

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis

Mention : INFORMATIQUE

Présentée et soutenue par

André KALAWA

Migration des applications vers les tables interactives par recherche d'équivalences

Thèse dirigée par Michel RIVEILL

préparée au sein du laboratoire I3S, Equipe RAINBOW

soutenue le 30 août 2013

Jury :

<i>Rapporteurs :</i>	Daniel HAGIMONT	- Professeur, INPT/ENSEEIH de Toulouse
	Jacky ESTUBLIER	- Professeur, LIG Grenoble
<i>Directeur :</i>	Michel RIVEILL	- Professeur, Université de Nice
<i>Co-Encadrante :</i>	Audrey RIVEILL	- Docteur, Université de Nice Sophia Antipolis
<i>Président :</i>	Mireille BLAY-FORNARINO	- Professeur, Université de Nice Sophia Antipolis

Résumé

Dans le domaine du génie logiciel pour les Interactions Homme Machine (IHM), la migration des interfaces utilisateurs (UI) est un moyen pour réutiliser des applications sur des plateformes ayant des modalités d'interactions différentes des environnements de départ. Les approches existantes de migration des UI sont manuelles dans le cadre des approches spécifiques, elles sont automatiques dans le cadre des services d'adaptation des UI aux contextes d'usage, ou elles sont semi automatiques dans le cadre d'une migration flexible dirigée par un concepteur.

Dans cette thèse nous nous intéressons à la migration semi automatique des UI vers une cible comme une table interactive dans l'objectif de transformer des UI Desktop en UI qui favorisent la collaboration et l'utilisation des objets tangibles. Les tables interactives sont des plateformes qui disposent des instruments d'interactions permettant de décrire des UI tangibles et multi-utilisateurs. En considérant que le noyau fonctionnel (NF) des applications de départ peut être réutilisés sur les cibles sans changement, les UI des applications sont caractérisées par la dimension des dialogues entre les utilisateurs et le système, la dimension de la structure et du positionnement des éléments graphiques et la dimension du style des éléments visuels. La migration d'une UI dans ces conditions consiste à transformer ou à recréer les différentes dimensions d'une UI de départ pour la cible tout en considérant les critères de conception des UI pour les tables interactives.

Nous proposons dans cette thèse un modèle d'interactions abstraites pour établir les équivalences entre les dialogues et la structure des UI indépendamment des modalités d'interactions des plateformes source et cible. Les primitives d'interactions et la structure des composants graphiques permettent de décrire des opérateurs d'équivalences pour retrouver et classer les éléments graphiques équivalents en prenant en compte les guidelines des tables interactives. Nous proposons aussi des règles de substitution et de concrétisation pour accroître l'accessibilité des éléments graphiques et favoriser l'utilisation des objets tangibles.

Mots clés : migration des interfaces utilisateurs, équivalences des plateformes, modalités d'interactions, critères de conception, guidelines

Abstract

In software engineering, in the field of human computer interaction (HCI), the migration of user interface (UI) is a way to reuse existing applications on platforms with different interactions modalities. The existing approaches for UI migration can be manual (for specific applications), they can be automatic (for services which adapt UI based on context aware), or they can be mix of the previous - semi automatic (providing a flexible migration process driven by the person in charge).

This thesis proposes a semi automatic process for migration of UI from a desktop to interactive table for the purpose of transforming the UI of desktop to support further collaboration and usage of tangible objects. The interactive tables are platforms with interactions instruments which allow the description of tangible and multi users UIs. Considering that the functional core (FC) of source applications can be reused on target platform without transformation, any UI can be characterized with three dimensions : the first dimension concerns the dialogues between the users and the system, the second dimension concerns the structure and the layout of graphical components, and the third dimension concerns the visual style of graphical elements. In this context, the problematic regarding the UI migration is how to transform or re inject these different dimensions of source UI into the target, while considering the UI design criteria for interactive tables.

This thesis proposes an abstract interactions model for establishing equivalences (independent of modalities of interactions) between the source and the dialogue and structure of the target. The primitives of interaction and the structure of graphical components are used to describe equivalence operators to find and to rank equivalent elements on interactive tables. Furthermore, this thesis proposes substitution and concretization rules to increase the accessibility of graphical elements and to facilitate the usage of tangible objects. The ranking process and the transformation rules are based on guidelines for UI migration to interactive tables which are interpreted form design criteria.

Keywords : user interface migration, equivalence of platform, design criteria, guidelines

Part I

Introduction

Introduction

Contexte

Le nombre grandissant de plateformes et surtout la très grandes variétés de dispositifs d'interactions dont ils disposent telles que les smartphones, les tablettes, les terminaux tactiles ou les tables interactives ont généralisé l'usage des applications ayant des modalités d'interactions [C93] beaucoup plus sophistiquées qu'une simple utilisation du clavier et de la souris. Les tables interactives par exemple offrent la possibilité de mettre en œuvre des applications réellement multi-utilisateurs mêlant interactions tactiles et objets tangibles ¹ sur une surface de prêt de 1 mètre carré.

Le domaine du génie logiciel pour les Interactions Homme Machine (IHM) propose plusieurs approches pour la conception d'une application. En particulier la tendance actuelle est de concevoir les Interfaces Utilisateurs (UI) selon les modèles du Framework de Référence Cameleon (CRF) [CCB⁺02] qui identifie différents niveau d'interaction. De la plus abstraite ou indépendante des dispositifs d'interactions disponibles sur la plateforme vers un niveau concret. Ceci permettant la mise en œuvre de l'UI selon la modalité d'interaction souhaitée et disponible sur la plateforme cible. Si l'interface utilisateur d'une application est construite selon cette approche, alors la migration de cette dernière vers des terminaux ayant des modalités différentes est abordé partiellement par [PSO11]. Notre travail aborde la possibilité de faire migrer entre plateformes ayant des modalités d'interactions différentes des applications qui n'ont pas été conçues selon les modèles du framework de référence (CRF).

La migration des UI est une activité de génie logiciel qui implique la transformation des différents aspects qui constituent une UI existante tels que les interactions (ou le dialogue entre l'utilisateur et le système) qu'il faut nécessairement préserver pour que l'utilisateur puisse toujours accomplir les mêmes tâches, les structures (organisations et types de données des éléments graphiques), le positionnement et les styles des composants graphiques qui doivent être adaptés pour être conformes aux spécificités de la plateforme cible et toujours satisfaire les utilisateurs finaux dans les choix de configuration qui ont pu être fait sur la source.

Au delà des problèmes liés à des différences possibles entre les environnements d'exécution des plateformes source et cible [TKB78] qui nécessite le portage du code, les différences des modalités d'interactions entre la source et la cible impliquent nécessairement la prise en compte des critères usuels de conception des UI. La transformation de l'UI de départ doit être guidée non seulement par les critères ergonomiques de conception mais aussi par les dispositifs d'interactions disponibles sur la plateforme cible qu'il peut être intéressant d'utiliser.

Évidemment, en complément de la volonté de rendre disponible des dispositifs d'interactions nouveaux, peut-être plus intuitifs, les critères ergonomiques de conception constituent un ensemble de principes à respecter pendant la mise en œuvre. Ils permettent de garantir l'utilisabilité d'une UI. Par exemple, l'utilisation d'une application pour desktop sur une table interactive sans aucune adaptation

¹ Les interfaces utilisateurs tangibles (TUI) permettent à une application de pouvoir interagir avec des objets physiques directement manipulable par les utilisateurs [UI97]. Les objets tangibles sont ceux pouvant être manipulés par une interface utilisateur tangible

pose des problèmes d'utilisabilité, car la simulation du clavier et de la souris n'est pas le meilleur mode d'interaction en terme d'utilisabilité. Si l'application le permet, on peut même imaginer une utilisation multi-utilisateurs avec une nouvelle UI à déduire de l'UI d'origine.

Enjeux de la migration

Dans le cadre de la migration des applications pour desktops vers les tables interactives par exemple, nous notons que les dialogues, la structure et le positionnement des éléments des UI doivent prendre en compte l'éventualité d'avoir plusieurs utilisateurs mais surtout la possibilité d'utiliser des objets tangibles. Travailler sur ces deux exemples va nous permettre d'avoir une réelle évolutivité des UI entre le terminal source et le terminal cible et permet de favoriser la collaboration dans un espace de travail co-localisé et multi-utilisateurs offert par la table interactive et donc de l'utiliser pleinement.

Nous faisons l'hypothèse que le cœur de l'application peut être migré sans modification et qu'il ne faut agir qu'au niveau de l'UI qu'il faut bien évidemment faire migrer. Cette migration, d'un terminal source vers un terminal cible donnée doit permettre de conserver dans la mesure du possible les dialogues, la structure et les positionnements relatifs des éléments de l'interface, le respect du style des éléments graphiques des UI de la source mais bien évidemment proposer une adaptation pour utiliser au mieux, sans perdre les utilisateurs, les nouveaux moyens d'interaction mis à la disposition des utilisateurs.

La transformation des dialogues d'une UI desktop vers une cible collaborative peut-elle garantir à chaque utilisateur des interactions qui favorisent la collaboration? De nombreux éléments sont à prendre en compte. En effet, chaque dialogue ou message d'erreur ne doit pas perturber les activités des autres utilisateurs; par exemple les ?feedbacks? ou les messages d'erreurs destinés à un utilisateur ne doivent pas bloquer un autre utilisateur si une réponse est attendue. Par ailleurs la transformation des dialogues peut-elle assurer que chaque dialogue reste cohérent avec l'intention et les actions des utilisateurs ? Par exemple la modification d'une valeur par deux utilisateurs distincts peut les perturber. Il faut en effet décider si le résultat est la somme des deux modifications, la moyenne, le min ou le max. Évidemment le contexte et la nature de la valeur peuvent guider sur le choix à effectuer.

La transformation des dialogues sur les UI et les fonctionnalités de l'application source peuvent provoquer de nombreux changements? En effet, la transformation des dialogues des UI desktops pour les tables interactives peuvent par exemple impliquer des modifications pour prendre en compte la présence de plusieurs utilisateurs ou d'objets tangibles.

La transformation de la structure et du positionnement des éléments d'une UI pour desktop vers une table interactive consiste à assurer l'accessibilité aux différents utilisateurs des éléments graphiques pertinents ². Par ailleurs la transformation des aspects structurels d'une UI doit elle aussi préserver au mieux l'utilisabilité de l'UI. En effet les solutions de transformations proposées doivent produire des UI conformes aux spécificités de la plateforme cible et satisfaisantes pour les utilisateurs finaux.

Pour terminer, la transformation du style a nécessairement un impact sur la collaboration ou sur l'utilisation des objets tangibles. Généralement, chaque application adopte une charte graphique qu'il faut soit respecter, soit rénover.

Les questions soulevées par les transformations des différents aspects des UI lors d'une migration sont usuellement traitées de différentes manières. En premier lieu, il est possible d'adopter une approche manuelle [WGM08]. Celle-ci permet d'avoir, au prix d'un travail minutieux, une UI conforme aux attentes des utilisateurs finaux car les transformations sont flexibles et donc la qualité de

²Un menu par exemple

L’UI produite dépend directement des compétences de celui qui effectue la migration manuelle. Mais cette approche est difficilement réutilisable, à moins de la consigner dans un document, que ce soit pour d’autres applications ou pour d’autres personnes car elle nécessite une bonne connaissance des critères de transformation des différents aspects et des technologies des UI des plateformes source et cible.

Il est aussi possible d’adopter une approche automatique basée ou non sur des modèles abstraits [Bes10] [PZ10] qui permet une transformation des différents aspects des UI. Bien que ces approches automatiques soient réutilisables, elles sont nécessairement moins flexibles car le concepteur qui ne peut pas intervenir pendant la transformation, adapte l’UI produite dans un second temps.

Entre ces deux approches, il est aussi possible d’adopter une approche semi automatique [Mar97] qui présente de nombreux inconvénients car non seulement elle induit un travail supplémentaire pour la personne en charge de la migration qui guide la transformation mais en permettant plus de flexibilité, il est assez difficile de capitaliser sur le travail effectué si la personne en charge de la migration change. Néanmoins, cette approche permet une transformation interactive des différents aspects de l’UI. L’intérêt d’une flexibilité dans l’approche de migration d’UI permet d’avoir des UI migrées proches des attentes des utilisateurs finaux.

Contribution de la thèse

Cette thèse a pour premier objet d’étudier la migration des UI entre plateformes ayant des dispositifs d’interactions différents. Nous avons fait le choix de cibler en particulier la migration des UI depuis une station possédant clavier et souris vers une table interactive. Nous souhaitons que cette migration se fasse au moindre coût pour les concepteurs ou les développeurs tout en prenant en compte les spécificités de la cible et, bien évidemment, les critères ergonomiques usuels utilisés pour la conception des UI.

