

Vers un modèle des tâches dans l'ambient

Asma Gharsellaoui

LIMSI-CNRS, Univ. Paris-Sud
B.P. 133, 91403 Orsay Cedex,
France

Asma.Gharsellaoui@limsi.fr

Yacine Bellik

LIMSI-CNRS, Univ. Paris-Sud
B.P. 133, 91403 Orsay Cedex,
France

Yacine.Bellik@limsi.fr

Christophe Jacquet

SUPELEC

3 rue Joliot-Curie 91192 Gif-
sur-Yvette Cedex, France

Christophe.Jacquet@supelec.fr

RÉSUMÉ

La modélisation des tâches a fait l'objet de plusieurs travaux de recherche dans le but d'analyser l'activité pour l'évaluation ou encore pour la conception d'un système. Elle constitue aussi une aide précieuse dans le processus de conception des interfaces graphiques. Dans cet article, nous nous intéressons à la modélisation des tâches dans le cadre des environnements ambiants. Nous mettons en évidence les principaux problèmes introduits par la modélisation des tâches dans l'ambient : relation entre une tâche et son emplacement spatial, relation entre une tâche et les dispositifs nécessaires à son accomplissement, prise en compte des erreurs possibles survenant lors de l'exécution d'une tâche, etc. Ceci nous permet alors d'identifier certaines contraintes qu'un modèle de tâche adapté à l'ambient doit satisfaire.

Mots clés

Modélisation des tâches utilisateur ; intelligence ambiante ; interactions dans l'ambient ; systèmes centrés sur l'utilisateur.

ACM Classification Keywords

H.5.2. Information interfaces and presentation

General Terms

Human Factors; Design

INTRODUCTION

L'intelligence ambiante vise à concevoir des environnements dotés de divers capteurs et effecteurs permettant de collecter des informations et d'agir sur l'environnement de l'utilisateur. Les objets du quotidien se voient ainsi augmentés de capacités de calcul, de communication sans fil et d'interaction avec l'être humain [4] dans le but d'assister ce dernier dans l'accomplissement de ses tâches quotidiennes.

Les buts que cherche à atteindre l'utilisateur sont généralement décomposés en tâches. Celles-ci représentent donc les actions à réaliser afin d'atteindre un objectif déterminé [13]. Si la modélisation des tâches dans le cadre des interfaces graphiques a été largement étudiée (voir section suivante), la recherche dans le domaine de l'assistance et la supervision des tâches dans les environnements ambiants demeure encore à un stade embryonnaire. Notre travail s'inscrit dans ce contexte et se focalise sur la définition de méthodes garantissant le bon déroulement des tâches utilisateur en mettant en œuvre une approche de modélisation des tâches adaptée aux

caractéristiques des environnements ambiants. Cette modélisation est réalisée dans l'objectif final de prédire au mieux l'activité de l'utilisateur afin de le guider et de lui venir en aide en cas de problème.

Dans la première section les spécificités des tâches dans l'ambient sont détaillées. La deuxième section de cet article présente une vue globale des principaux modèles de tâches existants. Une nouvelle catégorisation des tâches dans l'ambient est ensuite proposée. Dans une dernière partie, nous décrivons les différentes exigences qu'un modèle de tâches adapté à l'ambient doit satisfaire.

SPÉCIFICITÉS DES TÂCHES DANS L'AMBIANT

L'intelligence Ambiante

Le terme intelligence ambiante a été introduit initialement par la Commission Européenne en 2001 [4,11]. Elle représente une vision de l'environnement quotidien de l'utilisateur dans lequel des objets intelligents embarqués répondent aux besoins de l'utilisateur humain [12].

Contrairement à un poste de travail dont les actions ont un impact seulement sur lui-même et son environnement immédiat, un système ambient peut intégrer des composants lui permettant d'agir plus largement sur l'environnement physique (effecteurs) ou d'acquérir des informations sur cet environnement (capteurs). En plus des capteurs et des effecteurs, le système peut intégrer des dispositifs permettant l'interaction avec l'utilisateur. Les capteurs disponibles dans le système peuvent aussi procurer des informations sur le comportement des utilisateurs (position, orientation, déplacement, etc.). Ces informations pourraient ainsi être exploitées afin de s'assurer du bon déroulement des tâches accomplies par l'utilisateur et lui offrir une assistance en cas de besoin.

