

Planification pour la Composition Dynamique d'Interfaces Homme-Machine

Yoann Gabillon

Laboratoire d'Informatique de Grenoble
385, rue de la bibliothèque
BP53, 38041, Grenoble cedex 9, France
yoann.gabillon@imag.fr

RESUME

Aujourd'hui, une Interface Homme-Machine (IHM) est le fruit de transformations souvent mentales, appliquées à une représentation abstraite de l'IHM appelée modèle de tâches. Ces transformations font des compromis entre le contexte d'usage courant (<Utilisateur, Plate-forme, Environnement>) et les propriétés d'utilisabilité à satisfaire. En informatique ambiante, le contexte d'usage devient variable et imprévisible. Il est donc nécessaire de doter le système de capacités de raisonnement (modèles et transformations) pour lui permettre de conformer l'IHM à ce contexte dynamique. Ma thèse explore la Planification pour la construction d'un système interactif conforme à un objectif utilisateur donné (un modèle de tâches partiel) et un contexte d'usage donnés. L'Ingénierie Dirigée par les Modèles (IDM) est utilisée comme passerelle entre ces deux communautés.

MOTS CLES : Composition dynamique, Contexte d'usage, IHM, Planification, IDM.

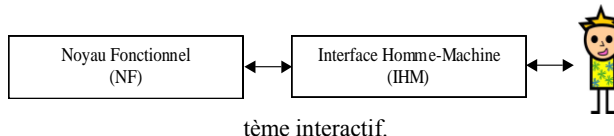
INTRODUCTION

Ma thèse traite de la composition dynamique de systèmes interactifs par planification. La décomposition fonctionnelle minimale d'un système interactif distingue le Noyau Fonctionnel (NF) de l'Interface Homme-Machine (IHM) (Figure 1). Le NF regroupe l'ensemble des traitements indépendants de toute représentation à l'utilisateur. L'IHM fait les choix de représentation compte tenu du contexte d'usage ciblé et des propriétés ergonomiques à satisfaire. Par contexte d'usage, on entend l'utilisateur (ses caractéristiques, compétences, préférences, etc.), sa plate-forme d'interaction (capacités de calcul, de communication et ressources d'interaction disponibles en entrée et en sortie) ainsi que l'environnement physique/social de l'utilisateur (niveaux lumineux, sonore, etc.).

C'est l'analyse des besoins qui permet de cerner la couverture fonctionnelle à donner au NF. S'ensuit alors un processus de développement en V, spirale ou autre. L'architecture logicielle en est une étape clé : elle couvre l'identification des composants, processus et ressources physiques (dont les processeurs) ainsi que l'allocation

des fonctions (NF et IHM) aux composants, des composants aux processus et des processus aux ressources physiques [1]. Si jusqu'ici ces identifications (fonctions, composants, processus et ressources physiques) et allocations (fonctions aux composants, ..., processus aux ressources physiques) étaient immuables, figées à la conception, elles doivent désormais être dynamiques. Cette dynamique permet de répondre à la variabilité et imprévisibilité du contexte d'usage en informatique ambiante.

Figure 1 : Décomposition fonctionnelle minimale d'un sys-



tème interactif.

Ma thèse explore les algorithmes de planification de tâches tels qu'étudiés en Intelligence Artificielle (ici appelés *planification* pour simplifier) pour la composition dynamique d'IHM. La planification produit des plans exécutables pour un objectif et un état du monde donnés. Le plan est un ordonnancement d'actions ou d'opérateurs (par exemple, déplacer un cube). Traditionnellement, la planification est étudiée en robotique. Elle est ici appliquée de façon originale à l'ingénierie des Interfaces Homme-Machine (IHM).

Cet article identifie les verrous scientifiques de l'étude puis présente l'approche adoptée ainsi que les premiers résultats. L'ensemble est illustré sur un cas d'étude implémenté dans un premier démonstrateur « en largeur ».

