suppression. Ces opérations se font le plus souvent grâce à des Widgets accrochés aux objets 3D. Mine [18] les place dans la main virtuelle de l'utilisateur. Il a également mis en place une technique pour supprimer les objets. Dans ce cas, l'utilisateur sélectionne un objet et le jette par-dessus son épaule, pour indiquer la suppression de l'objet 3D dans la scène.

Les premiers travaux concernant les menus en environnement 3D ont consisté à adapter les menus utilisés en environnements 2D. Les challenges de cette adaptation sont d'éviter le phénomène de flottement rendant difficile leur sélection et d'éviter les occlusions de parties de la scène 3D [21]. Des solutions basées sur la présence permanente d'un bouton fixe dans le champ de vision [18] constituent une solution.

Le prototype de simulation d'un habitat intelligent présenté dans la section suivante tire profit de cette structuration des techniques d'interaction en environnement virtuel 3D pour proposer des techniques d'interactions ubiquitaires de différentes natures en

¹ Technique de « cliquer glisser » utilisé avec un dispositif de pointage ou sur une surface tactile

fonction des interactions requises sur cet environnement. Nous nous appuyons sur un processus de conception pour présenter les fonctionnalités, les techniques d'interaction et l'implémentation du prototype Home Simu^{3D}.

3. HOME SIMU^{3D}

Home Simu^{3D} est un outil d'aide à la conception, au déploiement et à la découverte de systèmes de maintien à domicile. Dans un exemple d'appartement virtuel 3D, Home Simu^{3D} permet la mise en place de dispositifs (capteurs) servant aux maintiens à domicile de patients. De plus les patients et leurs familles ont la possibilité de prendre connaissance de l'environnement ainsi équipé, d'obtenir des informations relatives aux capacités des capteurs ; l'environnement leur permet donc de se projeter dans le futur environnement et d'être rassuré sur l'efficacité du maintien à domicile.

Les objectifs du système étant définis, nous explicitons dans un premier temps le processus de conception suivi afin d'établir les fonctionnalités du système et les techniques d'interaction proposées. Nous détaillons ensuite les principaux aspects de son implémentation.

3.1 Processus de conception

A notre connaissance, le contexte de notre intervention ne s'inscrit pas dans une activité existante : une classique analyse de l'activité ne peut donc pas initier le processus de développement. La démarche de conception que nous avons adoptée est centrée sur l'utilisateur et itérative [11]. Elle est composée de trois phases principales : recueil d'informations, définition fonctionnelle, et plusieurs cycles itératifs de conception et de développement.

3.1.1 Recueil d'informations

Cette première phase d'analyse a pour but de se familiariser avec le contexte de l'application à concevoir. Dans notre cas, elle se découpe en deux étapes principales.

La première étape a consisté en une analyse de besoins aux moyens d'entretiens menés auprès de différents experts. Ce sont notamment des personnels médicaux (service de médecine physique et réadaptation du CHU de Rangueil) et des acteurs du développement d'habitat intelligent (maison intelligente de Blagnac - MIB) qui ont été impliqués. Cette étape a conduit à l'identification des profils d'utilisateurs de notre système. Nous avons ainsi posé comme hypothèse que les patients ne sont pas des cibles prioritaires du système afin de ne pas réduire les techniques d'interaction aux éventuels handicaps physiques ou moteurs de ces utilisateurs. Au contraire les différents utilisateurs potentiels mis en évidence sont :

- les médecins / infirmiers : ils ont pour rôle de vérifier si les systèmes installés chez leurs patients sont assez performants pour effectuer le suivi visé ;
- les concepteurs de systèmes de maintien à domicile : ils ont pour rôle de mettre en place les capteurs dans l'environnement de simulation 3D. Ce métier n'existe pas à l'heure actuelle, ce serait donc soit une équipe médicale soit des techniciens ;
- les accompagnants et l'entourage du patient : ils ont comme intérêt de découvrir l'environnement, d'obtenir des informations et s'assurer que le suivi est correct ;
- les opérateurs de maintenance : ils ont comme intérêt dans la plateforme de prendre connaissance de l'environnement dans lequel ils vont intervenir afin d'installer ou de réviser les capteurs.

Nous avons pu remarquer que les tâches des utilisateurs se recoupent. En effet, les opérateurs de maintenance et les accompagnants ont tous deux besoin d'explorer l'environnement 3D afin d'en prendre connaissance. De plus le métier de concepteur de systèmes de maintien à domicile n'existe pas aujourd'hui et ce sont les médecins connaissant les besoins et les contraintes de leurs patients qui doivent assurer ce rôle. Par conséquent la conception de l'outil Home Simu^{3D} s'est focalisée sur deux types d'utilisateurs supposés en bonne santé et n'ayant pas d'handicap : *les médecins* (conception et vérification du système de maintien à domicile), *les accompagnants* (découverte de l'environnement, obtention d'informations et assurance du suivi correct).

La seconde étape de la phase de recueil d'informations a eu pour but de cerner le domaine d'intervention en étudiant en particulier une revue des techniques d'interaction avancées en environnement virtuel 3D (cf. 2.2).

3.1.2 Définition fonctionnelle

Cette phase conduit à l'élaboration de la proposition initiale et se découpe en deux étapes distinctes. La première vise à extraire des fonctionnalités génériques, exprimant les besoins fonctionnels que le système doit offrir. La seconde consiste à mettre en place le canevas interactif de l'application : il permet de structurer l'enchaînement entre les différentes tâches et ainsi de faire le lien entre les utilisateurs et la liste des fonctionnalités.

Tâches génériques exprimant les besoins fonctionnels

Les besoins fonctionnels ont été établis en fonction de chaque utilisateur et ont été associés à deux activités génériques indépendantes de la thématique du système : l'identification de ces activités génériques permet d'amplifier la réutilisabilité des solutions de conception élaborées et d'accroître la cohérence entre les solutions d'interaction retenues pour une application donnée. La première activité générique concerne les deux groupes d'utilisateurs cible : « **Explorer et obtenir des informations sur un environnement 3D et les objets qui le composent.** » La seconde concerne plus spécifiquement les médecins : « **Placer des objets dans un environnement 3D et effectuer des réglages précis sur leurs paramètres internes et externes.** ». Néanmoins ces activités génériques ne précisent pas encore les moyens d'interaction requis pour réaliser ces activités. C'est l'objet de la définition de l'étape suivante.

Canevas interactif

Dans un premier temps il convient de définir les **principes fonctionnels** que l'application devra respecter. Un principe fonctionnel est associé à une seule tâche générique. Il permet d'identifier les fonctions et données minimales requises pour rendre interactive cette activité, c'est-à-dire pour permettre la réalisation de cette activité par l'utilisateur. Dans notre cas, les deux principes fonctionnels sont :

- pour explorer l'environnement, le système doit fournir une représentation de l'environnement virtuel 3D et des moyens de déplacement dans les différentes dimensions. Pour prendre des informations sur cet environnement virtuel 3D et les objets qui le composent, le système doit également contenir des moyens de sélection et de manipulation sur les objets, ainsi que des moyens de visualisation des informations pertinentes ;
- pour placer des objets dans un environnement virtuel 3D et effectuer des ajustements précis, le système requiert un catalogue d'objets, un moyen de sélection d'un objet du catalogue et les moyens supports à la manipulation en 6D de

ces objets. De plus la perception de ces objets dans l'environnement virtuel 3D est nécessaire.

Pour compléter les principes fonctionnels, la définition de **principes directeurs** (guidelines) est requise : ils permettent de guider l'adaptation des principes fonctionnels à la thématique du projet en induisant des choix de mise en scène. Ils peuvent par exemple s'appuyer sur des recommandations ergonomiques [1] liées à l'utilisabilité des techniques d'interactions. Dans notre cas, les critères principaux retenus sont la lisibilité, la concision, l'incitation, la flexibilité, le critère de contrôle, le critère d'actions minimales et le feed-back immédiat. Ils seront ensuite utilisés pour les choix de conception et d'analyse des tests utilisateurs. Par ailleurs compte tenu du contexte occasionnel de l'utilisation de cet environnement, de la faible expertise en termes d'interaction avec les environnements virtuels 3D, l'utilisation de techniques d'interactions tangibles a été retenue.

L'étape suivante consiste à écrire des scénarios d'utilisation et des modèles de tâche des actions des utilisateurs. Ils reprennent les besoins utilisateurs, la liste des fonctionnalités et les principes directeurs. Ils ont pour but de clarifier et de valider les attentes des utilisateurs et constituent une base pour les cycles de développement.

Cette phase se termine par la réalisation d'une proposition initiale récapitulant au sein d'un premier artefact de conception l'ensemble des décisions de conception prises dans les étapes précédentes. Compte tenu du processus de conception suivi, des décisions liées à la conception des techniques d'interaction n'ont pas été approfondies et ne sont donc pas incluses dans cette proposition initiale. Seuls des principes directeurs recommandent l'utilisation d'une forme particulière d'interaction : les systèmes tangibles. Leur conception constitue le cœur des itérations menées dans la phase suivante.

3.1.3 Cycle de conception itératif et incrémental

Un cycle de conception est constitué de quatre parties : analyse, conception, prototypage, évaluation. L'utilisateur final du système est impliqué dans chaque étape du cycle. Nous avons réalisé deux cycles de conception pour arriver à un prototypage logiciel abouti qui sera présenté dans les parties suivantes. Le but des cycles est d'itérer sur les techniques d'interaction répondant à l'analyse de notre application.

Brainstormings

Les participants impliqués dans les séances de brainstorming répondaient à différents profils : accompagnants, spécialistes interactions avancées, spécialistes IHM. Deux séances distinctes ont permis d'aborder les problématiques suivantes : comment aller vous interagir avec l'application afin d'évoluer dans cet appartement virtuel 3D ? Quelle séquence d'action allez-vous réaliser pour l'exploration ? Quelles vues souhaiteriez-vous avoir ? Quelle technique d'interaction va-t-on utiliser pour changer de vue et naviguer dans l'appartement ? Quels outils d'aides à l'exploration ? Quelle technique d'interaction pour naviguer et changer de point de vue ? Quelles séquences d'actions ? Quels dispositifs ?

Ces séances de conception participatives ont permis de faire émerger deux formes majeures de solutions concernant la navigation dans l'environnement 3D :

- un mode « avatar » dans lequel l'utilisateur adopte une vue à la première personne dans l'environnement 3D, il déplace une représentation numérique d'un humain ;

- un mode « caméra libre » dans lequel l'utilisateur peut déplacer une caméra dans toutes les directions, sans contrainte, et téléporter celle-ci vers un endroit précis.

Il est également apparu incontournable d'établir un couplage en sortie entre une vue 3D de l'environnement et une vue 2D sous la forme d'une carte. En termes de techniques d'interaction plusieurs solutions sont apparues notamment l'utilisation d'une dalle tactile, de détecteur de mouvement pour que l'avatar suive les mouvements de l'utilisateur (torse, épaules), une poupée pour orienter le regard ...

Suite à l'analyse de ces brainstormings, nous avons retenu certaines solutions suivant des critères de faisabilité dans le temps imparti et les dispositifs disponibles dans notre équipe. Pour mettre en œuvre ces solutions une étape intermédiaire basée sur l'utilisation de prototypes papiers et vidéos a été mise en œuvre.

Prototypage papier / vidéo

L'intérêt de réaliser ce genre de prototypes est qu'ils sont faciles à concevoir, qu'ils demandent moins de temps qu'un prototypage logiciel et que l'on peut facilement itérer et faire évoluer les propositions. Ils sont donc rapides et jetables et respectent les scénarios de conception établis lors du canevas interactif. Néanmoins ces prototypes ne sont pas interactifs, et nécessitent souvent le recours à la technique du magicien d'Oz pour faire évoluer l'environnement (évolution réalisée par un humain et non une machine). Nous avons mis en œuvre cette combinaison de prototypes et techniques de magicien d'Oz pour réaliser des évaluations sur ces prototypes par des utilisateurs finaux. Ces expérimentations ont permis de valider des points complémentaires pour la suite du développement : l'utilisateur doit rester assis car le système doit être utilisé par des médecins qui travailleront toute la journée sur l'application ; le système doit être disposé sur une table afin que l'interaction puisse utiliser des objets tangibles ; la mise en œuvre d'une table tactile et d'un objet tangible pour déplacer l'avatar.

