

Outils de modélisation des tâches utilisateurs : exigences du point de vue utilisation

Vincent Lucquiaud et Dominique Scapin

I.N.R.I.A.
Domaine de Voluceau - Rocquencourt, B.P. 105
78153 Le Chesnay Cedex, France
{Vincent.Lucquiaud.Dominique.Scapin}@inria.fr

Francis Jambon

LISI/ENSMA
Téléport 2, BP 40109,
86961 Futuroscope cedex, France
Francis.Jambon@ensma.fr

RESUME

L'analyse de la tâche de l'utilisateur est une étape primordiale du processus d'ingénierie des systèmes interactifs. Les nombreux modèles de tâches actuellement disponibles peuvent laisser supposer que cette étape est correctement supportée. Or ces modèles sont peu ou pas outillés. Cet article propose de faire le point sur les capacités attendues, du point de vue de leur utilisation, des outils de modélisation des tâches utilisateur. Ces exigences sont structurées en trois catégories : l'aide à la description, l'utilisation de la description et les services communs sur les données du modèle. Chacune de ces catégories est illustrée par des exemples d'outils actuellement disponibles : IMAD, Euterpe, CTTE et Tamot.

MOTS CLES : Analyse des tâches utilisateur, outil, besoins utilisateur

ABSTRACT

User task analysis is a critical step in the a design process of interactive systems. The large set of user task models available today may lead to assume that this step is well supported. However, these models are little or not tool-supported. This paper focuses on user requirements the user task analysis tools. These requirements are organized into three categories: description support, model data use and model data utilities. Each category is illustrated by examples coming from available tools: IMAD, Euterpe, CTTE and Tamot.

KEYWORDS: User tasks analysis, tool, user requirements

INTRODUCTION

La prise en compte des facteurs humains est désormais reconnue comme étant nécessaire dès les premières

étapes du processus d'ingénierie des systèmes interactifs. Cette prise en compte fait appel, dans la grande majorité des cas, à une description de la tâche de l'utilisateur. Il existe actuellement de nombreuses techniques de spécification des tâches utilisateur dont les caractéristiques sont parfois très différentes. Ainsi, a-t-on cherché à les comparer.

En effet, ces techniques de spécification peuvent être perçues selon différents points de vue tels que leurs origines, leur caractère formel, leurs objectifs, leur pouvoir d'expression, etc. Différentes classifications de ces techniques ont déjà été réalisées comme celles de Brun [5], Jambon [8][9] ou encore Balbo [1][2].

Celles de Brun [5] et de Jambon [8] par exemple classent ces techniques suivant leurs origines. Elles nous informent de leur caractère plus ou moins formel lequel permet, entre autres, l'élaboration de preuves. Les auteurs font également remarquer que l'évaluation des aspects issus des sciences cognitives n'est pas prise en compte dans les spécifications issues des mathématiques. En outre, ces classifications ne s'intéressent qu'au pouvoir d'expression théorique des spécifications, et elles ne précisent pas l'utilisation pratique qui peut en être faite par l'intermédiaire d'outils.

Celles de Balbo [1][2] ou Jambon [9] comparent ces techniques suivant différents critères. Ces critères ont été choisis pour recouvrir un éventail important de techniques de spécification, engendrant ainsi des critères parfois beaucoup trop généraux pour mettre en avant leurs compétences dans le cadre d'une utilisation pratique. Par exemple, pour le critère de preuve, la classification ne permet pas de savoir quels types de preuves peuvent être réalisées pour chacune de ces techniques. Ainsi, elles ne permettent pas de dégager avec précision les implications et les compétences de ces techniques lors de leur utilisation pratique.

