

## Migration des interfaces utilisateurs vers les tables interactives

André Kalawa

Financement BDE: Région PACA/LudoTIC

13 décembre 2012

Motivations

Domaine d'étude

État de l'art des approches de migration

Modélisations

Mécanismes de migrations

Implémentation

Conclusions et perspectives

## Pourquoi migrer les applications existantes ?

- Refactoring des applications existantes
  - Réduire le temps et le **coût de développement**
  - Faciliter le travail des développeurs en garantissant le respect des **critères ergonomiques**

## Pourquoi migrer vers les tables interactives ?

- Nouveaux moyens d'interactions
  - Interactions tactiles et tangibles
- UI multi utilisateurs et travail collaboratif

## Pourquoi migrer les applications existantes ?

- Refactoring des applications existantes
  - Réduire le temps et le **coût de développement**
  - Faciliter le travail des développeurs en garantissant le respect des **critères ergonomiques**

## Pourquoi migrer vers les tables interactives ?

- Nouveaux moyens d'interactions
  - Interactions tactiles et tangibles
- UI multi utilisateurs et travail collaboratif

# Cas d'une application BD sur une table interactive

Une table interactive permet aux dessinateurs

- profiter d'un écran plus large et de nouveaux moyens d'interactions,
- avoir un nouvel espace de conception qui facilite la collaboration

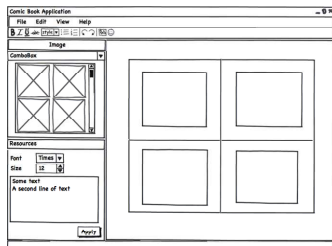


FIGURE : Fenêtre principale

# Cas d'une application BD sur une table interactive

Une table interactive permet aux dessinateurs

- profiter d'un écran plus large et de nouveaux moyens d'interactions,
- avoir un nouvel espace de conception qui facilite la collaboration

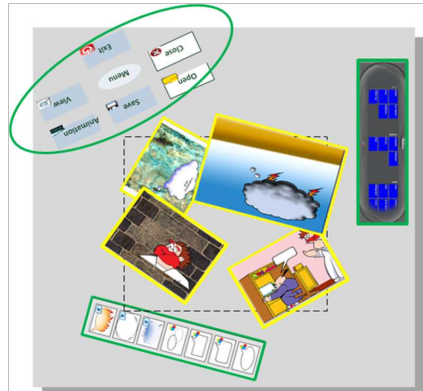
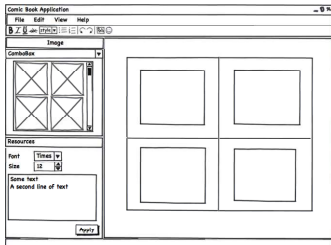


FIGURE : Fenêtre principale

# Périmètres de la migration

## Hypothèses :

Application = interface utilisateur(UI) + noyau fonctionnel(NF)

Considérons que le NF de l'application à migrer est réutilisé

⇒ Hétérogénéité des plateformes [TKB78]

## Pistes de migration d'UI :

1-Nouvelle conception de l'UI en prenant compte les principes de conception

Coûts de mise en œuvre

2-Déduire de l'UI à partir du NF [KKM03]

Non Respect des critères ergonomiques

3-Adapter les éléments de l'UI de départ par rapport à la cible

Réutilisation et Respect des critères ergonomiques

⇒ Comment adapter la Structure, le Positionnement (Layout), le Style, les

Interactions ?

# Périmètres de la migration

## Hypothèses :

Application = interface utilisateur(UI) + noyau fonctionnel(NF)

Considérons que le NF de l'application à migrer est réutilisé

⇒ Hétérogénéité des plateformes [TKB78]

## Pistes de migration d'UI :

1-Nouvelle conception de l'UI en prenant compte les principes de conception

**Coûts** de mise en œuvre

2-Déduire de l'UI à partir du NF (KKM03)

Non Respect des critères ergonomiques

3-Adapter les éléments de l'UI de départ par rapport à la cible

Réutilisation et Respect des critères ergonomiques

⇒ Comment adapter la Structure, le Positionnement (Layout), le Style, les

Interactions ?



# Périmètres de la migration

## Hypothèses :

Application = interface utilisateur(UI) + noyau fonctionnel(NF)

Considérons que le NF de l'application à migrer est réutilisé

⇒ Hétérogénéité des plateformes [TKB78]

## Pistes de migration d'UI :

1-Nouvelle conception de l'UI en prenant compte les principes de conception

Coûts de mise en œuvre

2-Déduire de l'UI à **partir du NF** [KKM03]

Non Respect des **critères ergonomiques**

3-Adapter les éléments de l'UI de départ par rapport à la cible

Réutilisation et Respect des **critères ergonomiques**

⇒ Comment adapter la Structure, le Positionnement (Layout), le Style, les

Interactions ?

# Périmètres de la migration

## Hypothèses :

Application = interface utilisateur(UI) + noyau fonctionnel(NF)

Considérons que le NF de l'application à migrer est réutilisé

⇒ Hétérogénéité des plateformes [TKB78]

## Pistes de migration d'UI :

1-Nouvelle conception de l'UI en prenant compte les principes de conception

Coûts de mise en œuvre

2-Déduire de l'UI à partir du NF [KKM03]

Non Respect des critères ergonomiques

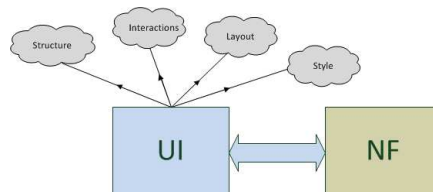
**3-Adapter** les éléments de l'UI de départ par rapport à la cible

Réutilisation et Respect des **critères ergonomiques**

⇒ Comment adapter la Structure, le Positionnement (Layout), le Style, les

Interactions ?

