



**HAL**  
open science

# Couplage de Ressources d'Interaction en Informatique Ambiante

Nicolas Barralon

► **To cite this version:**

Nicolas Barralon. Couplage de Ressources d'Interaction en Informatique Ambiante. Interface homme-machine [cs.HC]. Université Joseph-Fourier - Grenoble I, 2006. Français. NNT : . tel-00344173

**HAL Id: tel-00344173**

**<https://theses.hal.science/tel-00344173>**

Submitted on 3 Dec 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE  
présentée par

**Nicolas Barralon**

pour obtenir le titre de  
DOCTEUR de L'UNIVERSITE JOSEPH-FOURIER-GRENOBLE I  
(arrêtés ministériels du 5 juillet 1984 et du 30 mars 1992)  
Spécialité : **Informatique**

## **Couplage de Ressources d'Interaction en Informatique Ambiante**

date de soutenance : 8 décembre 2006

### **Composition du Jury :**

Directeur de thèse :	Mme. Joëlle Coutaz
Président :	Mr. Jean Caelen
Rapporteurs :	Mme. Jocelyne Nanard Mr. Michel Beaudouin-Lafon
Examineurs :	Mr. Jean Vanderdonckt Mr. Gilles Privat

Thèse préparée au sein du laboratoire  
Communication Langagière et Interaction Personne-Système  
Fédération IMAG  
Université Joseph Fourier - Grenoble I



# *Remerciements*

En comparaison à la rédaction de cette thèse, je crois que ces remerciements sont encore plus difficiles à écrire, non pas que j'y mette de la mauvaise volonté, mais c'est plutôt la peur d'oublier de remercier certains d'entre vous. Pour avoir bonne conscience : « merci à tous! ». Essayons maintenant d'entrer dans le détail... mais ne voyez pas de signification particulière dans l'ordre de ces remerciements.

Mes premiers remerciements vont à Joëlle, tout d'abord sur un plan scientifique, car sans elle cette thèse n'en serait pas une. J'ai réellement apprécié de travailler avec elle, toujours droit au but, remettre en cause son travail, lire, relire,... Sur un plan un peu plus personnel, je salue sa disponibilité et sa gentillesse car je suis persuadé que bien souvent, dans son planning, d'autres tâches que ma thèse étaient prévues.

En deuxième lieu, je tiens à remercier Michel Beaudouin-Lafon et Jocelyne Nanard d'être rapporteurs de cette thèse, en particulier pour les diverses remarques qu'ils m'ont prodiguées et qui me permet de prolonger ma réflexion sur certains points de cette thèse. J'associe à ces remerciements Jean Caelen, Jean Vanderdonckt et Gilles Privat qui m'ont fait l'honneur d'accepter d'être examinateur de mes travaux.

Je salue maintenant l'équipe IIHM dans laquelle j'ai effectué mes travaux de thèse, mon DEA et mon stage de maîtrise. En tout premier lieu, les permanents, Joëlle (c'est déjà fait), Gaëlle, Laurence, Sophie, François et Renaud avec une mention toute particulière pour Gaëlle sans laquelle je n'aurais jamais mis les pieds dans l'équipe IIHM ni même dans la recherche. Pour les non-permanents (dont certains le sont maintenant mais ailleurs.... que c'est compliqué), Philippe, Yann, Gaëtan, Kris, Jullien, Benoît, Alex, Lionel, Tung, Xavier, Jean-Sébastien, David, Anne, Damien, oulala, j'en oublie, c'est sûr, car aucun homme n'est jamais assez fort pour ce calcul. Pour plus de sûreté, tous les locataires des bureaux : B202 (si si y'en a eu), B203, B204 (en particulier les membres de [gossip@imag.fr](mailto:gossip@imag.fr)), B205, B206, B207, B209, B302. J'associe à ces remerciements les membres de l'IUT2 que j'ai pu rencontrer pendant mes deux années (supers) d'ATER. Quel beau métier, professeur !

---

Sur un plan plus personnel, je tiens à remercier, pour leur soutien et leur aide, ma Famille au sens large du terme, mes parents, mon frère, ma belle-soeur et ma petite femme bien sûr : pas la jeune informaticienne qui compilait le C à longueur de journée, l'autre celle qui supporte toutes mes sautes d'humeur, et une relectrice bienveillante. Je lui associe « le 0,5 qui fait de nous 2,5 ; un autre bonheur à venir ». Je remercie également toute ma belle-famille.

Je terminerai comme l'a fait un de mes illustres confrères par une analogie avec l'escalade (aucun lien avec « l'escalope sur une belle salade »). Pour les grandes expéditions, d'avoir un très bon second (ou un très bon premier) de cordée, d'avoir toute confiance et de pouvoir toujours compter sur lui. J'ai cette chance depuis plus de 27 ans. Merci.

Enfin, une petite phrase pour illustrer cette thèse : « Vive le couplage ! Vive le couplage ! Rien n'est jamais assez beau quand il s'agit de grandeur :- ) ».