CAS D'ETUDE ET DEMONSTRATEUR MAPPING

Le cas d'étude est celui d'un chercheur (Bob) inscrit à une conférence. Bob arrive en retard à une session d'intérêt. Déçu, il reste dans le couloir, faute de place dans l'amphithéâtre. Il suit la présentation à distance grâce à un mur écran installé dans le couloir, mais au moment des questions il aimerait se manifester. Il décide alors d'utiliser le système MAPPING (Models, Artificial Perception and Planning for INteractive systems Generation) pour pouvoir poser sa question au conférencier. Il

spécifie son objectif « I want to ask questions » en langage naturel (l'anglais) dans une fenêtre dédiée (Figure 2).

Le système MAPPING analyse l'objectif utilisateur. Il établit alors un modèle des tâches incomplet : dans l'exemple, il est réduit au seul objectif utilisateur (« Poser des questions »). Aucune procédure (i.e., façon d'accomplir la tâche pour atteindre l'objectif) ne lui est associée. Grâce à un annuaire de services stockant des descriptions de systèmes interactifs en termes de tâche utilisateur offerte et de ressources d'interaction requises, MAPPING identifie deux procédures possibles pour atteindre le but : requérir à un outil de communication synchrone (par exemple, Messenger) ou asynchrone (par exemple, le courrier électronique) (Figure 2). Les deux plans sont proposés à l'utilisateur.

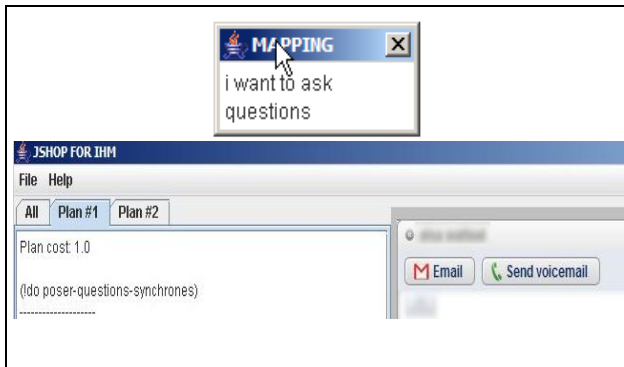


Figure 2 : Spécification de l'objectif utilisateur en langage naturel. Obtention de deux plans selon le mode synchrone ou asynchrone de la question.

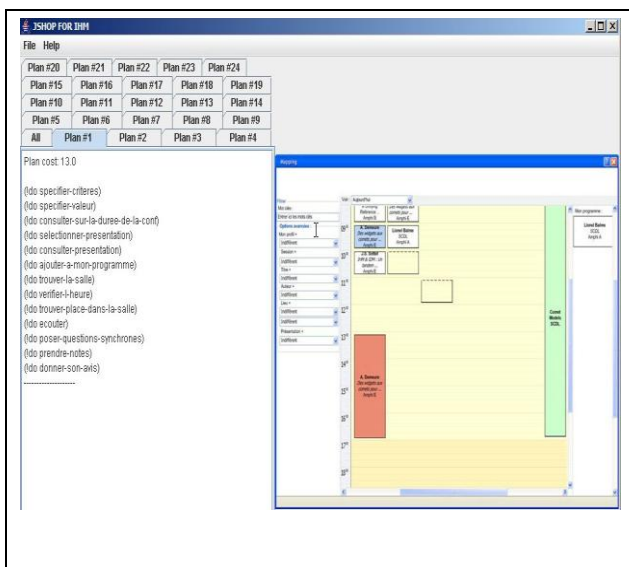


Figure 3 : Vingt-quatre plans possibles pour l'élaboration du programme scientifique de Bob. L'IHM du plan sélectionné (#1) s'affiche en partie droite.

La Figure 3 montre l'application de MAPPING à un exemple où la combinatoire est plus élevée : la tâche uti-

lisateur est plus complexe (assister à des présentations sur un sujet donné) et la plate-forme plus variée : réduite à un seul PC en Figure 2, elle est ici étendue à un PC et un PDA, allumés ou éteints, branchés ou non à un clavier physique. L'utilisateur (anglophone) et l'environnement (sonore) restent fixés. Vingt-quatre plans sont trouvés et présentés à l'utilisateur.