Prototypage logiciel

Le deuxième prototype que nous avons réalisé est un prototype logiciel. Il permet de rajouter toutes les interactions avec l'utilisateur. Un des objectifs de ce prototype est d'obtenir des retours de la part des utilisateurs, lors de séances de tests et d'identifier des erreurs de conception pouvant déclencher une nouvelle itération : il s'agit donc d'un processus évolutif. Suite à ces itérations successives, un prototype final a été réalisé : ses fonctionnalités, les techniques d'interactions proposées et l'implémentation de la solution logicielle sont présentées dans la section suivante.

3.2 Fonctionnalités de Home Simu ^{3D}

Dans cette version du prototype du système Home Simu ^{3D}, deux dispositifs pertinents pour le maintien à domicile ont été intégrés : des capteurs de pressions et des caméras. Les médecins ont la possibilité d'explorer l'appartement virtuel de leurs patients à l'aide d'un avatar et/ou d'une caméra libre. Ils peuvent choisir des dispositifs (capteurs) adaptés aux patients dans un catalogue, les positionner dans l'appartement, les sélectionner, les configurer et régler leur orientation. Ils peuvent visualiser les caractéristiques des dispositifs (descriptions, usages), ajouter des notes et visualiser les notes laissées par les accompagnants.

Les accompagnants disposent quant à eux des mêmes fonctionnalités que les médecins à l'exception de la possibilité de rajouter et de déplacer les capteurs. Ils peuvent aussi laisser des notes aux médecins afin d'en apprendre davantage sur le système.

Nous détaillons dans la section suivante les techniques d'interaction proposées aux utilisateurs et implémentés dans le prototype logiciel. Ces techniques sont regroupées en quatre tâches : exploration de l'environnement 3D interactif, repérage des capteurs équipant le domicile, sélection des capteurs et visualisation de leurs principales caractéristiques, manipulation des capteurs pour les positionner et les orienter.

3.3 Techniques d'interaction proposées

La solution (Figure 1) que nous proposons est constituée :

- d'une zone verticale, écran d'ordinateur ou projection à partir d'un vidéoprojecteur, représentant l'environnement virtuel 3D interactif ;
- d'une table tactile inclinée devant l'utilisateur où celui-ci peut visualiser un plan 2D, vue de dessus, de l'environnement 3D et utiliser des moyens de contrôle de l'application ;
- deux objets tangibles (détecteurs de mouvements), un permettant l'orientation du regard et l'autre servant à la sélection.

3.3.1 Exploration de l'environnement 3D interactif

La tâche d'exploration utilise une interaction mixte basée sur la table tactile et un objet tangible localisé en 6D. Dans les deux modes décrits par la suite, cet objet tangible (Figure 1) est en fait une métaphore de l'orientation du regard de l'objet manipulé, il se présente sous la forme d'une tête d'un personnage.

L'utilisateur dispose de deux points de vue pour explorer l'environnement 3D interactif : le point de vue d'un avatar qu'il déplace selon un mode dit à la 1^{ère} personne, ou le point de vue d'une caméra libre selon un mode dit à la 3^{ème} personne. La position de l'avatar et de la caméra sont représentées sur le plan 2D de l'environnement par des icônes distinctives (Figure 2) : la sélection tactile de l'une ou l'autre de ces icônes active le mode de vue correspondant et met à jour la vue 3D.

Le déplacement de la caméra dans le mode caméra libre s'accomplit sur le plan 2D en effectuant du drag'n drop de l'icône de la caméra libre. Cela permet un déplacement horizontal de la vue. Pour effectuer des mouvements verticaux, un slider (Figure 2) apparaît et permet de modifier l'altitude de la caméra dans l'environnement 3D, provoquant alors un comportement similaire à un zoom/dé zoom.

Le déplacement de l'humain virtuel dans le mode avatar s'effectue à l'aide d'un joystick tactile (Figure 2). Une translation du cercle se situant au centre du joystick, déclenche une translation de l'avatar dans la même direction que celui-ci.

Dans les deux modes, l'orientation des caméras se fait avec une technique d'interaction tangible (métaphore du regard). De plus dans le cas du déplacement en mode avatar, le déplacement suit le regard : par exemple en avançant tout droit avec le joystick tactile alors que le regard est tourné vers la droite, l'avatar se déplacera vers la droite de la scène : la vue en 3D laissera paraître un déplacement en ligne droite conforme à celui appliqué par le joystick tactile.

Si l'utilisateur souhaite se déplacer plus rapidement vers un endroit précis de l'appartement, il est possible de téléporter l'une ou l'autre des caméras en touchant le plan 2D à l'emplacement souhaité.

3.3.2 Repérage des capteurs équipant le domicile

Cette version intègre deux types de dispositifs : des capteurs de pression et des caméras. Ils sont représentés par des icônes sur le plan 2D (Figure 2) et par des formes géométriques différentes dans l'environnement 3D interactifs. Enfin une liste récapitulative

des capteurs du domicile est présente sur le plan 2D : elle est organisée selon la pièce où les capteurs sont installés (Figure 2).

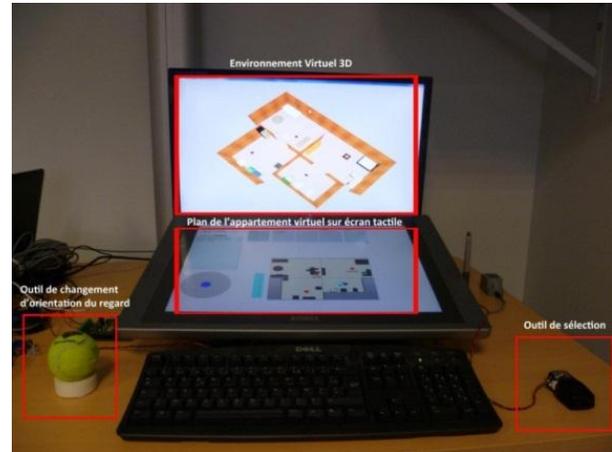


Figure 1 : Prototype logiciel Home Simu^{3D}

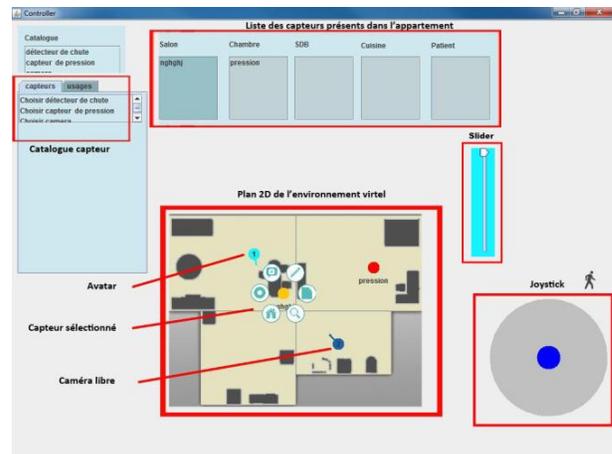


Figure 2 : Prototype logiciel du plan 2D de l'environnement

3.3.3 Sélection des capteurs et visualisation de leurs principales caractéristiques

Une fois que l'utilisateur a repéré le capteur qui l'intéresse, deux options s'offrent à lui pour le sélectionner. Il peut toucher l'icône correspondant au capteur sur la carte 2D affichée sur l'écran tactile (Figure 2) ou bien utiliser le second objet tangible. Cet objet est une métaphore de la main de l'Homme. Une main apparaît alors dans l'environnement 3D. L'utilisateur la manipule pour pointer sur la représentation du capteur dans l'environnement 3D. Pour sélectionner le capteur, l'objet tangible est doté d'un bouton de validation. Dans les deux scénarios de sélection, le capteur est mis en surbrillance sur le plan 2D et dans l'environnement virtuel 3D.

Une fois la sélection réalisée, l'utilisateur peut visualiser diverses informations sur le plan 2D et réaliser des actions. Il les choisit à l'aide d'un pie-menu présent autour du capteur (Figure 3). Pour chaque capteur, il est possible de visualiser sa description, de connaître son usage exprimé de manière textuelle sur le plan 2D, de visualiser les zones d'influences du capteur sur le plan 2D, de l'éditer ou de le supprimer, de téléporter l'avatar ou la caméra libre à côté du capteur et de



Figure 3 : Pie-menu

visualiser le champ de vision du capteur si celui-ci est une caméra. Ces deux dernières actions se réalisent sur le plan 2D et se répercutent sur l'environnement 3D.

3.3.4 Ajout et manipulation des capteurs pour les positionner et les orienter

Un des buts de notre application est d'aider la mise en place virtuelle de systèmes de maintien à domicile. Nous avons développé un composant qui permet de choisir et d'ajouter un nouveau dispositif au domicile. Ce composant est présent sur la carte 2D et permet de choisir un nouveau capteur à ajouter à partir de deux critères : le nom du capteur ou son usage.

Après avoir été ajouté à l'environnement virtuel le capteur doit être positionné et orienté par l'utilisateur. Pour cela l'utilisateur doit sélectionner le capteur et choisir le mode édition sur la carte 2D (cf. 3.3.3). La manipulation sur les capteurs utilise une interaction tactilo-tangible. En effet l'utilisateur positionne le capteur en effectuant du drag'n drop sur l'icône représentant le capteur sur la carte 2D. Pour la troisième dimension, déplacement vertical, l'utilisateur utilise un slider. Enfin pour l'orientation l'utilisateur interagit avec l'objet tangible « visage » représentant le regard d'un personnage. La métaphore d'interaction est identique à celle utilisée pour l'orientation du regard de la caméra : en effet l'orientation du « visage » tangible modifie l'orientation du capteur.

Il est aussi possible de déplacer le capteur directement dans l'environnement virtuel 3D interactif. L'utilisateur se sert de l'objet tangible représentant la main de l'Homme pour la sélection (cf. 3.3.3). Il peut ensuite déplacer le pointeur en forme de main dans une autre position dans l'environnement 3D. En actionnant le bouton du dispositif, le capteur se déplace sur la position de la main virtuelle.

La dernière section décrit brièvement les ressources technologiques utilisées pour l'implémentation de ce prototype logiciel.

3.4 Implémentation

L'architecture logicielle adoptée permet de séparer les techniques d'interactions de l'environnement 3D afin de rendre celui-ci ouvert à de nouvelles techniques. C'est pour cela que l'environnement 3D, la carte 2D et les capteurs d'orientation 6D sont trois blocs logiciels indépendants. Un bus logiciel d'échanges de données (Figure 4) permet la communication entre ces différents éléments. Pour cela nous avons utilisé le bus IVY qui a l'avantage d'être multiplateforme, multi langage et facile à mettre en œuvre.

L'environnement 3D a été codé en C++ avec le moteur 3D open source Irrlicht [15]. L'un des avantages de ce moteur est qu'il permet de décrire l'environnement 3D sous la forme d'un graphe de scène stocké au format XML. Un outil de prototypage 3D est également disponible, Coppercube [8], permettant de générer rapidement un graphe de scène modulable en XML.

La carte 2D a été codée en Java avec la librairie Swing. Les différentes données du système, liste et paramètres des capteurs, notes des utilisateurs, sont stockés sous la forme d'un fichier XML.

Les capteurs d'orientation 6D sont des détecteurs de la marque *Polhemus* appelés Patriot Wireless [26]. Les 2 marqueurs utilisés permettent la détection de mouvement suivant 6 degrés de liberté (orientation et rotation). Un driver écrit en C++ a été développé dans l'équipe : il permet notamment d'envoyer les données des marqueurs sur le bus IVY.

Pour terminer le bouton présent sur l'un des objets tangibles est relié à une matrice Phidgets [23]. La détection d'une action sur le bouton se réalise avec une API en C++.

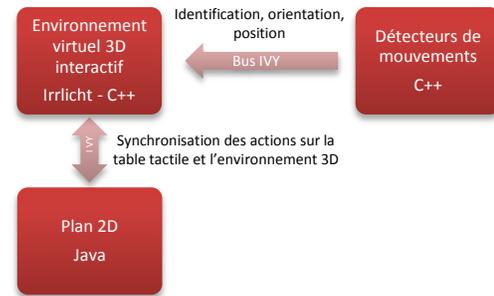


Figure 4 : Architecture logicielle

4. DISCUSSIONS ET CONCLUSIONS

Nous avons présenté dans cet article un environnement 3D interactif pour la simulation d'un habitat ambiant, support au maintien à domicile. L'objectif de cet environnement est principalement d'aider les patients, leurs accompagnants et les équipes médicales à découvrir et comprendre le futur lieu de vie du patient. Ainsi des techniques d'interactions mixtes, une forme de techniques ubiquitaires, ont été implémentées pour permettre la navigation et la manipulation de l'environnement.