# Table des matières

---

	REMERCIEMENTS	3
	TABLE DES MATIÈRES	5
	TABLE DES FIGURES	9
	TABLE DES TABLEAUX	13
CHAPITRE I	INTRODUCTION	17
	1.Sujet .....	19
	2.Motivations.....	22
	3.Objectifs et Approche.....	23
	4.Portée de l'étude.....	24
	5.Organisation du manuscrit.....	25
CHAPITRE II	TEC : UNE TAXONOMIE POUR LE COUPLAGE DE RESSOURCES D'INTERACTION	27
	1.Couplage d'entités : définitions .....	28
	1.1.Définition du couplage d'entités .....	29
	1.2.Entités mixtes et ressources d'interaction .....	30
	1.3.Composition de couplages.....	32
	2.Analyse des taxonomies de l'état de l'art .....	33
	2.1.L'espace de conception des briques.....	33
	2.2.L'espace de conception des interfaces tangibles (Fishkin) .....	34
	3.TEC : une nouvelle taxonomie pour le couplage .....	38
	3.1.Caractérisation des entités .....	38
	3.2.Caractérisation des fonctions.....	42

---

	TABLE DES MATIÈRES	5
--	--------------------	---

	3.3.Technique d'interaction .....	43
	3.4.Caractérisation de l'instanciation du couplage .....	48
	4.Synthèse.....	49
<b>CHAPITRE III</b>	<b>COUPLAGE DE RESSOURCES D'INTERACTION ET SON ÉTAT DE L'ART SELON TEC</b>	<b>51</b>
	1.Hétérogénéité de superclasses : mixage PN et MN.....	52
	1.1.Couplage PN : MetaDesk, MCRpd et DataTiles.....	52
	1.2.Couplage PN : AmbientRoom et le cube de contrôle télévision .....	54
	1.3.1Couplage MN : MediaBlocks, Sentient Data Access et Augmented Surfaces	57
	1.4.Couplage MN : Bricks, Table Magique et Sensetable .....	60
	2.Homogénéité de superclasses : PP et MM .....	63
	2.1.Couplage PP : Geospace, un vrai couplage physique.....	63
	2.2.Couplage MM : Pebbles, Dynamo, Programmable Beads, Smart-Its Friends,	64
	Icon.....	
	2.3.Couplage MM : Triangles, Building blocks, et la Table Magique .....	68
	2.4.Couplage MM : Augmented Surfaces, Synchronous Gesture, Connectables,	72
	DataTiles .....	
	2.5.Couplage MM : Smoblet et GAS .....	75
	3.Analyse synthétique.....	78
	3.1.Entités : généralité et hétérogénéité.....	78
	3.2.L'ensemble des fonctions cibles : cardinalité et généralité.....	80
	3.3.Couplage et instanciation .....	81
	3.4.Techniques d'interaction.....	82
	4.Synthèse.....	82
<b>CHAPITRE IV</b>	<b>CADRE D'ANALYSE POUR LA DYNAMIQUE DES COUPLAGES DE RESSOURCES D'INTERAC-</b>	<b>85</b>
	<b>TION</b>	
	1.Cycle de vie d'un couplage .....	87
	1.1.Couplage de ressources d'interaction : rappel de la définition .....	87
	1.2.Un état : trois critères de vérité.....	87
	1.3.Transitions .....	88
	1.4.Cycle de vie .....	88
	1.5.Illustrations .....	89
	1.6.Analyse .....	97
	2.Propriétés ergonomiques d'un couplage .....	101
	2.1.Souplesse de l'interaction .....	101
	2.2.Robustesse de l'interaction .....	102
	3.Génération de couplages.....	104
	3.1.Exemple de base .....	104
	3.2.Notation graphique .....	104
	3.3.Algorithme de génération .....	105
	3.4.Application à l'exemple .....	106
	3.5.Formalisation mathématique .....	111
	3.6.Les fonctions et la compatibilité.....	115
	4.Synthèse.....	120
<b>CHAPITRE V</b>	<b>TECHNIQUES D'INTERACTION POUR LE COUPLAGE DE RESSOURCES D'INTERACTION</b>	<b>121</b>

---

CHAPITRE VI

1.Introduction .....	122
2.Notation .....	122
2.1.Les éléments de la notation .....	122
2.2.Grammaire d'une désignation .....	124
2.3.Illustration.....	129
2.4.Classes de désignation .....	133
2.5.Couplage dans I-AM .....	136
3.Aspect fonctionnel d'une technique d'interaction .....	141
3.1.Technique monofonctionnelle / plurifonctionnelle .....	141
3.2.Techniques génériques .....	143
4.Synthèse.....	144
<hr/>	
CONCLUSION .....	145
1.Synthèse de la contribution .....	146
1.1.Définition du couplage et TEC : Taxonomy For Entity Coupling .....	146
1.2.Cadre d'analyse pour la dynamique des couplages.....	148
1.3.Technique d'interaction .....	148
1.4.Implémentation.....	149
2.Limites et perspectives .....	149
2.1.Définition du couplage et TEC.....	149
2.2.Cadre d'analyse du couplage .....	149
2.3.Technique d'interaction .....	150
2.4.Implémentation.....	151
3.Synthèse.....	152
1.I-AM (Interaction Abstract Machine) .....	154
1.1.Principe.....	154
1.2.I-AM Description .....	154
1.3.Les contexteurs .....	156
2.Ambient-Desktop .....	156
2.1. Couplage d'écrans.....	157
2.2.Gestion de fichiers .....	161
3.Synthèse.....	162
<hr/>	
BIBLIOGRAPHIE .....	163





