École Nationale Supérieure de Mécanique et d 'Aérotechnique Université de Poitiers

Laboratoire d 'Informatique Scientifique et Industrielle

ENSMA - Téléport 2-1 Avenue Clément Adre - BP 40109 - 80961 FUTUROSCOPE CHASSENEUIL Cedex - France
Tél : (+33/0) 5 49 49 80 63 - Fax : (+33/0) 5 49 49 80 64 - E_mail : rault@ensma.fr - web : http://www.lisi.ensma.fr

Introduire des techniques de Programmation sur Exemple dans une boîte à outils : une étude de besoin

État de l 'art

La Programmation sur Exemple (PbD) requiert des techniques de programmation approfondies (différences entre variables et constantes, structures de contrôles, nomination des objets, niveau d'enregistrement des actions, …). L'intégration des mécanismes de PbD nécessite le plus souvent une recomposition de l'application ciblée afin de pouvoir capturer les actions utilisateur à un niveau proche du noyau fonctionnel.

Il n'existe pas de plate-forme robuste (système de création ou de gestion de programmes, architecture déployée) pour la construction de système de PbD. On ne dispose pas d'outils spécifiques permettant la réalisation ou la réimplantation de ces systèmes.

Deux types de solutions existent actuellement

développés selon des approches radicalement différentes :

- une **approche interne** (intégrer la PbD au cœur de l'application pendant sa construction) : AIDE Project

- une **approche externe** (espionner une application déjà développée) faisant le choix de l'établissement d'hypothèses préalables sur les noyaux fonctionnels ou ne considérer que des interactions de bas niveau : PbDScript.

Auteurs :
- **Loé SANOU** (LISI/ENSMA) loe.sanou@ensma.fr
- **Patrick GIRARD** (LISI/ENSMA)
- **Laurent GUITTET** (LISI/ENSMA)

Notre contribution

La présente contribution est une première étape vers la réalisation d 'une telle plate-forme, susceptible de mettre la technique de la PbD à la portée de tous programmeurs.

- ✓ Classification des systèmes de PbD,
- ✓ Définir les principaux besoins,
- ✓ Appui sur l'exemple du système Eager[Cypher 91] pour fournir les besoins d 'une boît à outils.

Trois classes de systèmes (en fonction de l'utilisation de la PbD)

– **L'assistance** : automatiser des tâches utilisateurs.

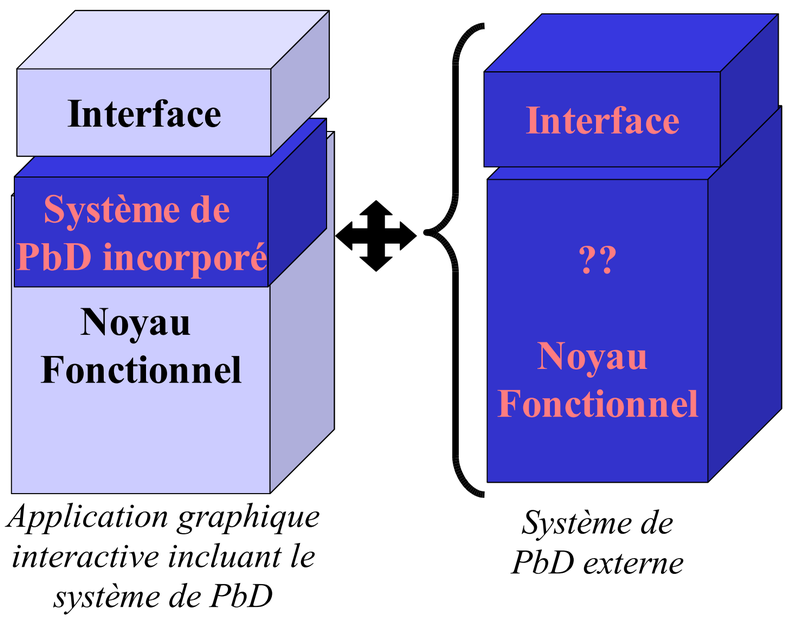
– **Les outils de conception** : produire un résultat qui découle directement de l'utilisation de la PbD.

– **L'apprentissage de la programmation** : apprendre à programmer.

2 systèmes non classifiés car leur objectif est de construire des applications de PbD : The AIDE Project [Pernot 1993] et PbDScript [Christophe 2003]

Ces 2 systèmes peuvent constituer une 4éme catégorie : **les outils de PbD**

Vision architecturale



Application graphique interactive incluant le système de PbD

Système de PbD externe

BESOINS POUR UNE BOITES A OUTILS (BO)

Du point de vue du programmeur : recenser et minimiser les actions de programmation explicite nécessaire pour construire ses fonctionnalités de PbD.

Une application de PbD devra permettre l'enregistrement et le rejeu des interactions de l'utilisateur, auxquelles s'ajoutent des capacités de généralisation.

Opérations de base : l'enregistrement et le rejeu.

Les service à fournir
Les bases de la technique
Les opérations de bases

Enregistrement ***Rejeu*** ***Généralisation***

Enregistrement

S'appuie sur la définition à priori de commandes (en tant que design pattern) tout en s'abstrayant des tâches articulatoires constituant une part importante du dialogue homme-machine.

Deux types d'actions à espionner : commandes explicites programmées par le programmeur (exemple : clic sur un bouton) et commandes implicites généralement implantées par défaut dans les widgets (exemple : utilisation de menu ou de raccourcis clavier).

Rejeu

La BO doit permettre deux modes de fonctionnement : ré-activation des interactions de l 'utilisateur en les mettant en évidence et aussi la non nécessité de rejouer très exactement ces interactions lorsque la tâche automatisée doit être complétée.

Principe de réalisation à base de Swing :

Nous développons actuellement un prototype de boîte à outils sur la base de Swing.

L 'utilisation de l 'abonnement est faite pour satisfaire le besoin d 'espionnage des commandes explicites.

Sous-classer les widgets de la BO support serait nécessaire pour gérer les actions implicites. Contrôler finement le rendu des widgets pour ajouter les primitives nécessaires.

Web :
- http://www.lisi.ensma.fr/members/sanou/
- http://www.lisi.ensma.fr/ihm/