Caractéristiques des tâches dans l'ambient

Dépendance vis-à-vis des ressources

Les systèmes ambiants peuvent comporter de nombreux dispositifs physiques et services logiciels. La réalisation de certaines tâches peut nécessiter l'utilisation de dispositifs ou de services bien définis. Or un environnement ambient est un environnement hautement dynamique au sein duquel les dispositifs et les services sont susceptibles d'apparaître ou de disparaître à tout moment. Lorsque l'utilisateur a besoin d'une ressource particulière, le système doit donc pouvoir l'invoquer ou à défaut invoquer une ressource équivalente. Cela nécessite une catégorisation des ressources nécessaires aux tâches en fonction des services qu'ils pourraient offrir. Par exemple pour téléphoner, un

téléphone peut être nécessaire, mais un ordinateur équipé de Skype pourrait également convenir.

Dépendance vis-à-vis de la localisation physique

Certaines tâches ne peuvent avoir lieu que dans certains endroits particuliers et ne sont pas réalisables ailleurs (par exemple la préparation d'un plat cuisiné ne peut se faire dans une salle de bain, de même qu'un bain ne peut se prendre dans une cuisine). De plus, il peut arriver que certaines ressources requises par la tâche ne soient pas mobiles. L'exécution de la tâche correspondante ne sera donc possible que dans un champ bien spécifique de l'espace physique.

Dynamicité

Les environnements ambiants du futur seront capables, à partir de l'observation du comportement et des habitudes de l'utilisateur, d'apprendre de nouvelles tâches. Par exemple un environnement ambiant pourrait être capable d'allumer automatiquement la télévision lorsque l'utilisateur rentre le soir chez lui, s'il a observé un penchant de l'utilisateur pour cette activité à ce moment. Le modèle des tâches dans l'ambiant revêt ainsi un caractère dynamique avec des possibilités d'évolutions dans le temps.

Autres caractéristiques

Les tâches dans l'ambiant partagent également certaines caractéristiques avec les tâches classiques : (1) plusieurs types de relations temporelles peuvent lier deux tâches dans l'ambiant ; (2) la gestion des erreurs de l'utilisateur et des dysfonctionnements du système doit également être prise en compte lors de la mise en place du modèle. On peut noter que la présence des capteurs constitue une aide précieuse pour la détection aussi bien des erreurs de l'utilisateur que des éventuelles pannes du système [9].

LIMITES DES MODÈLES DE TÂCHE EXISTANTS

De nombreux modèles de tâches ont été décrits dans la littérature. Nous présentons, dans cette section, une brève analyse critique des principaux d'entre eux.

Hierarchical Task Analysis (HTA)

Le modèle HTA propose de représenter les tâches de façon hiérarchique [1]. La modélisation démarre avec l'identification des buts. Ensuite la décomposition est répétée d'une manière récursive jusqu'à l'obtention de sous-tâches observables liées à l'utilisateur ou à l'interface [8]. Ce formalisme ne permet pas de distinguer entre la modélisation d'une action du système et celle de l'utilisateur. De plus ce modèle n'offre pas suffisamment d'opérateurs temporels (par exemple il ne comporte pas de symbole spécifique permettant de modéliser des tâches parallèles : la seule possibilité est de choisir une séquence non-ordonnée).

Groupware Task Analysis (GTA)

Le modèle GTA décrit une tâche en se focalisant sur trois concepts principaux : agents, travail et situation [14]. Les agents représentent généralement des personnes (individuelles ou en groupes) et dans certains cas des acteurs

non-humains ou des systèmes comprenant une collaboration entre personnes et machines. Le modèle GTA ne permet donc pas de vérifier si une tâche est allouée à un système, une personne ou les deux à la fois (dans le cas de tâches interactives).

Concurrent Task Trees (CTT)

Dans [10], les auteurs décrivent CTT comme un modèle possédant une représentation structurée en arbres. Ils distinguent 4 types de tâches: les tâches utilisateur, les tâches de l'application, les tâches d'interaction, et enfin les tâches abstraites (ayant besoin d'être raffinées). Ce modèle offre une grande variété de relations temporelles entre les tâches. Il est considéré comme l'un des modèles les plus riches. Sa principale limitation (qui s'applique également aux modèles précédents) concerne son caractère statique et sa non prise en compte ni du lieu de réalisation de la tâche ni des ressources nécessaires à son accomplissement.