# Table des Figures

---

	REMERCIEMENTS	3
	TABLE DES MATIÈRES	5
	TABLE DES FIGURES	9
	TABLE DES TABLEAUX	13
CHAPITRE I	INTRODUCTION	17
	De l'informatique conventionnelle vers l'informatique ambiante (d'après [Lyytinen et Yoo 2002]).....	20
	Le lapin Nabaztag [Nabaztag]. .....	21
	Exemples de couplage opportuniste de ressources d'interaction. ....	22
	La ressource d'interaction au centre de l'interaction. Les flèches indiquent le sens des échanges. ....	25
CHAPITRE II	TEC : UNE TAXONOMIE POUR LE COUPLAGE DE RESSOURCES D'INTERACTION	27
	Relations entre les différentes classes d'entités, physiques, numériques ou mixtes et ressources d'interaction.....	31
	Composition de couplages.....	32
	Couverture des trois classes d'entités physiques par la taxonomie de Fishkin, d'après [Fishkin 2004] .....	37
	Vue globale des éléments taxonomiques de TEC selon les quadrants entités, fonctions, technique d'interaction et couplage. ....	37
	Classification des entités selon leur niveau de généralité.....	39
	Type de mixage ou hétérogénéité de superclasses. ....	40
	Hétérogénéité de classes.....	41
	Hétérogénéité de niveaux d'abstraction. ....	42
	Classement des techniques de désignation. ....	43

---

	TABLE DES FIGURES	9
--	-------------------	---

	Deux désignations dans Synchronous Gesture [Hinckley 2003]. .....	44
	Différentes formes de désignation instrumentale et de représentation de l'entité désignée. ....	46
	Classement des techniques de représentations d'entité.....	47
<hr/>		
CHAPITRE III	COUPLAGE DE RESSOURCES D'INTERACTION ET SON ÉTAT DE L'ART SELON TEC	51
	Manipulation du phicon représentant le Dôme du MediaLab, sur le MetaDesk. ....	53
	A gauche, différentes tuiles pré-imprimées ou gravées. A droite, les mêmes tuiles en action sur la surface dédiée.....	53
	Ambient Room. ....	55
	Couplage du cube physique et du cube virtuel.....	56
	Vue d'ensemble du prototype Augmented Surfaces.....	58
	Blocks et Media Sequencer. ....	58
	Utilisation du PDA dans Sentient Data Access.....	59
	Une brick servant pour la manipulation d'un objet virtuel. ....	60
	Utilisation simultanée de quatre jetons sur la Table Magique. ....	61
	Taxonomie appliquée aux Bricks, à la Table Magique et à Sensetable. ....	62
	Couplage physique de deux phicons dans le MetaDesk.....	63
	Couplage dans Peebles. ....	65
	Les points d'interaction dans Dynamo.....	66
	Couplage de Programmable Beads pour la confection d'un collier. ....	67
	Configuration ICon pour le couplage d'un clavier avec une souris pour la fonction "Modifier".....	67
	Composition de Triangles pour former une histoire.....	69
	Couplage de BuildingBlocks. ....	70
	Couplage de jetons dans la Table Magique.....	71
	Couplage dans Augmented Surfaces. ....	72
	Couplage de tablettes dans Synchronous Gesture. ....	73
	Couplage de deux Connectables.....	73
	Couplage de deux DataTiles.....	74
	Couplage de DataTiles par un geste. ....	74
	Couplage et coopération de Smart Objects. ....	76
	Couplage d'un eBook avec une eLamp. ....	77
<hr/>		
CHAPITRE IV	CADRE D'ANALYSE POUR LA DYNAMIQUE DES COUPLAGES DE RESSOURCES D'INTERACTION	85
	Automate du cycle de vie du couplage (r1, c, r2).....	89
	Le SCP ; le couplage d'une caméra et d'un vidéo projecteur actionnés par un moteur. ....	90
	Les surfaces planes de l'environnement du SCP. ....	90
	Le Portable Display. ....	91
	Automate résultant de la « demi-fermeture transitive ». ....	92
	Interaction avec une liste de surfaces. ....	92
	Automates des couplages pour la technique à l'aide de menus d'après [Borkowski 2006].....	93
	Interface graphique pour l'interaction au laser. ....	94
	Automates des couplages pour la technique d'interaction à distance [Borkowski 2006] ....	95
	Automates des couplages pour la technique avec PDS d'après [Borkowski 2006]... ..	96
	Symétrie graphique des automates de la technique de déplacement entre surfaces.. ..	98
	Nouveaux automates pour la technique de déplacement entre surfaces.....	99
	Structure commune à deux automates.....	100
	Notation graphique appliquée au cas 2 de l'exemple.....	105