# Migration des UI sur des tables interactives



## Problèmes

- Quels principes de conception (guidelines) suivre pendant la migration ?  
⇒ Quels types d'UI pour tables interactives ?
- Comment adapter les différents aspects des UI à migrer ?  
⇒ Quelles approches pour la migration d'UI ?

# Tables interactives [Mit, UI97, Mic11]

Tables interactives	Utilisateurs	Instruments d'interactions	Type d'UI
<b>DiamondTouch</b> [SVFR04]	1 à 4 Utilisateurs	Ecran non Capacitif	UI Tactile, UI Collaborative (Identification d'utilisateurs)
<b>metaDesk</b> [UI97]	1 ou Plusieurs Utilisateurs	Écran, Camera Infrarouge, Objets Tangibles Uniquement	UI Tangible, UI Collaborative
<b>Microsoft PixenSense</b> [Mic09]	1 ou Plusieurs Utilisateurs (Limité par le nombre de points de contact)	Écran Capacitif, Reconnaissance de Tags et formes	UI Tactile, UI Collaborative, UI Tangible,

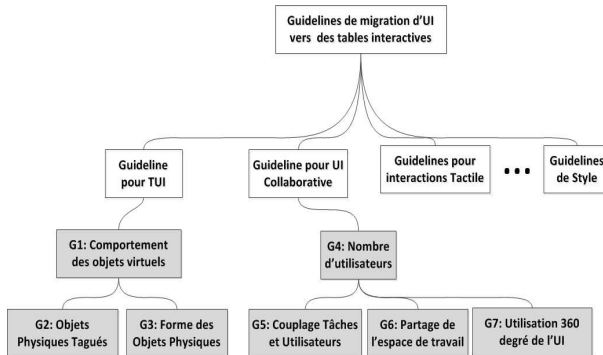
$$Type_{UI} = f(Utilisateurs, Instrument_{Interaction})$$

Utilisateurs (Nombre et Position)

Instruments d'interactions = {Dispositifs d'interactions, Bibliothèques Graphiques}

# Tables interactives [Mit, UI97, Mic11]

Corpus de guidelines [Mic11]



$$Type_{UI} = f(Utilisateurs, Instrument_{Interaction})$$

Utilisateurs (Nombre et Position)

Instruments d'interactions = {Dispositifs d'interactions, Bibliothèques Graphiques}

11 9

## Caractéristiques des approches de migration d'UI

$$ApprochesMigrationUI = \langle Source, Cible, Mécanismes \rangle$$

## Mécanismes

- Équivalences entre les instruments d'interactions
- Adaptation des aspects de l'UI source
- Prise en comptes des guidelines

⇒ Ces mécanismes peuvent être manuels, automatiques ou semi automatiques

# Caractéristiques des approches de migration d'UI

$ApprochesMigrationUI = \langle Source, Cible, Mécanismes \rangle$

## Mécanismes

- Équivalences entre les instruments d'interactions
- Adaptation des aspects de l'UI source
- Prise en comptes des guidelines

⇒ Ces mécanismes peuvent être manuels, automatiques ou semi automatiques

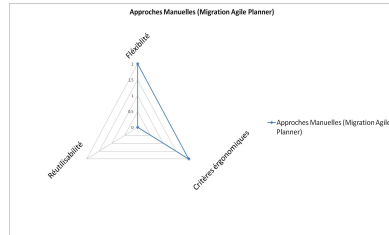
## Evaluer chaque approche en tenant compte de

- la **flexibilité** des mécanismes ou du degrés d'intervention du concepteur pendant la migration,
- la prise en compte des guidelines pour assurer une UI respectant les **critères ergonomiques**, [Van97]
- la **réutilisabilité** du mécanisme d'adaptation ou d'équivalence qui est garantie par son indépendance par rapport aux applications ou plateformes



# Migration de l'application AgilePlanner sur une table interactive [WGM08]

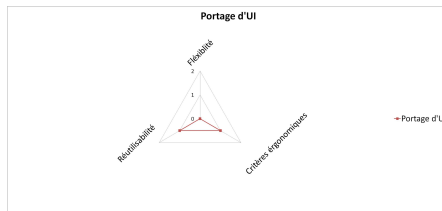
- Approche ad hoc et manuelle de migration d'une UI
- Processus complet de migration en plusieurs étapes
  1. Analyse
  2. Identification des guidelines
  3. Migration
  4. Évaluation



⇒ Mécanismes non réutilisables mais un résultat conforme aux attentes des utilisateurs finaux car flexibles

# Portage d'UI sur tables interactives [Bes10]

- Migration sans re conception d'UI source
- Approches spécifiques à des instruments d'interactions (Bibliothèques graphiques)



⇒ Mécanismes peu flexibles, réutilisables pour un ensemble applications et produisant des UI peu utilisables

# Services de migration d'UI [PSS09]

CRF [CCB<sup>+</sup>02] : Modèles pour la conception d'UI multi plateformes

- Tâches et concepts, AUI, CUI, FUI
- Implémentations : USIXML, MARIAXML

Mécanismes de migration automatique d'UI

Basés sur des modèles de conception

⇒ Modèles de l'UI source sont fournies avec  
l'application à migrer

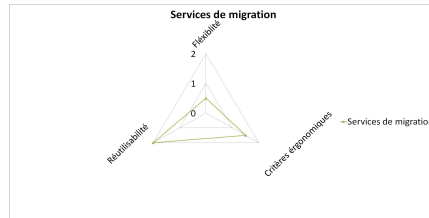
Basés sur une activité de Reverse  
Engineering(RE)

⇒ Modèles partiels de l'UI source sont  
retrouvés à partir de l'application à migrer