Cet environnement et ses techniques d'interaction n'ont pas encore donné lieu à des évaluations utilisateur. Il sera toutefois nécessaire d'étudier l'apport des techniques d'interaction ubiquitaires associées à la vue égocentrique (environnement virtuel 3D) versus exocentrique (carte 3D) dans l'exploration de l'environnement équipé ou encore d'analyser l'utilisabilité de la modalité tactilo-tangible pour la manipulation d'objets dans un environnement virtuel 3D.

Pour donner une suite à nos travaux, d'autres techniques d'interaction seront conçues : la vue 2D pourrait être un plan 2D imprimé couplé à du suivi de main plutôt qu'une table tactile. Cette version serait encore plus ubiquitaire. Des questions se posent également sur la disponibilité des informations : les menus sont exclusivement affichés sur la carte 2D et pas dans l'espace 3D. Est-ce opportun alors que les majorités des interactions ubiquitaires se passent dans l'environnement virtuel 3D ? De même comment visualiser efficacement les notices textuelles et/ou graphiques des capteurs pour qu'elles répondent aux critères de lisibilité, d'accessibilité... ? Quelles alternatives en fonction d'éventuels handicaps des utilisateurs ?

Un autre problème se pose sur l'adéquation des techniques d'interaction 3D. En effet, le joystick tactile combiné à la modification de l'orientation du regard fonctionne uniquement si l'utilisateur regarde l'environnement virtuel 3D. S'il regarde le plan 2D le comportement de l'avatar paraîtra incohérent. Caractériser et étudier le lien entre les caractéristiques des objets virtuels 3D et les caractéristiques des techniques d'interaction paraît alors primordial.

Enfin, le prototype proposé, Home Simu^{3D} permet de concevoir et simuler des habitats intelligents en s'appuyant sur un environnement virtuel 3D interactif. Cette première forme d'abstraction d'un environnement ubiquitaire de maintien à domicile permet de faciliter les itérations de conception nécessaires pour mettre en place différents systèmes de maintien adaptés aux contraintes des patients. Une couche d'abstraction supplémentaire consisterait à décrire de façon abstraite l'environnement virtuel 3D interactif ainsi que les techniques d'interaction utilisées. Cette description dresserait la structure de l'environnement 3D mais ajouterait également des données liées à la sémantique des interactions permises par chaque élément de la scène 3D. Elle faciliterait le couplage de techniques d'interaction ubiquitaires avec un

environnement 3D interactif. Une telle approche contribuera donc à catalyser le choix de techniques d'interactions ubiquitaires pour manipuler des environnements virtuel 3D interactifs et ainsi démocratiser l'usage de ces environnements virtuel 3D pour représenter différents habitats, des environnements propres à des Serious Game, la médiation culturelle en milieu muséal, etc... Par ailleurs, une telle abstraction permettrait de rendre accessible les environnements 3D interactifs à des déficients visuels : elle constituerait une base permettant l'élaboration d'une représentation de l'environnement par des moyens non visuels tels qu'un rendu textuel dont l'architecture serait structurée de manière adéquate. Pour ce profil d'utilisateurs, nous pourrions par exemple faire appel aux travaux sur l'accessibilité aux informations visuelles dans des pages web pour les non-voyants [37].

5. RÉFÉRENCES

1. AccesSim, <http://www.accessim.ceremh.org/>
2. Barrientos F. Canny J., *Cursive: Controlling Expressive Avatar Gesture Using Pen Gesture*, CVE'02, 2002.
3. Bastien Scapin, *Critères ergonomiques pour l'évaluation d'interfaces utilisateurs*, Le Chesnay : INRIA, 1993.
4. Bouguila L. Ishii M. Sato M., *Realizing a New Step-in-place Locomotion Interface for Virtual Environment with Large Display System*, EGVE '02, 2002.
5. Bowman D. Hodges L., *An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments*, I3D '97, 1997
6. Bowman D. Kruijff E. LaViola J. Poupyrev I., *3D User Interfaces: Theory and Practice*, ISBN 978-0-201-75867-2, Addison-Wesley, Boston, 2004.
7. Cohen P. Johnston M. McGee D. Oviatt S. Pittman J. Smith I. Chen L. Clow J., *Quickset: Multimodal Interaction for Distributed Applications*, MULTIMEDIA '97, 1997.
8. CopperCube - a 3D editor for Flash and WebGL, Mac OS X and Windows, <http://www.ambiera.com/coppercube/>
9. Darken R. Cockayne W. Carmein D., *The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds*, UIST '97, 1997.
10. Davies C. Harrison J., *Osmose: Towards Broadening the Aesthetics of Virtual Reality*, Computer Graphics 30(4): 25-28, 1996.
11. Dubois E., *Conception, Implémentation et Evaluation de Systèmes Interactif Mixtes : une Approche basée Modèles et centrée sur l'Interaction*, HDR, Université de Toulouse, 2009.
12. Forsberg A. Herndon K. Zeleznik R., *Aperture Based Selection for Immersive Virtual Environments*, UIST, 1996.
13. Frees S. Kessler G. Kay E., *PRISM Interaction For Enhancing Control in Immersive Virtual Environments*, ACM Transactions on Computer-Human Interaction, Volume 14 Issue 1, May 2007.
14. Igarashi T. Kadobayashi R. Mase K. Tanaka H., *Path Drawing for 3D Walkthrough*, UIST '98, 1998.
15. Irrlicht Engine - A free open source 3d engine, <http://irrlicht.sourceforge.net/>
16. Iwata H. Yano H. Tomioka H., *Powered Shoes*, SIGGRAPH '06, 2006.
17. LaViola J. Acevedo Feliz D. Keefe D. Zeleznik R., *Hands-Free Multi-Scale Navigation in Virtual Environments*, I3D '01, 2001.
18. Mine M. Brooks F. Sequin C., *Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction*, SIGGRAPH '97, 1997.
19. Noury Norbert et al. (au nom de l'équipe du projet Ailisa), *Ailisa plateformes d'évaluations pour des technologies de télésurveillance médicale et d'assistance en gérontologie*, Gérontologie et société, 2005/2 n° 113, p. 97-119.
20. Orange Healthcare, <http://www.orange.com/sante>
21. Ouramdane N. Otmane S. Mallem M., *Interaction en Réalité Virtuelle*, 2009.
22. Ouramdane N. Devesne F., *Follow-Me: a new 3D interaction technique based on virtual guides and granularity of interaction*, VRCIA '06, 2006.
23. Phidgets Inc. - Unique and Easy to Use USB Interfaces, <http://www.phidgets.com/>
24. Pierce J. Forsberg A. Conway M. Hong S. Zeleznik R. Mine M., *Image Plane Interaction Technique In 3D Immersive Environments*, I3D '97, 1997.
25. Pierce J. Steams B. Pausch R., *Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments*, I3D, 1999.
26. Polhemus - PATRIOT Wireless, http://www.polhemus.com/?page=Motion_PATRIOT%20Wireless
27. Robert-Bobée I., *Projections de population 2005-2050 pour la France métropolitaine*, Insee Résultats n°57 société, 2006.
28. Slater M. Usoh M. Steed A., *Steps and Ladders in Virtual Reality*, VRST '94, 1994.
29. Slater M. Steed M. Usoh M., *The Virtual Treadmill: A Naturalistic Metaphor for Navigation in Immersive Virtual Environments*, VE '95, 1995.
30. Song C. Kwak N. Jeong D., *Developing an efficient technique of Selection and Manipulation in Immersive VE*, VRST '00, 2000.
31. Stoakley R. Conway M. Pausch R., *Virtual Reality on a WIM: Interactive Worlds in Miniature*, CHI '95, 1995.
32. Sturman D. Zeltzer D. Pieper S., *Hands-on Interaction with Virtual Environments*, UIST '89, 1989.
33. Tse E. Hancock M. Greenberg S., *Speech-Filtered Bubble Ray: Improving Target Acquisition on Display Walls*, ICMI '07, 2007.
34. Ware C. Osborne S., *Exploration and Virtual Camera Control in Virtual Three Dimensional Environments*, I3D '90, 1990.
35. Weiser M., *The computer for the 21st century*, Scientific American, 3(265): 94-104, 1991.
36. Wells M. Peterson B. Aten J., *The Virtual Motion Controller: A Sufficient-Motion Walking Simulator*, VRAIS '97, 1996.
37. Bou Issa Y. Mojahid M. Oriola B. Vigouroux N., *Analysis and Evaluation of the Accessibility to Visual Information in Web Pages*, ICCHP 2010, Vienna, juillet 2010.

Adaptation des processus de prise de décision au contexte dynamique des ressources en informatique ubiquitaire. Application en eSanté.

Ebrahim Nageba

INSA-Lyon, Université Lyon 1,
UMS3443 SFR Santé Lyon EST,
69677 Bron, France

ebrahim.nageba@insa-lyon.fr

Jocelyne Fayn

INSERM, Université Lyon 1,
UMS3443 SFR Santé Lyon EST,
69677 Bron, France

jocelyne.fayn@insa-lyon.fr

Paul Rubel

INSA-Lyon, Département
Informatique, 69100 Villeurbanne,
France

paul.rubel@insa-lyon.fr

RESUME

L'un des défis à relever dans le domaine des systèmes d'information pervasifs avancés est l'adaptation des processus de prise de décision aux changements dynamiques des conditions de fonctionnement et/ou d'utilisation des ressources, des tâches et des environnements hétérogènes. De plus, le degré de disponibilité et les capacités des services, des dispositifs, des sources de connaissances, etc, qui sont généralement distribués, hétérogènes, et en constante évolution rendent difficiles la découverte et la mobilisation des ressources requises par les tâches utilisateurs. Dans cet article, nous proposons un Framework basé sur des modèles ontologiques en vue de faciliter l'automatisation des processus en informatique ubiquitaire. Nous présentons aussi un exemple d'application dans le domaine de l'eSanté pour démontrer comment le Framework proposé peut rendre plus efficace les processus de prise de décision en accord avec les spécificités des tâches utilisateurs, et en fonction du changement des conditions de mise en œuvre des ressources découvertes en cours d'utilisation, en termes de disponibilité, d'accessibilité et de capacité.

Mots-Clés

Systèmes Pervasifs, Sensibilisation au Contexte, Modélisation des connaissances, Ontologie, Raisonnement à base de règles, eSanté.

ABSTRACT

One of the challenges in the advanced ubiquitous information systems domain is the adapting of decision processes to the dynamically changing conditions of heterogeneous resources, tasks and environments. In addition, the availability and capabilities of the devices, services, knowledge sources, etc, which are usually distributed, constantly changing and heterogeneous, make the discovery and mobilization of the appropriate resources required by user tasks difficult. In this paper, we propose a framework based on ontological models to empower the automation of processes in ubiquitous computing. We present an application example from the eHealth domain to demonstrate how the proposed framework can support the decision making processes according to user tasks specifications, and to discovered resources conditions in terms of availability, accessibility, and capability.

Categories and Subject Descriptors

D.2.1 [Requirements/Specifications]: Methodologies, H.4.2 [Types of Systems]: Decision Support, I.2.6 [Learning]: Knowledge Acquisition, I.2.4 [Knowledge Representation

Formalisms and Methods]: Representations (Procedural and rule-based), Semantic networks, J.3 [Life and Medical Sciences]: Medical Information Systems.

General Terms

Management, Performance, Design.

Keywords

Pervasive Systems, Context-awareness, Knowledge Modeling, Ontology, Rule-based Reasoning, eHealth.

1. INTRODUCTION

Dans un processus donné, les acteurs ayant des attitudes et des compétences différentes, effectuent plusieurs tâches en fonction de leurs profils. En outre, chaque tâche est décrite par ses propres spécifications et peut être exécutée dans des environnements dynamiques, parfois hostiles. Or, de nombreuses ressources hétérogènes, telles que les sources de connaissances, les appareils, dispositifs, services, etc, peuvent être requises par ces tâches. En général, ces ressources et connaissances sont distribuées, possédées ou gérées par des organisations différentes. Dans certains scénarios d'applications, les conditions de ressources, en termes de disponibilité, accessibilité, et compétence, changent en permanence.