La solution que nous proposons repose sur un processus semi automatique de migration d’UI. Celui-ci comporte plusieurs étapes interactives pour que les concepteurs ou les développeurs puissent effectuer des choix conformes aux critères de conception qu’ils souhaitent privilégiés. Nous avons fait le choix de l’interactivité, basé sur des choix simples pour garantir simultanément la réutilisabilité, minimiser les coûts mais aussi pour accroître la flexibilité et garantir ainsi l’UI la plus pertinente.

Il est primordial de prendre en compte, lors de la migration des interfaces utilisateurs, les critères ergonomiques de conception. Nos travaux utilisent des concepts issus de plusieurs domaines de recherche.

Dans le domaine de l’utilisabilité, nous nous sommes intéressés aux travaux qui modélisent les critères ergonomiques de conception pour les traduire en règles opérationnelles et utilisables pendant la conception. L’objectif est de pouvoir intégrer ces règles dans la plateforme de migration dans le but de réduire la charge de travail des personnes en charge de la migration.

Dans le domaine de l’ingénierie des modèles, nous nous sommes intéressés aux travaux permettant de modéliser une plateforme dans le but d’effectuer la migration à un niveau abstrait : de concept à concept. L’objectif est de rendre notre travail réutilisable si la source et la cible évoluent. Cette approche nous permet d’abstraire non seulement les UI mais aussi les interactions. Nous proposons ainsi un modèle d’interactions basées sur les primitives d’interactions [KOR⁺11] pour décrire les actions atomiques qui constituent les dialogues entre l’utilisateur et le système.

Plan du manuscrit

Notre manuscrit est structuré de la manière suivante :

Le chapitre 2 présente l'espace des problèmes liés à la transformations des différents aspects d'une UI. Il délimite le périmètre de la migration des UI d'un poste fixe vers une table interactive. Il fixe nos objectifs.

Le chapitre ?? présente une étude du modèle d'interactions des tables interactives dans le but d'identifier les spécificités et les critères ergonomiques de conception à intégrer pour la migration des UI vers cette cible.

Le chapitre ?? est un état de l'art des approches de migration des UI. Dans ce chapitre nous décrivons les critères nécessaires pour atteindre nos objectifs et nous évaluons les différentes approches à l'aide de ces critères. Ce chapitre se termine par une synthèse des différentes approches présentées.

Le chapitre ?? propose un modèle d'UI qui prend en compte deux aspects des UI : leurs interactions et leur structure. Les objectifs de ce modèle d'UI sont de décrire les UI à migrer indépendamment des plateformes mais aussi de décrire des opérateurs d'équivalences entre les dispositifs d'interactions des plateformes source et cible.

Le chapitre ?? résume les mécanismes de transformation de la structure et des interactions de l'UI source vers la cible. L'objectif de ce chapitre est de décrire la prise en compte effective des critères ergonomiques de conception sous forme de guidelines par les mécanismes de transformation.

Le chapitre ?? présente le prototype que nous avons réalisé. Il constitue une preuve de concept des mécanismes proposés. Plusieurs applications ont été migrées afin de valider les éléments des différents modèles, mettre en évidence le respect des critères ergonomiques et surtout mettre en évidence les bénéfices que l'on peut retirer d'une telle approche.

Le chapitre ?? est une conclusion. Elle rappelle les objectifs que nous souhaitons atteindre, met en évidence nos contributions et leurs apports. Elle donne aussi quelques éléments sur des travaux complémentaires qui pourraient être menés pour parfaire nos travaux.

Part II

Domaine d'étude et état de l'art

Migration des applications

Sommaire

2.1	Motivations	9
2.2	Problèmes liés à la migration des UI vers une table interactive	10
2.2.1	Cas d'une application desktop	10
2.2.2	Problèmes liés à la migration du dialogue	12
2.2.3	Problématiques liées à la migration d'UI vers une table interactive	13
2.3	Spécificités de la migration d'UI vers les tables interactives	14
2.3.1	Modèle d'interactions d'une table interactive	15
2.3.2	Propriétés caractéristiques du modèle d'interactions des tables interactives	21
2.3.3	Principes de conception d'UI pour les tables interactives	23
2.4	Concepts pour la migration d'UI	28
2.4.1	Architecture des applications à migrer	28
2.4.2	Modèle d'interactions abstraites	30
2.5	Conclusion	32

Dans le but de décrire les différentes étapes et les problèmes liés à la migration des interfaces Hommes-Machines en général, et vers une table interactive en particulier, nous avons pris le parti d'utiliser un exemple "fil rouge". Les motivations du choix de l'étude de cette cible sont décrites dans la section 2.1, puis la section 2.3 décrit pas à pas l'application initialement conçue pour un desktop que nous souhaitons mettre en œuvre une table surface. La section 2.4 présente les objectifs de cette thèse.

2.1 Motivations

Les tables interactives comme les desktops et les smartphones sont utilisées dans plusieurs domaines d'activités[Kub11]. Une table interactive peut servir comme plateforme de jeux, pour la consultation de photos, de cartes, de dessins, pour noter les idées lors d'une session de brainstorming, etc. Les tables interactives permettent de mettre en œuvre un espace de travail collaboratif.

Néanmoins, on remarque que les tables interactives ne se sont pas démocratisées comme les tablettes ou les smartphones. Évidemment, le prix de vente en est une des raisons peut constituer un frein à l'achat. Il est par exemple d'environ 6600\$ pour la version 2.0 commercialisées par Microsoft. Même si nous observons une nette baisse du prix par rapport à la première version celui-ci constitue indiscutablement un frein à l'achat. Par ailleurs, la faible diffusion de cette plateforme fait que les applications pour les tables interactives sont loin d'être aussi nombreuses que pour les tablettes ou les smartphones. Selon des données provenant de Distimo¹, en mars 2012 nous avons noté plus de 350.000 applications développées aux Etats-Unis pour des desktops, 150.000 applications pour les

¹Distimo : www.distimo.com

smartphones et presque 100.000 spécifiquement pour les tablettes 4 (cf figure 2.4) et aucune pour les tables interactives.

Pour accroître le nombre d'applications disponibles et toucher un éventuel public, nous pensons que la migration des applications existantes peut être une piste sérieuse si la migration utilise effectivement les divers dispositifs d'interactions disponible sur la table. Dans cette thèse, nous étudions par conséquent les divers problèmes posés par le processus de migrations en espérant pouvoir en complément enrichir les applications d'une dimension collaborative et tangible.

Nous présentons donc à la section 2.2 les problèmes liés à la migration d'une application desktop vers une table interactive puis les objectifs de nos travaux. Les hypothèses et les différentes approches permettant de faire migrer des UI sont présentés dans la section 2.3. L'ensemble de ce chapitre nous permet de décrire le périmètre de nos travaux.

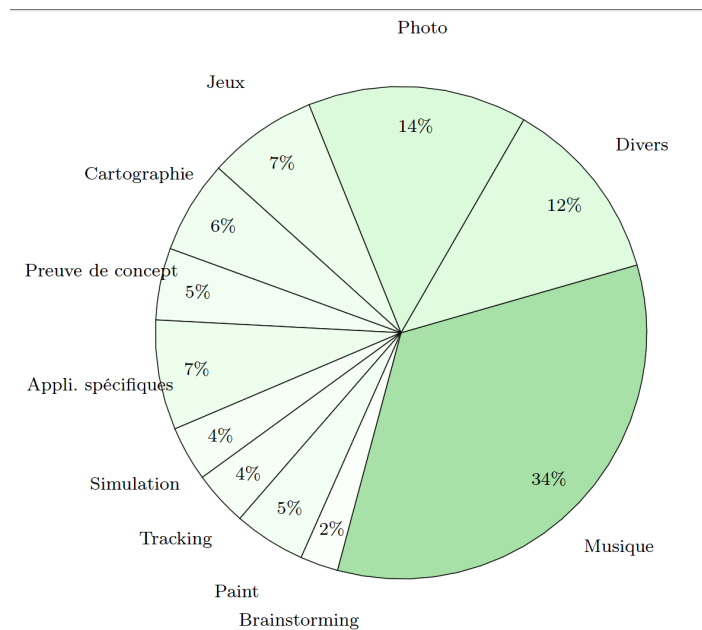


Figure 2.1: Applications disponibles sur les marchés de téléchargement par plateformes en mars 2012 aux États Unis, source Distimo

2.2 Problèmes liés à la migration des UI vers une table interactive

Cette section décrit le fonctionnement d'une application permettant l'élaboration de bande dessinée que nous souhaitons faire migrer vers une table interactive. Dans un second temps, nous présentons les problèmes liés à cette migration et les différents objectifs que nous souhaitons atteindre sont décrits à la fin de cette section.

2.2.1 Cas d'une application desktop

Considérons l'exemple suivant : soit une application conçue pour un desktop qui permet l'élaboration des bandes dessinées (BD) telle que *Comics Book Application* (CBA). Cette application est conçue pour être utilisée par un dessinateur de BD qui est assis devant son écran en utilisant sa souris et son

clavier. La fenêtre principale de l'application CBA décrite par la figure 2.2 nous permet d'identifier trois zones majeures que sont la zone des menus correspondant aux composants graphiques situés en haut de la fenêtre principale, l'espace de travail qui contient les cadres d'une page de bande dessinée et la zone de légende correspondant aux groupes, aux formulaires et listes à gauche de la fenêtre.

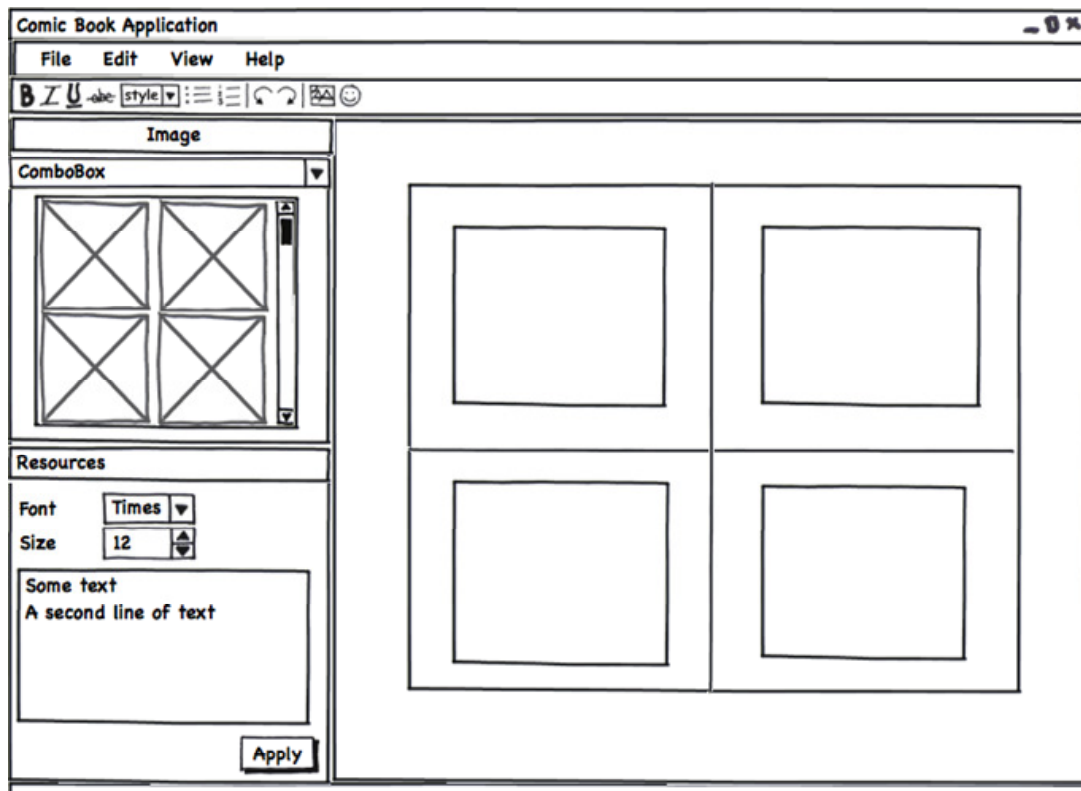


Figure 2.2: Fenêtre principale de l'application CBA

Nous considérons de manière globale que les applications à migrer doivent respecter une décomposition minimale permettant de séparer aisément l'UI et le Noyau Fonctionnel (NF). Si cette hypothèse est vérifiée, les différents aspects des UI des applications peuvent être décrites de manière générale comme suit :

- le **dialogue** entre l'utilisateur et le système permet d'organiser les interactions et les comportements d'une UI graphique,
- la **structure** et le **positionnement** des éléments graphiques qui décrivent l'organisation et les types de données d'une UI,
- le **style** pour décrire les tailles, les couleurs et les polices d'une UI graphique.