User Action Notation (UAN)

UAN [6,3] consiste principalement en une notation textuelle qui représente les tâches utilisateur sous forme d'une structure quasi-hiérarchique de tâches asynchrones. Chaque tâche élémentaire est associée à une table à trois colonnes indiquant : (a) les actions d'utilisateur; (b) les réponses du système et (c) l'état interne du système. Bien que ce modèle offre plusieurs relations temporelles, il ne permet en général de représenter que des interactions spécifiques aux interfaces graphiques (appui de touches au clavier, opérations de glisser-déplacer à la souris, etc.).

K-MAD

Le modèle K-MAD [7] repose sur un formalisme en arbre qui définit les relations temporelles entre une tâche mère et ses sous-tâches. Ses tâches possèdent un ensemble varié de caractéristiques (nom, numéro, but, fréquence, modalité, importance...). Ce modèle distingue quatre type de tâche : utilisateur, système, interactive et abstraite. Il n'offre que les opérateurs de décomposition suivant : séquentiel, alternatif, parallèle, pas d'ordre. On note que ce modèle dispose d'un simulateur K-MADe mais qui n'intègre pas toutes les fonctionnalités du modèle mais ce modèle ne permet pas la gestion des erreurs ainsi que son caractère statique.

PROPOSITION D'UNE TAXONOMIE POUR LES TÂCHES DANS L'AMBIANT

Nous introduisons ci-dessous une nouvelle taxonomie des tâches dans l'ambiant. Pour justifier le besoin d'une telle taxonomie, considérons le scénario suivant :

« Anne part travailler à 8h du matin. Une fois qu'elle quitte sa maison, le chauffage s'éteint. Comme elle ne rentre à la maison qu'après 18h, elle a programmé l'enregistrement de sa série télévisée préférée qui passe à 17h. En cours de journée, le système "décide" de lui enregistrer un programme de cuisine en tenant compte de son goût pour ce genre de programmes. Le système remet le chauffage en route 30 minutes avant qu'elle ne rentre (en prenant en compte le moment où elle quitte son bureau, la distance entre la maison et le bureau et l'état du trafic) afin que la maison soit bien chauffée à son arrivée. Aussitôt rentrée, elle allume son téléviseur. Celui-ci étant situé en face de la

fenêtre ; le système décide de baisser légèrement les stores pour offrir une meilleure visibilité de l'écran. »

Nous proposons de catégoriser les tâches ambiantes selon deux axes en fonction de :

(a) la façon dont elles sont déclenchées:

- 1/ Tâche automatique : le système démarre la tâche de lui-même sans aucune intervention de l'utilisateur.
- 2/ Tâche interactive explicite : la demande de l'utilisateur est une condition nécessaire pour le lancement de la tâche.
- 3/ Tâche interactive implicite : l'interprétation des actions de l'utilisateur par le système conduit au déclenchement de la tâche sans que l'utilisateur ait conscience d'être en interaction avec le système.
- 4/ Tâche utilisateur: il s'agit d'une tâche accomplie entièrement par l'utilisateur.

(b) la façon dont elles sont acquises:

- 1/ Tâches programmées: c'est l'utilisateur ou le concepteur qui a défini la tâche.
- 2/ Tâches apprises: à partir de l'observation des habitudes de l'utilisateur, le système apprend à réaliser à la place de ce dernier certaines tâches répétitives.
- 3/ Tâches déduites : à partir du profil et des préférences de l'utilisateur, le système décide de réaliser certaines tâches par anticipation des besoins de l'utilisateur.

Le tableau 1 fournit des exemples extraits du scénario ci-dessus ou d'un scénario proche.