L'intérêt d'une technique de spécification étant de pouvoir être utilisée dans le processus d'ingénierie, nous orientons directement notre travail non pas vers les compétences théoriques des techniques comme le font nombre de classifications, mais vers leurs outils. Cette

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.
IHM 2002, November 26-29, 2002, Poitiers, FRANCE.
Copyright 2002 ACM 1-58113-615-3/02/0011...\$5.00.

approche nous permet notamment d'évaluer ces techniques en fonction de ce qu'elles peuvent réellement apporter à l'utilisateur au cours du processus d'ingénierie. Nous proposons ainsi un ensemble structuré d'exigences qui se basent sur les besoins des utilisateurs des techniques d'analyse de tâches, contrairement à Bentley [3] qui exprime les exigences des outils à travers ce qu'ils doivent être capables de représenter. Son travail est tourné vers leurs contenus, par exemple les opérateurs temporels ou la coopération de tâches, alors qu'ici elles ne sont pas toujours en lien avec leurs contenus.

Nous présentons ces exigences suivant trois catégories qui correspondent aux étapes que suit l'utilisateur lors de l'utilisation pratique d'un outil. Par ailleurs, chacune de ces exigences est illustrée, quand cela est possible, par une compétence présente dans l'un des outils représentatifs examinés. Ces outils sont: IMAD (version 1) [10], Euterpe (version 1.00.91) [7][18], CTTE (version 1.5.3) [6] et Tamot (version 3v2.0e) [16]. Ces outils implémentent respectivement les modèles de tâche MAD* [14], CTT [12], GTA [17] et Diane+ [15]. Notons que ces modèles et outils ont inspiré nombre d'autres outils recouvrant d'autres étapes du cycle de développement.

EXIGENCES SUR LES OUTILS

La première de ces exigences concerne la description, c'est-à-dire les informations que l'utilisateur fournit pour décrire l'activité qu'il souhaite représenter. La seconde a trait à l'utilisation de la description, c'est-à-dire aux compétences liées à la manipulation et à l'évaluation de ces informations. Enfin la dernière concerne les services communs sur les données des modèles et les exigences relatives à leur gestion, c'est-à-dire la gestion de fichiers, la génération de documents et leur comparaison.

Aide à la description

L'utilisateur doit pouvoir décrire l'ensemble de l'activité avec les concepts présents dans le formalisme du modèle. Par exemple le concept de tâche qui est central pour décrire l'activité doit être descriptible en détail au moyen de l'outil. De plus, il s'agit de minimiser le risque d'erreur et de parvenir au plus vite à une description cohérente, tout en tenant compte de la réalité du terrain. L'aide à la description comprend cinq types d'assistance qui vont du recueil des données à la vérification de leur cohérence syntaxique.

Recueil des données : Avant d'envisager une description, l'utilisateur doit tout d'abord recueillir les données appropriées. Ces données sont issues d'interviews, de questionnaires ou/et d'entretiens. Cette étape est d'autant plus facilitée (et proche du formalisme) que des techniques structurées de recueil d'information sont disponibles (par exemple entretiens semi-directifs [13], pour utiliser IMAD).

Transformation des données initiales : Afin de faciliter le passage des notes du recueil à la description elle-même, des outils automatiques ou semi-automatiques d'extraction existent (cf. Brassier [4]) où est proposée une méthode d'extraction à partir de mots clefs, en utilisant un réseau de transitions augmenté. Concernant les outils, seul CTTE prend en charge les notes textuelles durant la description. L'utilisateur peut ainsi décrire le modèle en visualisant le fichier contenant les notes de terrain.

Support à la description : La démarche de description de l'activité étant dépendante du contexte du recueil, il est nécessaire qu'elle puisse être menée de façon flexible et itérative. L'outil doit donc permettre au moins trois modes de construction :

- Le mode « top-down » permet à l'utilisateur de décrire l'activité de la racine aux feuilles de l'arbre, comme le font tous les outils.
- Le mode « bottom-up » permet à l'utilisateur de décrire l'activité des feuilles à la racine de l'arbre. Seul CTTE le permet.
- Le mode « opportuniste » permet à l'utilisateur de décrire l'activité en fonction des données disponibles. L'outil ne contraint pas l'utilisateur à un ordre de description. L'utilisateur peut ainsi décrire des tâches dont la complétude, plus particulièrement l'ordonnancement, est partielle. Par exemple, dans IMAD, un « presse-papier actif » lié à la description en cours correspond en partie à cette exigence. De plus, ceci permet la constitution de descriptions provisoires, lesquelles peuvent ensuite être réintégrées dans une description plus complète.