Le contexte précité limite la capacité des systèmes basés sur le web de fournir aux utilisateurs des informations de haute qualité, surtout dans des domaines d'application tels que l'eSanté où les processus de la prise de décision, pouvant être d'ailleurs très compliqués, ne sont pas nécessairement exprimables par des algorithmes précis, et peuvent exiger des solutions de compromis intelligentes intégrant à la fois des connaissances sur les objets interrogés, y compris leurs contextes, tout en tenant compte de leur éventuelle imprécision, de leur degré de vraisemblance, et des risques attachés aux objets et aux connaissances a priori sur les possibilités de chacune des ressources disponibles.

Par conséquent, nous avons besoin de compléter les systèmes pervasifs basés sur le web par des modèles de connaissances pour que ces systèmes soient capables d'acquérir des connaissances sur les acteurs, les tâches qu'ils effectuent, les ressources hétérogènes et distribuées requises par les tâches ainsi que sur les organisations qui disposent de ces ressources. Basés sur de tels modèles, des outils peuvent alors être développés afin d'adapter les systèmes pervasifs aux contextes des tâches, des environnements, et des ressources, afin de faciliter les processus

de prise de décision et d'améliorer leur qualité et leur performance.

Dans cet article, nous proposons un Framework basé sur des modèles ontologiques qui permet, d'une façon efficace, d'adapter le processus de prise de décision aux ressources découvertes dans un contexte dynamique et d'améliorer la qualité des informations fournies, quelque soit le type de scénario, simple ou complexe, et la mise en œuvre d'applications basées sur le web. Le Framework proposé est piloté par la tâche utilisateur et est basé sur des ontologies décrivant les différentes entités des environnements pervasifs ainsi que les associations entre ces entités. Il comprend également une base de règles combinant un ensemble de formulations qui spécifient les relations entre chaque tâche et les ressources requises par cette tâche, et un moteur d'inférence pour effectuer un raisonnement à base de règles.

La suite de cet article est organisée comme suit. La section 2 fournit une synthèse bibliographique des travaux connexes. La section 3 fournit une description des principaux composants de l'architecture du Framework proposé. En section 4 on examine, au travers d'un exemple d'application en eSanté, comment le Framework proposé peut être utilisé pour adapter le processus de la prise de décision au contexte dynamique des ressources.

2. Travaux connexes

Par définition, un processus est composé d'un ensemble de tâches pouvant invoquer des services lors de leurs exécutions [3]. Une tâche peut être définie comme un ensemble d'activités, par exemple la recherche de données, l'accès aux données, l'échange de messages, etc, organisées en étapes successives. Les systèmes basés sur le web exécutent de multiples tâches pour fournir à l'utilisateur des informations pertinentes à sa requête [2] [11]. Habituellement, les processus et les tâches sont hétérogènes et leurs descriptions diffèrent d'un domaine à l'autre. En plus, l'exécution des tâches exige de découvrir et de mobiliser diverses ressources. Le challenge de la découverte dynamique de ressources hétérogènes, distribuées, et étant en changement permanent, a été introduit par Liu [14]. Peu d'efforts de recherche ont été faits pour aborder le problème de l'allocation automatisée des ressources et le support du processus décisionnel, en particulier en informatique médicale, sensible au contexte [9] [1]. Un Framework architectural orienté services et basé sur des ontologies ainsi que la technologie d'agent ont été proposés par Sasa et al. [16] pour supporter l'exécution des tâches. En outre, un modèle ontologique qui couvre plusieurs domaines de santé a été présenté par Dang et al. [7] pour capturer les connaissances nécessaires pour la personnalisation des scénarios concernant les soins des patients, les politiques d'assurance, et les prescriptions des médicaments. Par ailleurs, le problème de la mobilité des utilisateurs et le changement de l'environnement d'exécution a été adressé par Kim et al. [12], qui ont introduit une méthodologie pour décrire sémantiquement et découvrir la variété des services dans des environnements différents.

En ce qui concerne la gestion du contexte en informatique ubiquitaire, divers travaux ont été effectués pour modéliser le contexte [5] [6] [17]. Une approche hybride a été proposée par Lee and Meier [13] pour modéliser l'information contextuelle. Elle intègre les avantages de la gestion et la communication des modèles de contexte traditionnels orientés objet, tout en profitant des avantages de la sémantique et de l'inférence des modèles de

contexte à base d'ontologie. Une revue de la sensibilisation au contexte en eSanté a été fournie par la référence [4].

Malheureusement, dans les systèmes d'information avancés, la question de la découverte et de la mobilisation des ressources en termes de disponibilité, accessibilité et compétence n'a pas été convenablement traitée par les travaux cités ci-avant. Aucune méthodologie bien définie n'a été établie pour lier les spécifications des tâches avec les ressources. Dans notre Framework ce problème a été traité en profondeur pour fournir une solution plus générale qui met en relation, d'une part les caractéristiques, les spécificités et les diversités des tâches et, d'autre part, les ressources découvertes.

3. Architecture du Framework

La figure 1 montre une vue de l'ensemble des composants de l'architecture du Framework que nous proposons pour réaliser la gestion et l'exécution des tâches dépendant du contexte et en fonction de la disponibilité et des capacités des ressources. L'architecture est principalement organisée en trois couches : base de connaissances, gestion des tâches, et interfaces.

La base de connaissances est organisée autour d'un méta-modèle de connaissances composé de plusieurs modèles ontologiques génériques et indépendants des domaines, tels que le modèle ontologique d'acteur, de tâche, de ressource, etc, pouvant être étendus à différents domaines comme eGouvernement ou eSanté. Ces modèles ontologiques décrivent des concepts et des relations. Par exemple, Acteur effectue une tâche, une tâche concerne un objet, un objet a un état, une tâche exige des ressources, une organisation gère des ressources, etc. L'objectif de ces modèles ontologiques est de capturer et d'organiser les connaissances sur les différentes entités constituant l'environnement pervasif, comme les tâches, les services, les ressources, les organisations, les acteurs, etc. Une base de règles a été créée pour permettre au moteur d'inférence d'effectuer des raisonnements sur les connaissances stockées dans les ontologies.

Pour formaliser les modèles ontologiques afin qu'ils puissent être interprétés par un ordinateur, OWL-DL (Ontology Web Language- Description Logic) a été utilisé. En effet, l'extension de la logique de description fournit un riche ensemble de primitives, telles que, intersection \wedge , union \vee , complément \neg , etc. Il permet au concepteur de l'ontologie de définir des restrictions et conditions sur les classes des ontologies. Il est donc possible de déterminer automatiquement la hiérarchie de classes et de détecter des inconsistances dans les ontologies.

Le composant clé de notre architecture est le gestionnaire des tâches. Il instancie les modèles ontologiques selon le domaine d'application et intègre un requêteur (moteur de requêtes) SPARQL pour l'interrogation des données, stockées dans les ontologies, et décrites en format RDF.

En outre, le gestionnaire des tâches effectue un filtrage des solutions inférées par le moteur d'inférence en fonction des facteurs contextuels liés, par exemple, à l'endroit où se trouve l'utilisateur, les caractéristiques techniques de l'outil de communication dont il dispose, les possibilités des ressources découvertes, etc.

La couche d'interface comprend différents éléments dont, notamment, une interface homme/machine, des composants pour la découverte des différents services web fournis par les diverses organisations et pouvant être invoqués et exécutés pour récupérer

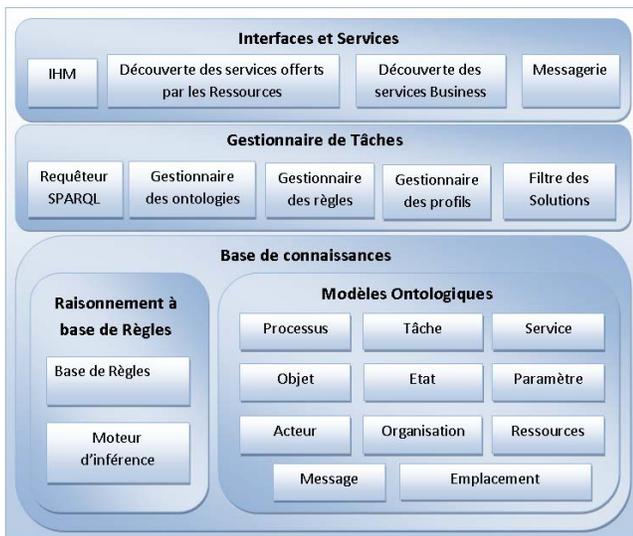


Figure 1. Vue d'ensemble de l'architecture de notre Framework

des données concernant les ressources possédées ou gérées par ces organisations, et une messagerie pour gérer les messages en provenance du gestionnaire des tâches. La couche interface permet d'échanger des données et des connaissances entre le système et les acteurs externes, qu'ils soient des acteurs humains, par exemple des utilisateurs, ou des acteurs logiciels, par exemple un système d'information communiquant. A ce niveau de l'architecture, une politique de transmission de messages a été mise en œuvre pour optimiser l'échange des messages. Cette politique est basée sur des règles de stratification des messages selon leurs niveaux de priorité.

4. Mise en œuvre dans le domaine de l'eSanté

4.1 Exemple d'application

Les scénarios d'application en eSanté sont variés. Certains d'entre eux sont bien connus et basés sur des protocoles et des standards prédéfinis, par exemple le suivi d'un patient à domicile, la téléconsultation, la télé-radiologie, etc. Mais d'autres scénarios d'applicabilité sont contextuels et plus compliqués, par exemple, la prise de décision pour orienter ou transférer un patient vers un hôpital. C'est ce scénario que nous avons adopté pour illustrer ce papier. Il s'agit d'orienter une personne victime d'un malaise vers l'hôpital le plus approprié qui correspond au mieux à ses conditions cliniques et sociales, ou de transférer un patient déjà hospitalisé vers un autre hôpital disposant des ressources matérielles, logistiques et humaines disponibles pour réaliser certains traitements spécifiques pour le patient. En outre, ces scénarios deviennent plus compliqués lorsque les tâches effectuées par les professionnels de santé se déroulent dans des environnements hostiles comme des zones géographiquement isolées et à risque, qui manquent d'infrastructures pour la télé-médecine [15].

Le scénario d'orientation du patient nécessite donc une prise de décision en fonction des facteurs contextuels tels que l'état clinique du patient, ses antécédents médicaux, sa situation administrative, etc, ainsi que la disponibilité et les capacités des ressources hétérogènes nécessaires pour la prise en charge du patient. Par conséquent, le système à base de connaissances doit fournir des solutions pour que le professionnel de santé puisse

prendre la bonne décision en fonction des éléments contextuels précités.

4.2 Instanciation des modèles et description des règles

Selon le domaine de spécialisation, les modèles ontologiques peuvent être automatiquement instanciés. Ensuite, les différents types de tâches abstraites constituant un ensemble de fonctionnalités d'un domaine d'application spécifique, peuvent être définis. Les tâches concrètes seront des instances des tâches abstraites. Le processus d'instanciation est effectué en deux étapes. La première est l'instanciation des modèles ontologiques du Framework pour obtenir les modèles ontologiques du domaine de l'eSanté. La deuxième étape vise à créer des instances des ontologies du domaine de l'eSanté pour traiter le scénario étudié, i.e. l'orientation ou le transfert du patient.

La base de règles a été créée pour composer explicitement différentes propriétés des entités du domaine de l'eSanté. Les règles sont classées en deux groupes principaux. Le premier est le groupe de règles organisationnelles permettant d'allouer des ressources différentes en fonction de la tâche en question. Le second groupe porte sur la politique de transmission des messages. Il s'agit des règles de communication qui permettent de prioriser et stratifier les messages générés par la tâche en question tout en tenant compte du profil utilisateur. Pour décrire les règles, nous avons utilisé le langage SWRL (Semantic Web Rule Language). SWRL est une recommandation du W3C basé sur une combinaison d'OWL-DL et d'OWL-Lite. Le tableau 1 présente un exemple de deux règles décrites en SWRL pour traiter la situation contextuelle du scénario orientation du patient.