	Automatique	Interactive implicite	Interactive explicite	Utilisateur
Programmée	Enregistrer un programme TV programmé par l'utilisateur.	Mettre le chauffage en marche quand l'utilisateur rentre chez lui.	Enregistrer un programme TV immédiatement après l'ordre de l'utilisateur.	L'utilisateur demande au système de lui rappeler de faire une certaine tâche.
Apprise	Enregistrement TV régulier appris à partir des habitudes de l'utilisateur.	Mettre le chauffage en marche quand l'utilisateur rentre chez lui parce qu'il l'a demandé à plusieurs reprises.	Modification de la tâche planifiée suite à une interaction avec l'utilisateur. Le système apprend comment répondre à de nouveaux ordres.	Le système réalise qu'à chaque fois que l'utilisateur allume le climatiseur il ferme la fenêtre. Cette tâche doit être ajoutée au modèle de tâche pour la rappeler à l'utilisateur.
Déduite	Enregistrer un programme TV en raisonnant sur les goûts de l'utilisateur.	Mettre le chauffage en marche 30 minutes avant l'arrivée de l'utilisateur (durée calculée en fonction du trafic et de la distance entre la maison et le bureau).	Quand l'utilisateur allume la télévision, le système descend les stores puisqu'il juge que la lumière extérieure gêne la vue.	Une métarègle stipule que si le climatiseur est allumé, les fenêtres doivent être fermées. Si l'utilisateur ouvre les fenêtres sans éteindre le climatiseur, le système lui demande de l'arrêter manuellement.

Tableau 1: Illustration des catégories de tâches à travers des exemples

CONTRAINTES POUR UN MODÈLE DE TÂCHES DANS L'AMBIANT

Tenant compte des spécificités des tâches dans l'ambiant, nous pensons qu'un modèle de tâches adapté à l'ambiant doit répondre aux exigences suivantes:

Prise en considération de la dépendance vis-à-vis de l'emplacement et des ressources

Le modèle de tâche doit inclure la possibilité d'étiqueter une tâche selon les contraintes spatiales (certaines tâches ne peuvent être réalisées qu'à certains endroits) et les

ressources physiques et logiques nécessaires à sa réalisation. Plus particulièrement, on évitera si possible de spécifier une instance particulière de dispositif pour une tâche mais on se référera plutôt à une classe ou à un type de dispositifs. L'utilisation d'une ontologie pour la classification de ces dispositifs est recommandée [5], de façon à ce qu'une tâche puisse être rattachée à une classe abstraite de dispositifs. Lorsqu'une tâche est sollicitée le système pourra alors trouver une instance concrète de dispositif compatible et disponible.

Catégorisation des tâches

L'utilisation de notre catégorisation des tâches peut être utile dans plusieurs cas. Par exemple, lorsque les tâches à réaliser nécessitent simultanément une ressource donnée, le modèle doit permettre d'établir des priorités d'exécution. Dans ce cadre, les tâches *déduites* devraient par exemple être moins prioritaires car elles ne sont pas explicitement demandées par l'utilisateur. Les tâches *programmées* présentent un plus haut niveau de confiance et de priorité.

Modèle dynamique

Le modèle de tâche proposé devra être évolutif afin de permettre la représentation des tâches apprises et déduites. Il évolue lorsque le système acquiert de nouvelles connaissances sur le profil de l'utilisateur et apprend de nouvelles tâches. Par exemple, si on suppose que le modèle est sous forme d'arbre, des nœuds pourraient être ajoutés, modifiés ou retirés dynamiquement.

Traitement des erreurs

Dans un environnement ambiant, les erreurs de l'utilisateur et les défaillances des effecteurs peuvent être détectées grâce à l'aide des capteurs (par exemple un capteur de luminosité permet de détecter si une ampoule est défectueuse). Le modèle de tâche doit intégrer des alternatives qu'il sera possible de proposer à l'utilisateur.

Granularité de la décomposition des tâches

Le principe de base des modèles de tâches consiste à décomposer une tâche complexe en sous-tâches plus fines

jusqu'à l'obtention de tâches élémentaires non décomposables. Quel doit être la limite de décomposition d'une tâche dans un environnement ambiant ? Nous proposons d'arrêter la décomposition lorsque des services sont invoqués afin de garder un certain niveau d'abstraction. À partir de ce niveau, une décomposition plus fine reste possible en utilisant les modèles classiques.

Autres contraintes

Certains modèles de tâches existants présentent des avantages qu'un modèle de tâche pour l'ambiant doit également offrir. Par exemple il peut être utile de disposer d'une représentation graphique, aussi bien pour les concepteurs que pour les utilisateurs finaux.