Bibliothèque de tâches : Dans un objectif de réutilisation de tâches déjà décrites, une bibliothèque contenant des structures de tâches génériques est à prendre en considération. Ceci est suggéré dans IMAD, mais n'est pas implémenté.

Vérification syntaxique de la cohérence : Les aides mis en œuvre pour vérifier la cohérence des données doivent être proactives (en cours de description) et réactives (après la description).

Pour faciliter la description et donc l'apprentissage de l'outil, il doit être possible d'adapter les attributs de la description en fonction de ceux déjà exprimés. Cela permet de limiter les erreurs et d'informer l'utilisateur des liens qui existent entre les attributs. Par exemple, dans IMAD si une tâche est terminale, les autres opérateurs changent d'apparence.

Après avoir décrit une partie de l'activité, il convient de vérifier la cohérence du modèle, c'est-à-dire que la description respecte bien la grammaire du modèle (par exemple, qu'un minimum est plus petit qu'un maximum, qu'une tâche a bien un exécutant). Au niveau des outils Euterpe permet, à l'aide d'un formulaire, de vérifier les

contraintes de cardinalité (par exemple: les tâches n'ayant pas de responsable ; les objets utilisés par aucune tâche ; ou encore les rôles sans agents). CTTE propose une commande qui contrôle la description. Les résultats de cette analyse permettent d'indiquer à l'utilisateur les éventuels « warnings » ou erreurs à la manière d'un compilateur. Par exemple il indique qu'une tâche abstraite, donc décomposable, est utilisée comme tâche terminale. IMAD contrôle l'expression de toutes les conditions des tâches.

L'ensemble de ces exigences d'aide à la description correspondent aux services que l'outil dispose pour aider l'utilisateur. Certains de ces services, comme la cohérence des données sont liés au formalisme du modèle, contrairement à d'autres services comme les bibliothèques. À ce stade, il n'y a pas de traitement sur les données ; ceci fait l'objet de la partie suivante.

Utilisation de la description

Une fois les données vérifiées, l'utilisateur de l'outil peut avoir divers objectifs de type interrogation, simulation, validation, ainsi que des objectifs d'évaluation et de conception.

Interrogation : Elle est réalisée par des critères préalablement établis selon les objectifs de l'utilisateur. Cela permet notamment de vérifier que la description est bien conforme à celle désirée. L'utilisateur peut ainsi demander, comme le fait Euterpe, les tâches utilisant tel objet ou les tâches ayant plus de x sous-tâches. Les résultats sont visibles à partir de la description, mais il n'est pas possible de les enregistrer.

Pour garantir l'intégrité des données tout au long de la description ou étudier les conséquences d'une modification, la comparaison de deux fichiers contenant le résultat d'une requête doit être réalisable. Cette exigence n'est réalisée par aucun des outils.

De plus, à partir d'une interrogation, il doit être possible de faire de l'évaluation sur le résultat de cette requête. Par exemple, plusieurs requêtes peuvent être effectuées, suivant plusieurs contextes, afin de trouver la longueur du chemin utilisé par l'utilisateur et ainsi comparer leur temps d'exécution.

Simulation : Seule la simulation du modèle de tâche permet à l'utilisateur de contrôler l'ordonnement exact des tâches. Pour cela la simulation doit être réalisée étape par étape en proposant à l'utilisateur les tâches pouvant être exécutées selon le contexte. L'utilisateur doit avoir la possibilité de modifier les attributs influençant l'ordonnement afin de visualiser directement les conséquences. IMAD et CTTE le permettent, mais avec CTTE les modifications ne sont pas réalisables. Par ailleurs, la simulation doit pouvoir être faite automatiquement et permettre la simulation des effets de bord sur les objets afin de tenir compte des

conditions. CTTE et IMAD proposent une simulation mais sans tenir compte des conditions. En outre, ils permettent d'enregistrer le scénario de la simulation. CTTE autorise sa relecture.

