Programmation d'applications sur PDA

l'exemple de Waba

Introduction

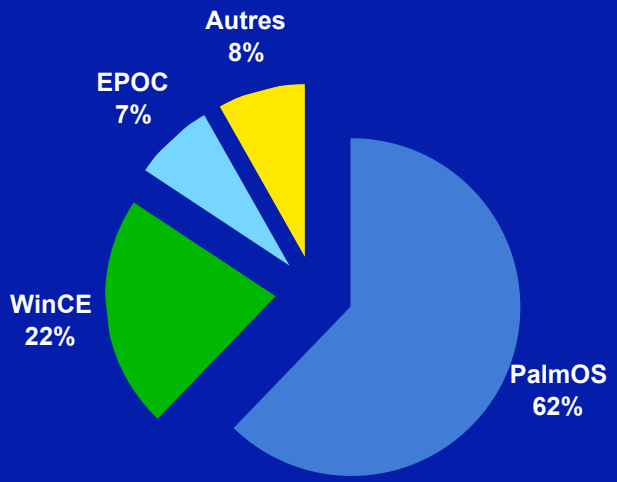
Introduction

- Généralisation des PDAs
- Utilisation spécifique des PDAs
- Projet originel de Java
- Présentation de Waba

Introduction (1) - Présentation

- Généralisation des PDAs
  - 1989: Portfolio d'Atari (MS-DOS)
  - 1992: John Sculley définit le PDA
  - 1993: Apple Newton (NewtonOS)
  - Mars 1996: Palm Pilot 1000 (PalmOS)
  - Novembre 1996: Microsoft lance Windows CE
  - 1997: Symbian lance EPOC
  - Aujourd'hui: environ 15 millions d'utilisateurs

Introduction (2) - Présentation



Introduction (3) - Présentation

- Utilisation spécifique des PDAs
  - Affichage et traitement de données
  - Échange avec les ordinateurs de bureau
    - HotSync de Palm
    - ActiveSync de WinCE
  - PDA = « Portable Data Access » ?

Introduction (4) - Présentation

- Projet originel de Java
  - Langage pour les clients dans une architecture avec des gros serveurs: adapté en théorie pour les machines peu puissantes
  - Langage orienté objet et applicatif (possibilité de Frameworks)

Introduction (5) - Présentation

- Waba
  - Une machine virtuelle Java (enfin, presque)
  - Des APIs spécifiques
  - Utilisation des compilateurs Java habituels
  - Un langage et un modèle pour la programmation d'applications pour PDA

Introduction (6) - Plan

- Waba: un paradigme de la programmation sur PDA
- Programmer une application pour PDA avec Waba (avec un exemple)
- Limites et ouvertures: la machine virtuelle et le projet Waba

Introduction (7) - Objectifs

- Objectifs théoriques
  - Spécificité de la programmation sur PDA (développement croisé, limites des PDAs)
  - Spécificité des applications sur PDA (modèle événementiel, interfaces graphiques, stockage, communication)

Introduction (8) - Objectifs

- Objectifs pratiques
  - Le langage Waba (comme sous-ensemble du langage Java)
  - Le modèle événementiel appliqué à Waba
  - Présentation générale des APIs Waba
  - Pourquoi développer (et ne pas développer) avec Waba

Waba: paradigme de la programmation sur PDA

Waba comme paradigme

- Le langage Java/Waba
- Les APIs Java et les APIs Waba
- Le problème de la mémoire

Le langage Waba/Java (1)

- Les différents langages de programmation sur PDA
  - Palm (C, C++, Java, …)
  - Newton (C++, NewtonScript, Java, Basic, …)
  - WinCE (C++, Basic, Java, …)

Le langage Waba/Java (2)

- Les machines virtuelles Java
  - Waba (Newton, Palm, WinCE, MS-DOS)
  - KVM (Palm) & PersonalJava (WinCE) de Sun
  - IBM J9 VM (Palm)
  - Kada VM (Palm)
  - Jeode (Zaurus)

Le langage Waba/Java (3)

- Les avantages de Java pour les PDAs
  - Code aussi petit que possible
  - A priori pour des machines pas trop puissantes
  - Portabilité
  - Avantage d'un langage interprété pour les plateformes fragiles (e.g. Palm)