Condifiguration initiale. ....	106
Configuration terminale du cas 2. ....	107
Couplages évalués pour les chemins de type s-écran1-écran2 en haut, écran1-écran2-s en bas. ....	108
Couplages évalués pour les chemins de type s-clavier1-écran2 en haut, clavier1-écran2-s en bas. ....	109
Configuration terminale pour le cas 3. ....	110
Couplages évalués pour le couplage initiateur c5. ....	110
Ecran1, souris1 et clavier1 vus comme une seule entité ; écran2 et souris2 également. ....	116
Génération complète de couplages. ....	117
En haut, l'écran1 et l'écran2 sont considérés comme une seule entité. En bas, les couplages terminaux. ....	117
Configuration terminale en considérant l'écran1, souris1, clavier1 comme privés, l'écran2 et la souris2 comme publics. ....	118

CHAPITRE V

TECHNIQUES D'INTERACTION POUR LE COUPLAGE DE RESSOURCES D'INTERACTION 121

Récapitulatif des types d'entités et des relations.....	123
Colocalisation de symboles. ....	124
Exemple d'interaction entre une souris manipulée par un utilisateur et un écran ...	124
L'entité physique à désigner joue également le rôle d'instrument de désignation..	126
Cinq types de désignations : Physique-Physique direct, Numérique-Physique, Physique-Physique, Numérique-Numérique, Physique-Numérique. ....	127
Chaînes de représentations : iprp, inrn, eprp, enrn. ....	128
Description graphique d'une technique avec modalité orale. ....	130
Forme condensée de la notation graphique. ....	130
Description graphique d'une technique avec modalité orale comportant une complémentarité. ....	131
Description graphique du couplage dans Synchronous Gesture avec une tablette mobile et une autre fixe. ....	132
Description graphique du couplage dans Synchronous Gesture avec deux tablettes mobiles. ....	132
Description graphique du couplage dans Synchronous Gesture avec deux tablettes mobiles tenues par deux utilisateurs.....	133
Quatre grandes classes de désignation.. ....	134
Classes de techniques d'interaction non présentes dans les prototypes de recherche... ..	136
Technique Click'n'Couple basé sur deux clics synchronisés pour effectuer le couplage. ....	137
Feedback de couplage en deux écrans.....	138
Technique Click'n'Couple permettant l'ajustement de la taille des passerelles.....	139
Ajustement dynamique des points de liaison entre surfaces (partie gauche) et évaluation a priori de l'espace unifié potentiel (partie droite). ....	139
Description de la désignation de la technique Click'n'Couple réalisée au doigt. ....	140
Description de la désignation de la technique Click'n'Couple réalisée à l'aide de deux souris.....	140
Description de la désignation de la technique Click'n'Couple réalisée à l'aide de deux stylets.....	141
Configurateur de surfaces sous Mac Os X pour les fonctions « extension de la surface d'affichage » ou « mirroring ». ....	143

CHAPITRE VI

CONCLUSION 145

Principe d'I-AM. ....	155
-----------------------	-----

---

Un espace d'interaction dont les deux surfaces sont couplables.....	157
Problèmes d'interprétation avec le Click'n'Couple. ....	158
Portables équipés de notre capteur de proximité. ....	159
Configurateur avec deux surfaces, leurs bords physiques et une liaison.....	160
Surfaces couplées. ....	162

---

BIBLIOGRAPHIE

163

# *Table des tableaux*

---

	REMERCIEMENTS	3
	TABLE DES MATIÈRES	5
	TABLE DES FIGURES	9
	TABLE DES TABLEAUX	13
CHAPITRE I	INTRODUCTION	17
CHAPITRE II	TEC : UNE TAXONOMIE POUR LE COUPLAGE DE RESSOURCES D'INTERACTION	27
CHAPITRE III	COUPLAGE DE RESSOURCES D'INTERACTION ET SON ÉTAT DE L'ART SELON TEC	51
CHAPITRE IV	CADRE D'ANALYSE POUR LA DYNAMIQUE DES COUPLAGES DE RESSOURCES D'INTERACTION	85
CHAPITRE V	TECHNIQUES D'INTERACTION POUR LE COUPLAGE DE RESSOURCES D'INTERACTION	121
CHAPITRE VI	CONCLUSION	145
	BIBLIOGRAPHIE	163

---



---

A ma Famille,





---

*Chapitre I      Introduction*



---

## *Avant-propos*

---

Sous le règne d'Henri III (1551-1589), se joue une grande révolution : la fourchette s'impose sur les tables. Un demi-siècle plus tôt, François Ier (1494-1547) lançait la mode de l'assiette. Quelques années plus tard, le Cardinal de Richelieu (1585-1642), excédé de voir les gens se curer les dents avec la pointe de leur poignard, fait arrondir les lames des couteaux. À la suite de ces événements, la fourchette et le couteau sont utilisés conjointement. Certes, ces instruments conservent leur fonction originale, mais ils sont associés, soit pour couper (la fourchette tenant les aliments), soit pour manger (le couteau poussant les aliments dans la fourchette). Au-delà de l'aspect historique de l'utilisation conjointe du couteau et de la fourchette, il nous arrive fréquemment d'associer des objets pour obtenir de nouvelles fonctions. Par exemple, tracer des traits rectilignes par l'usage conjoint du crayon (qui, seul, permet de dessiner à main levée) et de la règle (conçue pour mesurer). Si l'association d'objets est un phénomène naturel de la vie courante, il est légitime d'en étudier le prolongement en Interaction Homme-Machine (IHM) dans l'esprit de l'informatique ambiante, notamment.