DIFFICULTES ET APPROCHE

D'un point de vue fonctionnel, deux difficultés se posent :

1. Percevoir les exigences (l'objectif utilisateur et les propriétés d'utilisabilité attendues) et les ressources disponibles (dispositifs d'entrée et de sortie, unités de calcul, capacités de communication, composants réutilisables) ;
2. Calculer un NF et une IHM conformes aux exigences et ressources.

Dans MAPPING, l'objectif utilisateur est formulé en langage naturel par l'utilisateur lui-même. Les exigences d'utilisabilité ne sont aujourd'hui pas formulées. Les ressources sont déclarées. Le calcul se fait par planification : c'est l'approche suivie dans ce travail pour répondre au problème de la combinatoire.

Mon approche (Figure 4) est la planification. Comme passerelle entre les communautés IHM et Planification, je pratique l'Ingénierie Dirigée par les Modèles (IDM). De façon plus précise, l'approche s'articule en cinq temps :

1. Spécification du problème dans le domaine IHM : l'objectif utilisateur est analysé et reformulé en un modèle de tâches incomplet. Dans MAPPING, l'analyse se fait via ConceptNet [4], un moteur de recherche de sens.
2. Réécriture du problème dans le domaine de la planification [5] : le modèle de tâches est transformé en un problème de planification ; les ressources définissent l'état du monde.
3. Calcul de plans par un planificateur : le calcul s'appuie sur les composants réutilisables (modèles ou code relevant du NF ou de l'IHM). Dans MAPPING, une IHM est associée à chaque modèle de tâches. Les modèles et codes sont entreposés dans un annuaire de services. Le planificateur est JSHOP [3]. Il combine les modèles de tâches pour réaliser l'objectif utilisateur compte tenu des ressources disponibles. JSHOP renvoie un plan : une séquence d'actions physiques.
4. Réécriture des plans en IHM : le plan est reformulé en un modèle de tâches.

5. Transformation en IHM pour construire une IHM ergonomique : le modèle de tâches peut être restructuré par ajout de tâches abstraites pour typiquement éviter des « rateaux » qui aboutiraient à des IHM « timbres postes » ; des concepts peuvent être factorisés pour un meilleur Guidage / Groupement-distinction entre items [6] ; etc.

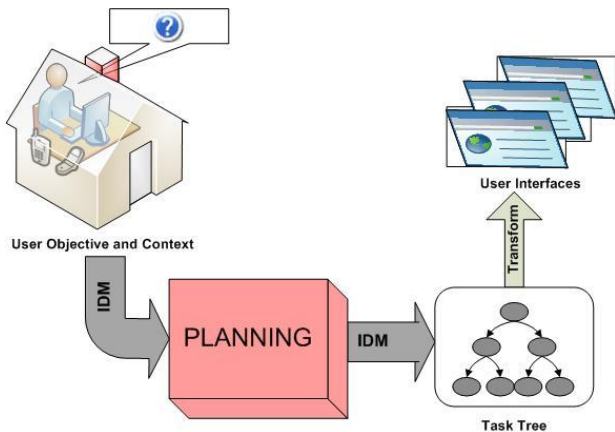


Figure 4 : Notre approche.

MAPPING est un démonstrateur « en largeur ». De nombreuses questions restent ouvertes tant en description (quels langages de tâches, de ressources, ... ?) qu'en algorithmes (SHOP, GraphPlan, ... ?). Aujourd'hui, mon travail s'est porté sur la compréhension du problème et la passerelle entre les deux communautés.

PASSERELLE ENTRE IHM ET PLANIFICATION

Les transformations entre IHM et planification sont fondées sur des métamodèles explicites : d'une part, une (méta)modélisation des tâches utilisateur [7] et du contexte d'usage [2] et d'autre part de la planification. La figure 5 donne mon métamodèle de tâches. Par rapport à l'existant, l'originalité de ce métamodèle est de permettre une estampille de tout élément (tâches, opérateurs entre tâches, décorations de tâches) du contexte d'usage dans lequel il fait sens.