Tableau 1. Exemples de règles SWRL en eSanté

Règle	Description
Règle 1: Règle organisationnelle	$T\grave{a}che_T\grave{e}l\grave{e}m\grave{e}d\grave{e}c\grave{e}n\grave{e} (?Tele-Task) \wedge Patient (?P) \wedge Douleur_Poitrine (?DP) \wedge Patient_a_\acute{e}t\grave{a}t (?P,?DP) \wedge T\grave{a}che_C\grave{o}nc\grave{e}r\grave{e}_Patient (?Tele-Task,?P) \wedge Unit\acute{e}_S\grave{o}i\grave{n}_I\grave{n}t\acute{e}n\acute{s}_I\grave{f} (?USI) \rightarrow Requ\acute{e}r\grave{i}r (?Tele-Task,?USI)$
Règle 2: Règle de communication	$Urgentiste (?U) \wedge T\grave{a}che_T\grave{e}l\grave{e}m\grave{e}d\grave{e}c\grave{e}n\grave{e} (?Tele-Task) \wedge Message (?Mes) \wedge Effectuer (?EP,?Tele-Task) \wedge G\acute{e}n\acute{e}r\acute{e}r (?Tele-Task, ?Mes) \rightarrow Niveau_Priorit\acute{e}_Message (?Mes, "High")$

La première règle décrit le contexte suivant. Si la tâche de télé-médecine ($?Tele-Task$) concerne un patient ($?P$), et si ce dernier a comme état clinique une douleur à la poitrine ($?DP$), alors cette tâche exige comme ressource matérielle, par exemple, une place dans une unité de soins intensifs ($?USI$). De la même façon, d'autres règles peuvent être définies pour allouer les différentes ressources requises par la tâche.

La deuxième règle dit : si un urgentiste ($?U$) effectue une tâche de télé-médecine ($?Tele-Task$) et si cette tâche génère des messages ($?Mes$), alors les messages générés par la tâche doivent avoir un niveau de priorité élevé. De cette manière, tous les messages générés par les tâches effectuées par différents professionnels de santé peuvent être priorisés en fonction des profils de ces derniers.

4.3 Algorithme d'exécution des tâches

Le tableau 2 montre l'algorithme de l'exécution des tâches. L'algorithme décrit les instructions à exécuter pour fournir à l'utilisateur les multiples solutions qui correspondent à sa requête et à ses besoins. Les entrées de l'algorithme sont une instance d'acteur, par exemple un professionnel de santé, une instance de tâche, par exemple la tâche d'orientation du patient, une instance de l'objet traité par la tâche, par exemple le patient, et une instance de l'état de l'objet, c'est à dire l'état clinique du patient, par exemple une crise cardiaque, arythmie cardiaque, douleur thoracique, hypertension artérielle, etc, ainsi que des données contextuelles concernant la tâche en question, par exemple les caractéristiques techniques de l'outil de communication dont peut disposer l'utilisateur, l'emplacement ou se trouve ce dernier, des données sur l'environnement. Les sorties de l'algorithme sont une liste des ressources nécessaires pour atteindre les objectifs de la tâche ainsi que la liste des organisations qui disposent de ces ressources disponibles.

Tableau 2. Algorithme d'exécution des tâches

<p>Entrées de l'algorithme: des instances de tâche, acteur, objet, état d'objet, et d'autres instances liées au contexte, emplacement, outil de communication, environnement, etc.</p>
<p>01: Begin 02: For each actor, task, and object's status 03: Create instance 04: Insert instances into the relevant ontologies 05: End for 06: Get actor context 07: Get the context of task concerned object 08: Call reasoning 09: SPARQL Query (select the list of resources required by the running task) 10: task) 11: For each resource instance in the resources list 12: Call service: get resource availability 13: Call service: get resource competence 14: Call service: get resource organization manager 15: 16: End for 17: SPARQL Query (select the list of organizations managing the required resources) 18: For each organization 20: If organization location = location selected by actor 21: Select organization 22: Return a list of organization 23: End for 24: Generate the task messages 25: Send messages to the concerned actors and organizations 26: End</p>
<p>Sorties de l'algorithme: liste des ressources requises pour atteindre l'objectif de la tâche en question, les organisations qui gèrent ces ressources, et des données sur la disponibilité et les capacités de ces ressources.</p>

Durant la phase de raisonnement, le moteur d'inférence applique les règles de la base de règles pour déduire des nouvelles connaissances liées à la tâche en cours d'exécution. Plus précisément, le moteur d'inférence infère les connaissances relatives aux ressources requises par la tâche d'orientation du patient. Les ressources matérielles et logistiques requises peuvent être, à titre d'exemple, la disponibilité d'un lit dans l'unité de soins intensifs, ou tout autre équipement de pointe spécifique nécessaire pour effectuer des traitements spécifiques au patient. Pour récupérer les résultats des raisonnements, c'est-à-dire les

axiomes déduits par le moteur d'inférence, des requêtes SPARQL sont formulées pour interroger la base de connaissances. Les solutions inférées sont classées et filtrées en fonction du contexte de l'utilisateur, c'est-à-dire, en fonction de son emplacement, son environnement, ses conditions, etc.

4.4 Expérimentation

Pour éditer, encoder, et évaluer les ontologies, nous avons utilisé l'environnement de développement Protégé [18] ainsi que le raisonneur intégré Pellet pour tester la consistance de la hiérarchie des classes inférées. Le raisonnement sur les ontologies a été effectué par le moteur d'inférence JESS (Java Expert System Shell) [10] en appliquant un ensemble de règles SWRL. Les règles ont été implémentées en utilisant SWRL-Tab. Ce dernier est une extension pour le plug-in protégé OWL qui permet de gérer des règles décrites en SWRL. Les règles sont interprétées et appliquées par JESS pour inférer des nouvelles connaissances relatives aux concepts décrits dans les ontologies. Concrètement, le raisonnement par JESS est réalisé en trois étapes principales. La première consiste à convertir les axiomes OWL en axiomes JESS et à transformer les règles SWRL en règles JESS. La deuxième étape réalise l'exécution du raisonnement, et l'étape finale consiste à convertir les connaissances inférées par JESS en axiomes décrits en OWL.

Un démonstrateur a été créé pour implémenter la tâche du transfert du patient (figure 2). Une fois que l'utilisateur commence une nouvelle tâche et que les données correspondantes sont enregistrées, plusieurs instances sont créées. La partie supérieure de la figure 2 affiche les données nécessaires pour qualifier le patient, par exemple son nom, prénom, âge, numéro de sécurité sociale, état clinique, etc. En appliquant les règles, plusieurs axiomes liés à la tâche en cours d'exécution peuvent être déduites par le moteur d'inférence. Ces axiomes décrivent les différentes ressources requises pour prendre une décision concernant l'orientation du patient (cf partie inférieure de la figure 2). Des services web peuvent être invoqués et exécutés pour récupérer des données relatives à la disponibilité, aux capacités et à l'accessibilité des ressources.

En fonction des données contextuelles, un processus de filtrage des résultats de la requête SPARQL peut être réalisé en incluant le modificateur FILTER dans la syntaxe de la requête SPARQL pour obtenir seulement les hôpitaux qui sont situés dans la zone géographique choisie par l'utilisateur.

5. Conclusion

Dans cet article, nous avons proposé un nouveau Framework basé sur des modèles ontologiques, un raisonnement à base de règles, et des composants orientés objet de gestion de tâches en informatique ubiquitaire. L'objectif est d'adapter les processus de prise de décision aux contextes dynamiques inhérents aux changements de conditions des ressources hétérogènes requises par les différentes tâches des systèmes pervasifs. Le Framework proposé a été implémenté sous la forme d'un prototype dans le domaine de l'eSanté pour résoudre un problème sociétal : l'orientation des patients victimes d'un incident de santé dans des environnements hostiles.

Le Framework proposé constitue un élément clé pour créer des applications basées sur des connaissances et des services. Il a la capacité d'acquérir des connaissances sur les différents entités considérées (ressources, services, etc.) et sur leurs

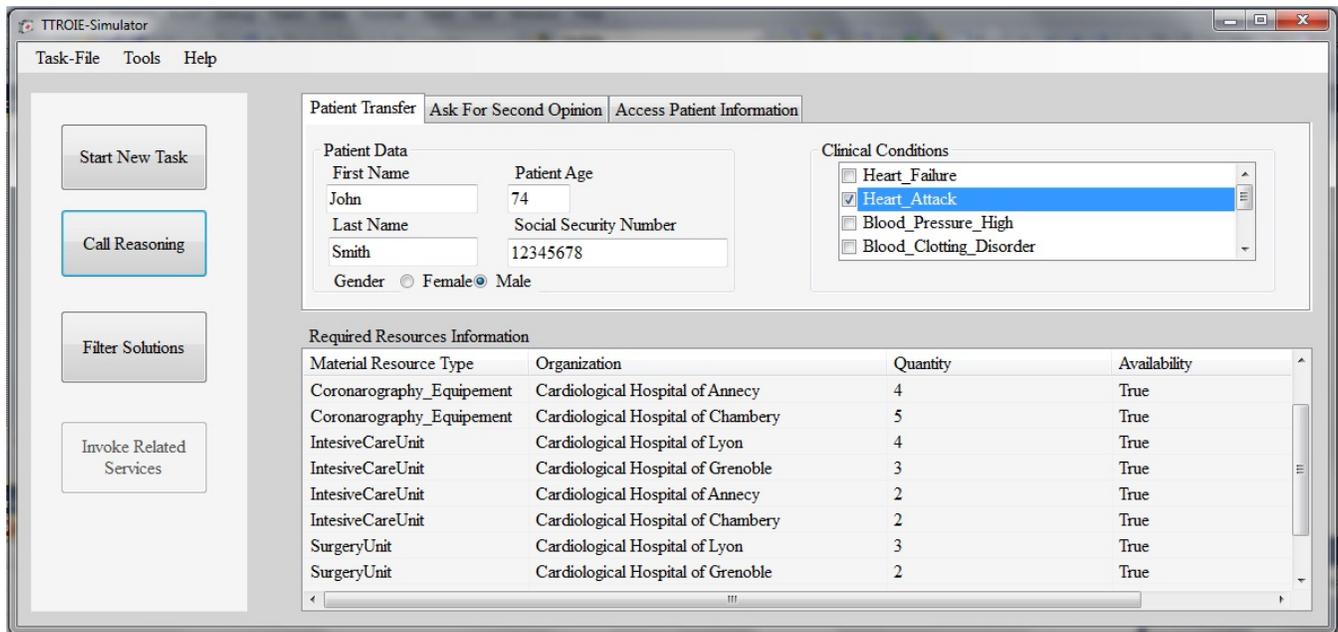


Figure 2. Impression d'écran du démonstrateur de la tâche « Orientation du Patient »

environnements opérationnels. Il fournit une structure de référence qui facilite la représentation, le stockage, et l'échange des données. La particularité de notre Framework réside dans sa capacité à effectuer des raisonnements à base de règles sur les connaissances d'un domaine donné pour fournir une exploitation intelligente des ressources disponibles. Ainsi, il contribue à l'amélioration de la qualité des informations échangées entre les acteurs dans des situations contextuelles hétérogènes et difficiles.

Enfin, notre Framework ouvre de nouvelles perspectives au domaine de l'eSanté dans la continuité du projet EPI-MEDICS [8] réalisé et validé par notre équipe de recherche, où les modèles ontologiques de la télémédecine permettront aux professionnels de santé d'effectuer de multiples tâches lors de scénarios complexes où les processus de prise de décision ne seront pas nécessairement standardisés en raison du contexte dynamique des ressources, de la variabilité de l'état des objets traités par les tâches, des connaissances tacites de l'utilisateur, de son comportement, ainsi que des conditions relatives à l'environnement.