# Services de migration d'UI [PSS09]

CRF [CCB<sup>+</sup>02] : Modèles pour la conception d'UI multi plateformes

- Tâches et concepts, AUI, CUI, FUI
- Implémentations : USIXML, MARIAXML



⇒ Mécanismes réutilisables et génériques pour des plateformes et des applications mais produisant des UI peu conformes aux critères ergonomiques car moins flexibles

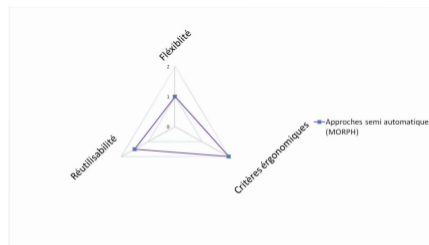
# MORPH [MR97]

## *Model Oriented Reengineering Process for HCI*

- Processus de migration en trois phases
  1. Détection
  2. Transformation
  3. Génération
- Approche basée sur des modèles abstraits (Structure, Tâches d'interactions)
  - Ajout manuel du layout et du style à l'UI migrée
- Mécanismes de prise en compte des guidelines peuvent être incorporés dans la base de connaissances
- Équivalences dynamiques basées sur un modèle de connaissances

# MORPH [MR97]

## *Model Oriented Reengineering Process for HCI*



⇒ Processus réutilisable, produisant des UI conformes aux critères ergonomiques et plus flexible que les services de migrations d'UI mais n'offrant pas une assistance aux concepteurs

# Synthèse

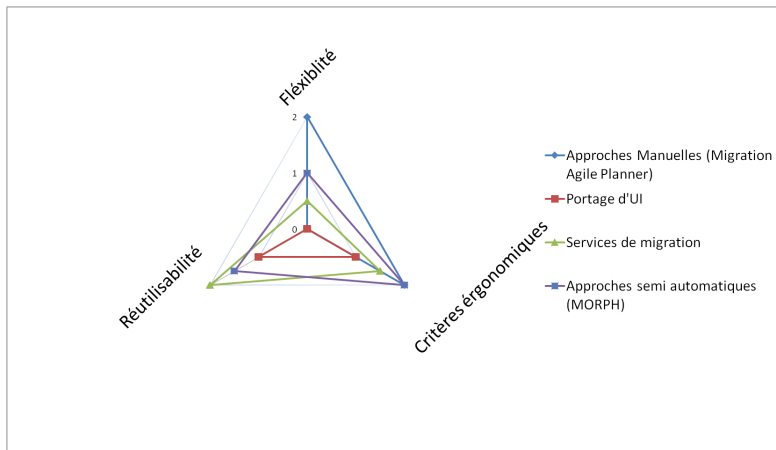


FIGURE : Synthèse des approches de migration d'UI

# Objectifs et Moyens

Migration des UI vers les tables interactives en respectant les critères ergonomiques de la cible

- Mécanismes de reverse engineering d'UI (réutilisables)
- Mécanismes d'aide à la migration d'UI plus flexible à moindre **coût**

## Modélisation

Proposer un modèle d'interactions abstraites qui permet

- la **flexibilité** de la prise en compte des guidelines pour décrire les équivalences

## Mécanismes de migrations d'UI vers les tables interactives

- prise en compte des guidelines pendant la migration pour le respect des **critères ergonomiques**
- formalisation de la prise en compte des guidelines pour plus de **réutilisabilité**



## Objectifs et Moyens

## Migration des UI vers les tables interactives en respectant les critères ergonomiques de la cible

- Mécanismes de reverse engineering d'UI (réutilisables)
- Mécanismes d'aide à la migration d'UI plus flexible à moindre coût

## Modélisation

Proposer un modèle d'interactions abstraites qui permet

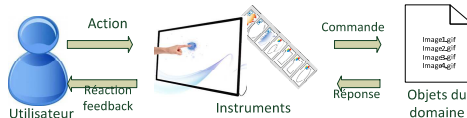
- la **flexibilité** de la prise en compte des guidelines pour décrire les équivalences

## Mécanismes de migrations d'UI vers les tables interactives

- prise en compte des guidelines pendant la migration pour le respect des **critères ergonomiques**
- formalisation de la prise en compte des guidelines pour plus de **réutilisabilité**

# Interactions

## Interactions



⇒ Exprimer les actions, les commandes, les réponses et les feedback de l'UI par des interactions atomiques sur les **composants graphiques** : **Primitives d'interactions**

# Interactions

## Primitives d'interactions

*ComposantGraphique* =

$\langle \textit{Propriétés graphiques}, \textit{Contenu}(s), \textit{Lien avec NF} \rangle$

### Propriétés graphiques

- Widget Move
- Widget Rotation
- Widget Resize
- Widget Selection,  
Navigation
- Widget Display

### Contenu(s)

- Data Edition
- Data Selection
- Data Move In
- Data Move Out
- Data Display

### Lien avec le NF

- Activation

## Interactions

## Primitives d'interactions

$$ComposantGraphique =$$

*⟨Propriétés graphiques, Contenu(s), Lien avec NF⟩*

## Propriétés graphiques

Contenu(s)

Lien avec le NF

- Widget Move
- Widget Rotation
- Widget Resize
- Widget Selection, Navigation
- Widget Display

- Data Edition
- Data Selection
- Data Move In
- Data Move Out
- Data Display

# Interactions

## Primitives d'interactions

*ComposantGraphique* =

$\langle \textit{Propriétés graphiques}, \textit{Contenu}(s), \textit{Lien avec NF} \rangle$

Propriétés graphiques

- Widget Move
- Widget Rotation
- Widget Resize
- Widget Selection,  
Navigation
- Widget Display

Contenu(s)