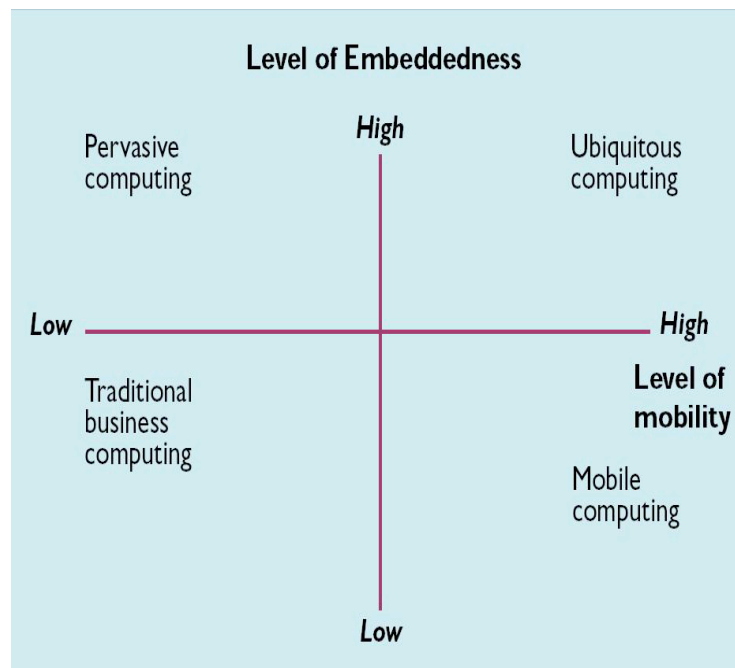
### *1. Sujet*

---

Ces travaux de recherche doctorale s'inscrivent dans le domaine de l'Interaction Homme-Machine (IHM) en informatique ambiante avec, comme sujet d'étude, le couplage d'une classe particulière d'objets : les ressources d'interaction.

D'après Lachenal [ACCORD], une ressource d'interaction est un rôle (c'est-à-dire une fonction) qu'un objet physique assure lorsqu'il sert de support aux échanges d'information entre un système informatique et un utilisateur. Dans notre étude, le rôle de ressource d'interaction résulte du couplage d'un objet matériel avec une entité numérique. En informatique conventionnelle, le clavier, la souris et l'écran jouent le rôle de ressources d'interaction s'ils sont couplés aux pilotes correspondants du système d'exploitation. Aujourd'hui, la convergence des réseaux ad hoc sans fil, des micro- et nanosystèmes, et la maturation de trente années de recherche en réseau et système répartis, en perception artificielle et en IHM, permettent d'envisager le passage d'une informatique confinée et statique, à une informatique ambiante reconfigurable [Myers 2002]. L'IHM en informatique ambiante n'implique pas nécessairement la disparition de l'ordinateur tel que nous le connaissons aujourd'hui. Il convient néanmoins d'en étudier l'évolution.

D'après Lyytinen et Yoo [Lyytinen et Yoo 2002], l'évolution vers l'informatique ambiante se résume par les degrés de mobilité et d'intégration dans le monde physique (Figure 1). La mobilité désigne la capacité pour l'utilisateur d'avoir accès en tout lieu, au service numérique souhaité. Le service suit l'utilisateur (ou en donne l'impression). Cette possibilité résulte de la miniaturisation des objets qui, de confinés au bureau, deviennent portables et connectables au réseau sans fil. Toutefois, un service purement mobile ne tient pas compte de son environnement. Cette dépendance à l'environnement relève du degré d'intégration. Un système à forte intégration avec le monde physique est en mesure d'extraire de l'information sur ce monde pour fournir des services adaptés. Réciproquement, l'environnement sait détecter les allées et venues des unités de calcul et des objets, forgeant ainsi une dépendance étroite entre les mondes physique et numérique. Dans l'espace de classification de la Figure 1, l'informatique conventionnelle correspond au cadre « absence de mobilité, absence d'intégration ». Nos travaux de recherche visent le cadre extrême à « forte mobilité, forte intégration ».



**Figure 1**  
De l'informatique conventionnelle vers l'informatique ambiante (d'après [Lyytinen et Yoo 2002]).

La convergence « mobilité-intégration » à haut degré signifie aussi que l'utilisateur n'est plus limité à une configuration imposée de ressources d'interaction (la station de travail usuelle), mais devient façonneur de son espace interactif où se fondent les mondes numérique et

physique. Ainsi, l'évolution vers l'informatique ambiante doit aussi se voir selon deux points de vue complémentaires :

- celui des objets physiques en tant que ressources d'interaction,
- celui du façonnage de l'espace par assemblage de ces objets.



**Figure 2**

**Le lapin Nabaztag [Nabaztag].**

Le lapin relié à Internet par Wi-Fi indique par des changements de couleurs, des mouvements des oreilles ou des sons, toutes sortes d'informations. L'informatique ambiante étend cet usage à tous les objets qui nous sont familiers.