6. REFERENCES

- [1] Anya, O. et al. 2010. Context-aware knowledge modelling for decision support in e-health. *The 2010 International Joint Conference on Neural Networks (IJCNN)* (2010), 1-7.
- [2] Bardram, E. 2005. Activity-based computing: support for mobility and collaboration in ubiquitous computing. *Personal Ubiquitous Comput.* 9, 5 (2005), 312-322.
- [3] BPMN 1.1: 2008. <http://www.omg.org/spec/BPMN/1.1/>.
- [4] Bricon-Souf, N. and Newman, C.R. 2007. Context awareness in health care: a review. *International Journal of Medical Informatics.* 76, 1 (2007), 2-12.
- [5] Chen, H. et al. 2004. SOUPA: standard ontology for ubiquitous and pervasive applications. *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (2004), 258-267.
- [6] Christopoulou, E. and Kameas, A. 2005. GAS ontology: an ontology for collaboration among ubiquitous computing devices. *Int. J. Hum.-Comput. Stud.* 62, 5 (2005), 664-685.
- [7] Dang, J. et al. 2008. An ontological knowledge framework for adaptive medical workflow. *Journal of Biomedical Informatics.* 41, 5 (2008), 829-836.
- [8] Fayn, J. and Rubel, P. 2010. Towards a Personal Health Society in Cardiology. *IEEE Transactions on Information Technology in Biomedicine.* 14, 2 (2010), 401-409.
- [9] Hsieh, F.-S. 2007. Context-aware Workflow Driven Resource Allocation for e-Healthcare. *9th International Conference on e-Health Networking, Application and Services* (2007), 34-39.
- [10] Jess, the Rule Engine for the Java Platform. <http://www.jessrules.com/jess/index.shtml>.
- [11] Jiang, F. et al. 2008. A User-centric Task Computing Architecture for Pervasive Computing. *Third International Conference on Pervasive Computing and Applications* (Alexandria, 2008), 491-469.
- [12] Kim, W. et al. 2006. Activity Policy-Based Service Discovery for Pervasive Computing. *Lecture Notes in Computer Science.* 4254/2006, (2006), 756-768.
- [13] Lee, D. and Meier, R. 2009. A hybrid approach to context modelling in large-scale pervasive computing environments. *Proceedings of the Fourth International ICST Conference on COMMunication System softWare and middlewaRE* (Dublin, Ireland, 2009), 1-12.
- [14] Liu, J. 2005. World Wide Wisdom Web (W4) and Autonomy Oriented Computing (AOC): What, When, and How? 3776/2005, *Lecture Notes in Computer Science* (2005), 157-159.
- [15] Nageba, E. et al. 2011. A Telemedicine System for Hostile Environments. *Advances in Telemedicine: Technologies, Enabling Factors and Scenarios.* InTech. 397-412.

- [16] Sasa, A. et al. 2008. Service-Oriented Framework for Human Task Support and Automation. *IEEE Transactions on Industrial Informatics*. 4, 4 (2008), 292-302.
- [17] Strassner, J. and O'Sullivan, D. 2009. Knowledge management for context-aware, policy-based ubiquitous computing systems. *Proceedings of the 6th international workshop on Managing ubiquitous communications and services* (Barcelona, Spain, 2009), 67-76.
- [18] The Protégé Ontology Editor and Knowledge Acquisition System: 1997. <http://protege.stanford.edu/>.

Mise au point d'un fauteuil roulant manuel intelligent et communiquant pour le suivi et l'assistance de la personne à mobilité réduite : un projet pédagogique d'intelligence ambiante collaboratif, inter-filière et pluridisciplinaire

Nicolas Vuillerme, Quentin Mourcou, Pierre-Yves Gumery,
Sophie Chareyron, David Eon, Didier Pradon*,
Nicolas Palix, Olivier Richard, Skandar Basrou, Didier Donsez
Université Grenoble 1

*Centre d'Investigation Clinique Innovation Technologique APHP R. Poincaré Garches

RESUME

L'émergence et le développement des technologies de l'intelligence ambiante peuvent apporter des solutions non seulement pour conseiller, mais également pour assister ces personnes handicapées. Ce cas d'application de l'intelligence ambiante est particulièrement motivant pour enseigner ces technologies auprès d'élèves ingénieurs issus de plusieurs disciplines. Cet article décrit le déroulement d'un projet pédagogique inter-filière d'intelligence ambiante mené à Polytech'Grenoble pour la réalisation d'un fauteuil roulant manuel instrumenté et communiquant pour le suivi et l'assistance de la personne à mobilité réduite

1. INTRODUCTION

On estime à environ 1,5 millions de personnes « handicapées motrices » en France. Parmi elles, 340 000 utilisent un fauteuil roulant [1]. Les pathologies concernées liées ou non au vieillissement sont nombreuses : myopathies, lésions de la moelle, traumatismes crâniens, accidents vasculaires cérébraux, maladies dégénératives (sclérose en plaques...), infirmité motrice cérébrale, séquelles de poliomyélite... Curieusement, le monde médical s'intéresse peu à l'utilisation de ces fauteuils roulants, qui est pourtant, considérés comme des dispositifs médicaux, font partie de la liste des produits de prestation. Il faut cependant rappeler la loi du 11 Février 2005 pour l'égalité des droits et des chances pour les personnes handicapées, qui souligne le droit à la compensation du handicap. Aussi, des centres référents (Fondation Garches, CERAH à Woippy) ou centres de rééducations (CHU Raymond Poincaré APHP Garches, CHU Henri Gabriel HCL Lyon, CHR Lille, ...) ont mis en place une prise en charge adaptée aux besoins, aux spécificités et aux demandes de l'utilisateur. Cette prise en charge est pour l'essentiel curative lors des soins et préventive lors de la rééducation. Actuellement, l'utilisation de cette aide au déplacement est prise en charge uniquement dans ces différents centres. Aussi, l'utilisateur se trouve la plupart du temps dépourvu de toute assistance, mais aussi de conseils et d'informations au quotidien et hors de ces centres. Il lui est par conséquent difficile d'appliquer les mouvements appris lors de la rééducation, d'adopter des stratégies d'évitement ou de contournement d'obstacles et autres difficultés dans ces déplacements quotidiens.

L'émergence et le développement des technologies de l'intelligence ambiante [2] peuvent apporter des solutions non seulement pour conseiller, mais également pour assister ces personnes dans l'utilisation quotidienne de leur fauteuil roulant. C'est dans ce contexte que le projet AMO vise l'assistance et le suivi de la personne à mobilité réduite se déplaçant à l'aide d'un fauteuil roulant manuel ou à assistance électrique à la propulsion, sur ses lieux de vie (logement, travail, site urbaine, transport en commun) au moyen des technologies de l'informatique ambiante.

La suite de cet article est organisée de la manière suivante. La section 2 présente les motivations du projet AMO. La section 3 décrit les fonctionnalités attendues des composants de l'architecture du système. La section 4 présente le déroulement de la réalisation du projet par les élèves ingénieurs de Polytech'Grenoble. La section 5 conclut par les enseignements tirés de cette première édition du projet pédagogique AMO.

2. MOTIVATIONS

Les motivations du projet AMO décrites dans cet article sont à la fois médicales et pédagogiques.

D'une part, nous souhaitons offrir aux médecins un dispositif innovant, fiable, valide, utile et utilisable visant à l'optimisation de l'usage du fauteuil roulant pour les personnes handicapées qu'ils suivent.

D'autre part, nous souhaitons conduire une expérience pédagogique de co-développement et de prototypage rapide d'un système cyberphysique complexe par des élèves ingénieurs d'année 4 provenant des filières de l'école d'ingénieurs Polytech'Grenoble [6] suivis par une dizaine d'enseignants-chercheurs et des professionnels de santé.

Sur un plan clinique, le projet AMO a pour but de permettre aux personnes atteintes de graves restrictions de mobilité consécutives à une lésion de la moelle épinière, à une maladie neurologique ou musculaire congénitale ou acquise, à un traumatisme des membres inférieurs, ou au vieillissement, de préserver leur autonomie et leur qualité de vie lié au déplacement en fauteuil roulant manuel:

- i. en favorisant le choix et le réglage approprié du fauteuil à leurs caractéristiques, besoins et attentes,

- ii en minimisant le risque d'apparition de troubles musculo-squelettiques,
- iii. en optimisant leur prise en charge fonctionnelle thérapeutique,
- iv en fournissant des informations pertinentes, fiables et valides aux collectivités locales pour la conception, l'aménagement et l'évaluation de l'accessibilité aux personnes à mobilité réduite dans le cadre d'activités quotidiennes (logement, établissements recevant du public, installations ouvertes au public, bâtiments d'habitation et ses annexes, moyens de transports collectifs conventionnels ...) ou des informations correspondantes au déplacement et à l'utilisation d'un lieu public.

3. ARCHITECTURE

Le projet AMO combine à la fois des technologies de l'instrumentation médicale et des technologies de l'information pour les éléments constituant le système : a savoir, l'instrumentation de l'utilisateur, l'instrumentation du fauteuil, le terminal mobile de l'utilisateur et le centre serveur.

La figure 1 schématise l'architecture générale du projet.

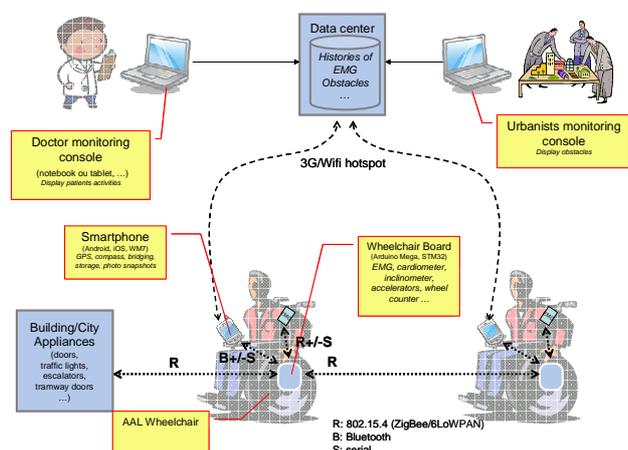


Figure 1 : Architecture du projet AMO

3.1 Instrumentation usager

L'instrumentation de l'usager vise à fournir, de manière synchrone, des informations relatives non seulement à l'utilisateur du fauteuil roulant (données physiologiques essentiellement), mais également au fauteuil roulant lui-même (données cinématiques essentiellement). Plus précisément, la personne est 'instrumentée' afin de monitorer son activité dans le fauteuil. L'effort requis pour effectuer ses déplacements est quantifié au moyen, d'une part, d'un cardio-fréquence-mètre et, d'autre part, de 4 EMG mesurant les contractions musculaires des biceps et triceps de chaque bras. L'utilisation de centrales inertielle permet également de reconstituer la posture des membres supérieurs.

3.2 Instrumentation fauteuil

Le fauteuil roulant est instrumenté dans le but d'assurer le monitoring de contexte d'utilisation du fauteuil et le confort de la personne. L'utilisation d'accéléromètres et de capteurs à effet Hall permet d'obtenir les informations sur la cinématique du fauteuil (distance parcourue, vitesse moyenne, inclinaison...). Le fauteuil est également équipé d'une détection des pieds sur les repose-pieds, d'un capteur de vibration et de plusieurs capteurs de

pression assurant la fonction de détecteur de présence et pouvant éventuellement fournir des informations préventives à la formation d'escarres. La synchronisation des informations physiologiques de la personne instrumentée avec les données collectées sur le fauteuil est réalisée à l'aide d'une liaison sans fil basée sur l'utilisation de modules ZigBee ; l'émetteur situé sur le bras de la personne transmet régulièrement au récepteur, localisé sur le fauteuil, l'ensemble des données physiologiques.

3.3 Terminal mobile usager

L'ensemble des signaux acquis et centralisés sur le fauteuil est collecté par le téléphone 3G/Wifi de l'utilisateur via une liaison Bluetooth. La géolocalisation embarquée dans le téléphone (GPS et/ou cellule GSM) fournit également les positions géographiques pour le suivi spatial de l'utilisateur [3]. D'autres technologies sont envisagées pour sa localisation à l'intérieur de bâtiment [8].

L'appareil photo du téléphone sert à dénoncer des obstacles que rencontre l'utilisateur dans ses trajets. Là où les photos géolocalisées sont prises ; les obstacles sont typés et éventuellement annotés. Les types d'obstacles les plus courants sont prédéfinis pour une annotation rapide. Ce sont par exemple les bâtiments non accessibles, les 'incivilités' (voitures sur le trottoir, sur passage piéton, terrasses de café et extension gênante d'une boutique) ou bien les 'travaux' (trous, tranchées, ...). Ces informations une fois remontées au centre serveur peuvent être partagées avec les autres personnes à mobilité réduite utilisant l'application AMODroid.

Les mesures et les dénonciations sont remontées via des liaisons 3G/Wifi vers un centre serveur qui les archive.

3.4 Centre serveur

Le centre serveur restitue aux médecins ou aux urbanistes, selon l'application voulue du fauteuil roulant manuel intelligent et communicant, les mesures individuelles ou collectives concernant les utilisateurs dans un objectif de support à la prise de décision (par exemple, choix ou réglage du fauteuil, éducation thérapeutique, aménagement des accès handicapés sur les sites).

