La programmation sur exemple pour l'automatisation des tests d'interfaces

Loé Sanou, Laurent Guittet, Sybille Caffiau

La Programmation sur Exemple pour l'Automatisation des Tests d'Interface Homme Machine

Loé SANOU – Laurent GUITTET – Sybille CAFFIAU

LISI / ENSMA Téléport 2 – 1 avenue Clément Ader BP 40109
86961 Futuroscope, France
{sanou, caffiaus, guittet}@ensma.fr

RESUME

La PsE est de plus en plus utilisée dans les applications interactives. L'utilisation de ses techniques d'enregistrement et de rejeu s'avère prometteuse pour la vérification et la validation des applications interactives. Une des utilisations est l'incorporation de ce principe dans la génération de scénarii pour les tests de comformité d'une IHM à son modèle de tâches prescrite.

MOTS CLES : Programmation sur Exemple, Interface homme machine, Test d'interface, Modèle de tâches.

ABSTRACT

The PsE is more and more used in the interactive applications. The use of recording and replaying techniques of the PsE turns out promising for the check and the validation of the interactive applications. One uses is the incorporation of this principle in the generation of scénarii for testing an IHM conformity to its task model.

KEYWORDS : Programming by Demonstration, Human Computer Interface, Interface testing, Task model.

INTRODUCTION

Les logiciels ont globalement évolué vers une augmentation des fonctionnalités, surtout avec l'avénément des interfaces graphiques. Cette course en avant sur les fonctionnalités a aboutit à imposer des interfaces plus lourdes. La recherche de solutions à ce problème constitue le point de départ des travaux sur le « End-User programming ». Plusieurs solutions ont été proposé : les préférences, les scripts, la personnalisation. Ces techniques présentent l'inconvénient de manquer de puissance d'expression. Des techniques d'enregistrement de macros ont alors vu le jour, tout particulièrement dans le domaine des systèmes iconiques [1]. L'une des dernières est la « Programmation Visuelle ». Cependant, une compréhension des principes de la programmation est là encore nécessaire. Au-delà de la simple utilisation de l'image, c'est sur celle de l'exemple que se sont concentrés les travaux. L'idée générale est que tout raisonnement autour de l'exemple est plus intuitif qu'un raisonnement abstrait. C'est ainsi qu'ont vu le jour les systèmes « sur exemple » [4]. La programmation sur exemple (PsE) a été expérimentée dans des domaines très variés. Certains champs d'application ont exploité ses fondements avec beaucoup de réussite, comme la conception technique par exemple. A ces champs d'application en expension, s'ajoute l'automatisation des tests d'IHM.

DOMAINES D'APPLICATION DE LA PSE

L'un des premiers systèmes relevant de la PsE est Pygmalion dont l'objectif est de permettre la construction d'un programme en s'appuyant sur un exemple. Le domaine des macros est un domaine privilégié de la programmation par l'utilisateur final. SmallStar [3] est une des premières tentatives pour adapter une technique graphique de PsE à cette problématique. C'est le premier système de PsE destiné au grand public dans le but d'automatiser les tâches répétitives. Le domaine de conception technique (incluant en particulier la CAO) a pleinement assimilé les techniques de la PsE, au point qu'aujourd'hui, les systèmes dits paramétriques, les plus utilisés au plan industriel, fondent leur modélisation sur les principes de la PsE. EBP [4] est un exemple d'environnement de développement de programmes paramétrés, utilisant des techniques de PsE permettant la création de composants standards portables par des utilisateurs de systèmes de CAO. Un autre domaine est celui portant sur l'enseignement, l'éducation et la simulation. Fournir une assistance, par l'exemple, à la compréhension des mécanismes de programmation est un thème important dans la PsE. Un environnement d'initiation à la programmation et basé sur l'exemple est MELBA [2] dont la cohérence est assurée par des mécanismes de PsE. Enfin, l'assistance à l'utilisateur est un autre domaine emblématique de la PsE, bien représenté par Eager [4] qui permet d'automatiser les tâches répétitives. Il observe en permanence le comportement de l'utilisateur pour essayer de prédire ses prochaines commandes.

CHAMPS D'APPLICATION DE LA PSE

La PsE a donné lieu à de nombreux systèmes utilisés dans des champs d'application divers :

- **Fonction d'assistance.** De nombreux éditeurs de logiciels ont ainsi développé des solutions s'appuyant sur les techniques de la PsE pour enrichir les possibilités d'adaptation considéré comme un besoin majeur par les utilisateurs. Les techniques de la PsE, utilisées de façon ponctuelle dans certaines situations d'interaction, sont de nature à apporter de nouvelles fonctions d'assistance à l'utilisateur.

- **Pédagogie.** L'objectif des systèmes utilisant la pédagogie ou la simulation est très varié. Ces systèmes peuvent concerner l'apprentissage de la programmation ou plus généralement de la logique. L'effort principal se porte dans ces systèmes sur l'aide à l'explicitation des princi-

pes de généralisation. Des projets avancés sont actuellement en cours (LegoSheets[1]).

- **Outils de conception et de validation.** C'est certainement là où la PsE est intégrée le plus. Les outils graphiques modernes utilisent l'analyse temps-réel des interactions pour fournir une assistance au placement des composants (alignement, etc.). On trouve ces fonctionnalités dans des outils de présentation (Keynote[2]) ou encore les GUI-Builders.