- Data Edition
- Data Selection
- Data Move In
- Data Move Out
- Data Display

Lien avec le NF

- Activation

# Primitives d'interactions et Composants graphiques

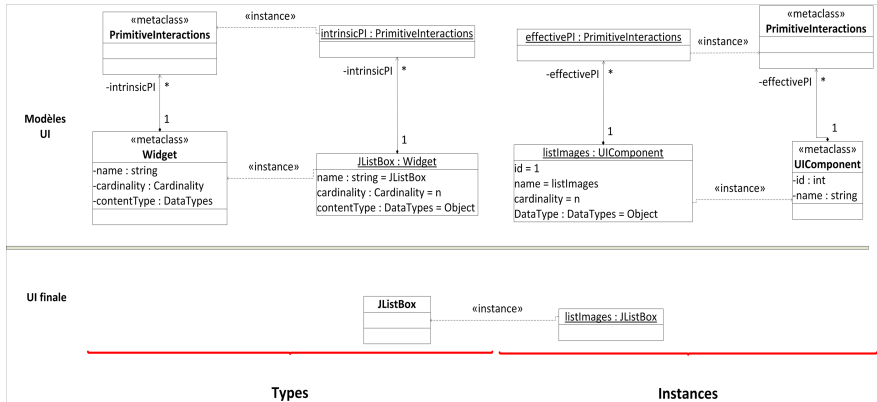


FIGURE : Modèles abstraits d'UI et UI finales

# Primitives d'interactions et Composants graphiques

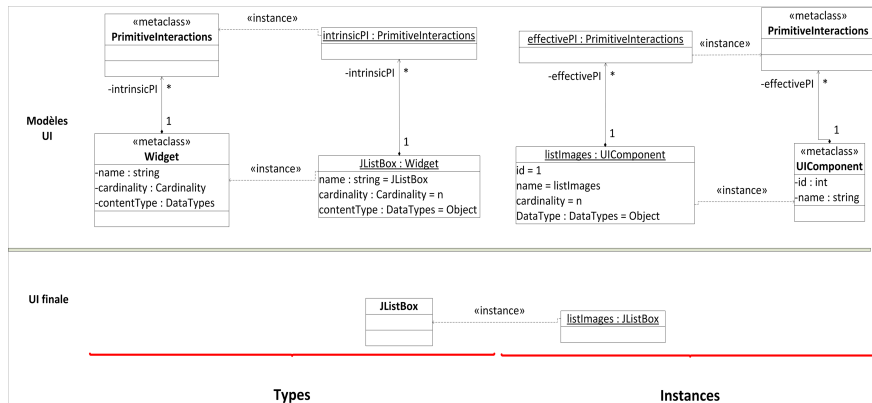
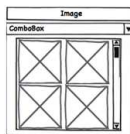


FIGURE : Modèles abstraits d'UI et UI finales

# Exemple de Primitives d'interactions

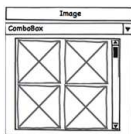


Composants graphiques	Primitives d'interactions
Liste d'images	Widget Selection, Navigation, Widget Display, Data Selection, Data Move Out, Activation Data Display
Liste déroulante	Widget Selection, Navigation, Data Selection, Data Display, Activation

- Les **primitives d'interactions** permettent de décrire les interactions entre UI et l'utilisateur **indépendamment des instruments d'interactions**
- Comment décrire les **équivalences des composants graphiques** avec les primitives d'interactions?



# Exemple de Primitives d'interactions



Composants graphiques	Primitives d'interactions
Liste d'images	Widget Selection, Navigation, Widget Display, Data Selection, Data Move Out, Activation Data Display
Liste déroulante	Widget Selection, Navigation, Data Selection, Data Display, Activation

- Les **primitives d'interactions** permettent de décrire les interactions entre UI et l'utilisateur **indépendamment des instruments d'interactions**
- Comment décrire les **équivalences des composants graphiques** avec les primitives d'interactions ?

## Équivalences des composants graphiques

$$ComposantGraphique = \langle Cardinalité, TypeDeDonnée, PrimitivesInteractions \rangle$$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

$$\left\{ \begin{array}{c} \text{JButton} \equiv \text{SurfaceButton} \\ \text{Navigation} \\ \text{WidgetSelection} \\ \text{Activation} \end{array} \right\} \equiv \left\{ \begin{array}{c} \text{Navigation} \\ \text{WidgetSelection} \\ \text{Activation} \end{array} \right\}$$

# Équivalences des composants graphiques

$ComposantGraphique = \langle Cardinalité, TypeDeDonnée, PrimitivesInteractions \rangle$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

$$JFrame \cong DSFrame$$

$$\left\{ \begin{array}{l} Navigation \\ WidgetSelection \\ WidgetDisplay \\ WidgetMove \\ WidgetResize \\ Activation \end{array} \right\} \leq \left\{ \begin{array}{l} Navigation \\ WidgetSelection \\ WidgetDisplay \\ WidgetMove \\ WidgetResize \\ Activation \\ \textcolor{red}{WidgetRotation} \end{array} \right\}$$

Opérateurs d'équivalences basés sur des PI et les types de données

- $\cong$  Équivalence stricte des PI et type de données différentes
- $\leq$  Équivalence large des PI et type de données différentes
- $\geq$  Équivalence faible des PI et type de données différentes

# Équivalences des composants graphiques

*ComposantGraphique* =  $\langle \text{Cardinalité}, \text{TypeDeDonnée}, \text{PrimitivesInteractions} \rangle$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

$$\begin{array}{c} \text{ListBox (Éditable)} \geq \text{SurfaceTextBox} \\ \left\{ \begin{array}{c} \textit{Navigation} \\ \textit{WidgetSelection} \\ \textit{WidgetDisplay} \\ \textit{DataEdition} \\ \textit{DataSelection} \\ \textit{Activation} \end{array} \right\} \supseteq \left\{ \begin{array}{c} \textit{Navigation} \\ \textit{WidgetSelection} \\ \textit{WidgetDisplay} \\ \textit{DataEdition} \\ \textit{Activation} \end{array} \right\} \end{array}$$