Nous considérons ces trois aspects comme autant de dimensions qui permettent de construire l'espace des problèmes lié à la migration d'une UI prévue pour un desktop pour que l'application s'exécute sur une table interactive. Pour chacune de ces dimensions nous étudions les problèmes soulevés par l'utilisation des objets tangibles/physiques comme moyen d'interactions. Dans le cadre

des tables interactives, une **interaction tangible** consiste à utiliser des objets physiques (généralement en contact avec l'écran ou la surface d'affichage) comme des dispositifs de manipulation directes ou comme des moyens d'accès à des fonctionnalités. Pour une application "conception de bandes dessinées" (CBA) sur une table interactive, il est envisageable d'utiliser un stylo pour écrire dans les bulles de dialogue d'une bande dessinée. Dans ce cas le stylo est un **objet tangible** de manipulation directe. Par ailleurs, il est également possible que les concepteurs des bandes dessinées souhaitent utiliser un cube comme **objet tangible** dans le but sélectionner des images (représentant des personnages par exemple). Chaque face du cube sera associée à une catégorie de liste d'images ou d'objets graphiques utilisables dans une bande dessinée.

Nous étudions aussi les problèmes soulevés par la transformation d'une UI mono-utilisateur en UI multi-utilisateurs co-localisées. Dans le cadre des tables interactives, les applications peuvent naturellement être utilisées par plusieurs personnes ce qui est plus difficilement imaginable sur un smartphone. Il est alors nécessaire de re-concevoir l'UI de l'application qui est alors destinée à plusieurs personnes (UI multi-utilisateurs) mais qui partagent le même écran (utilisateur co-localisé). L'application "conception de bandes dessinées" que l'on souhaite faire migrer sur table interactive peut être utilisée simultanément par plusieurs créateurs de bandes dessinées qui ainsi collaborent pour concevoir ensemble une nouvelle BD.

Dans les sections 2.2.2, 2.2.3 et 2.2.4, nous illustrons ces problèmes pour les différents aspects de l'application CBA décrite par la figure 2.2 ci-dessus.

2.2.2 Problèmes liés à la migration du dialogue

Le dialogue est une dimension des UI de départ qui varie fortement dans le cadre de la migration vers les tables interactives. En effet le nombre d'utilisateurs et les modalités d'interactions changent entre la source et la cible. Dans ce cadre, la transformation du dialogue de départ soulève des questions par rapport à l'utilisation des objets tangibles, à l'équivalence des interactions et au nombre d'utilisateurs. Cette section expose ces questions.

2.2.2.1 Utilisation des objets tangibles comme moyen d'interactions

Dans le cadre de l'application CBA, les dessinateurs de BD peuvent utiliser différents objets (cube, disque, etc.) pour afficher un menu de manière contextuelle dans le but d'effectuer une tâche de changement de couleur ou de police d'un texte. De manière générale, il est indispensable d'identifier dans l'UI départ les composants graphiques ou les fonctionnalités que les utilisateurs utiliseront par des **interactions tangibles**.

Les objets physiques peuvent aussi servir d'instruments de manipulations directes ou pour entrer des données. L'exemple d'un stylo pour sélectionner des options ou pour écrire dans le cadre de l'application CBA est envisageable. Dans ce cas, la migration des dialogues impacte à la fois l'UI et le NF. En effet, les **interactions tangibles** peuvent aussi nécessiter l'ajout dans le NF des traitements inexistant dans l'application de départ et la transformation de l'UI. Par exemple l'écriture à main levée avec un stylo à main levée peut nécessiter l'ajout d'un module de reconnaissance d'écriture si la plateforme cible n'offre pas cette fonctionnalité.

L'utilisation des objets tangibles comme moyen d'interaction pour les UI de la cible soulève un sous-ensemble de la dimension des problèmes liés à l'équivalence des interactions entre la source et la cible.

2.2.2.2 Équivalences des interactions entre la source et de la cible pendant la migration

Pour établir des équivalences entre les plateformes source et cible, il est indispensable d'étudier les instruments qui permettent aux utilisateurs de dialoguer avec les UI sur ces plateformes. En effet les différents instruments d'interactions offrent différentes modalités aux utilisateurs des applications. Dans le cas de la migration de l'application CBA du desktop vers une table interactive, il est indispensable de transposer par les interactions du clavier et de la souris sur les tables par des interactions tactiles ou par un objet tangible.

La transposition des dialogues de la source sur une table interactive impose des transformations des interactions initiaux présentes entre l'UI desktop et l'utilisateur et entre l'UI et le noyau fonctionnel. Dans le but d'assurer la cohérence des dialogues après la migration, il est alors indispensable de s'assurer que les nouveaux dialogues permettent toujours une utilisation de l'application. Pour ce faire, nous pensons que les équivalences d'interactions doivent :

- Garantir que l'ensemble des fonctionnalités de l'application desktop restent accessible sur les tables interactives après la migration. En effet, une migration ne doit pas altérer les fonctionnalités de l'application initiale sauf si la personne en charge de la migration le souhaite.
- Assurer que les dialogues modifiant les données d'une application prennent en compte les interactions de plusieurs utilisateurs.

2.2.2.3 Transformation du dialogue de la source pour une plateforme multi-utilisateurs et co-localisée**2.2.3 Problématiques liées à la migration d'UI vers une table interactive**

Dans le cas spécifique d'une migration d'UI vers une table interactive, les problématiques évoquées à la section ?? peuvent être précisées d'avantages. En considérant l'application CBA (cf section ??), la migration de son UI sur une table interactive se fait en soulevant les questions suivantes:

- comment placer et organiser les éléments de l'UI de départ pour une table interactive?
En effet, la fenêtre principale de l'application CBA est conçue suivant la métaphore de bureau qui permet de décrire les applications pour les ordinateurs personnels [App95]. Les menus sont placés en haut et à des positions fixées, les outils ou les légendes sont toujours placés à gauche ou à droite et la zone de travail au centre. Cette structuration des zones permet aux utilisateurs des desktops de se retrouver facilement quelque soit l'application.
Sur une table interactive, cette structuration n'est pas recommandée car elle ne permet pas une utilisation par plusieurs utilisateurs et elle fixe l'orientation des composants graphiques de chaque zone. Pour l'application CBA, les différentes zones de la structure de départ peuvent être conservées mais chaque zone n'est plus associée à un espace géographique spécifique de l'écran. Sur la droite de la figure 2.3, les différentes zones de l'UI CBA de départ n'ont plus de position fixe sur l'UI CBA de la table interactive à droite de la même figure.
- comment utiliser les interactions (tactiles, tangiles) d'une table interactive avec l'UI de départ?
- comment prendre en compte la taille de la surface d'affichage d'une table interactive?
- comment prendre en compte le nombre d'utilisateurs pendant la migration de l'UI de CBA?
Cette problématique consiste aussi à transformer une UI mono utilisateur en UI collaborative dans le cas où le nombre d'utilisateur est supérieur à un. Le projet CoWord [XSS⁺04] permet à

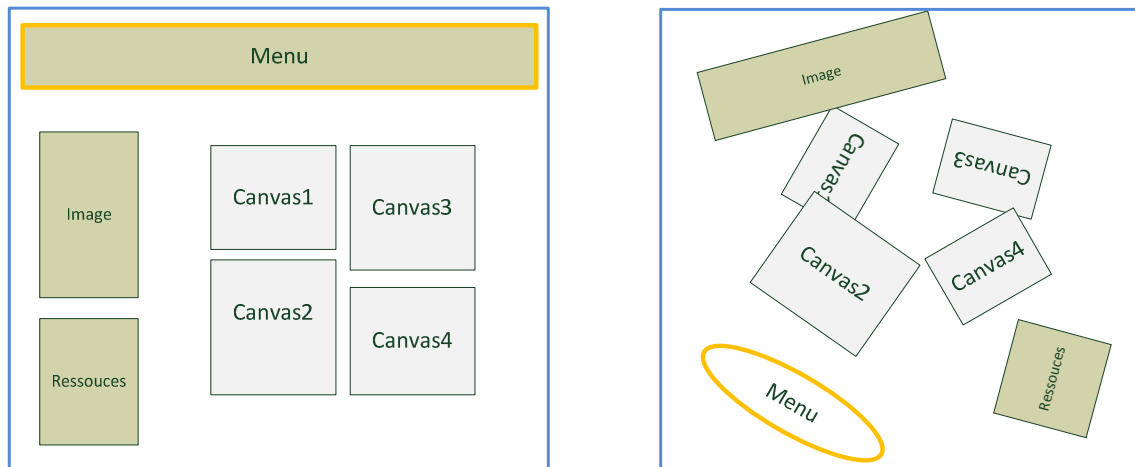


Figure 2.3: Migration de la structure d'une UI Desktop sur une table interactive

plusieurs utilisateurs d'utiliser le logiciel Microsoft Word en même temps mais pas sur un même support, en effet les utilisateurs sont dispersés géographiquement et ils ont chacun un ordinateur de bureau. Dans notre cas les utilisateurs devront partager une même surface d'affichage qui est l'écran de la table interactive ciblée.

Dans la section suivante, nous détaillerons ces problématiques en présentant les spécificités des tables interactives à travers leur modèle d'interaction et leurs principes de conception des UI.

2.3 Spécificités de la migration d'UI vers les tables interactives

Les tables interactives sont des surfaces qui offrent la possibilité d'interagir avec des systèmes interactifs à travers des UI. Les spécificités d'interactions entre les utilisateurs et ces tables interactives peuvent être décrites avec des interactions instrumentales [BL00]. Une interaction instrumentale est une action d'un utilisateur à l'aide de dispositifs physiques (écran tactile par exemple) et de composants graphiques (boutons, scrolls par exemple) pour modifier ou accéder à des objets d'un domaine (images, données numériques, etc.). Ensuite une interaction en sortie est une réponse traduite en une réaction ou un feedback par les instruments d'interaction en sortie. La figure 2.4 montre les différents éléments de ce modèle d'interaction, les instruments servant de moyen d'interactions pour l'utilisateur sont constitués de l'écran tactile et des composants graphiques. La migration d'une UI décrite suivant un autre modèle d'interactions vers ce modèle implique l'adaptation des objets du domaine et des interactions de l'UI de départ aux nouveaux instruments qui constituent la plateforme d'arrivée. L'adaptation des différents aspects de l'UI de départ à la table interactive, se fait suivant un ensemble de principes et des recommandations liés aux tables interactives, dans l'objectif d'avoir une UI respectant ses critères ergonomiques.

Dans l'objectif d'identifier et de caractériser les principes de conception des UI pour les tables interactives, cette section étudie les différents éléments qui composent le modèle d'interaction d'une table interactive (instruments matériels, instruments logiciels, représentation des interactions en entrée et en sortie entre les utilisateurs et les objets du domaine). Elle présente l'ensemble des principes de conception des UI liées aux tables interactives nécessaires pour la migration.

2.3.1 Modèle d'interactions d'une table interactive

Les instruments d'interactions constituent l'ensemble des dispositifs matériels et logiciels d'une table interactive qui permettent d'interagir avec un SI. Les dispositifs matériels d'interactions sont des moyens d'interactions en entrée (actions) ou en sortie (réactions et feedback) cf. figure 2.4. Et les bibliothèques graphiques sont des instruments logiciels pour décrire des interfaces utilisateurs graphiques dans le cadre des tables interactives. Dans ce paragraphe nous nous appuierons sur des tables interactives (Microsoft PixelSense, TangiSense, DiamondTouch, etc.) pour caractériser les dispositifs matériels et d'interaction, les bibliothèques graphiques et les modalités d'interactions des tables interactives.



Figure 2.4: Modèle d'interaction instrumentale d'une table interactive

2.3.1.1 Les dispositifs matériels d'interaction d'une table interactive

Dans ce paragraphe nous étudions les instruments d'interactions de trois tables interactives. Nous étudions d'abord DiamondTouch [DL01] qui est l'une des première table interactive utilisée dans un cadre non expérimental, elle fut industrialisée par MERL [Mit], nous l'étudions car elle comporte une bibliothèque graphique DiamondSpin et permet de décrire des interactions multi utilisateurs, collaboratives et tactiles. Ensuite, l'on s'intéresse à metaDesk [UI97] qui supporte que des objets tangibles comme moyen d'interaction. Enfin nous nous intéresserons aux tables Microsoft PixelSense 1.0 et 2.0 [Mic11] pour leurs bibliothèques graphiques et leurs moyens d'interactions, contrairement aux deux autres tables précédentes, les tables Surface supportent à la fois les interactions collaboratives, tangibles et tactiles.