Le lapin de la Figure 2 donne un exemple d'évolution des objets de la vie quotidienne vers le rôle de ressource d'interaction. Le lapin en plastique, joue le rôle de ressource d'interaction dès l'instant où il est couplé à des services Internet. L'exemple de la Figure 3 illustre le façonnage : deux amis se rencontrant par hasard, assemblent leur PDA en les plaçant côte à côte (Figure 3 à gauche). Ils obtiennent ainsi une surface d'affichage suffisamment grande pour engager une collaboration. Ou encore, un voyageur à proximité d'un mur actif, obtient des renseignements que l'environnement transfère automatiquement au bon format sur le dispositif personnel (Figure 3 à droite). Les assemblages d'objets de la Figure 3 (les deux PDA d'une part, le mur actif et le dispositif

personnel, d'autre part) sont des couplages car il en résulte, pour l'utilisateur, de nouvelles fonctions (agrandissement de la surface d'affichage, d'une part, renseignements situés, d'autre part).



**Figure 3**  
Exemples de couplage opportuniste de ressources d'interaction.

Nous venons d'introduire notre sujet : le couplage de ressources d'interaction en informatique ambiante. Quelle en est la motivation ?

## 2. Motivations

En vérité, le couplage de ressources d'interaction n'est pas un phénomène nouveau. Dans les IHM graphiques conventionnelles, la souris et l'écran sont couplés pour fournir le service de pointage. De même, la souris est liée au clavier pour désigner le focus, lieu d'insertion des caractères. Toutefois, ces couplages existent par conception : ils sont immuables et les objets physiques tenant lieu de ressources d'interaction sont dédiés et conçus pour cela (c'est-à-dire pour pointer, pour saisir des caractères, etc.). En conséquence, il suffit au programmeur de développer les drivers idoines et à l'utilisateur d'apprendre un nombre limité de conventions.

En informatique ambiante, mobilité et intégration ont deux conséquences sur le couplage. Avec la mobilité, l'espace d'interaction est extrêmement dynamique. Des ressources peuvent disparaître ou devenir

disponibles. Elles peuvent être prêtées, empruntées, assemblées au gré des besoins. Avec l'intégration, tous les objets du monde réel peuvent jouer le rôle de ressource d'interaction. La diversité de ces objets et leur couplage dynamique ouvrent un large champ d'expérience. A contrario, cette ouverture pose au concepteur le problème des choix dans l'espace des solutions, au développeur, des problèmes de mise en œuvre technique, et à l'utilisateur, des difficultés d'appropriation.

En conception, l'approche la plus souvent adoptée est de nature exploratoire : de l'expérience, on espère tirer des leçons, par exemple, des taxonomies pouvant servir de guides de conception. Nous verrons au chapitre suivant que ces taxonomies négligent le plus souvent de considérer la nature des liaisons possibles entre objets, de même la composition de couplages. Sur le plan technique, des solutions commencent à émerger : la boîte à outils ICon permet de configurer des ressources d'interaction d'entrée selon un modèle flux de données [Dragicevic 2004]. Ubit [Lecolinet 2003] est capable de gérer plusieurs écrans et plusieurs pointeurs au-dessus de X Window, ou encore I-AM permet la construction dynamique d'un espace d'interaction multi-écrans, multipointeurs, multiclaviers [Lachenal 2004]. Mais il s'agit de couplage d'objets dédiés de l'informatique conventionnelle. iStuff, qui permet de connecter des objets, va un peu plus loin, mais la spécification de couplages entre iStuff's est, pour l'instant, réservé aux seuls experts informaticiens [Ballagas et al. 2003].

Quant à l'utilisateur naïf, comprend-il qu'un couplage est possible entre deux objets donnés ? Peut-il en prédire les effets ? Connaît-il la technique d'interaction pour faire ou défaire un couplage particulier ? Autant de questions que Bellotti et ses co-auteurs posent à propos des systèmes en informatique ambiante [Bellotti et al. 2002].

Les difficultés de conception, mise en œuvre et les problèmes d'utilisabilité que pose le couplage de ressources d'interaction en informatique ambiante, justifient une étude rationnelle de cette notion.

### *3. Objectifs et Approche*

---

Notre objectif général est de comprendre la notion de couplage de ressources d'interaction en informatique ambiante. Cet objectif implique les étapes suivantes :

- Fournir une définition formelle du couplage de ressources d'interaction et de là,



- Produire un cadre conceptuel de raisonnement permettant de définir des requis logiciels et des requis centrés utilisateurs. En particulier,
- Ouvrir des pistes pour la définition de nouvelles techniques d'interaction d'expression du couplage.

Pour atteindre ces objectifs, nous avons opté pour une démarche ascendante : en l'absence de travaux théoriques sur le sujet, nous avons analysé les exemples de la littérature. Puis, nous avons procédé par abstraction et généralisation vers un modèle génératif pour l'expression du couplage.