Le langage Waba/Java (4)

- Les inconvénients de Java pour les PDAs
  - Lenteur de Java (e.g. threads)
  - Gourmand en mémoire
- Les différences avec Waba
  - Absence de thread
  - APIs légères

Le langage Waba/Java (5)

- Le langage Waba: un sous-ensemble strict de Java
  - Même format pour les classes, les éléments inutiles (e.g. tables pour les exceptions) sont ignorés.
  - Les opcodes 0-201 (« standard ») sont supportés sauf ceux pour les threads, les longs, les doubles et les exceptions

Les APIs Java/Waba (1)

- L'exemple de l'interface graphique: complexité des APIs Java usuelles
  - Le cas des menus
  - La complexité des Layouts (idée que la dimension peut être variable)
  - Les listeners Java & surcharge des méthodes sur Waba

Les APIs Java/Waba (2)

- Les PDAs requièrent des APIs très spécifiques: les données sur PDA
  - Les catalogues: le stockage des données
  - Les conduits: l'échange de données

Les APIs Java/Waba (3)

- Le cas du son
  - Applications Java: Java Sound API (javax.sound.*) MIDI, échantillons
  - Applets Java: APIs documentées depuis la JDK 1.2
  - waba.fx.Sound (bips, impulsions à une fréquence donnée) & waba.fx.SoundClip (échantillons)

Le problème de la mémoire (1)

- Limitation essentielle du parc actuel des PDAs:
  - Palm: de 512 Ko (Personal) à 8 Mo (derniers modèles) pour l'exécution et le stockage
- Coût & avantage du ramasse miette
  - Programmation plus aisée
  - Peu de fuites dans les programmes Waba
  - Un peu plus coûteux en mémoire/temps

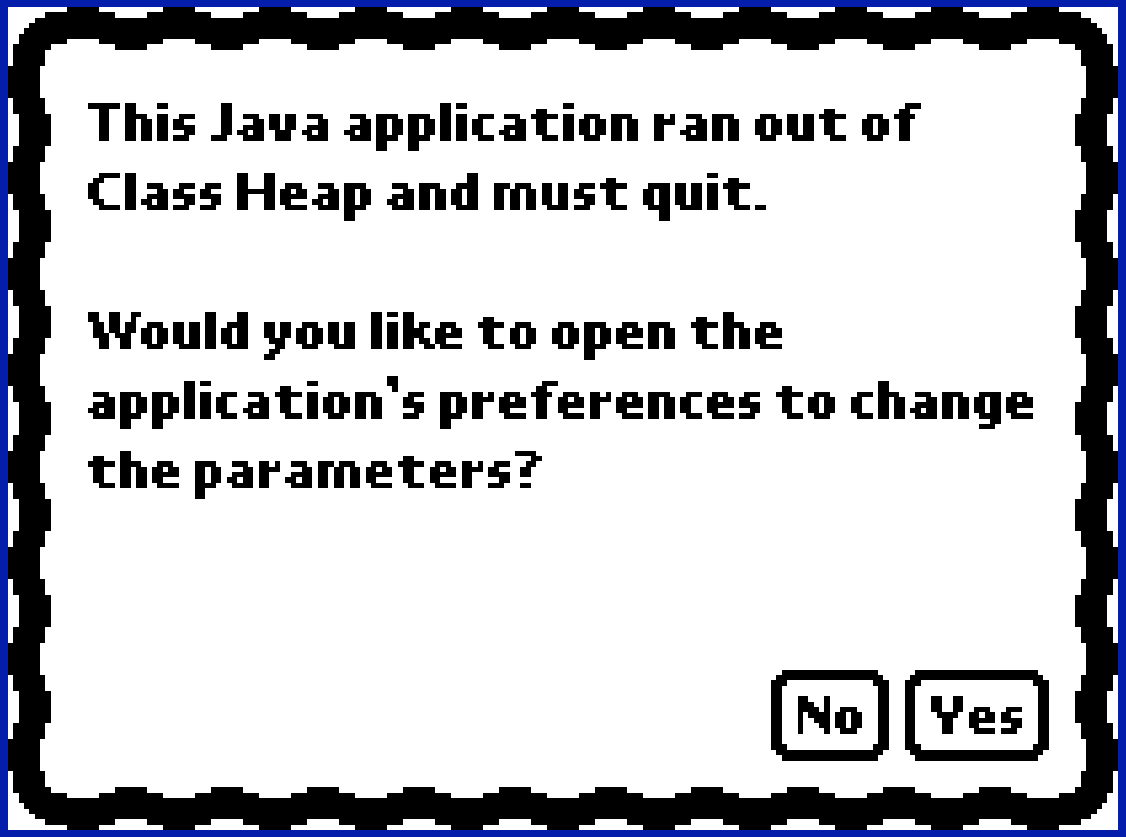
Le problème de la mémoire (2)