DiamondTouch - Les moyens d'interaction de la table DiamondTouch sont une surface tactile non capacitive pour les interactions en entrée et un vidéo projecteur pour les interactions en sortie. Cette table supporte des interactions multi utilisateurs et elle est capable d'identifier les contacts de chaque utilisateur autour de la table. Elle ne supporte pas les interactions tangibles car elle ne peut pas détecter des objets ou des tags. Cette table permet de décrire des UI collaboratives. Les UI collaboratives pour les tables interactives ont pour objectif de permettre à plusieurs utilisateurs de partager en même temps une UI. Dans le cadre de la table interactive DiamondTouch les utilisateurs d'une UI collaborative partagent un même espace (la table interactive) et il est possible de créer à chaque utilisateur son espace de travail. Cette solution implique une répartition géographique fixe autour de la table. En effet pour migrer l'UI de l'application CBA par exemple sur une table DiamondTouch, il est indispensable de connaître le nombre d'utilisateurs de l'UI migrée car chaque utilisateur doit être assis sur une chaise pour utiliser la table(cf. figure 2.5). DiamondTouch ne permet aux utilisateurs d'être debout pendant l'utilisation de la surface d'affichage.

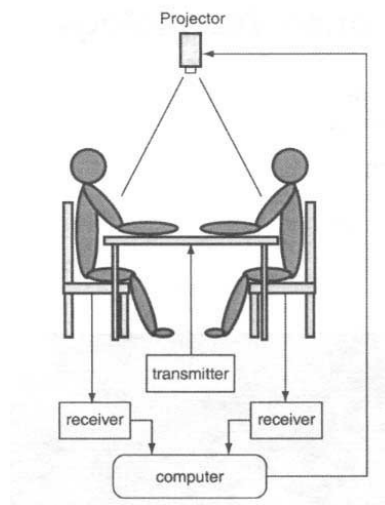


Figure 2.5: Table interactive DiamondTouch

metaDESK - Les moyens d'interaction de la table metaDESK sont constitués des objets tangibles et de caméra infrarouge. Ces objets permettent à chaque utilisateur de manipuler des objets virtuels associés aux objets physiques. Elle ne limite pas le nombre d'utilisateurs comme la table Diamond-Touch et permet à chaque utilisateur d'être mobile autour de la table. Cette Table permet de décrire des UI tangibles (TUI).

Ulmer et Ishii [IU97] définissent un TUI comme des systèmes interactifs qui utilisent des objets physiques pour représenter et utiliser des informations digitales. Le mapping entre le monde physique et digital peut se faire en représentant les différents composants graphiques d'une UI graphique à l'aide des objets concrets. La conception de TUI pour la table metaDESK [IU97], les concepteurs associent une lentille à une fenêtre, un plateau à un menu, etc. comme l'indique la figure 2.6 d'instanciation physique des éléments GUI de metaDESK de la figure 2.6. Ce type d'association permet de concrétiser des composants graphiques virtuels à travers des objets physiques.

Dans le cadre de la migration d'une UI desktop vers la table metaDESK, toutes les interactions de l'UI de départ doivent être adaptées ou émulées avec des objets tangibles. En considérant l'UI de l'application CBA par exemple, le formulaire *Ressources* doit être affiché et rempli avec des objets tangibles, la sélection de la taille ou de la police peuvent être émuler avec des objets circulaires en les tournant par exemple. Par ailleurs l'émulation d'un clavier à l'aide d'un objet tangible pour la saisie des textes est possible mais n'est pas facilement utilisable [?]. Les tables interactives uniquement tangibles comme metaDESK ou TangiSense [KLL⁺09] sont en générale conçues pour faire de la réalité augmentée pour des applications de cartographie par exemple.

Microsoft PixelSense Les moyens d'interaction de la table interactive Microsoft PixelSense sont un écran tactile permettant la reconnaissance des objets physiques, un clavier virtuel et des dispositifs sonores. Elles permettent de décrire des UI tactiles et tangibles par l'utilisation des tags. Elles supportent 50 contacts simultanés et permet donc de décrire des UI multi utilisateurs dans la limite des contacts supportés. En effet une UI ne peut pas supporter plus de 50 points de contacts simultanés, cette contrainte et la taille de l'écran limite le nombre d'utilisateurs utilisant une application sur cette table au même moment.

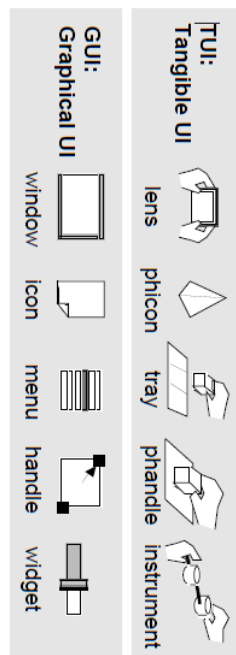


Figure 2.6: Instanciation physique des éléments GUI dans TUI

Cette plateforme permet la description de TUI par la reconnaissance des Tags et la forme des objets. L'utilisation de tags permet de ne pas limiter les objets à utiliser dans la description des UI. Les tags sont des codes barres à deux dimensions qui sont facilement identifiables par la table surface. La Surface offre deux types tags : Identity tags et Byte tags². Ils peuvent être utilisés pour reconnaître des objets physiques ou les distinguer parmi plusieurs, pour déclencher une commande ou une action (afficher un menu ou une application par exemple), pour pointer et orienter une application par un objet tagué peut être utilisé pour le contrôle de volume car il peut détecter le changement d'angle d'un objet.

Résumé Les tables interactives de manière générale ont deux types de dispositifs physiques d'interactions: les dispositifs d'interaction en entrée qui peuvent être tactiles et/ou tangibles et les dispositifs d'interaction en sortie qui sont des surfaces d'affichage. Ces surfaces sont de tailles variables et de différentes formes (rectangulaire, circulaire, etc.). Elles peuvent être disposées de manière horizontale (pour DiamondTouch, TangiSense, Microsoft PixelSense 1.0 et 2.0) ou de manière verticale (pour Microsoft PixelSense 2.0, etc.).

Le nombre d'utilisateurs et leurs dispositions autour de la surface d'affichage sont des éléments qui permettent de caractériser les interactions des tables interactives. En effet ces deux caractéristiques impactent la conception des UI car une UI destinée à plusieurs personnes doit permettre l'accessibilité des différentes fonctionnalités à tous les utilisateurs et elle doit aussi prendre en compte le partage des éléments de l'UI (menus, visualisation des contenus, etc.)

²Les Identity tags sont des tags sur avec une plage de valeurs sur 128 bits, les Byte tags on une plage de valeurs sur 8 bits

Les dispositifs matériels des tables interactives permettent de décrire des UI multi utilisateurs, co-localisées, tactiles et tangibles. La migration des UI des applications qui ne prennent pas en compte ces caractéristiques soulève des problématiques liées à la prise en compte des différents moyens d'interactions en entrée et en sortie des tables interactives, du nombre d'utilisateurs et de la disposition des utilisateurs par rapport à la surface d'affichage.

2.3.1.2 Bibliothèques graphiques des tables interactives

Les bibliothèques graphiques sont des boîtes à outils logiciels qui contiennent des éléments pour décrire des UI graphiques; dans le modèle d'interactions des tables interactives, les composants graphiques sont des instruments logiciels qui permettent de faire le lien entre les dispositifs matériels d'interactions et les objets d'un domaine. Elles offrent des composants graphiques adaptés aux moyens d'interactions d'une plateforme spécifique. Dans cette section nous présentons quelques bibliothèques graphiques spécifiques aux tables interactives.

DiamondSpin [SVFR04] est une bibliothèque graphique pour table interactive qui offre des composants graphiques adaptés à plusieurs utilisateurs. En effet, elle offre des composants graphiques qui permettent : une manipulation des documents visuels, la manipulation directe des éléments d'UI, la possibilité d'utiliser ses doigts, un stylet ou un clavier comme moyen d'interaction, la rotation des composants graphiques, et la possibilité de créer et de gérer des espaces privés pour chaque utilisateur. Les composants graphiques *DSContainer*, *DSPanel*, *DSWindow*, *DSFrame* (cf. figure 2.7) par exemple permettent d'avoir un container qui regroupe un ensemble de composants graphiques qu'un utilisateur peut déplacer en fonction de sa position.

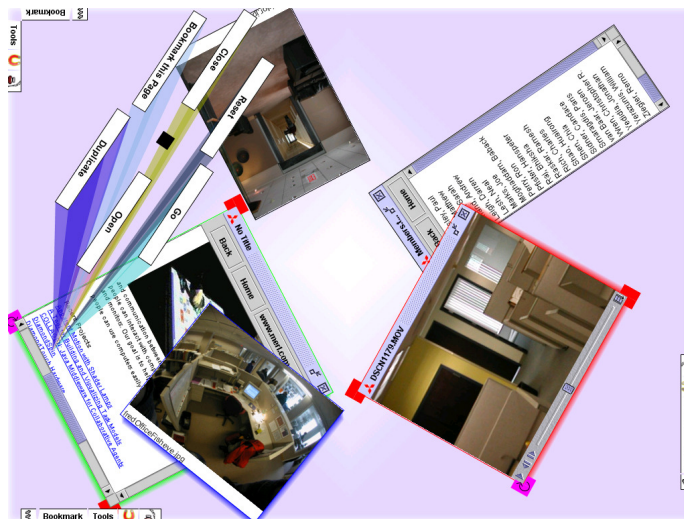


Figure 2.7: Instances des container DiamondSpin

Surface SDK 1.0 et 2.0 [Mic12a] sont des API et des boîtes à outils pour développer des applications pour une Table interactive Microsoft (1.0 et 2.0). Ils font partie du Framework .Net et offrent des bibliothèques graphiques pour concevoir des UI WPF [Mic12b] et XNA [Mic12c]. Ces bibliothèques graphiques offrent des composants graphiques permettant la reconnaissance des formes

d'objets, l'utilisation des tags, 50 points de contacts simultanés, la détection de l'orientation des touches, etc. Ces facilités par rapport à la bibliothèque graphique DiamondSpin évitent au programmeur la gestion du déplacement ou la rotation des composants graphiques par exemple. En effet un *ScatterView* (Figure 2.8) définit le déplacement ou la rotation de manière intrinsèque.

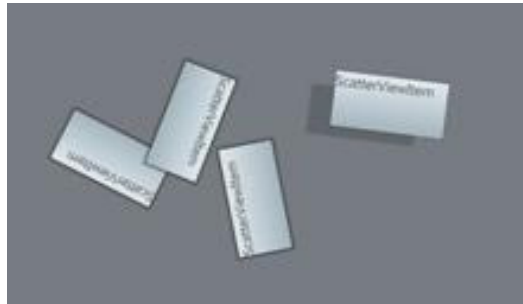


Figure 2.8: Exemple de ScatterView

Résumé La bibliothèque DiamondSpin permet d'utiliser la table DiamondTouch à l'instar d'un bureau virtuel par plusieurs personnes assises autour de la table. Elle offre aussi des composants graphiques basés sur la métaphore du papier [BRNB07]. La métaphore du papier permet l'utilisation des éléments graphiques d'un container comme une feuille de papier en ayant la possibilité de plier, retourner, dupliquer ou déchirer le container, sur une table interactive cette métaphore permet de conserver les actions qu'on réalise sur une table de travail concrète.

La bibliothèque Surface SDK permet de décrire des TUI à l'aide des tags. De manière générale, les bibliothèques graphiques des tables interactives offrent des composants graphiques qui ont des interactions (rotation, redimensionnement, déplacement, tags, etc.) adaptées pour la description des UI pour les tables interactives. Elles permettent d'affiner les caractéristiques des tables interactives en précisant si une table interactive supporte ou non des interactions tangibles, si elle a des composants graphiques accessibles par tous les utilisateurs par exemple.

Dans le cadre de la migration, le changement de bibliothèque graphique implique une nouvelle conception de l'UI de départ pour rendre accessible ses fonctionnalités sur la plateforme d'arrivée en utilisant les composants graphiques adaptés aux dispositifs physiques de la table interactive.

2.3.1.3 Modalités d'interactions

Nigay [NC94] propose une définition de la modalité d'interaction comme un couple $\langle d, l \rangle$ constitué d'un dispositif d'interaction et d'un langage d'interaction [NC94]: - d désigne un dispositif physique (par exemple, une souris, une caméra, un écran, un haut-parleur), - l dénote un système représentationnel, c'est-à-dire un système conventionnel structuré de signes assurant une fonction de communication (par exemple, un langage pseudo naturel, un graphe, une table).