Opérateurs d'équivalences basés sur des PI et les types de données

- $\cong$  Équivalence stricte des PI et type de données différentes
- $\leq$  Équivalence large des PI et type de données différentes
- $\geq$  Équivalence faible des PI et type de données différentes

# Équivalences des composants graphiques

$ComposantGraphique = \langle Cardinalité, TypeDeDonnée, PrimitivesInteractions \rangle$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

Opérateurs d'équivalences basés sur des PI et les types de données

- $\cong$  Équivalence stricte des PI et type de données différentes
- $\lesssim$  Équivalence large des PI et type de données différentes
- $\gtrsim$  Équivalence faible des PI et type de données différentes

$$\begin{array}{c}
 \text{ListBox} \cong \text{LibraryBar} \\
 \left\{ \begin{array}{c} \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{DataSelection}[t1] \\ \text{DataMoveIn/Out}[t1] \\ \text{DataDisplay}[t1] \\ \text{Activation} \end{array} \right\} \cong \left\{ \begin{array}{c} \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{DataSelection}[t2] \\ \text{DataMoveIn/Out}[t2] \\ \text{DataDisplay}[t2] \\ \text{Activation} \end{array} \right\} \\
 t1 = \text{String}, t2 = \text{MediaElement}
 \end{array}$$

# Équivalences des composants graphiques

$ComposantGraphique = \langle Cardinalité, TypeDeDonnée, PrimitivesInteractions \rangle$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

Opérateurs d'équivalences basés sur des PI et les types de données

- $\cong$  Équivalence stricte des PI et type de données différentes
- $\lesssim$  Équivalence large des PI et type de données différentes
- $\gtrsim$  Équivalence faible des PI et type de données différentes

$$\begin{array}{c} \text{Button} \lesssim \text{Image} \\ \left\{ \begin{array}{c} \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{Activation} \end{array} \right\} \lesssim \left\{ \begin{array}{c} \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{DataDisplay}[t] \\ \text{Activation} \end{array} \right\} \\ t = \text{MediaElement} \end{array}$$

# Équivalences des composants graphiques

$ComposantGraphique = \langle Cardinalité, TypeDeDonnée, PrimitivesInteractions \rangle$

Opérateurs d'équivalences basés sur des PI(Primitives d'Interactions)

- $\equiv$  Équivalence stricte des PI
- $\leq$  Équivalence large des PI
- $\geq$  Équivalence faible des PI

Opérateurs d'équivalences basés sur des PI et les types de données

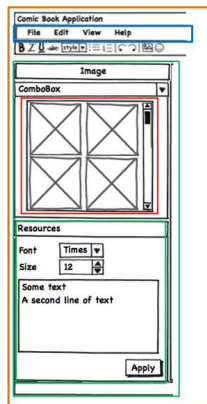
- $\cong$  Équivalence stricte des PI et type de données différentes
- $\lesssim$  Équivalence large des PI et type de données différentes
- $\gtrsim$  Équivalence faible des PI et type de données différentes

$$\left\{ \begin{array}{c} \text{Image} \\ \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{DataDisplay[t]} \\ \text{Activation} \end{array} \right\} \gtrsim \left\{ \begin{array}{c} \text{SurfaceButton} \\ \text{Navigation} \\ \text{WidgetSelection} \\ \text{WidgetDisplay} \\ \text{Activation} \end{array} \right\}$$

$t = MediaElement$

# Exemple d'équivalences

*Source = XAML et Cible = XAMLSurface*



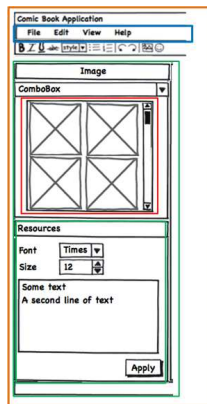
	$\equiv / \approx$	$\leq / \lesssim$	$\geq / \gtrsim$
Window	-/-	-/-	SurfaceWindow/-
ListBox	SurfaceListBox / LibraryBar	-/-	ElementMenu, SurfaceMenu / DataGrid

- Quels composants graphiques sélectionner si plusieurs correspondants ?
- Comment transformer l'UI source en prenant en compte les guidelines ?



# Exemple d'équivalences

Source = XAML et Cible = XAMLSurface



	$\equiv / \approx$	$\leq / \lesssim$	$\geq / \gtrsim$
Window	-/-	-/-	SurfaceWindow/-
ListBox	SurfaceListBox / LibraryBar	-/-	ElementMenu, SurfaceMenu / DataGrid

- Quels composants graphiques **sélectionner** si plusieurs correspondants ?
- Comment **transformer** l'UI source en prenant en compte les guidelines ?

# Transformations de la structure d'UI source

UI cible obtenue en remplaçant les éléments de la source

## Substitution $1 \rightarrow 1$

- Remplacer un élément de l'UI source par son **équivalent** sur la cible (conformément aux opérateurs d'équivalences)
- Exemple(JavaSwing vers DiamondTouch) : JFrame correspond à DSFrame

# Transformations de la structure d'UI source

UI cible obtenue en remplaçant les éléments de la source

## Substitution $1 \rightarrow 1$

- Remplacer un élément de l'UI source par son **équivalent** sur la cible (conformément aux opérateurs d'équivalences)
- Exemple(JavaSwing vers DiamondTouch) : JFrame correspond à DSFrame

## Substitution $n \rightarrow m$ avec $n \geq 1, m \geq 1$

Remplacer un **groupe** d'éléments d'UI par un ou plusieurs élément(s) équivalent

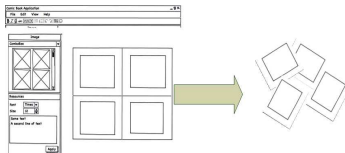
⇒ Comment choisir les groupes ?