Validation sémantique du modèle : Elle a deux exigences qui sont : la vérification des contraintes et la vérification d'ordonnement. La première permet de détecter qu'une contrainte est contradictoire par rapport à celle exprimée en amont. Par exemple, une précondition d'une sous-tâche n'est pas conforme à une post-condition de sa tâche mère. La seconde permet de garantir que toutes les tâches sont accessibles au moins une fois. Les modèles ne disposent actuellement pas suffisamment de sémantique pour remplir ces exigences.

Évaluations et Conception : A partir des données de la description, diverses évaluations peuvent être effectuées, comme le temps d'exécution des tâches, ou l'évaluation de la charge de travail. Ces évaluations dépendent étroitement du contenu de la description. Le temps d'exécution ne peut être évalué que si le formalisme du modèle prend en compte le temps. Pour l'évaluation, d'autres outils peuvent, en récupérant les données issues de la description, prendre en charge cette partie. Par exemple EMA [2] évalue l'interface à partir des données de IMAD.

De plus, au niveau conception, du code ou des interfaces peuvent être générés en liaison avec un autre outil comme par exemple SSI [10] qui traduit les éléments de la tâche MAD* en éléments d'interface. Ou bien, PetShop [11] qui utilise les données de CTTE dans le but de faciliter le passage entre la modélisation des tâches et le modèle du système.

Services communs sur les données du modèle

S'agissant des données, nous distinguons trois exigences : la gestion de versions, la génération de documents et leur comparaison deux à deux.

Gestion de version : Quand l'utilisateur a terminé une partie de la description ou bien lorsqu'il s'apprête à enrichir la description de façon significative, la sauvegarde classique des données dans un fichier comme le font les quatre outils examinés ne suffit pas car il n'y a pas moyen de suivre des modifications consécutives qui ont été faites ainsi que leurs justifications. C'est pourquoi la présence d'un système d'archivage est important. De plus, elle facilite la gestion du travail en équipe.

Génération de documents : Ceci regroupe tous les services liés à la production de documents. Ceux-ci sont souvent nécessaires, par exemple sur le terrain ou lors de réunions de conception, sous forme papier ou sous forme électronique (avec des hyperliens, des photos de terrain, des vidéos ou du texte justifiant un choix). Tamot, CTTE et Euterpe génèrent des fichiers HTML ou XML, mais seul Euterpe ajoute des supports multimédias. De plus,

CTTE peut créer des images de l'arbre des tâches ou générer un fichier correspondant au format LOTOS.

Comparaison : la comparaison de deux descriptions permet à l'utilisateur de comparer le contenu de deux fichiers de données. La comparaison se fait à partir de critères comme le nombre de tâches, d'utilisateurs. CTTE réalise cela et donne des informations sur les éventuelles différences. Néanmoins deux descriptions ne peuvent être comparées qui si elles utilisent les mêmes rôles.

Nos trois catégories d'exigences regroupent les compétences qui nous semblent incontournables pour prétendre décrire convenablement une activité ainsi que manipuler et extraire les données des descriptions. Néanmoins, certaines exigences sont fortement liées aux concepts présents dans les modèles. En effet les exigences relatives à la vérification syntaxique de la cohérence et toutes celles présentées dans la deuxième section ne peuvent être évaluées qu'en fonction de ce que le modèle est à même de représenter. À l'évidence, si l'outil n'a pas la notion d'objet ou de temps, il ne lui est pas possible de simuler des tâches en tenant compte de l'état des objets ou de prédire le temps d'exécution.

CONCLUSION

Cet article est une première approche qui tente d'élaborer des exigences basées sur l'utilisation d'un outil de terrain modélisant les tâches utilisateur. Ces exigences font apparaître que les outils, bien qu'ils tentent tous de modéliser l'activité, ont des compétences et des services fort différents, voire disjoints. En outre, l'utilisation d'un outil détermine pour l'utilisateur l'aide dont il peut disposer pour décrire l'activité et garantir la cohérence et la validité de sa description. De plus, le potentiel de l'outil permet de garantir l'intégrité des données et de leur facilité la consultation.