La migration des UI vers les tables interactives implique un changement des dispositifs d'interactions et la possibilité de décrire des équivalences entre les modalités d'interactions des plateformes de départ et celles des tables interactives.

Dans cette section nous abordons la problématique liée aux changements des modalités d'interactions des UI à migrer. En effet comment décrire les interactions de l'UI de départ à l'aide des modalités d'interactions de la plateforme d'arrivée? Pour répondre à cette question nous étudions au paragraphe 2.3.1.1 les problèmes liés aux changements de modalités d'interactions pendant la migration. Et au paragraphe 2.3.1.2 nous présentons quelques modèles d'interactions abstraite qui permet de décrire les interactions indépendamment des modalités d'interactions.

Changement de modalité d'interaction - Considérons comme plateforme de départ un desktop composé d'un écran comme moyen d'interaction en sortie et d'un clavier et d'une souris comme moyen d'interaction en entrée.

- La modalité d'interaction en sortie M1 du desktop est décrite par l'écran et par le langage de description des UI 2D (L1), $M1 = \langle \text{Ecran}, L1 \rangle$. Le langage L1 correspond aux composants graphiques tels que labels, champ de texte, boîte de dialogue, etc. d'une bibliothèque graphique.
- La modalité d'interaction en entrée M2 du desktop est décrite par le clavier et par le langage des commandes (L2), $M2 = \langle \text{Clavier}, L2 \rangle$. Le langage L2 décrit pour chaque actions les tâches réalisables à l'aide d'un clavier sur une UI graphique, ces actions sont par exemple : copier avec Ctrl+C, couper avec Ctrl+X, coller avec Ctrl+V, saisir un texte avec les touches alphanumériques, valider avec la touche entrée, etc.
- La modalité d'interaction en entrée M3 du desktop est décrite par la souris et par le langage de manipulation directe d'une UI 2D (L3), $M3 = \langle \text{Souris}, L3 \rangle$. Le langage L3 décrit les actions réalisables à l'aide d'une souris sur une UI graphique, ces actions sont par exemple : cliquer et valider pour sélectionner un composant graphique, cliquer et déplacer pour sélectionner un texte, déplacer un élément, redimensionner, etc.

Et considérant maintenant que la table interactive cible est composée d'un écran tactile, d'un clavier virtuel et de la reconnaissance des objets physiques comme moyens d'interactions.

- La modalité d'interaction en sortie M'1 de la table interactive est décrite par l'écran tactile et par le langage de description des UI 2D (L'1), $M'1 = \langle \text{Ecran Tactile}, L'1 \rangle$. Le langage L'1 correspond à la bibliothèque graphique de la table interactive qui contient les composants graphiques tels que labels, champ de texte, images, fenêtre, etc.
- La modalité d'interaction en entrée M'2 de la table interactive est décrite par l'écran tactile et par le langage de manipulation tactile d'une UI 2D (L'2), $M'2 = \langle \text{Ecran Tactile}, L'2 \rangle$. Le langage L'2 décrit les actions utilisateurs sur un écran tactile. Ces actions sont par exemple : toucher avec un doigt pour activer ou sélectionner un élément, toucher avec deux doigts et déplacer pour agrandir, réduire, tourner des composants graphiques, etc.
- La modalité d'interaction en entrée M'3 de la table interactive est décrite par le clavier virtuel et par le langage de commande (L'3), $M'3 = \langle \text{Ecran Tactile}, L'3 \rangle$. Le langage L'3 décrit les actions utilisateurs réalisables avec un clavier virtuel tel que saisir un texte.
- La modalité d'interaction en entrée M'4 de la table interactive est décrite par l'écran tactile et par le langage de manipulation des objets tangibles (L'4), $M'4 = \langle \text{Objets Tangibles}, L'4 \rangle$. Le langage L'4 décrit les actions utilisateurs sur un écran tactile à l'aide des objets tangibles. Ces actions sont par exemple : poser un objet pour afficher un menu ou un formulaire, déplacer un objet physique pour déplacer l'objet virtuel associé, tourner un objet physique pour sélectionner une fonctionnalité, etc.

Les langages d'interaction L1 et L'1 permettent de décrire les interactions en sortie des UI des applications source et cible. Ces langages peuvent être modélisés en se basant sur des boîtes à outils indépendantes des dispositifs d'interactions en sortie. Crease dans [Cre01] propose une boîte à outils décrivant des widgets multimodales et indépendantes des dispositifs d'interactions en sortie. Les widgets de Crease [Cre01] restent cependant liées aux dispositifs d'interactions en entrée tels que le clavier et la souris.

Les langages d'interaction L2, L3, L'2, L'3 et L'4 quant à eux permettent de décrire et d'interpréter les interactions en entrée des utilisateurs des plateformes de départ et d'arrivées. Ces langages d'interaction permettent d'associer à chaque action de l'utilisateur un comportement ayant un sens dans l'UI de l'application. Les actions utilisateurs telles que cliquer, sélectionner, Ctrl+C, poser un objet, etc. dépendent des dispositifs d'interactions tandis que les comportements de l'UI dépendent du type d'UI et de l'interprétation souhaitée par le concepteur.

Équivalences des modalités d'interactions La migration de la plateforme desktop vers une table interactive peut être considérée comme un processus de changement de modalités d'interactions. En effet dans le but de réutiliser l'UI d'une application de départ avec les dispositifs d'interactions de la plateforme d'arrivée, il est possible d'établir des équivalences entre les dispositifs d'interaction et les langages d'interactions des plateformes de départ et d'arrivée.

Pour décrire les interactions d'une UI de départ à l'aide des dispositifs d'interaction d'une table interactive, l'une des approches à envisager peut être la mise en correspondance des dispositifs d'interactions en établissant des équivalences entre les différentes modalités d'interactions des plateformes source et cible. Ce qui consiste par exemple à décrire des équivalences d'abord entre les modalités d'interactions en sortie M1 et M'1 et ensuite entre les modalités d'interactions en entrée M2, M3 et M'2, M'3 M'4.

L'équivalence entre M1 et M'1 consiste à comparer les deux dispositifs qui sont des écrans qui peuvent afficher des UI graphiques et les langages L1 et L'1 qui ont des composants graphiques appartenant à des bibliothèques graphiques différentes. L'équivalence entre les modalités d'interactions en entrée n'est possible que si l'on peut comparer les langages L2, L3, L'2, L'3 et L'4. Ces langages sont décrits en fonction des dispositifs d'interaction et aussi en fonction des applications, en effet chaque concepteur peut décrire un langage propre à son application, l'objectif de ces langages étant de faciliter l'interaction entre un dispositif physique (clavier, souris, écran tactile, etc.) et l'UI de l'application.

2.3.2 Propriétés caractéristiques du modèle d'interactions des tables interactives

Les éléments du modèle d'interactions instrumentales des tables interactives tels que les dispositifs matériels d'interactions en entrée et en sortie, les bibliothèques graphiques ou les modalités d'interactions nous permettent d'identifier des propriétés caractéristiques des tables interactives qui impactent la conception des UI pour des tables interactives.

2.3.2.1 Dispositifs d'interactions en entrée

Ces dispositifs influencent la conception des interactions. Dans le cadre des tables interactives nous identifions deux propriétés qui correspondent aux moyens d'interactions tangibles et tactiles.

Propriété 1 *Tangibilité des interactions*

Les dispositifs d'interactions en entrée permettent d'associer des objets physiques aux fonctionnalités³ ou aux composants graphiques⁴ d'une UI.

Propriété 2 *Tactibilité des interactions*

Les dispositifs d'interactions en entrée des tables interactives permettent de décrire des manipulations directes des UI telles que la sélection, l'édition, le redimensionnement, le déplacement, etc.

2.3.2.2 Dispositifs d'interactions en sortie

Ce sont les surfaces d'affichage des tables interactives, les propriétés caractéristiques liées à la taille et à la disposition des tables interactives qui impactent la conception des UI.

Propriété 3 *Taille de la surface d'affichage*

Cette propriété permet d'adapter la taille des composants graphiques par rapport à la taille de l'écran pour faciliter l'utilisation de l'UI.

Propriété 4 *Disposition de la surface d'affichage*

La surface d'affichage peut être disposée de manière horizontale comme une table de travail ou de manière verticale comme un tableau collaboratif. Ces dispositions influencent l'orientation et l'utilisation des composants graphiques d'une UI.

2.3.2.3 Utilisateurs des tables interactives

Les utilisateurs influencent la conception de l'UI par leur nombre et leur répartition autour de la surface d'affichage.

Propriété 5 *Nombre d'utilisateurs*

Cette propriété permet de savoir comment disposer les éléments de l'UI pour faciliter l'accessibilité en fonction du nombre d'utilisateurs.

Propriété 6 *Répartition des utilisateurs*

Cette propriété permet de savoir comment décrire la collaboration entre les utilisateurs autour de la table. La répartition de l'espace de travail de chaque utilisateur peut se faire par une division géographique de la surface ou permettre aux utilisateurs d'accéder à toute la surface.

2.3.3 Principes de conception d'UI pour les tables interactives

Le paragraphe 2.3.1.3 nous montre les différences de modalités d'interactions entre un desktop et une table interactive. Ces différences impactent aussi la conception d'UI pour ces deux plateformes. Besacier et *al.* montrent que la réutilisation des applications desktop sur les tables interactives en adaptant les éléments de l'UI aux métaphores du papier par exemple facilite l'utilisation des UI [BRNB07]. Par ailleurs, une réutilisation d'une application desktop sur des tables interactives sans prise en compte de ses spécificités pose deux problématiques majeures: la transformation de l'UI de départ en UI collaborative d'une part et la transformation d'une GUI en TUI d'autre part. Ces deux caractéristiques font parties de l'ensemble des principes qui guident la conception des UI pour les tables interactives.

Les principes de conception d'UI pour une plateforme constituent un ensemble de recommandations pour les concepteurs d'UI qui indiquent comment décrire les aspects tels que les interactions instrumentales, le layout (ou le placement des éléments graphiques), les activités d'une UI et aussi les styles de présentations (couleurs, polices, tailles, etc).

Les principes de conceptions sont des recommandations de haut niveau qui doivent être traduites en règles formelles utilisables pendant la migration [Van97].

Dans cette section nous caractérisons les principes de conception pour la migration des UI vers les tables interactives en trois catégories: les principes de conception pour les UI tangibles, les principes de conception pour les UI collaboratives et colocalisées et enfin les autres principes de conception d'UI sur une grande surface d'affichage telles que la taille et la position de la surface d'affichage, l'utilisation en 360 degré de l'UI et le style de l'UI.

2.3.3.1 Corpus des principes de conception d'UI pour Microsoft PixelSense

Les principes de conception d'UI pour une Microsoft PixelSense sont décrits sous forme de guidelines (ou recommandations) dans le document *User Experience Design Guideline* [Mic11]. Ces guidelines sont le fruit des expériences des utilisateurs ayant développé des UI pour cette plateforme. L'objectif de ces guidelines est de faciliter la conception des interfaces utilisateurs naturelles [Ste12] et intuitives. Ces guidelines couvrent plusieurs aspects du processus de conception des UI tels que la conception des interactions entre les UI et les utilisateurs finaux, les guides de styles pour une cohérence visuelle, les guides d'utilisation des textes, etc.

2.3.3.2 Guidelines pour TUI

Cette catégorie regroupe les recommandations qui permettent de décrire le comportement des éléments de l'UI qui sont des objets virtuels d'une part et l'association entre ces objets virtuels et les objets tangibles d'autre part. L'association peut se faire par des tags qui permettent de marquer les objets physiques ou en se basant sur la forme des objets physiques. Les guidelines de cette catégorie sont inspirées par la propriété 1 de tangibilité des interactions.

Guideline 1 *Comportement des objets virtuels*

Les comportements des éléments d'une TUI pendant leurs utilisations doivent correspondre aux objets physiques. Les éléments (ou objets virtuels) de l'UI à migrer doivent être associés à des objets physiques dans le but d'afficher des menus, des formulaires ou des fenêtres et aussi dans le but d'activer ou d'utiliser des fonctionnalités.

Guideline 2 *Objets physiques tagués*

Un tag est associé à un objet virtuel d'une TUI (les objets virtuels sont identifiés grâce à la guideline 1). Le déplacement, l'orientation et la position du tag sur l'écran peuvent être associés à des interactions en entrée ou à des comportements de l'objet virtuel associé.