Une structure hiérarchique pourrait également être utile pour permettre une identification claire des étapes nécessaires à la réalisation d'une tâche pour atteindre un objectif donné.

Le modèle devrait également inclure un ensemble riche d'opérateurs temporels pouvant exprimer les relations possibles entre les tâches (suite ordonnée, suite non ordonnée, parallélisme, suspension, attente de la terminaison d'une autre tâche, reprise, etc.).

CONCLUSION

La modélisation des tâches a été largement étudiée dans le cadre des interactions classiques. Dans cet article, nous nous sommes intéressés à la modélisation des tâches dans un environnement ambiant. Après avoir analysé les spécificités des tâches dans l'ambiant, nous avons proposé une nouvelle taxonomie de ces tâches selon deux dimensions : déclenchement et acquisition. Cette nouvelle taxonomie a été illustrée à travers des exemples tirés d'un scénario dans l'ambiant. Enfin, nous avons identifié les contraintes qu'un modèle de tâches adapté à l'ambiant doit satisfaire.

Nos futurs travaux concerneront l'élaboration d'un modèle de tâches conforme aux exigences dégagées dans cet article, ainsi que son instrumentation logicielle afin de pouvoir superviser les tâches utilisateur dans l'ambiant et lui apporter assistance dans la réalisation de ces tâches si nécessaire.

RÉFÉRENCES

1. Annett, J. Hierarchical Task Analysis. In the handbook of task analysis for human computer interaction, (2004), 67–116.
2. Bonino, D., Fulvio, C. Dogont-Ontology, Modeling for Intelligent Domestic Environments. In Proc. of the 7th International Conference on The Semantic Web, in Karlsruhe, Germany (2008), 790-803.
3. Coutaz, J., Paterno, F., Faconti, G., Nigay, L. Comparison of approaches for specifying multimodal interactive systems, in Proc. of the ERCIM Workshop on Multimodal Human-Computer Interaction, (1993), 165-174.
4. Ducatel, K, Bogdanowicz, M. Scapolo, F. Leijten, J and Burgelman, J-C. Scenarios for Ambient Intelligence in 2010. Final report, Information Society Technologies Advisory Group (ISTAG), European Commission, february 2001.
5. Gaggioli, A. Optimal experience in ambient intelligence, in Ambient Intelligence, IOS Press, (2005), <http://www.ambientintelligence.org>.
6. Hartson, H. R., Siochi, A. C., Hix, D. The UAN: A user-oriented representation for direct manipulation interface designs. In ACM Transactions on Information Systems, July 1990, 8(3):181–203.
7. Lucquiaud, V., Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateur, in the proceeding of the 17th french-speaking conference on Human-computer interaction, In Toulouse, France 2005: 83-90.
8. Mills, S. Contextualising design: Aspects of using usability context analysis and hierarchical task analysis for software design, in the Journal of Behaviour & Information Technology archive, Vol 26, Issue 6, November 2007, 499-506.
9. Mohamed, A., Jacquet, C, Bellik, Y. Real-time Diagnosis of Ambient Environments Using a Modeling of Physical Effects Combined with Temporal Logic, in Proc. of The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2011), Nov 2011.
10. Paterno, F., Mancini, C., and Meniconi, S. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Human-Computer Interaction, Interact'97, (1997), 362-69.
11. Riva, G., Loreti, P., Lunghi, M., Vatalaro, F., Davide, F. Presence 2010: The Emergence of Ambient Intelligence, in: G. Riva, W.A IJsselsteijn (Eds.), Being There: Concepts, effects and measurement of user presence in synthetic environments, IOS Press, Amsterdam, (2003), 60-81.
12. Sadri, F. Ambient intelligence: A survey in ACM: computer survey, Vol.43, Issue.4, (October 2011), 36–66
13. Schulungbaum, E. Model-based user interface software tools-current state of declarative models. Graphics, visualization and usability centre, Georgia Institute of Technology, GVU Tech #96 #30,1996.
14. Van der Veer, G. C., Lenting, B. F., Bergevoet, B.A.J. GTA: Groupware task analysis - modeling complexity, in Acta Psychologica, 1996, vol 91, 297-322.