- **Tests d'interface.** L'utilisation des techniques d'enregistrement-rejeu de la PsE s'avère extrêmement prometteuse pour la vérification et la validation des applications interactives. Alors que les techniques de tests unitaires font de plus en plus d'émules[3], et que leurs outils sont de plus en plus utilisés, tester les interfaces s'avère encore un processus difficile. Aujourd'hui, ce sont les approches intégrant la PsE qui semblent émerger. Ainsi, Jacareto[4] permet d'enregistrer les interactions de bas-niveau et autorise-t-il l'automatisation de tests d'interface. Pour rendre ces outils plus puissants, un couplage avec les approches à base d'analyse de tâches, comme dans [6], s'avère très efficace.

AUTOMATISATION DES TESTS A L'AIDE DE LA PSE

La validation des systèmes interactifs est et demeure un problème difficile. Au-delà des propriétés ergonomiques classiques, le point crucial consiste à établir la conformité de l'application livrée avec les besoins du client. À la suite de Norman, de nombreux travaux ont été menés dans le domaine de l'analyse de l'activité, autour du concept de modélisation des tâches. Plusieurs notations ont ainsi été développées. Diverses tentatives pour incorporer une sémantique précise ont été menées, avec par exemple les travaux sur CTT[5], ou plus récemment sur K-MAD[6]. Ces approches permettent d'envisager une utilisation plus rigoureuse de ces outils. Cependant, utiliser les modèles de tâches pour aider à concevoir les applications est paradoxalement resté relativement peu développé. Quelques approches ont tenté de générer le contrôle des applications à partir des modèles de tâches, comme TERESA par exemple. Malheureusement, la méthode utilisée ne garantit pas le respect des propriétés lors des transformations. Pourtant, si l'on établit un lien entre l'application réalisée et le modèle de tâches, il serait possible de vérifier des propriétés importantes comme la conformité de l'application aux tâches prescrites. C'est à ce problème que la PsE apporte un solutionnement par son principe d'enregistrement-rejeu [5].

On part de l'idée que la vérification de la conformité d'une application à son modèle de tâches prescrites était une activité qui, si elle ne pouvait s'appuyer sur une preuve formelle, pouvait aisément relever du domaine du test. On envigeage de réaliser des tests dont les résultats pourront être confrontés au modèle de tâches. Ces tests sont enregistrés, pour éventuellement servir dans une approche de non-régression. En s'appuyant sur l'un des principes de la PsE , l'enregistrement-rejeu, qui permet l'incorporation aisée de fonctionnalités d'espionnage et de rejeu des interactions dans les applications interactives, on définit les bases d'une méthode constructive de test basée sur l'enregistrement des interactions de l'utilisateur. Ainsi, un développeur peut outiller de façon simple son application interactive pour permettre de vérifier la conformité de l'IHM avec un modèle de tâches. À partir de l'interface applicative, en exécution, des scénarii sont générés. Ces scénarii peuvent ensuite être chargés dans l'environnement de simulation pour être testés. Au final, des réponses claires sont fournies sur la conformité de l'application par rapport à son modèle de tâches.

CONCLUSION

La programmation sur Exemple permet de relier une application concrète avec un modèle de tâche censé la représenter à partir de sa technique d'enregistrement-rejeu. La vérification de la conformité de l'application au modèle de tâche permet de montrer de façon indubitable des erreurs de conception. Afin d'obtenir un outil permettant d'effectuer des tests de non-régression, il faudrait être en mesure de rejouer un scénario validé sur l'application ; l'utilisation de la programmation sur exemple pour construire le scénario entrouvre de nombreuses portes. Enfin, il convient de déterminer une méthode pour obtenir d'autres résultats que la conformité aux tâches, comme la complétude des tâches ou la simulation simultanée dans un environnement comme K-MADe par exemple lors de l'exécution de l'application.

BIBLIOGRAPHIE

1. Glinert, E.P. *Visual programming environments: paradigms and systems*.1990, 660 pages
2. Guibert N, Girard. P, Guittet L. *Example-based Programming: a pertinent visual approach for learning to program*. In Proceedings of *Advanced Visual Interfaces* 2004, Gallipoli, Italy, acm press,
3. Halbert, D. *SmallStar : Programming by Demonstration in the Desktop Metaphor*. Watch What I Do : Programming by Demonstration, 1993, pp 102-123.
4. Lieberman, H. *Your Wish is my command*. Morgan Kaufmann, 2001, 416 pages.
5. Sanou L. *Validation directe de la conformité d'une application interactive avec son modèle de tâches*. In Proceedings of l'IHM 2007, Paris, France, AFIHM, pp 249-252
6. Tarby, J.-C. *Evaluation précoce et conception orientée évaluation*. In Proceedings of Ergo'IA 2006 2006, Bidart/Biarritz France, pp 343-346

[1] http://l3d.cs.colorado.edu/systems/legosheets/
[2] http://www.apple.com/iwork/keynote/
[3] http://www.extremeprogramming.org/, http://www.junit.org/
[4] http://jacareto.sourceforge.net/
[5] http://giove.isti.cnr.it/CTTE/
[6] http://www-rocq.inria.fr/merlin/kmade/