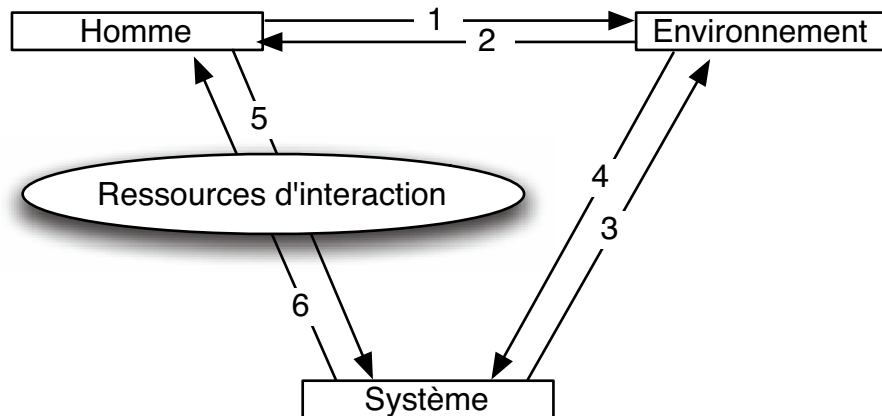
Dans la section qui suit, nous précisons la couverture de notre étude.

#### *4. Portée de l'étude*

---

Comme le montre la Figure 4, l'Homme, l'Environnement physique et le Système sont en interaction. Et l'interaction a lieu par le biais d'objets physiques jouant le rôle de ressources d'interaction. Trois flux d'interactions sont possibles :

- Homme - Environnement : Dans ce cas, le Système est absent et l'utilisateur peut agir (via des ressources d'interaction) sur l'Environnement et réciproquement. Par exemple (1), en éteignant une lampe (ressource d'interaction), un utilisateur diminue la luminosité de son environnement. A contrario (2), le vent qui souffle dans la voile (ressource d'interaction) d'un véliplanchiste exerce une force sur ce dernier. C'est le type d'interaction que nous effectuons dans la vie quotidienne sans intervention de l'acteur Système.
- Environnement - Système : Le système peut agir sur l'Environnement par le biais de ressources d'interaction, par exemple (3) un gestionnaire d'énergie [Thevenin et al. 1999] d'une maison peut ajuster la température des radiateurs. Cette réaction peut être programmée ou peut être initiée par le système si un capteur de température extérieure permet de détecter des baisses anormales de température (4).
- Homme - Système : En informatique conventionnelle, l'Homme peut par exemple à l'aide d'une souris modifier la position d'une fenêtre (5). En réponse, grâce à l'écran, le Système indique à l'Homme que la modification de la position de la fenêtre de son traitement de texte est effective (6).



**Figure 4**  
**La ressource d'interaction au centre de l'interaction. Les flèches indiquent le sens des échanges.**

Dans notre étude, nous considérons les interactions entre l'Homme et le Système par le biais de ressources d'interaction. Une ressource d'interaction est un rôle tenu par une entité physique qui permet l'interaction entre l'Homme et le Système.

Dans la suite de l'exposé, le terme « entité » désigne aussi bien un objet physique qu'un objet numérique.

## 5. Organisation du manuscrit

La structure du manuscrit reflète notre démarche.

- Au chapitre suivant, nous proposons une définition formelle de la notion de couplage complétée par la définition d'une nouvelle taxonomie : TEC (Taxonomy for Entity Coupling) dont l'objectif est d'intégrer la diversité des exploitations actuelles de la notion de couplage, voire détecter de nouvelles niches d'exploration.
- Au chapitre 3, nous analysons l'état de l'art en matière de couplage selon les dimensions de TEC. L'objectif est ici de dégager une synthèse sur les tendances actuelles et d'identifier de nouvelles pistes.
- Au chapitre 4, nous proposons un cadre formel plus fin d'analyse avec la définition d'un cycle de vie et de relations de dépendance

entre couplages : un couplage n'est pas nécessairement un phénomène spontané, mais passe par des étapes que l'utilisateur doit être en mesure d'évaluer et de contrôler (au sens de la Théorie de Norman [Norman 1986]). Le cycle de vie que nous proposons pour les couplages rend ces étapes explicites. La création d'un couplage peut avoir des effets de bord sur des couplages existants ou en créer d'autres. Nous proposons une notation permettant au concepteur (ou au système) de raisonner sur ces dépendances.

- Le chapitre 5 est orienté vers les techniques d'interaction dont le rôle est de permettre à l'utilisateur d'exprimer son désir de coupler des entités. Nous proposons ici une notation pour représenter ces techniques en sorte de les évaluer de manière prédictive à haut niveau d'abstraction. Nous appliquons cette notation aux techniques d'interaction de l'état de l'art, mais aussi à notre propre proposition pour le couplage d'écrans.
- Une conclusion résume les contributions et s'ouvre sur les prolongements possibles de ces travaux de thèse.
- On trouvera en annexe, la présentation des principes de mise en œuvre de l'intergiciel I-AM auquel nous avons participé. I-AM (Interaction Abstract Machine) permet aux utilisateurs de coupler des écrans gérés chacun par des machines susceptibles d'exécuter des systèmes d'exploitation différents (par exemple, MacOS et Windows). Deux écrans couplés offrent à l'utilisateur un espace d'affichage plus grand et uniforme (quelles que soient la définition et l'orientation de ces écrans).