Le concept clé du métamodèle est le concept de «Task». Ce concept représente l'objectif utilisateur (ex : poser une question) et la procédure qui lui est associée (une décomposition en sous-tâches par le biais d'opérateurs). Une tâche possède un nom, une catégorie (par exemple, tâche abstraite si elle est décomposée en sous-tâches de catégories hétérogènes, tâche mentale ou tâche d'interaction), un type (comparaison, édition, etc.) et un ensemble d'informations telles qu'une description, son caractère collaboratif, des éventuelles pré-conditions à satisfaire avant toute réalisation, etc.

Les tâches sont liées entre elles par des opérateurs, donnant globalement lieu à un arbre des tâches. Le sommet est le concept de «TaskModel». Les « opérateurs » peu-

vent être logiques (et, ou) ou temporels (séquence, entrelacement, etc.)

Une tâche manipule des concepts («Concept»). Ce sont les entités du domaine. Chaque concept a une certaine centralité dans la tâche. Cette information est déterminante pour la composition de l'IHM. Par exemple, un concept central devra être affiché de facto sans que l'utilisateur ait à naviguer (liens hypertextes, onglets, boutons, etc.) pour y accéder. Une navigation nuirait à l'ergonomie de l'IHM. Les exigences non fonctionnelles de l'utilisateur sont donc à considérer dans le choix de la transformation composant l'IHM.

Les tâches peuvent être décorées («Decoration»). Les décorations expriment classiquement la criticité d'une tâche (tâche irréversible), sa fréquence, son caractère itératif (la tâche peut être exécutée plusieurs fois par l'utilisateur) ou optionnel (tâche facultative). Des contraintes temporelles peuvent être formulées (durée minimale, maximale et moyenne).

Le concept de «Context Of Use» comporte des informations sur l'utilisateur, la plate-forme et l'environnement. Les éléments du modèle de tâches («Task», «Decoration», «Operator», «Concept») peuvent être estampillés du contexte d'usage qu'ils requièrent. Ainsi, peut-on spécifier que la tâche « poser une question » nécessite un ordinateur connecté à Internet, ou que cette tâche devient optionnelle sur PDA dès lors que l'utilisateur est au travail.

Le métamodèle de la planification est conforme à la BNF de JSHOP. Il couvre les notions classiques de domaine, d'actions, etc. Nous constatons déjà que la correspondance entre communautés est loin d'être immédiate :

- Divergences terminologiques : la notion d'opérateur, par exemple, est un faux ami. En IHM, il dénote les relations logiques ou temporelles entre tâches. En planification, il s'apparente à un prédicat (par exemple, allumer(X) où X est une plate-forme) ;
- Divergences sémantiques : l'opérateur OU, par exemple, en IHM est exclusif. Il signifie que l'utilisateur a le choix (par exemple, se déclarer présent ou absent à une conférence). En planification, c'est le système qui tranche entre les deux alternatives ;
- Absence de correspondants : c'est le cas, par exemple, de l'opérateur temporel « entrelacement » sans équivalent en planification où le seul opérateur temporel est la séquence. C'est également le cas de la décoration itération (l'utilisateur peut refaire la tâche) non intégrée en planification ;
- Préoccupations différentes : dans son calcul, le planificateur connaît les tâches utilisateur abstraites mais ne les fournit pas en sortie. Il restitue une séquence