## Exemples

Substitution 4 à 4 en

appliquant la guideline d'UI

collaborative



# Transformations de la structure d'UI source

UI cible obtenue en remplaçant les éléments de la source

## Substitution $1 \rightarrow 1$

- Remplacer un élément de l'UI source par son **équivalent** sur la cible (conformément aux opérateurs d'équivalences)
- Exemple(JavaSwing vers DiamondTouch) : JFrame correspond à DSFrame

## Substitution $n \rightarrow m$ avec $n \geq 1, m \geq 1$

Remplacer un **groupe** d'éléments d'UI par un ou plusieurs élément(s) équivalent

⇒ Comment choisir les groupes ?

## Exemples

Substitution 1 à 2 en

appliquant la guideline d'UI

Tangible



&



## Transformations de la structure d'UI source

UI cible obtenue en remplaçant les éléments de la source

Substitution  $1 \rightarrow 1$ 

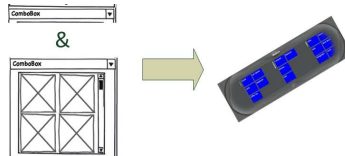
- Remplacer un élément de l'UI source par son **équivalent** sur la cible (conformément aux opérateurs d'équivalences)
- Exemple(JavaSwing vers DiamondTouch) : JFrame correspond à DSFrame

Substitution  $n \rightarrow m$  avec  $n \geq 1, m \geq 1$

Remplacer un **groupe** d'éléments d'UI par un ou plusieurs élément(s) équivalent

$\Rightarrow$  Comment choisir les groupes ?

## Exemples



Substitution 2 à 1

# Modèle de structure

Exprime une instance d'UI

- Primitives d'interactions effectives
- **Données** des composants graphiques (Type de données, Cardinalité)
- Liens de contenance entre composants graphiques : Pour exprimer les **groupes**
  - **UIComponent** : élément d'UI simple
  - **Container** : élément d'UI contenant d'autres, exprime un **groupe** d'élément d'UI

Contenu	Type de container
$\{UIComponent\}$	<b>Homogène</b> : données du même type <b>Hétérogène</b> : données de types différents
$\{UIComponent\} \cup \{Container\}$	<b>Hétérogène</b> : si a un parent
$\{Container\}$	<b>Récuratif</b> : si contient des containers

# Modèle de structure

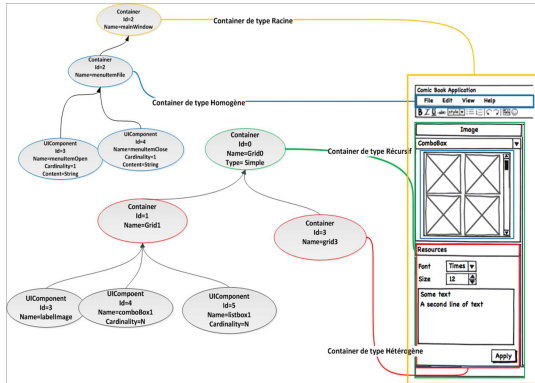
Exprime une instance d'UI

- Primitives d'interactions effectives
- **Données** des composants graphiques (Type de données, Cardinalité)
- Liens de contenance entre composants graphiques : Pour exprimer les **groupes**
  - **UIComponent** : élément d'UI simple
  - **Container** : élément d'UI contenant d'autres, exprime un **groupe** d'élément d'UI

Contenu	Type de container
$\{UIComponent\}$	<b>Homogène</b> : données du même type <b>Hétérogène</b> : données de types différents
$\{UIComponent\} \cup \{Container\}$	<b>Hétérogène</b> : si a un parent
$\{Container\}$	<b>Récuratif</b> : si contient des containers

Type Container= **Racine** si pas de parent

# Exemple Modèle de structure

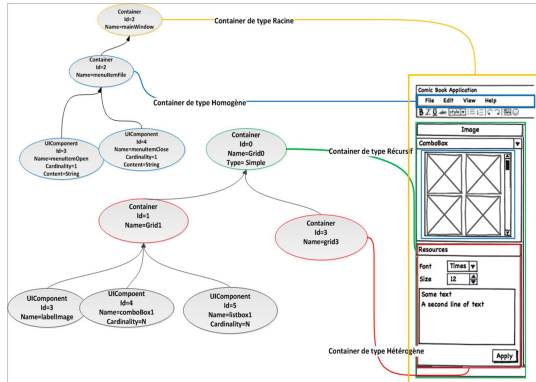


⇒  $ModèleUI = Modèle\ de\ Structure + Primitives\ Interactions$

Transformations du modèle d'UI en prenant en compte les guidelines



# Exemple Modèle de structure

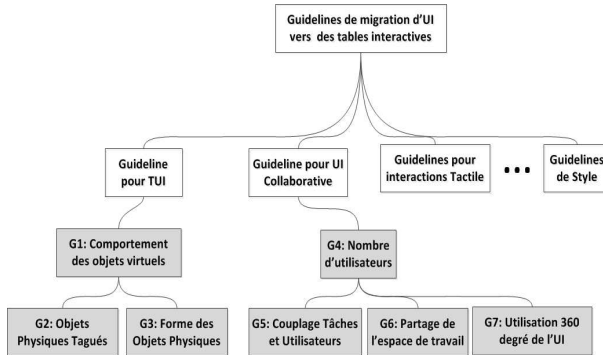


⇒  $ModèleUI = Modèle\ de\ Structure + Primitives\ Interactions$

**Transformations** du modèle d'UI en prenant en compte les **guidelines**

# Rappel des guidelines

Corpus de guidelines [Mic11]



## Prise en compte des guidelines : Transformations

### Règle 1 : Container de type Récursif

Appliquer la guideline d'utilisation en 360 degrés de l'UI (UI Collaborative)