Guideline 3 *Forme des objets physiques*

La forme d'un objet physique peut être associée à un objet virtuel ou à une fonctionnalité. Le déplacement, l'orientation et la position d'un objet physique sur l'écran peuvent être associés à des interactions en entrée ou à des comportements de l'objet virtuel associé. L'utilisation de la forme des objets physiques comme moyen d'interaction en entrée nécessite un module de reconnaissance de la forme d'un objet tangible.

En considérant l'UI de l'application CBA à la figure 2.2, le menu principal et le formulaire *Ressources* par exemple peuvent être associés à un objet physique pour les afficher facilement sur l'écran. Les règles formelles issues de la guideline 1 permettront d'identifier et de transformer les éléments concrets de l'UI

Les tags permettent par exemple d'utiliser deux objets de la forme avec des couleurs différentes et marqués des deux différents tags pour afficher un menu ou un formulaire.

2.3.3.3 Guidelines pour UI collaborative

Cette catégorie regroupe les recommandations pour la conception d'une UI collaborative et colocalisée pour une table interactive. Les guidelines sont liées à la propriété 5 caractérisant le nombre d'utilisateurs et à la propriété 6 qui caractérise leur répartition autour de la surface d'affichage. Elles sont aussi liées à la propriété 3 et à la propriété 4 qui caractérise la taille et la disposition (horizontale ou verticale) de la surface d'affichage des tables interactives.

Guideline 4 *Nombre d'utilisateurs de l'UI migrée*

Cette guideline préconise de prendre en compte le nombre d'utilisateurs de l'UI après la migration. Les UI d'une table interactive peuvent être collaboratives (plusieurs utilisateurs) ou non (dans le cas d'un seul utilisateur). Elle impacte le couplage entre les tâches et les utilisateurs (qui est spécifié par la guideline 5), l'organisation de la surface de travail (qui est spécifiée par la guideline 6) et l'accessibilité des éléments d'une UI collaborative (qui est spécifiée par la guideline 7).

Guideline 5 *Couplage tâches et utilisateurs d'une UI*

Elle est une spécification de la guideline 4, car elle préconise dans le cas d'une UI multi utilisateurs :

- d'éliminer toutes les boîtes de dialogues bloquantes pour les autres utilisateurs de l'UI.
- de dupliquer certains éléments de l'UI pour permettre à tous les utilisateurs d'accéder aux éléments de l'UI.

Guideline 6 *Partage de l'espace de travail*

Cette guideline spécifie la guideline 4 en préconisant des composants graphiques qui facilitent le partage de l'écran dans le cas d'une UI multi utilisateurs. L'espace de travail doit:

- permettre à une personne d'utiliser une UI sans avoir l'aide d'autres utilisateurs,
- permettre à un utilisateur d'utiliser l'UI sans interrompre les utilisateurs présents.

Le partage de l'espace de travail est implémenté de diverses manières en fonction des tables interactives. La table DiamondTouch [DL01] par exemple permet aux concepteurs d'associer un utilisateur à une zone de l'écran. Cependant la table Microsoft PixelSense quant à elle ne permet pas une division de l'écran en zones associées aux utilisateurs ou à des fonctionnalités. La figure 2.9 illustre une division de l'écran en quatre zones, ce type de partage d'écran n'est pas autorisé par les guidelines de la table Surface. Dans le cadre de la migration d'UI vers les tables interactives, nous pensons que la division de l'écran en plusieurs zones ne permet pas de décrire des UI collaboratives si elle ne permet pas d'échanges entre les utilisateurs et leur mobilité autour de l'écran.

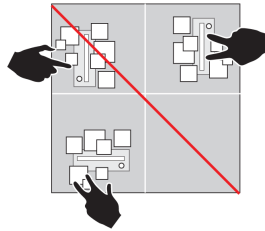


Figure 2.9: Illustration de partage d'espace entre plusieurs utilisateurs

Guideline 7 *Utilisation 360 degré de l'UI*

Cette guideline permet de spécifier la guideline 4 car elle préconise la sélection des composants graphiques pouvant être utilisés partout autour de la table. Elle rend accessible les éléments d'une UI migrée et renforce la collaboration entre les utilisateurs.

La figure 2.10 illustre à un cas d'utilisation des composants graphiques utilisables à 360°.

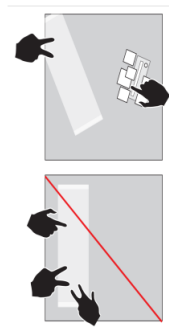


Figure 2.10: Illustration de la propriété 360°

En considérant que l'UI de l'application CBA (cf. figure 2.2) est migrée vers une table Surface pour quatre dessinateurs de BD par exemple. Les ressources (images, ballons, etc.) utilisés pour la conception des BD doivent être accessibles par les quatre utilisateurs. La guideline 6 préconisant le partage de l'espace de travail et la guideline 6 préconisant l'utilisation des objets 360 degré permettent de décrire une UI sans layout avec des groupes d'éléments utilisables à 360°. La guideline 5 par exemple permettra de supprimer les composants graphiques bloquants tels que les boîtes de dialogues de l'UI de l'application CBA.

2.3.3.4 Autres guidelines pour les UI sur surface

Les deux catégories de guidelines présentées ci-dessus qui permettent de décrire des UI tangibles et des UI collaboratives (cf. section 2.3.3.3) ne constituent pas le corpus des guidelines applicables pendant la migration d'une UI vers une table interactive. En effet les aspects de l'UI liés au style des

composants graphiques, des textes de l'UI et les aspects liés aux interactions tactiles ne sont pas pris en compte par ces deux catégories de guidelines. Cette troisième catégorie regroupe les guidelines de mise en œuvre des interactions tactiles et des styles de l'UI.

Guidelines pour les interactions tactiles - Les tables interactives permettent en général de décrire des interactions tactiles. L'ensemble des actions tactiles de l'utilisateur sont interprétées par le dispositif d'interaction en entrée. Dans le cadre des UI tactiles, il existe des actions tactiles standards pour des interactions de redimensionnement, de déplacement ou de rotation. Par exemple les guidelines des interactions tactiles pour une table Surface [Mic11] identifient l'ensemble des gestes recommandés aux concepteurs pour la sélection, l'activation, le déplacement, la rotation, le zoom, etc. Cette catégorie de guidelines est liée à la propriété 2 de tactibilité des interactions d'une table interactive.

Dans le cadre de la migration de l'UI de l'application CBA vers les tables interactives Surface, il est indispensable de pouvoir décrire des correspondances entre les interactions tactiles et les interactions du clavier et de la souris de l'UI départ.

Guidelines pour le styles - Cette sous catégorie regroupe l'ensemble des recommandations pour la personnalisation des aspects visuels d'une UI pour une table interactive. Ces guidelines peuvent être utilisées dans le cadre d'une migration faisant intervenir les concepteurs comme des exemples pour les inspirer du choix de la forme, des couleurs, des icons et aussi de la disposition des textes d'une UI. Dans le cas du scénario de migration, le formulaire *Ressources* par exemple peut être migré suivant comme l'indique le tableau 2.1.

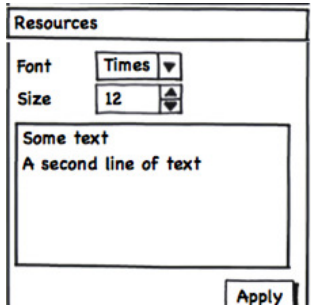
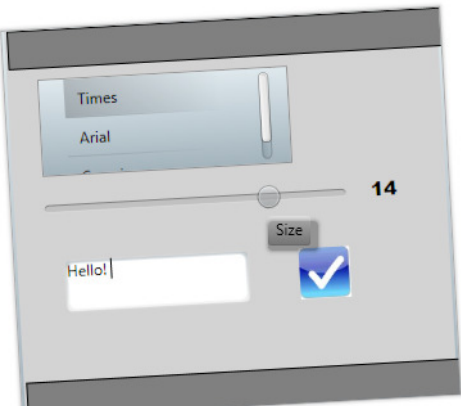
Formulaire de l'application de départ	Formulaire migré en prenant en compte les guidelines de styles
	

Table 2.1: Migration de l'aspect visuel d'un formulaire

2.3.3.5 Synthèse des guidelines pour la migration d'UI vers les tables interactives

Le corpus de guidelines décrit dans la section 2.3.3 sont des recommandations de haut niveau utiles et nécessaires pour la migration des UI vers une table interactive. Cette caractérisation de ce corpus a pour but de rendre une UI de départ tangible et collaborative pendant la migration tout en considérant les autres modes d'interactions des tables interactives et les aspects visuels de l'UI.

Les liens entre ces guidelines peuvent être synthétisés à l'aide du diagramme de la figure 2.11 qui présente les trois catégories de guidelines, les 7 guidelines pour les UI tangibles et collaboratives et les liens entre ces guidelines. Dans la catégorie des guidelines pour TUI, les guidelines G2 (Objets physiques tagués) et G3 (Forme des objets Physiques) spécifient les moyens d'interactions tangibles à choisir et à associer aux objets virtuels identifiés par la guideline G1.

Dans la catégorie des guidelines pour UI collaborative, la guideline nombre d'utilisateurs (G4) est spécifiée pour les aspects liés aux tâches par la guideline de couplage des tâches et utilisateurs (G5), la guideline spécifiant le partage de l'espace de travail (G6) entre les utilisateurs et la guideline préconisant l'utilisation 360° de l'UI (G7) permet l'accessibilité de l'UI pour tous les utilisateurs autour de la table.

Dans la catégorie comportant les autres guidelines pour les UI sur surface nous regroupons deux sous catégories; la première concerne les interactions tactiles et la seconde le style de l'UI migrée.

2.4 Concepts pour la migration d'UI

Dans les sections précédentes, nous avons identifié quels sont les problèmes liés à la migration d'UI vers une table interactives en partant d'un exemple d'application (à la section 2.1). Ensuite nous avons aussi identifié les spécificités des tables interactives qui sont importantes pour la migration des UI à travers ses propriétés caractéristiques et les guidelines. Dans cette section nous abordons les problématiques liées aux architectures des applications à migrer et les modèles d'interactions abstraites pour décrire des équivalences entre les modalités d'interactions.

2.4.1 Architecture des applications à migrer

Les applications à migrer sont des systèmes interactifs (SI) conçus en respectant un modèle d'architecture. Les modèles d'architectures sont des patrons de conception logiciel, ils préconisent des stratégies de répartition des services qui se traduisent par un ensemble de constituants logiciels. En général, la décomposition minimale des SI préconise une séparation entre le Noyau Fonctionnel (NF) et ceux de l'UI. Dans cette section nous présentons deux modèles d'architectures, d'abord le modèle ARCH [UIM92] qui se base sur des composants logiciels spécifiques et indépendants des plateformes, ensuite le modèle MVC [KP88] .

2.4.1.1 ARCH

ARCH [UIM92] est un modèle d'architecture qui se base sur des composants conceptuels du modèle de Seeheim [Pfa85]. Comme l'indique la figure 2.12, ce modèle permet une séparation entre le NF, le Contrôleur de Dialogue (CD) et la Présentation. Les deux pieds de l'arche sont des composants spécifiques à une plateforme; le composant NF décrit un domaine précis et les composants d'Interaction sont liés à des dispositifs du monde réel. Le CD gère l'enchaînement des tâches ainsi que les liens avec les objets des deux composants voisins. L'Adaptateur de domaine joue un rôle d'interface avec le composant NF pour corriger les différences de conceptions. Le composant Présentation est une boîte à outils virtuelle, telle que XVT [Val89] qui implémente les objets de présentations concrétisés finalement par les objets d'interaction de la bibliothèques graphique.

Migration d'une application respectant une architecture ARCH - Thevenin et al. [TC02] se basent sur l'architecture ARCH pour concevoir des applications avec des UI multi cibles et adaptables. Dans le cadre de la migration, cette approche consiste à remplacer les composants des interacteurs

logiques et physiques de l'UI à migrer pour qu'ils soient conformes aux principes de conception de la table interactive ciblée.

La migration des interacteurs physiques concerne la migration du composant d'interaction du modèle ARCH [TC02]. Elle consiste en un changement de bibliothèque graphique de l'application de départ et à utiliser celle de la plateforme d'arrivée. Ce type de migration permet de conserver la nature des composants graphiques mais leurs rendus sont distincts en fonction des plateformes et des bibliothèques graphiques. En considérant l'application CBA par exemple, la nature des éléments logiques de l'UI décrits dans le composant Présentation tels que Bouton, Liste d'images, etc. sont interprétés en fonction des bibliothèques graphiques (DiamondSpin, Surface SDK, etc.).