- La division de la mémoire de Waba en différentes zones
  - Object heap:mémoire pour les objets Java, RM
  - Class heap: table pour les classes, pas de RM
  - Pile Java (1 Ko)
  - Pile Machine virtuelle (300 octets)

Le problème de la mémoire (3)

- Waba alloue les quatre zones au démarrage du programme
- La quantité de mémoire est spécifiée à la compilation et ne peut être changée sur le PDA (sauf Newton)
- Les quatre zones sont libérées à la fin du programme: pas de fuite

Le problème de la mémoire (4)



Le problème de la mémoire (5)

- Malgré cette contrainte, Waba répond assez bien aux impératifs des PDAs:
  - Machine virtuelle très petite (60 Ko)
  - Programmes très petits (byte code Java)
  - Programmes peu gourmands en mémoire

Waba comme paradigme

- Langage & MV adaptés aux contraintes de la programmation sur PDA
  - Taille du code
  - Utilisation de la mémoire
- APIs adaptées au développement d'applications pour PDA

Programmer une application pour PDA avec Waba

Un exemple: logiciel de base de données pour une bibliothèque personnelle

Programmer une application

- Les stages de développement
- L'interface graphique
- Le modèle événementiel
- Le stockage et l'échange des données

Les stages de développement (1)

- Le développement croisé, typique sur PDAs
  1. Édition
  2. Compilation
  3. Correction des erreurs à la compilation
  4. Retour à l'étape 1
  5. Génération du paquet
  6. Installation
  7. Test
  8. Retour à l'étape 1

Les stages de développement (2)

- Le développement croisé avec Waba
  - Edition: n'importe quel éditeur (e.g. vim, emacs, CodeWarrior)
  - Compilation: javac, jikes ou autre
  - Génération du paquet
    - Warp (.exe ou classe Java)
    - Exegen (idem)
  - Installation: dépend du PDA

Les stages de développement (3)

- Utilisation de l'émulateur (Palm, WinCE)
  - Permet de réduire le temps d'installation et de test
  - Permet de tester sur plusieurs PDAs sans trop de difficultés
  - ROMs Palm accessibles aux développeurs enregistrés

Les stages de développement (4)

- La Waba SDK
  - APIs Waba émulées avec AWT
  - Non complètes mais en source libre
  - Classes requises pour la compilation

Les stages de développement (5)

- Avantages de la Waba SDK
  - Permet l'exécution du programme sur l'ordinateur de développement
  - Le programme est appelé comme le compilateur/le générateur de paquet
    - java waba.applet.Applet Biblio
  - Exécution dans un butineur
  - Programmes utilisables à la fois sur PDA et ordinateurs de bureau

Les stages de développement (6)

- VisualWaba de DMIC
  - Utilisation par glisser-déposer
  - Ecrit en Java et fonctionne sur toute plateforme
  - Logiciel gratuit, support technique payant
  - Un peu bogué

Les stages de développement (7)

- Inconvénients de Waba:
  - Pas de cruci-dévermineur
  - System.out.println n'existe pas (mais il y a des équivalents)
- La WabaSDK corrige ces problèmes, mais:
  - Elle ne donne pas une bonne idée de la gestion de la mémoire
  - Elle est incomplète

L'interface graphique (1)