### Règle 2 : Container de type Homogène

Appliquer les guidelines d'utilisation des Objets Tangibles (UI Tangible)

### Règle 3 : Container de type Hétérogène

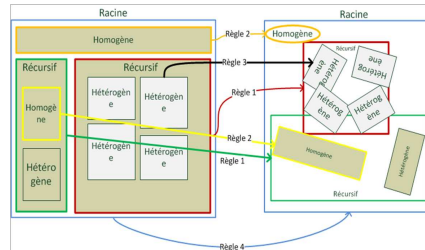
Si parent est un Récursif alors cf. Règle 1

### Règle 4 : Container de type Racine

Si tous les fils sont Container alors appliquer Règle 1

Sinon Appliquer la guideline de partage

d'espace de travail (G6) en fonction du nombre d'utilisateurs (G4)



## Prise en compte des guidelines : Transformations

### Règle 1 : Container de type Récursif

Appliquer la guideline d'utilisation en 360 degrés de l'UI (UI Collaborative)

### Règle 2 : Container de type Homogène

Appliquer les guidelines d'utilisation des Objets Tangibles (UI Tangible)

### Règle 3 : Container de type Hétérogène

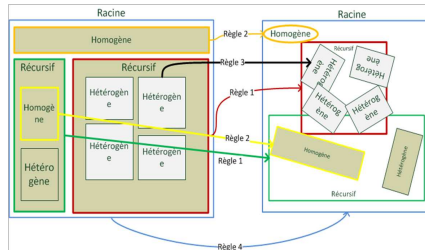
Si parent est un Récursif alors cf. Règle 1

### Règle 4 : Container de type Racine

Si tous les fils sont Container alors appliquer Règle 1

Sinon Appliquer la guideline de partage

d'espace de travail (G6) en fonction du nombre d'utilisateurs (G4)



## Prise en compte des guidelines : Transformations

### Règle 1 : Container de type Récursif

Appliquer la guideline d'utilisation en 360 degrés de l'UI (UI Collaborative)

### Règle 2 : Container de type Homogène

Appliquer les guidelines d'utilisation des Objets Tangibles (UI Tangible)

### Règle 3 : Container de type Hétérogène

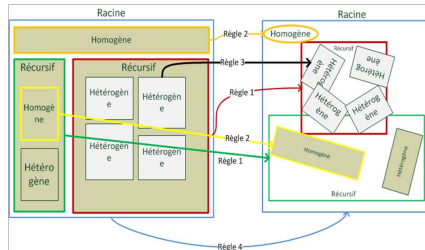
Si parent est un Récursif alors cf. Règle 1

### Règle 4 : Container de type Racine

Si tous les fils sont Container alors appliquer Règle 1

Sinon Appliquer la guideline de partage

d'espace de travail (G6) en fonction du nombre d'utilisateurs (G4)



## Prise en compte des guidelines : Transformations

### Règle 1 : Container de type Récursif

Appliquer la guideline d'utilisation en 360 degrés de l'UI (UI Collaborative)

### Règle 2 : Container de type Homogène

Appliquer les guidelines d'utilisation des Objets Tangibles (UI Tangible)

### Règle 3 : Container de type Hétérogène

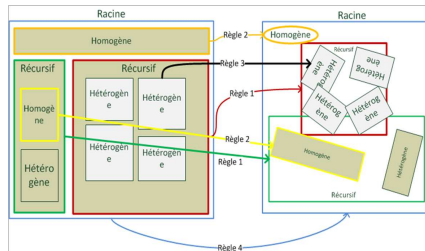
Si parent est un Récursif alors cf. Règle 1

### Règle 4 : Container de type Racine

Si tous les fils sont Container alors appliquer Règle 1

Sinon Appliquer la guideline de partage

d'espace de travail (G6) en fonction du nombre d'utilisateurs (G4)



⇒ Et la concrétisation du modèle transformé!

## Prise en compte des guidelines : Sélection des composants graphiques

Plusieurs opérateurs d'équivalence !

- Plusieurs composants graphiques cible équivalents
- Méthode de classement suivant les Utilisateurs finaux et le concepteur

## Prise en compte des guidelines : Sélection des composants graphiques

### Plusieurs opérateurs d'équivalence !

- Plusieurs composants graphiques cible équivalents
- Méthode de classement suivant les Utilisateurs finaux et le concepteur

### Classement des composants graphiques équivalents

#### Deux critères

- Conformité aux guidelines
- Coût de la mise en œuvre (Nouvelles ressources, Fonctionnalités supplémentaires, ...)

Importance d'un composant graphique = Conformité aux guidelines - Coût de la mise en œuvre

### Exemple





## Prise en compte des guidelines : Sélection des composants graphiques

### Plusieurs opérateurs d'équivalence !

- Plusieurs composants graphiques cible équivalents
- Méthode de classement suivant les Utilisateurs finaux et le concepteur

### Classement des composants graphiques équivalents

#### Deux critères

- Conformité aux guidelines
- Coût de la mise en œuvre (Nouvelles ressources, Fonctionnalités supplémentaires, ...)