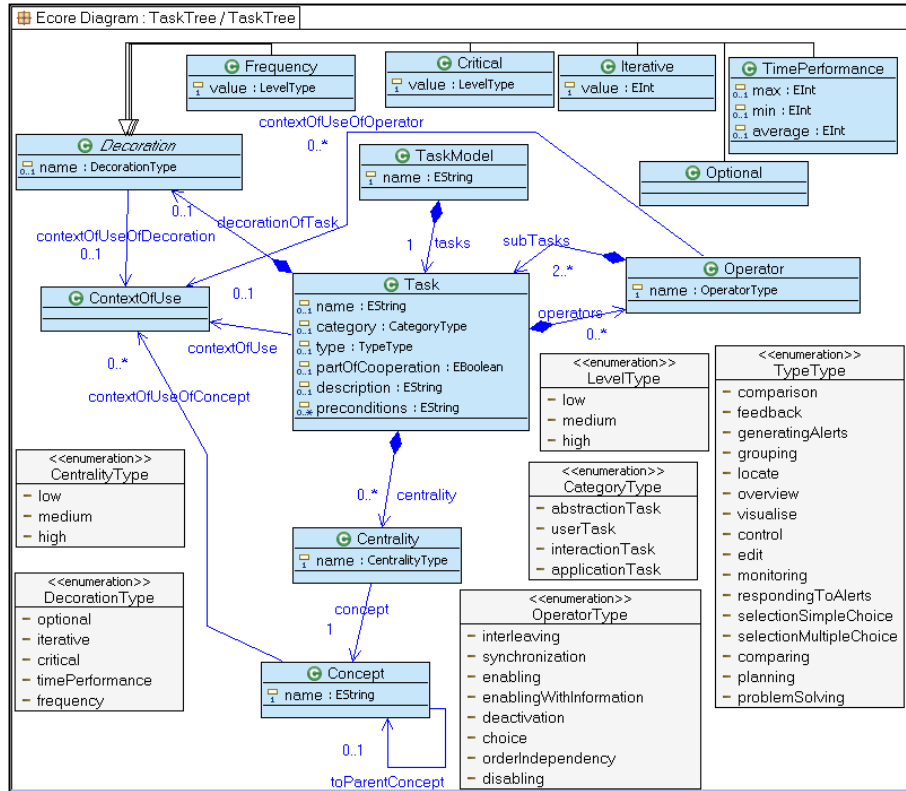


Figure 5 : Mon métamodèle de tâches pour une prise en compte du contexte d'usage.

d'actions physiques, les tâches abstraites devront être recalculées.

CONCLUSION ET PERSPECTIVES

L'article propose un parcours en largeur de mon approche pour la composition dynamique d'IHM. Elle est illustrée sur un premier prototype MAPPING dans lequel les efforts ont jusqu'ici été portés sur l'alignement des communautés IHM et planification. L'expérience montre que l'effort de mise en correspondance est loin d'être stérile. Il force à une meilleure compréhension en sémantique et complétude de chaque domaine. En perspective, je vise un travail « en profondeur » sur les descriptions et l'algorithme. Un point dur sera l'intégration des propriétés d'utilisabilité dans les langages et le processus de composition. MAPPING v2 fera l'objet d'évaluation pour comprendre le degré d'observabilité et de contrôle à donner à l'utilisateur dans le processus de composition.

REMERCIEMENTS

Je remercie la Région Rhône-Alpes, via le cluster ISLE, projet PRESENCE pour le financement de ma thèse.

BIBLIOGRAPHIE

1. Coutaz, J. Architectural Design for User Interfaces, The Encyclopedia of Software Engineering, J. Marciniak Ed., Wiley & Sons Publ., 2nde édition, 2001.
2. Florins, M. Graceful Degradation: A Method for De-

signing Multiplatform Graphical User Interfaces, Thèse de doctorat, Université catholique de Louvain, 2006.

3. Ilghami, O., Nau, D.S. A general approach to synthesize problem-specific planners, Technical report CS-TR-4597, UMIACS-TR-2004-40, University of Maryland, October 2003.
4. Liu, H., Singh, P. Focusing on ConceptNet's natural language knowledge representation, Commonsense Reasoning in and over Natural Language Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2004), Wellington, New Zealand, September 22-24, Lecture Notes in Artificial Intelligence, 2004.
5. Nau, D., Ghallab, M., Traverso, P. Automated Planning: Theory and practice, Morgan Kaufmann Publishers Inc., 2004.
6. Scapin, D.L., Bastien, J.M.C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems, Behaviour & Information Technology, 6(4-5), 1997, pp 220-231.
7. Sottet, J.S., Calvary, G., Favre, J.M. Towards mapping and model transformation for consistency of Plastic User Interfaces, Computer Human Interaction, Workshop on The Many Faces of Consistency in Cross-platform Design, 2006.