4. REALISATION

La réalisation de la première itération a été réalisée entre décembre 2010 et avril 2011. Elle a été envisagée comme une itération de spécification et de prototypage rapide.

La maîtrise d'ouvrage de ce projet a été confiée aux 24 élèves de la filière TIS (Technologies d'Information pour la Santé) organisés en 6 groupes. Cette phase de spécification des besoins fonctionnels et non-fonctionnels, réalisée à raison d'une douzaine d'heure de travail par élève et par semaine durant décembre 2010, a abouti à la rédaction d'un cahier des charges par groupe.

Début janvier 2011, 10 élèves des filières 3I (Informatique Industrielle et Instrumentation) et RICM (Réseaux Informatiques et Communication Multimédia) ont été missionnés pour réaliser un dénominateur commun des 6 spécifications figées au moyen de 2 réunions de concertation rassemblant les 36 élèves et les enseignants. La réalisation des deux parties (1) instrumentation de la personne handicapée et (2) instrumentation du fauteuil a été confié à 1 groupe de 6 élèves 3I, tandis que les parties téléphonie et centre serveur ont été confié aux 4 élèves RICM. Chaque groupe avait un représentant faisant la liaison notamment pour la

définition de formats échangés. La réalisation s'est faite jusqu'à mi avril à raison d'une douzaine d'heure de travail par élève et par semaine en parallèle de cours entrant dans leurs spécialisations respectives.

Les choix technologiques ont été fortement conditionnés par la priorité donnée à la production rapide d'un prototype [9] et sur le choix de technologies et de formats ouverts, non propriétaires et prises sur étagère [4]. Le choix s'est ainsi porté sur 2 cartes Arduino Lilypad équipées de modules Xbee pour les microcontrôleurs des 2 brassards EMG instrumentant l'utilisateur. La centrale de mesure attachée au fauteuil est une carte Arduino Mega équipée d'un accéléromètre, de 2 détecteurs de pression pour les repose-pieds, de capteurs à effet Hall sur les roues, d'un module XBee pour la communication avec les 2 brassards et d'un module Bluetooth pour la communication avec le téléphone. Le téléphone portable de l'utilisateur est un téléphone Google Android d'entrée de gamme pour limiter le coût global du système. Le centre serveur est une application Google App Engine (GAE) hébergée sur le nuage IaaS de Google. L'hébergement IaaS sera remis en cause lors du passage en phase d'exploitation du projet puisque les données stockées sont confidentielles du fait de leurs caractères médicaux. L'IHM RIA s'appuie sur le canevas Google Web Toolkit (GWT) ainsi que des extensions. Elle est destinée aux utilisateurs du fauteuil, aux médecins et aux urbanistes.

Coté échanges, la liaison entre le téléphone Android est Bluetooth ou USBSerial. La liaison entre le téléphone Android et le centre serveur GAE est basé sur HTTP/SOAP sur des hotspots Wifi publics et 3G. La remonté de données par HTTP/SOAP tolère les interruptions de connectivité de réseau. Aucun échange n'est sécurisé.

Fin avril, à la recette de cette première itération [7], les algorithmes de traitement des données capteurs (Figure 5) n'ont été que partiellement portés sur les cartes Arduino Lilypad. La jauge de déformation est fonctionnelle et permet de détecter la présence. Le calcul de la vitesse de déplacement du fauteuil, grâce aux capteurs à effet Hall situés sur la roue, est implémenté. Les signaux bruts de l'accéléromètre, branché directement sur l'Arduino Lilypad, sont filtrés et ceux de l'EMG sont traités afin d'être détectés sous forme d'enveloppe. Les données de l'accéléromètre couplées à celles des capteurs EMG permettent ainsi de détecter l'action de poussée ou de freinage du fauteuil (Figure 2). Le calcul du taux de co-contraction des muscles biceps et triceps n'est pas encore effectif. Bien que fonctionnelle, la liaison Bluetooth n'a pas permis de remonter les mesures des capteurs de la centrale d'acquisition vers le téléphone Android.

L'application AMODroid est cependant fonctionnelle. Elle présente les informations synthétiques et pertinentes pour l'utilisateur du fauteuil roulant. Ces dernières sont les indicateurs de la valeur actuelle des EMG du bras gauche avec un code couleur pour les co-contractions, la vitesse instantanée du fauteuil roulant, la position GPS du fauteuil roulant, l'inclinaison courante du fauteuil roulant et la présence actuelle des pieds sur le support (Figure 3, gauche). L'application stocke les données acquises dans la carte Flash du téléphone et remonte régulièrement les signaux capteurs mesurés ou simulés auprès du centre serveur. L'application permet également de prendre des photos et de saisir des commentaires brefs dénonçant des obstacles rencontrés par l'utilisateur du fauteuil roulant [8] (Figure 3, droite).

L'application RIA, qui a été développée avec le canevas Google Web Toolkit (GWT), offre aux médecins et aux urbanistes une première interface fonctionnelle et conviviale d'exploration des mesures et des dénonciations remontées par les téléphones AMODroid (graphes synthétiques et emplacement d'obstacles sur Google Maps).

Une validation du prototype de l'équipement est en cours d'évaluation au sein du CHU Raymond Poincaré APHP Garches, et de la Fondation Garches. Les résultats de cette première évaluation devraient permettre de juger de l'intérêt, la pertinence, la fiabilité et la validité des données collectées, ainsi que de l'ergonomie des IHM développées. Ces résultats devraient en outre favoriser la continuation de ce projet de septembre 2011 à décembre 2011 par la participation à un concours étudiant doté de récompenses.



Figure 2 : Exemple du fauteuil roulant instrumenté.



Figure 3 : IHM de l'application AMODroid (version 1)

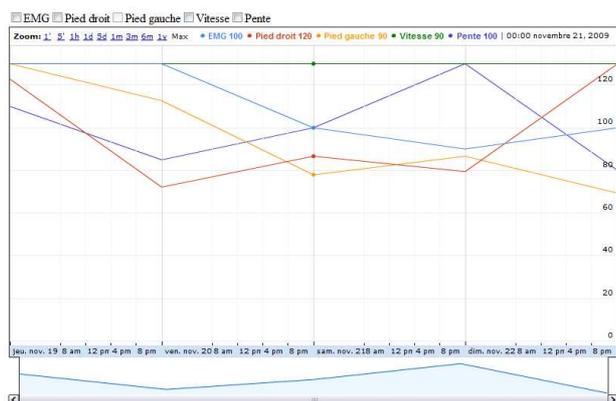


Figure 4 : IHM destinée aux médecins (version 1)

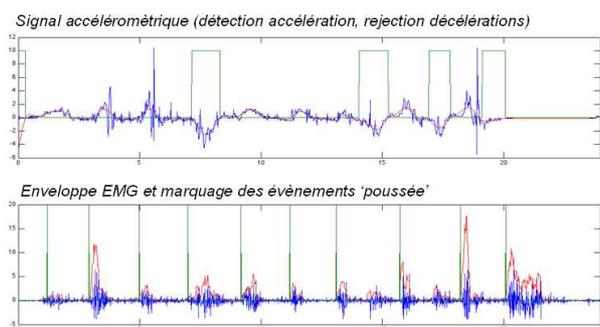


Figure 5 : Signaux accélérométriques et EMG traités.

5. CONCLUSION

Cet article décrit le projet AMO de fauteuil roulant manuel intelligent et communiquant pour le suivi et l'assistance de la personne à mobilité réduite dans le cadre d'un projet pédagogique d'intelligence ambiante collaboratif, inter-filière et pluridisciplinaire.

Plusieurs enseignements peuvent d'ores et déjà être retirés de la première itération de ce projet.

1. Ce type de projet est extrêmement motivant pour les élèves, car ils prototypent et réalisent une « chose » informatique qui sera potentiellement utilisable par de vrais « clients ».

2. Ce type de projet sensibilise plus particulièrement les élèves à la gestion de projet dont ils ne perçoivent pas la nécessité au démarrage. Ce type de projet leur permet également de comprendre les imbrications existants entre les différents cours que comportent leurs formations (IHM, Ergonomie, Systèmes, Systèmes réparties, Bases de Données, Génie Logiciel, Réseaux, Visualisation de données, Microcontrôleurs, Traitement du signal, Énergétique, ...) et de les mettre en pratique.

3. Les élèves ont du mal à appréhender la notion de prototypage rapide particulièrement requis dans le contexte de l'intelligence ambiante. Il est dès lors important que les élèves soient sensibilisés et formés sur ce point et que des enseignements orientés RAD soient introduits dans nos formations.

4. Les élèves provenant des filières différentes ont beaucoup de mal à considérer les « autres » comme des collègues travaillant vers un même objectif en y apportant des compétences à la fois spécifiques et complémentaires. Il est important que des enseignements vers la co-conception soient introduits dans nos formations dès les premières années des études universitaires.

5. L'investissement pour les enseignants encadrants ce type de projet est très important ce qui pourrait faire craindre des difficultés à préserver leur motivation et leur 'entraîné' pendant plusieurs années les encadrants.

Les points cités ci-dessus ne doivent cependant pas masquer l'intérêt et le réel plaisir que nous avons tous, étudiants, enseignants et clients à réaliser ce projet pédagogique inter-filière pluridisciplinaire. Nous pensons, en effet, que ce type de projet, comportant un enjeu médicale et/ou sociétal important, nécessitant une écoute, un dialogue, une collaboration, une entraide constitue une aventure pédagogique prometteuse et efficace pour la formation de nos étudiants des filières de l'école d'ingénieurs Polytech'Grenoble.

6. REMERCIEMENTS

Nous remercions l'ensemble des élèves ingénieurs qui ont réalisé cette première version du projet AMO, les équipes pédagogiques de Polytech'Grenoble, ainsi que la Fondation Garches pour son soutien.

7. REFERENCES

- [1] Vignier N, Ravaut JF, Winance M, Lepoutre FX, Ville I. (2008). Demographics of wheelchair users in France: results of national community-based handicaps-incapacités-dépendance surveys. *J Rehabil Med.* 40(3):231-9.
- [2] The Ambient Assisted Living (AAL) Joint Programme <http://www.aal-europe.eu/>
- [3] Dorothy McKinney, "Impact of Commercial Off-The-Shelf (COTS) Software and Technology on Systems Engineering", *INCOSE Chapters*, August 2001
- [4] AALOA Manifesto <http://aaloa.org/manifesto>
- [5] Filières TIS (Technologies de l'Information pour la Santé), 3I (Informatique Industrielle et Instrumentation) et RICM (Réseaux Informatiques et Communication Multimédia), <http://www.polytech-grenoble.fr>
- [6] Wiki du projet AMO http://air.imag.fr/mediawiki/index.php/Projet_%22AMO%22
- [7] Video de l'IHM d'AMOdroid <http://www.youtube.com/watch?v=hx4JFqrTQFA>
- [8] Hui Liu; Darabi, H.; Banerjee, P.; Jing Liu, Survey of Wireless Indoor Positioning Techniques and Systems *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Nov. 2007 Volume: 37 Issue:6 doi 10.1109/TSMCC.2007.905750
- [9] Tom Igoe, Making Things Talk: Practical Methods for Connecting Physical Objects, Make; 1 edition (September 28, 2007), ISBN-10: 0596510519

MARTS : Partage de données environnementales pour la découverte et l'observation de la terre*

Nehla Ghouaiel
ESTIA Recherche - IRIT
n.ghouaiel@estia.fr

Jean-Marc Cieutat
ESTIA Recherche - IRIT
j.cieutat@estia.fr

Jean-Pierre Jessel
IRIT
jessel@irit.fr

ABSTRACT

Le but de cet article est de montrer comment les techniques basées sur l'image et la Réalité Augmentée (RA) peuvent s'avérer être une assistance précieuse à la découverte de la terre et l'observation de l'évolution de l'environnement. Le support de l'étude est tout naturellement le Smartphone qui est devenu en quelques années l'appareil qui nous est le plus familier, qui nous accompagne quotidiennement partout où nous nous déplaçons.