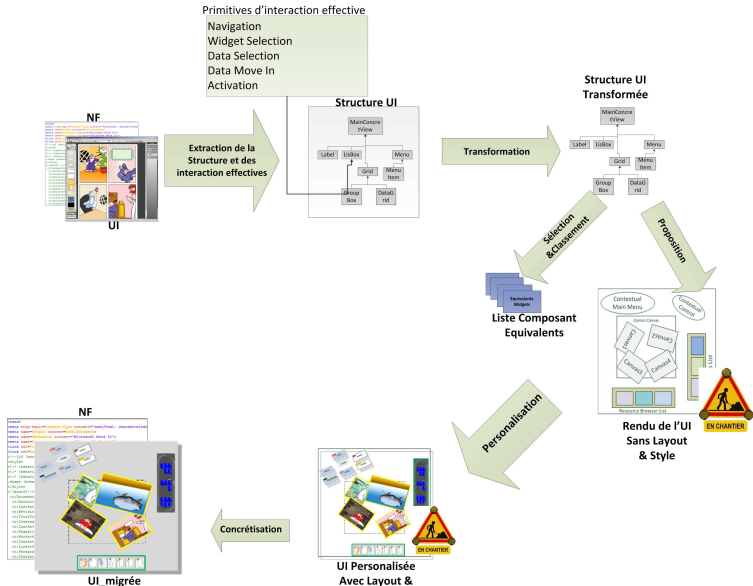
Importance d'un composant graphique = Conformité aux guidelines - Coût de la mise en œuvre

### Exemple



# Migration assistée d'UI vers les tables interactions

## Processus de migration



## Implémentations

# Équivalences des composants graphiques à base des primitives d'interactions

- Abstraction du modèle de structure et des primitives d'interactions d'une UI XAML
- Équivalences et classement des composants graphiques

## Editeur graphique pour table interactive

- Outil d’affichage et de manipulation d’UI à partir d’une instance du modèle d’UI
- Génération du fichier XAML pour Microsoft PixelSense

# Conclusions

## Conclusion de la thèse

- Les **primitives d'interactions** permettent de décrire des mécanismes d'équivalences **réutilisables** et **flexibles** en prenant compte les guidelines
- Les mécanismes de migrations basés sur le **modèle de structure** et les **primitives d'interactions** prennent en compte les guidelines spécifiques à la cible

# Conclusions

## Conclusion de la thèse

- Les **primitives d'interactions** permettent de décrire des mécanismes d'équivalences **réutilisables** et **flexibles** en prenant compte les guidelines
- Les mécanismes de migrations basés sur le **modèle de structure** et les **primitives d'interactions** prennent en compte les guidelines spécifiques à la cible

## Perspectives de cette thèse

- Extension des mécanismes de transformations de structure à d'autres plateformes
- Plateforme d'aide à la migration réutilisable pour un ensemble de plateformes

# État d'avancement de la rédaction

## État d'avancement de la rédaction

Chapitres	État
1-Tables interactives et Migration d'UI	Terminé (Lu par Audrey(2) et Gaetan(1))
2-Approches de migration d'UI	Terminé (Lu par Audrey(2))
3-Modélisation des interactions abstraites	En cours (Lu par Audrey(1), Philippe(2))
4-Prise en compte des guidelines	En cours (Lu par Audrey(1))
5-Prototype	70 % fait

# État d'avancement de la rédaction

## État d'avancement de la rédaction

Chapitres	État
1-Tables interactives et Migration d'UI	Terminé (Lu par Audrey(2) et Gaetan(1))
2-Approches de migration d'UI	Terminé (Lu par Audrey(2))
3-Modélisation des interactions abstraites	En cours (Lu par Audrey(1), Philippe(2))
4-Prise en compte des guidelines	En cours (Lu par Audrey(1))
5-Prototype	70 % fait

## Rédactions Perspectives

- Fin de la rédaction Début Janvier
- Soutenance février ou début mars 2013







*Interactions post-WIMP et applications existantes sur une table interactive.*  
PhD thesis, UNIVERSITÉ PARIS-SUD 11, 2010.

M. Kassoff, D. Kato, and W. Mohsin.



Microsoft Surface 2 Design and Interaction Guide.  
Technical Report July, 2011.



Using Knowledge Representation to Understand Interactive Systems.  
pages 60–67, May 1997.



MARIA : A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments.

Ranking 2015



# Bibliographie II



Andrew Tanenbaum S., Paul Klint, and Wim Bohm.

Guidelines for Software Portability.

*Software-Practice And Experience*, 8(6) :681–698, November 1978.



Brygg Ullmer and Hiroshi Ishii.

The metaDESK : Models and Prototypes for Tangible User Interfaces.

In *UIST '97 Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223 – 232, 1997.



Jean Vanderdonckt.

Conception assistée de la présentation d'une interface homme-machine ergonomique pour une application de gestion hautement interactive.  
1997.



Xin Wang, Yaser Ghanam, and Frank Maurer.

From Desktop to Tabletop : Migrating the User Interface of AgilePlanner.

In *Engineering Interactive Systems 2008*, pages 263–270, 2008.

## Detail 1 : Extraction des primitives d'interactions

## Detail 2 : Opérateurs d'équivalences

	$\equiv / \approx$	$\leq / \lesssim$	$\geq / \gtrsim$
Window	-/-	-/-	SurfaceWindow/-
Menu	ElementMenu, SurfaceMenu /	SurfaceListBox/ LibraryBar	
Grid	Grid / -	ScatterView / -	
ListBox	SurfaceListBox / LibraryBar	-/-	

## Detail 3 : Règles de transformations



## Detail 4 : Guidelines

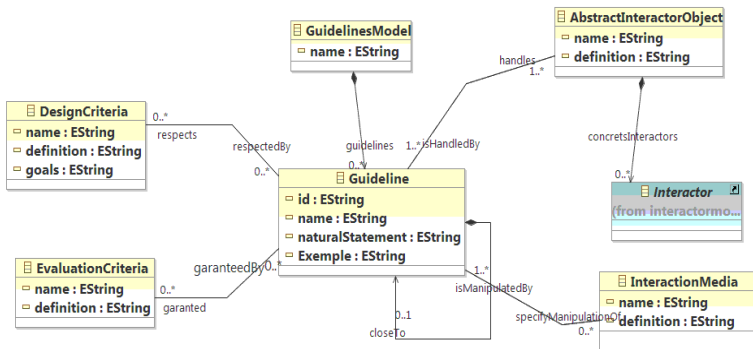


FIGURE :