- Biblio version 1: Hello World
  - Une fenêtre principale (et unique) pour l'application: MainWindow
  - La structure de Waba: des composants (classe Control et sous-classes) et des conteneurs (classe Container)
  - Un élément de base: les étiquettes (Label)

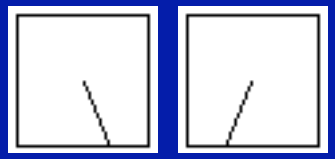
L’interface graphique (2)

- L’ajout d’éléments se fait en général en précisant les coordonnées
- Pas de Layout, mais:
  - RelativeContainer (précise la position relative d’un objet par rapport au précédent)
  - GridContainer

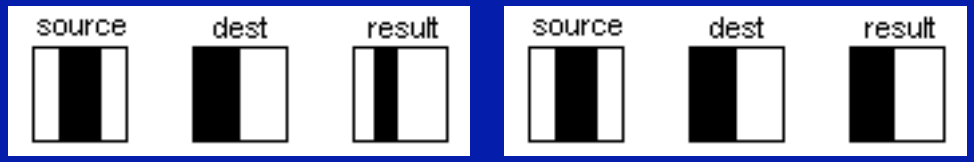
L'interface graphique (3)

- Dessin
  - Méthode onPaint appelée par la machine virtuelle
  - Texte, lignes, couleur, images, etc.
  - Mais bogues de PalmOS

Clip



AND



Le modèle événementiel (1)

- Boucle des événements gérée par l'OS
  - Elément essentiel des applications sur PDA
  - Permet des économies d'énergie
  - Incite à la coopération entre les applications

Le modèle événementiel (2)

- Avec Waba
  - Méthode onEvent à surcharger
  - Plusieurs types d'événements prédéfinis:
    - PenEvent & KeyEvent (viennent de la machine virtuelle)
    - ControlEvent: entre les objets Waba
  - Possibilité d'ajouter des événements

Le modèle événementiel (3)

- Synthèse: Biblio version 2
  - Composants: titre & menus
  - Dessin: boîte à propos
  - Gestion des événements: réponse à un menu

Le stockage des données (1)

- Les catalogues
  - Spécificité des PDAs: un catalogue par type de données/application
  - Synchronisation avec les ordinateurs de bureau
  - Accès depuis n'importe quelle application: intégration des données entre les programmes

Le stockage des données (2)

- Les catalogues dans Waba
  - Ouverture/création/suppression de catalogue
  - Ajout/suppression d'enregistrements
  - Accès octet par octet
  - Fermeture avec le ramasse-miettes

Le stockage des données (3)

- Biblio version 3: une petite base de données
  - L'interface Storable
  - La classe ObjectCatalog (et son extension ObjectCatalogX)
  - Limitation de l'occupation mémoire

L’échange des données

- Classes Socket et SerialPort
  - Lecture synchrone: grande simplicité
  - Difficultés pour le port série
    - Ajout de caractères sur Palm
    - Problèmes de l’accès synchrone sur Newton
  - Limitations pour les sockets TCP/IP
    - Synchrones: 1 seul échange de données à la fois
    - Pas de serveur possible

Limites et ouvertures: La machine virtuelle et le projet Waba

Limites et ouvertures

- L'empreinte de PalmOS sur Waba
- Une version trop restreinte de Java?
- Un projet Open Source: possibilités d'ouverture

L'empreinte de PalmOS (1)

- L'aspect de l'interface graphique
  - Les APIs Waba:
    - L'aspect visuel
    - Les menus à la Palm
  - La machine virtuelle:
    - La gestion des polices
    - La gestion des fenêtres
  - La reconnaissance d'écriture

L'empreinte de PalmOS (2)

- Le cas des catalogues
  - Une interface octets par octets
  - Conventions de nommage
  - Absence d'index
  - Aspect objet devant être inclus dans l'application (extra.ui.ObjectCatalog)
  - Non transactionnel

L'empreinte de PalmOS (3)

- Une limite de PalmOS: mono-application et mono-tâche
  - D'où aucune interaction avec les autres applications, pas de fenêtre, pas de threads
- Le cas des sockets TCP/IP et du port série
  - Bloquants
  - Gestion des noms des ports inexistante
  - Aucune gestion de la propriété des ports