La migration des interacteurs physiques permet certes d'utiliser la bibliothèque graphique de la plateforme d'arrivée mais elle ne prend pas en compte les guidelines de la plateforme ciblée.

En effet en migrant les interacteurs physiques de l'UI CBA desktop par les éléments de la boîte à outils d'une table interactive sans prendre en compte les guidelines pour les TUI ou les UI collaboratives, le layout et les interactions de UI résultantes par exemple ne seront pas conformes aux spécificités de la table interactive ciblée.

Prise en compte des principes de conceptions - Dans le cadre de la migration d'une application respectant le modèle d'architecture ARCH [Pfa85], les guidelines identifiées à la section 2.3.3 sont utiles pour transformer le Composant de présentation et pour la génération du composant d'interaction en permettant par exemple la sélection des composants graphiques. En considérant l'UI CBA et les guidelines pour UI collaboratives, la guideline pour le partage d'espace de travail (G6) permet par exemple de réorganiser le layout de l'UI CBA en transformant le composant de présentation et la guideline d'utilisation 360 degré permet de choisir des panels avec des mouvements de rotation, de déplacement pendant la génération du composant d'interaction.

2.4.1.2 MVC

MVC [KP88] est un modèle d'architecture pour la conception des systèmes interactifs réutilisables. Il divise les applications en trois types de composants : le modèle(M), la vue(V) et le contrôleur(C). Le modèle est une représentation du domaine d'une application, il peut contenir des données, des services, etc. et il fait partie du NF d'une application. La vue est la structure de l'UI d'une application, elle est constituée des éléments d'une bibliothèque graphique. Le contrôleur est une interface entre le modèle, la vue et les dispositifs d'interactions en entrée.

Migration d'une application respectant une architecture MVC - En considérant une partie de l'UI de l'application CBA, la figure 2.13 représente des instances des différents éléments du modèle MVC. La vue est composée de *JPanel*, *JLabel*, *JComboBox* et d'une *JList* contenant des images. La sélection d'une catégorie du sélecteur *JComboBox* avec un clavier ou une souris modifie la liste d'images en faisant d'abord appel au contrôleur de *JComboBox* ensuite au modèle de *JList* pour mettre à jour la vue. La migration de cet artéfact d'UI vers une table interactive qui supporte la bibliothèque graphique DiamondSpin par exemple implique la modification de la vue en remplaçant les composants graphiques par leurs équivalents et en adaptant le code du contrôleur et du modèle pour remplacer les variables correspondant aux composants graphiques de la vue. Par exemple dans le cas où *DSCombobox* correspond à *JComboBox* dans DiamondSpin, le type de la variable *cb* du contrôleur doit être remplacé dans le contrôleur et le modèle.

Par conséquent, la migration de l'UI vers la table interactive avec cette hiérarchie implique une modification à la fois du contrôleur, de la vue et du modèle. Pour éviter ces modifications, il faut que

le code des applications à migrer respecte des heuristiques qui favorisent la réutilisation des différents composants de l'application. Parmi ces heuristiques on peut citer: la séparation entre les éléments spécifiques à la plateforme (PSM) et les éléments indépendants de la plateforme (PIM) au niveau du contrôleur et de la vue d'une part, et la non utilisation des éléments PSM du contrôleur de la vue au niveau du modèle d'autre part. Cette séparation permettra de changer les contrôleurs de dialogue spécifiques aux dispositifs d'interaction et les vues spécifiques aux bibliothèques graphiques.

La prise en compte guidelines pendant la migration des éléments du modèle MVC conçus en respectant les heuristiques préconisant de séparation PIM et PSM se fait comme pour les composants du modèle d'architecture ARCH.

2.4.1.3 Résumé

Nous avons présenté dans cette section 2.4.1 le modèle d'architecture de ARCH qui permet la migration des composants de l'UI sans modification des composants du NF. Le modèle d'architecture ARCH facilite aussi la prise en compte des guidelines pendant la migration. Par ailleurs, En considérant un modèle d'architecture MVC avec des composants qui respectent une séparation entre les éléments PSM et les éléments PIM, il est possible de migrer les composants d'UI sans modifier les composants du NF et aussi de prendre en compte les principes de conception pendant la migration. Le modèle MVC permet en une séparation entre les interactions en entrée décrites au niveau des contrôleurs et les interactions en sortie. Cette séparation facilite le **changement des interactions** en entrée.

2.4.2 Modèle d'interactions abstraites

La migration d'une UI desktop vers les tables interactives est un processus qui implique un changement de modalité d'interaction tel que nous l'avons montré à la section 2.3.1.3. Le **changement des modalités d'interactions** doit préserver les actions de l'UI de départ et l'adapter aux dispositifs d'interactions de l'UI d'arrivée. Le changement de modalités d'interactions est possible en décrivant des équivalences entre les différentes modalités d'interactions d'une UI. Ce qui peut se faire en se basant sur un modèle indépendant des dispositifs d'interactions.

Le modèle d'interactions abstraites permet de décrire des équivalences entre deux modalités d'interactions en jouant un rôle de langage pivot.

Dans cette section nous présentons deux modèles d'interactions abstraites et nous en déduirons quelques caractéristiques d'un modèle d'interactions abstraites pour la migration.

2.4.2.1 Modèle de Vlist

Vlist et al. [VNHF11] proposent les primitives d'interaction (*interaction primitives*) qui constituent les plus petits éléments d'interactions adressables ayant une relation significative avec l'interaction elle-même. Selon eux, il est possible de décrire les capacités d'interactions des dispositifs d'interactions à l'aide des primitives d'interactions et ensuite la relation entre les dispositifs et l'UI sera décrite en faisant un mapping sémantique. Les primitives d'interactions sont associées à chaque dispositif d'interaction, elles sont choisies, identifiées et associées aux dispositifs par le concepteur de l'UI. Par exemple les primitives d'interactions Up, Down correspondent aux touches étiquetées '+' et '-', ces primitives d'interactions ont des types de données et des valeurs, elles sont mappées sur des composants graphiques tels que les contrôles de volume pour permettre l'utilisation

de composants avec ces touches. L'objectif de ce mapping sémantique est de faciliter son adaptation en fonction des contextes.

Dans le cadre de la migration, la mise en correspondance des dispositifs d'interactions des plateformes de départ et d'arrivée semble être possible en utilisant une description sémantique des capacités des dispositifs d'interaction d'une plateforme par les primitives d'interactions. Pour cela les primitives d'interactions doivent pouvoir décrire tous les dispositifs d'interactions en se basant sur un seul vocabulaire. En effet Up et Down sont certes adaptés pour les touches '+' et '-' mais en considérant un écran tactile par exemple il faut décrire les différents types de contacts (simple, double, multiple, toucher et glisser, etc.).

2.4.2.2 Modèle de Gellersen

Gellersen [GH95] quant à lui propose un modèle d'interactions abstraites qui a pour but de décrire les interactions en entrée et en sortie indépendamment des modalités d'interactions. Ce modèle est aussi indépendant d'un domaine ou d'une application. Il est présenté à la figure 2.14 et il décrit une hiérarchie des interactions en entrée et en sortie. Les interactions en entrée sont raffinées en deux catégories, les interactions d'entrée de données telles que *Editor*, *Valuator* (éditer du texte) et *Option* (sélectionner un élément d'une liste) qui sont des sous classes de *Entry* d'une part et les interactions de *Command*⁵ et de *Signal*⁶ d'autre part. Les interactions en sortie sont aussi de deux types : les messages (alertes, confirmation, etc.) et la vue qui permet l'affichage des données et des composants de l'UI.

Dans le cadre du changement de modalité d'interaction entraîné par la migration, ce modèle permet de décrire les interactions des différents composants graphiques de l'UI de manière indépendante des modalités d'interactions. Par exemple une fenêtre d'authentification comportant des champs de texte et des boutons, les champs de texte seront représentés par les Editor qui sont des interactions à la fois en entrée et en sortie et les boutons seront représentés par des Command dans un modèle abstrait.

Ce modèle identifie à la fois les interactions en entrée et en sortie de haut niveau et indépendamment des modalités d'interactions. Dans le cadre de la migration de l'UI CBA par exemple, les interactions telles que la rotation, le déplacement ou le redimensionnement qui sont liés aux guidelines de la table interactives doivent être interprétés comme des commandes. Cette interprétation n'est pas générique et dépend du concepteur des interactions, en effet le déplacement peut être considéré comme une rotation en modifiant l'angle par exemple. De manière générale, le modèle de Gellersen est un modèle d'interactions abstraites qui nécessite la spécification de certaines interactions indépendantes des modalités d'interactions pendant la migration.

2.4.2.3 Résumé

Deux caractéristiques sont essentielles pour les modèles d'interactions abstraites :

- leurs indépendances d'une modalité d'interaction
- et leur capacité à décrire toutes interactions du langage d'une modalité.

⁵Les Command sont des interactions en entrée de contrôle avec des paramètres (exemple copier texte, coller texte, etc.)

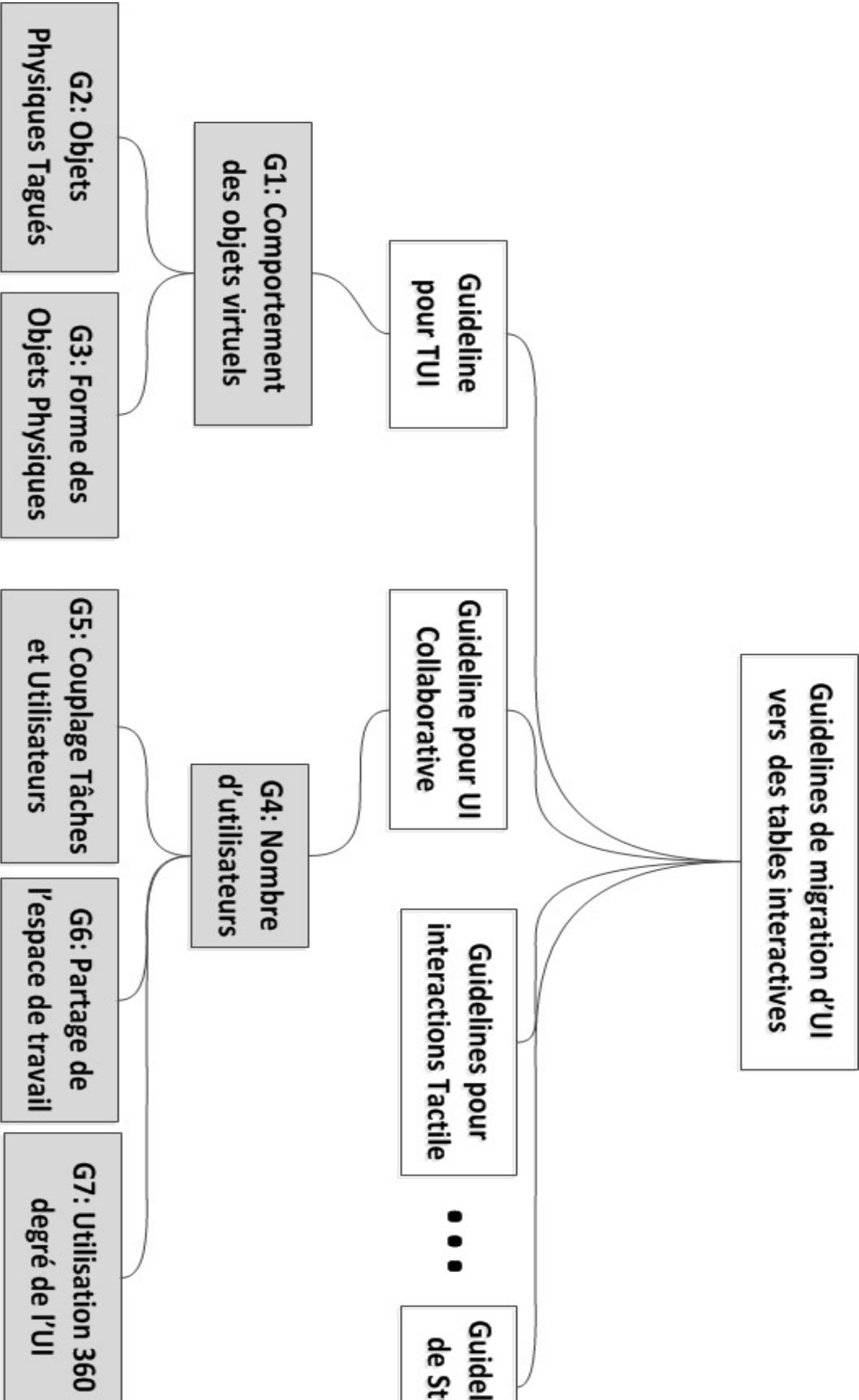
⁶Les Signal sont des interactions en entrée sans paramètres (exemple valider)

2.5 Conclusion

Dans ce chapitre nous avons étudié les problèmes liés à la migration d'UI vers une table interactive qui consiste à:

- transformer le layout de l'UI de départ conformément aux guidelines pour les UI collaboratives
- changer les modalités d'interactions de l'application de départ conformément aux guidelines pour les UI tangibles et tactiles
- adapter la taille et le style de l'UI de départ aux recommandations de la plateforme d'arrivée.