Categories and Subject Descriptors

H.4 [Applications des Systèmes d'Information]: [développement durable et environnement]

Keywords

Réalité Augmentée, interfaces mobiles, analyse d'image, synthèse d'image

1. INTRODUCTION

La miniaturisation des unités centrales, la réduction de la consommation d'énergie et la généralisation des réseaux conduit à une omniprésence de dispositifs informatiques nous accompagnant dans notre vie courante : Smartphone, ordinateur portable, ... Comme a pu le penser Mark Weiser [9, 10], la démultiplication des systèmes a changé notre façon de les utiliser. C'est dans cet esprit que s'inscrit cet article visant à promouvoir un nouvel usage de nos téléphones portables afin de collecter des données environnementales. Chaque promeneur peut ainsi joindre l'utile à l'agréable, bénéficier d'informations pour choisir et agréer son itinéraire mais également prendre des photos et des relevés qui serviront à l'observation de la terre. Le réseau informatique mondial qu'est internet devient tout naturellement le lieu privilégié pour collecter, partager et analyser toutes les données environnementales.

*(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM_PROC_ARTICLE-SP.CLS. Supported by ACM.

2. LES TRAVAUX ASSOCIÉS

Les changements environnementaux sont un sujet d'intérêt pour les chercheurs, les gestionnaires et le grand public. En effet, les changements dans l'environnement peuvent affecter la vie de chacun, et donc ils forment un grand sujet de préoccupation. J. Danado et al [7], présente un système mobile qui permet de visualiser la qualité d'eau et le niveau de la pollution dans les lacs artificiels et dans les cours d'eau naturels. Le système proposé s'appuie sur une architecture client serveur et comporte deux modules : un module de réalité augmentée et un autre module de géo-référencement. Le système d'information géographique (SIG) a été utilisé en plus de la réalité augmentée [11] pour modéliser les paysages naturels et montrer leurs évolutions au cours des âges. Le cas d'étude présenté dans [11] illustre la propagation des mauvaises herbes.

3. PARTAGE DE DONNÉES

Pour aider les scientifiques comme tout un chacun à mieux connaître notre environnement et surveiller ses transformations, les données environnementales doivent être mises à la disposition de tout le monde.

3.1 Echanges d'informations

Le web est un lieu idéal d'échange et de partage de l'information. Un outil comme Google Earth s'avère être par exemple très pratique pour préparer une sortie (trace GPS du parcours, difficultés, estimation du temps du parcours, ...).

Un repère dans Google Earth permet également d'enregistrer des données d'un lieu comme une rencontre avec des isards au cours d'une randonnée dans les Pyrénées qui est illustrée ci-dessus. Par souci d'échanges, c'est une information qui, ici, ne manquera pas d'inciter d'autres promeneurs à choisir cet itinéraire. Mais c'est aussi une information pour préserver notre environnement et sa diversité biologique : ne pas déranger ces animaux sauvages même si leur curiosité peut parfois surprendre.

3.2 Géo-positionnement d'une photo

Toutefois, à y regarder de près, les associations de photos avec les coordonnées du globe terrestre manquent cruellement de précision dans Google Earth. Cette imprécision, contradictoire avec la notion même d'observation, empêche toute utilisation à caractère scientifique de l'information.

Le moyen le plus immédiat pour que le géo-positionnement

d'une photo soit précis serait d'insérer un "géotag" [4], un marqueur à caractère géographique, directement dans le format de l'image. Cette balise de géolocalisation peut être constituée des coordonnées GPS [1] (latitude, longitude, hauteur), de la direction de la vue en coordonnées polaires, de la date et d'une saisie. Citons par exemple le format Exif (Exchangeable image file format) établi par le JEIDA (Japan Electronic Industry Development) qui permet d'intégrer automatiquement une balise dans un format image.

En l'absence de traitement automatique, il reste toujours également possible de "tatouer" la balise de géolocalisation directement dans l'image. Cette technique utilise généralement le bit de poids faibles (le moins significatif) des premiers octets de l'image pour insérer directement l'information dans l'image.



Figure 1: Rencontre inattendue au cours d'une randonnée dans les Pyrénées.

3.3 Réalité Augmentée et constitution de photothèques

L'étape suivante serait de pouvoir également effectuer des photos, complétées par d'autres relevés, dans des conditions strictement identiques mais étalées dans le temps pour mener à bien des observations (érosion du littoral, déforestation, urbanisation, pollution,...). Constituer des photothèques¹ présuppose alors d'être capable de reproduire des photos en des emplacements caractéristiques avec les mêmes directions de vue.

La technologie récente qu'est la Réalité Augmentée (RA) [2] peut répondre à ce besoin d'assistance pour la constitution de photothèques. La problématique sous-jacente à la RA est d'être capable de calculer la pose de la caméra pour mélanger le flux vidéo avec des informations virtuelles. Ce calcul est ici effectué à partir du GPS et des capteurs inertiels (accéléromètre, magnétomètre et gyroscope) de plus en plus présents dans les Smartphones[12]. En effet, la boussole numérique incluse dans les Smartphones fournit l'angle du lacet relatif au mouvement effectué et l'accéléromètre compense la boussole en calculant l'inclinaison résultante du dispositif. Le gyroscope de son côté, permet de calculer la vitesse de la rotation du Smartphone appelée aussi la vitesse angulaire. Les angles de rotation du Smartphone sont déterminés selon les trois axes : X(Roulis), Y(Tangage), Z(Lacet) comme le montre la figure 3. Par la suite, il suffit d'ajouter comme indices virtuels dans le flux vidéo des données vectorielles cartographiques pour se rendre en un emplacement précis ou encore ceux d'une mire et d'une cible pour orienter la direction de la vue comme cela est illustré sur la figure 2.

3.4 Réalité Augmentée et visualisation de repères

Une autre fonctionnalité de la Réalité Augmentée, dès lors où il est possible de calculer la pose de la caméra comme expliquée ci-dessus, est de pouvoir visualiser et partager avec d'autres directement des informations à travers le flux vidéo de son Smartphone ouvert comme une lucarne sur l'environnement (voir la figure 4).

1. banque de photos



Figure 2: Assistance pour le relevé d'un éboulement de terrain lors d'une promenade sur le sentier du littoral à Bidart.

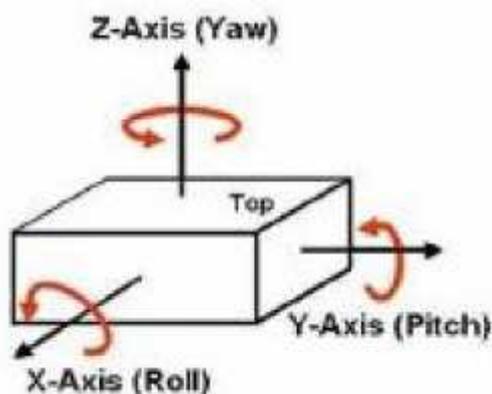


Figure 3: L'algorithme du filtre complémentaire permet de détecter le roulis, le tangage et le lacet.



Figure 4: La fameuse brèche de Rolland dans les Pyrénées.



Figure 5: Consultation de la photothèque.

4. ANALYSE ET INDEXATION D'IMAGE

Les deux disciplines que sont l'analyse et l'indexation d'images [8, 6] sont en perpétuelles évolutions avec un champ d'applications toujours plus vaste.

Les étapes de prétraitements, segmentation et de quantification sont les étapes générales à tout processus d'analyse d'images. Les critères employés pour la segmentation sont souvent complémentaires (couleurs, contours,...) et adaptés à la reconnaissance d'éléments naturels dans une image (relief, forêt, dunes, trait de côte,...).

L'indexation d'images est quant à elle particulièrement adaptée à la recherche d'un élément particulier dans une image (bâtiment, animal,...). Toute singularité est alors exploitée pour y parvenir (forme régulière, points d'intérêt,...).

5. DÉCOUVERTE ET OBSERVATION DE LA TERRE

Bien sûr, l'intérêt de la constitution de photothèques est de pouvoir les consulter et de les enrichir. Grâce aux techniques d'indexation d'images, il est alors possible de reconnaître tel ou tel animal qui paraît étranger, comme sur la figure 5 : quelle est cette race de chevaux qui paraît si petit ?

Dès l'instant où il devient possible de disposer de photos prises dans des conditions identiques mais à des époques différentes, on peut étudier l'évolution de l'environnement au cours du temps. Les techniques d'analyse et de synthèse d'images se prêtent volontiers à ce genre d'exercices.

Les environnements naturels subissent au cours du temps des transformations qui modifient continuellement leurs formes, tel est le cas des falaises d'Hendaye et d'Etretat des figures 6 et 7. Ainsi, il est très intéressant de permettre aux visiteurs d'un environnement naturel de visualiser ces changements. Dans ce cas de figure, les visiteurs pourraient par exemple tenir leurs dispositifs (PDA, Smartphone,...) et voir à ce que le paysage ressemblait dans les années passées à partir de leur point de vue. Sur le plan technologique, plusieurs techniques basées images sont capables de modéliser les transformations de l'environnement naturel. En analyse d'image, il est d'abord possible d'extraire les contours d'un relief, les dissocier du ciel et de la mer. Des travaux similaires détectent la ligne d'horizon [13, 3] en séparant la mer du ciel. Pour illustrer l'évolution au cours du temps des reliefs par exemple, la technique de "morphing" est très démonstrative [5]. Dans le cas du morphing qui consiste à transformer une

image 2D en une autre, les pourtours des reliefs sont détectés dans les deux images puis étirer d'une image à une autre, tout en créant des images intermédiaires (voir figure 6).



Figure 6: Erosion de la falaise à Hendaye.

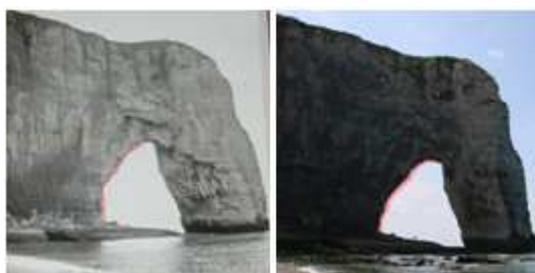


Figure 7: Erosion des falaises d'Etretat.

6. CONCLUSION

La réalité augmentée mobile sur Smartphone, comme le propose notre article, est une nouvelle façon de faire participer les scientifiques et les non scientifiques à observer l'environnement et à interagir avec lui. Cela a pour objectif d'augmenter la conscience écologique chez chacun. En plus de la visualisation des changements de l'environnement naturel, la réalité augmentée mobile et les techniques basées image peuvent être employées par exemple pour illustrer l'urbanisation au cours des âges.

7. REFERENCES

- [1] F.Evennou. *Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor*. Phd thesis, Université Joseph-Fourier, France, Mars 2007.
- [2] R. Azuma. A survey of augmented reality. *Presence : Teleoperators and Virtual Environments*, 6(4) :355–385, August 1997.
- [3] C. C.Jiang, H.Jiang and J.Wang. A New Method of Sea-Sky-Line Detection. In *Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 740–743. IEEE Computer Society, 2010.
- [4] a. M. C.P.Robert, D.P.Clough. Geo-tagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 77–82, 2007.
- [5] G.Wolberg. Recent Advances in Image Morphing. In *Proceedings of Computer Graphics International*, pages 64–71, June 1996.
- [6] H.Nezamabadi and S.Saryazdi. Object-Based Image Indexing and Retrieval in DCT Domain using Clustering Techniques. *World Academy of Science, Engineering and Technology*, 3 :98–101, 2005.
- [7] J. Danado, E. Dias, T. Romao, N. Correia, A. Trabuco, C. Santos, J. Serpa, M. Costa, and A. Camara. A multi-user mobile system to visualize environmental processes. In *Proc. 12th Int. Conf. on Geoinformatics - Geospatial Information Research : Bridging the Pacific and Atlantic*, pages 7–9. Geoinformatics, June 2004.
- [8] B. M.Obeid and M.Daoudi. Image indexing & retrieval using intermediate features. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 531–533. ACM, 2001.
- [9] M.Weiser. Human-computer interaction. chapter The computer for the 21st century, pages 933–940. 1995.
- [10] M.Weiser. Some computer science issues in ubiquitous computing. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3 :12–, July 1999.
- [11] P.Ghadirian and I. Bishop. Composition of Augmented Reality and GIS To Visualize Environmental Changes. *The Joint AURISA and Institution of Surveyors Conference*, November 2002.
- [12] D. S.Nasiri, S.Lin and J.Jiang. Motion Processing : The Next Breakthrough Function in Handsets. *InvenSense Inc*, 2009.

- [13] J. a. R. Z.Ji, Y.Su. Robust Sea-Sky-Line Detection Based on Horizontal Projection and Hough Transformation. In *Image and Signal Processing*, pages 978–990. IEEE, 2009.