Une version trop restreinte de Java? (1)

- Lacunes dans les classes standard
  - java.Object & java.String
  - Hiérarchie java.util.*

Une version trop restreinte de Java? (2)

- Absence des doubles
  - Pas de FPU sur Palm
  - Requièrent les longs
  - Pas de réel gain de vitesse ou de mémoire

Une version trop restreinte de Java? (3)

- Absence des exceptions
  - Choix de l'auteur
  - Simplifie grandement l'exécution (une seule pile)
  - Mais pas de réel gain de performance/mémoire
  - Ampute la programmation java
  - Une des fonctions les plus demandées

Une version trop restreinte de Java? (4)

- Absence des threads
  - Compromis pour la performance: le mauvais exemple de KVM
  - Héritage de PalmOS
  - Conflit avec la gestion des ports série/sockets

Un projet Open Source (1)

- Licences open source
  - GPL (pour Waba)
  - LGPL (pour SuperWaba)
  - BSD & IBM PL (pour la version Newton)
- Projet désormais sur SourceForge

Un projet Open Source (2)

- Possibilités de fonctions natives
  - Permettent de rajouter des APIs particulières
  - Permettent des traitements plus rapides
  - Mais besoin de coordination

Un projet Open Source (3)

- Les machines virtuelles alternatives (1)
  - Waba de Rick Wild
    - PalmPilot et WinCE
  - SuperWaba (Guilherme Campos et alii)
    - Palm et WinCE
    - Beaucoup plus rapide
    - APIs supplémentaires (très orientées Palm)
    - Longs et doubles
    - Meilleure gestion de l'unicode

Un projet Open Source (4)

- Les machines virtuelles alternatives (2)
  - Isao's WabaVM
    - Palm
    - Couleur avant le projet original
    - Multitâche coopératif (bancal)
  - Newton Waba (Sean Luke et alii)
    - APIs Newton
    - Gestion de la mémoire modifiable par l'utilisateur

Un projet Open Source (5)

- Les machines virtuelles alternatives (3)
  - WabaCE de Michael Brereton
  - Waba pour DOS
  - Waba pour TI
  - Waba pour iPaq

Un projet Open Source (6)

- Classes supplémentaires (1)
  - Waba extras de Rob Nielsen (désormais standard)
    - Désormais standard
    - Offrent un complément d'interface (e.g. titre & menus à la Palm)
    - ObjectCatalog & Storable
    - RelativeContainer & Container

Un projet Open Source (7)

- Classes supplémentaires (2)
  - MathFP
    - bibliothèque FPU comme celles de KVM mais sans les exceptions
  - ListBox
  - mWaba (TextAreas, etc.)
  - …

Limites et ouvertures

- Un projet très marqué par PalmOS
- Une version un peu trop réduite de Java (avec surtout le manque des exceptions)
- Mais un projet open source, d'où des possibilités d'évolution

Conclusion

Conclusion

- Programmation objet sur PDA
- Rapidité de programmation avec Waba
- Perspectives et alternatives

Conclusion (1)

- Programmation Java sur PDA
  - Par rapport au C (standard sur Palm): programmation par objet
  - Par rapport au C++: ramasse-miettes, taille du code
  - Bibliothèques de classes partagées (Newton seulement)

Conclusion (2)

- Rapidité de programmation avec Waba
  - Waba SDK: limite l'utilisation de l'émulateur pour Palm & WinCE (et pallie son manque sur d'autres plateformes)
  - Peu de risque de réinitialisation du PDA (surtout sur Palm)
  - Réutilisation du code avec les paquetages et facilitée par le modèle par événements

Conclusion (3)

- Perspectives
  - Création d'un framework multiplateforme
  - Extension de Waba à une version moins réduite de Java, possible avec la montée en puissance des PDAs
  - Utilisation avec les butineurs Web pour les applets (déjà le cas sur Newton avec Newtscape)

Conclusion (4)

- Alternatives
  - Généralisation de la KVM
  - Firmes plus enclines à suivre Sun ou IBM
- Un projet qui n'a que deux ans