Dans le chapitre suivant, nous étudions quelques mécanismes de migration des UI en faisant ressortir les problématiques liées à la prise en compte des guidelines et au changement de modalités d'interactions de ces mécanismes de migration.



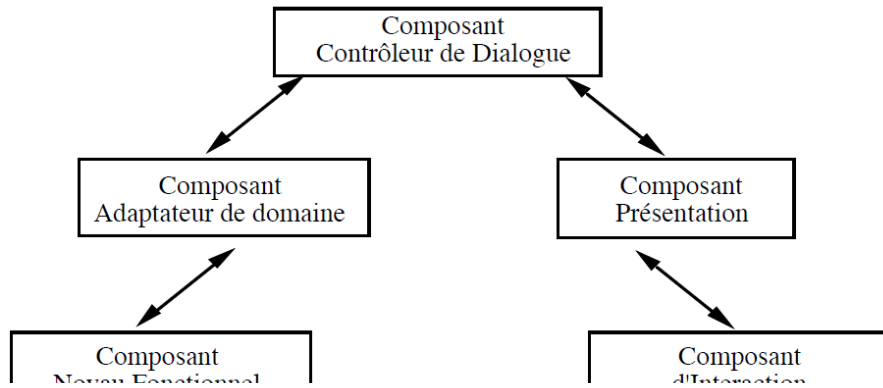


Figure 2.12: Composants du modèle ARCH

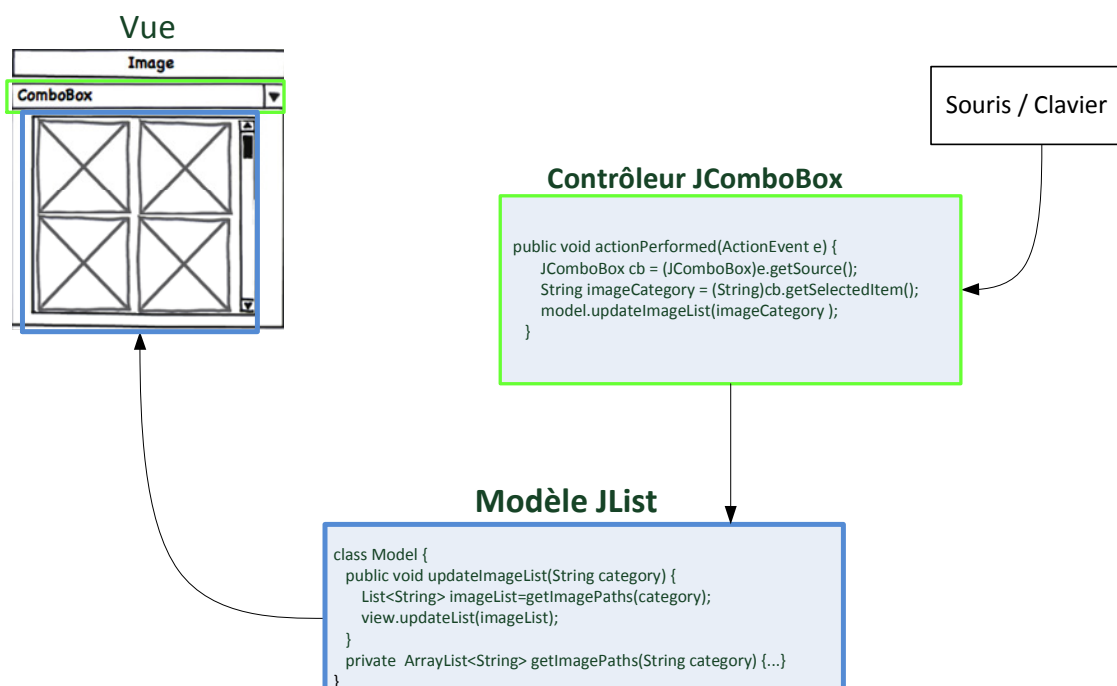


Figure 2.13: Modèle Vue Contrôleur

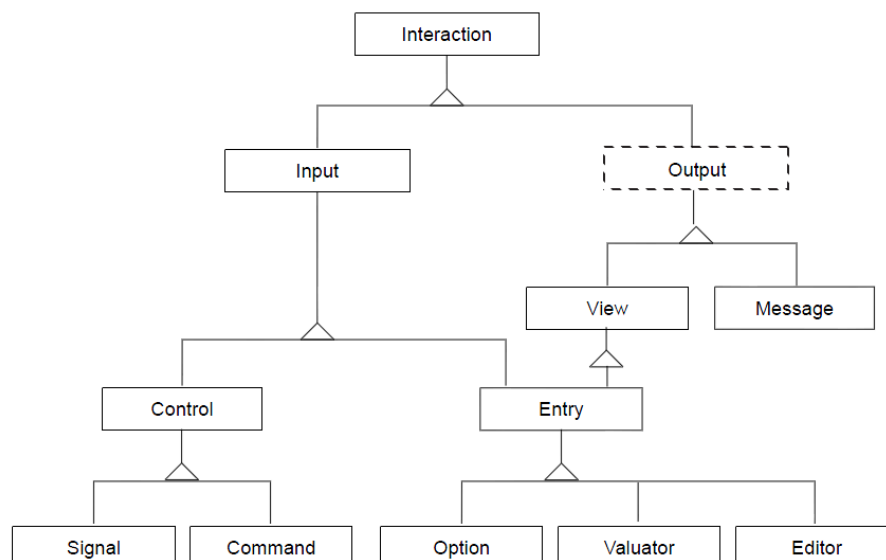


Figure 2.14: Modèle d'interactions abstraites de Gellerson

Tables interactives et Migration des UI

Part III

Contributions

Approches de migration des UI

Modélisation des UI

Mécanismes de migration des UI

Validations du processus

Part IV

Conclusion et Perspectives

Conclusions et perspectives

Bibliography

- [App95] Apple Computer Inc. *Macintosh Human Interface Guidelines*. Addison-wesley Publishing, 1995.
- [Bes10] Guillaume Besacier. *Interactions post-WIMP et applications existantes sur une table interactive*. PhD thesis, UNIVERSITÉ PARIS-SUD 11, 2010.
- [BL00] Michel Beaudouin-Lafon. Instrumental interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00*, pages 446–453, New York, New York, USA, April 2000. ACM Press.
- [BRNB07] Guillaume Besacier, Gaétan Rey, Marianne Najm, and Stéphanie Buisine. Paper Metaphor for Tabletop Interaction Design. In *HCII'07 Human Computer Interaction International*, pages 758–767, 2007.
- [C93] Laurence " and Joëlle Coutaz. A design space for multimodal systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93*, pages 172–178, New York, New York, USA, May 1993. ACM Press.
- [CCB⁺02] Gaëlle Calvary, Joëlle Coutaz, Laurent Bouillon, Murielle Florins, Quentin Limbourg, L. Marucci, Fabio Paternò, Carmen Santoro, N. Souchon, David Thevenin, and Jean Vanderdonckt. CAMELEON Project. Technical report, 2002.
- [Cre01] Murray Crease. *A Toolkit Of Resource-Sensitive, Multimodal Widgets Murray Crease*. PhD thesis, 2001.
- [DL01] Paul Dietz and Darren Leigh. DiamondTouch. In *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, page 219, New York, New York, USA, November 2001. ACM Press.
- [GH95] Hans-w Gellersen and Hans-W. Gellersen. Modality Abstraction : Capturing Logical Interaction Design as Abstraction from "User Interfaces for All". In *1st ERCIM Workshop on "User Inter- faces for All"*. ERCIM, 1995.
- [IU97] Hiroshi Ishii and Brygg Ullmer. Tangible bits. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97*, pages 234–241, New York, New York, USA, March 1997. ACM Press.
- [KKM03] M. Kassoff, D. Kato, and W. Mohsin. Creating GUIs for web services. *IEEE Internet Computing*, 7(5):66–73, September 2003.
- [KLL⁺09] Sébastien Kubicki, Sophie Lepreux, Yoann Lebrun, Philippe Dos Santos, Christophe Kolski, and Jean Caelen. *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*, volume 5612 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [KOR⁺11] André Kalawa, Audrey Occello, Michel Riveill, Nice Sophia, and Antipolis Cnrs. *SUIT : a framework for Supporting UI Translation*. 2011.

- [KP88] Glenn E Krasner and Stephen T Pope. A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. Technical report, 1988.
- [Kub11] S bastien Kubicki. *Contribution   la prise en consid ration du contexte dans la conception de tables interactives sous l'angle de l'IHM , application   des contextes impliquant table interactive RFID et objets tangibles*. PhD thesis, UNIVERSIT  DE VALENCIENNES, 2011.
- [Mar97] Panagiotis Markopoulos. A compositional model for the formal specification of user interface software. *Doctor*, (March), 1997.
- [Mic11] Microsoft. Microsoft Surface 2 Design and Interaction Guide. Technical Report July, 2011.
- [Mic12a] Microsoft. MSDN XAML, 2012.
- [Mic12b] Microsoft WPF. WPF, 2012.
- [Mic12c] Microsoft XNA. XNA, 2012.
- [Mit] Mitsubishi Electric Research Laboratoriesb. MERL .
- [NC94] Laurence Nigay and Jo lle Coutaz. *Conception et mod lisation logicielles des syst mes interactifs: application aux interfaces multimodales = Software design and implementation of interactive systems: a case study of multimodal interfaces*. PhD thesis, 1994.
- [Pfa85] G. E. Pfaff. User Interface Management Systems. June 1985.
- [PSO11] Fabio Patern , Carmen Santoro, and Rasmus Olsen. Migratory Interactive Applications for Ubiquitous Environments. pages 9–23, 2011.
- [PZ10] Fabio Patern  and Giuseppe Zichittella. Desktop-to-Mobile Web Adaptation through Customizable Two-Dimensional Semantic Redesign. pages 79–94, 2010.
- [Ste12] Gideon Steinberg. Natural User Interfaces. 2012.
- [SVFR04] Chia Shen, Fr d ric D. Vernier, Clifton Forlines, and Meredith Ringel. DiamondSpin: an extensible toolkit for around-the-table interaction. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, pages 167–174, New York, New York, USA, April 2004. ACM Press.
- [TC02] David Thevenin and Jo lle Coutaz. Adaptation des IHM : Taxonomies et Archi. Logicielle. In *IHM 2002*, pages 207–210, 2002.
- [TKB78] Andrew S. Tanenbaum, Paul Klint, and A. P. Wim B hm. Guidelines for software portability. *Softw., Pract. Exper.*, 8(6):681–698, 1978.
- [UI97] 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 technolog*, pages 223 – 232, 1997.
- [UIM92] UIMS. A METAMODEL FOR THE RUNTIME ARCHITECTURE OF AN INTERACTIVE SYSTEM The UIMS Tool Developers Workshop *. In *SIGCHI Bulletin*, number January, 1992.

- [Val89] Valdès. A Virtual Toolkit for Windows and the Mac. *Byte*, 11(3):209–216., 1989.
- [Van97] Jean Vanderdonckt. Conception assistée de la présentation d’une interface homme-machine ergonomique pour une application de gestion hautement interactive. 1997.
- [VNHF11] Bram J J Van Der Vlist, Gerrit Niezen, Jun Hu, and Loe M G Feijs. Interaction Primitives : Describing Interaction Capabilities of Smart Objects in Ubiquitous Computing Environments. Number September, pages 13–15, 2011.
- [WGM08] Xin Wang, Yaser Ghanam, and Frank Maurer. From desktop to tabletop: Migrating the user interface of agileplanner. In Peter Forbrig and Fabio Paternò, editors, *Engineering Interactive Systems*, pages 263–270, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [XSS⁺04] Steven Xia, David Sun, Chengzheng Sun, David Chen, Q Xia, D Sun, C Sun, and D Chen. Leveraging Single-user Applications for Multi-user Collaboration : the CoWord Approach Categories and Subject Descriptors. In *CSCW 2004*, pages 162–171, 2004.