Pour les exigences qui demandent une sémantique formelle comme la validité, la simulation des tâches avec des objets, en tenant compte des valeurs pouvant être non valides, aucun outil ne permet à ce jour de remplir totalement ces exigences. Il y a donc bien un manque de sémantique au sein des modèles de tâches, qui empêche de bénéficier des avantages des modèles mathématiques en termes de preuve et d'analyse. Cependant, il convient de s'intéresser également à l'essence même des descriptions, aux concepts et attributs, dans un contexte réel de terrain. Ceci est l'objet d'une étude à objectif descriptif et évaluatif, menée en parallèle, dans le cadre du transport ferroviaire.

BIBLIOGRAPHIE

1. Balbo S., Paris C., Ozkan N., *Characterising Task Formalisms : Towards a Taxonomy*, report N° CMIS 98/221, CSIRO-MIS Sydney
2. Balbo S., *Un pas vers l'évaluation automatique des interfaces homme-machine*, thèse en informatique, Université Joseph Fournier, septembre 1994

3. Bentley T., Johnson L., *Analyse of task model requirements*, in the proceedings of OzCHI, Fremantle, Australia, November 20-23, 2001
4. Brasser M., Vander Linden K., *Automatically eliciting task models from written task narrative*, Proceedings of the Fourth Conference on Computer-Aided Design of User Interfaces, 15-17 May, 2002, Valenciennes, France
5. Brun P., *XLT: une logique temporelle pour la spécification formelle des systèmes interactifs*, thèse en informatique, Université Paris XI, septembre 1998.
6. CTTE <http://giove.cnuce.cnr.it/tools2.html>
7. Euterpe <http://www.cs.vu.nl/~martijn/euterpe.html>
8. Jambon F., Brun P., Aït-Ameur Y., *Spécification des systèmes interactifs*, in Analyse et conception de l'IHM interaction homme-machine pour les SI 1, Kolski (dir), Hermes, Chapitre 6, p175-206, 2001
9. Jambon F., *Erreurs et Interruptions du point de vue de l'ingénierie de l'interaction homme-machine*, Thèse en Informatique, Université Joseph Fournier, 1996
10. Gamboa Rodriguez F., *Spécification et implémentation d'ALACIE: atelier logiciel d'aide à la conception d'interfaces ergonomiques*, thèse en informatique, Université Paris sud XI, octobre 1998
11. Navarre D., *Contribution à l'ingénierie en interaction Homme-Machine*, thèse en informatique, Université Toulouse I, juillet 2001
12. Paterno F., *Model-Based Design and Evaluation of Interactive Applications*, Springer Verlag, 1999
13. Sebillotte S., *Décrire des tâches selon les objectifs des opérateurs : de l'interview à la formalisation*, le travail Humain, 54(3), p.193-223, 1991
14. Scapin D.L., Bastien C., *Analyse des tâches et aide ergonomique à la conception: l'approche MAD**, in Analyse et conception de l'IHM interaction homme-machine pour les SI 1, Kolski (dir), Hermes, Chapitre 3, p85-116, 2001
15. Tarby J.C., Barthet M.F., *Analyse et modélisation des tâches dans la conception des systèmes d'information : la méthode Diane+*, in Analyse et conception de l'IHM interaction homme-machine pour les SI 1, Kolski (dir), Hermes, Chapitre 4, p117-144, 2001
16. Tamot <http://www.cmis.csiro.au/iit/Projects/Isolde/Tamot>
17. Van der Veer G.C., Lenting B.F., Bergevoet B.A.J., *GTA:Groupware Task Analysis - Modeling Complexity*, Acta Psychologica, 91, 1996, pp. 297-322
18. Van Welie M., Van der Veer G.C., Eliëns A., *Euterpe - Tool support for analyzing cooperative environments*, Proceedings of the Ninth European Conference on Cognitive Ergonomics , August 24-26, 1998, Limerick, Ireland