---

*Chapitre II*      *TEC : Une taxonomie pour le  
couplage de ressources  
d'interaction*

---

---

## *Avant Propos*

---

Le couplage de ressources d'interaction est inhérent à l'informatique ambiante : c'est par couplage que les frontières entre mondes physique et numérique s'estompent ; et c'est par couplage que les lieux de vie sont façonnés de manière opportuniste. La fusion des mondes physique et numérique se trouve portée par plusieurs courants de recherche : réalité augmentée ou mixte, interfaces utilisateurs tangibles, IHM manipulables, IHM incorporées, ou encore technologie calme . En particulier, pour Ishii et Ullmer, les IHM tangibles « augmentent le monde physique par couplage d'information numérique avec les entités physiques et les environnements du quotidien » [Ishii et Ullmer 1997]. L'étude du façonnage de l'espace est plus récente, centrée pour l'instant sur les salles de réunions [Johanson et al 2002] [Streitz et al. 1999] ou encore sur les services liés à la localisation (location-aware systems) et l'informatique portée (wearable computing). Par exemple, les ConnecTables s'appuient sur le couplage dynamique d'écrans pour la création d'espaces de travail partagés [Tandler et al. 2001]. La notion de couplage, on le voit, est utilisée de manière informelle à différentes fins sans que sa sémantique soit clairement établie. Après une phase exploratoire un peu touffue, le domaine commence à s'organiser par le biais de taxonomies [Fitzmaurice et al. 1995], [Holmquist et al. 1999], [Calvillo-Gamez et al. 2003], [Koleva et al. 2003], [Fishkin 2004], mais celles-ci sont plutôt centrées sur la fusion des mondes que sur le couplage qui sert de fondement technique à cette fusion.

Notre objectif, dans ce chapitre, est de fournir une définition formelle de la notion de couplage complétée par une taxonomie dont les éléments doivent intégrer la diversité des exploitations actuelles de la notion de couplage, voire détecter de nouvelles niches d'exploration. Ce chapitre est organisé comme suit : dans la section 1, nous proposons une définition formelle de la notion de couplage. Puis, dans la section 2, nous passons en revue les taxonomies actuelles pour justifier TEC (Taxonomy for Entity Coupling), un nouvel espace de classification que nous présentons en détail en 3.

---

### *1. Couplage d'entités : définitions*

---

Dans le langage courant, « coupler des choses, coupler quelque chose avec quelque chose d'autre, [c'est] les assembler, les réunir, les relier afin de constituer un seul élément. » [Encyclopédie Larousse, Tome 4, 1982]. En mécanique, le couplage est un assemblage de pièces de

manière à ce qu'elles opèrent conjointement. [Le Petit Robert Tome 1, 1989]. Ces deux définitions traduisent :

- d'une part, l'idée de composition unitaire : le résultat d'un couplage est une unité qui peut, à son tour, intervenir comme élément dans un autre assemblage,
- d'autre part, la notion d'opération conjointe : le résultat de l'assemblage est à même d'offrir de nouveaux services ou fonctions.

Notons aussi que le suffixe -age désigne aussi bien l'action que l'état résultant de l'action.

Dans cette section, nous proposons notre propre définition du couplage d'entités (section 1.1) que nous complétons en section 1.2, par un retour sur une classe particulière d'entités : les entités mixtes et leur lien avec les ressources d'interaction. En section 1.3, nous montrons comment, par composition de couplages, il est possible de former des assemblages de complexité quelconque.

### 1.1. DÉFINITION DU COUPLAGE D'ENTITÉS

Nous convenons que « le couplage de deux entités est l'acte de lier ces entités de manière à ce qu'elles opèrent conjointement pour offrir un nouvel ensemble de fonctions qu'elles ne pourraient offrir individuellement. C'est aussi le résultat de cette action, c'est-à-dire une nouvelle entité composée des deux entités liées.»

Dans cette définition, une entité peut être soit :

- physique, notée par le symbole  $P$  : il s'agit alors d'une entité purement matérielle,
- numérique, notée  $N$ , telle que des données, fonctions, processus, applications,
- mixte (ou  $M$ ) : elle résulte de couplage(s) entre entité(s) physique(s) et entité(s) numérique(s) ou entre entités mixtes.

Notre définition fait référence à l'acte de lier. Dans la vie courante, la nature de cet acte détermine les services obtenus. Par exemple, l'acte qui consiste à « mettre la chaise sur la table » (couplage de deux entités physiques) construit un assemblage qui permet d'atteindre un point élevé d'une pièce, alors que l'acte de « placer la chaise à côté de la table » offre d'autres perspectives. La nature de l'acte de lier a donc une importance sur le résultat. Si en informatique conventionnelle, les couplages sont définis par le concepteur, en informatique ambiante, l'acte de lier peut également être accompli par l'utilisateur. Dans ce cas, l'utilisateur doit disposer de techniques d'interaction (au sens de Foley [Foley et al. 1990] et de

Bowman [Bowman et Hodges 1999]). La nature de cette technique et des entités auxquelles elle s'applique, détermine la fonction obtenue. Notre définition du couplage se traduit comme suit dans la notation ensembliste. Soient :

- E, un ensemble d'entités physiques P, numériques N ou mixtes M,
- T, un ensemble de techniques d'interaction,
- F, un ensemble d'ensembles de fonctions.

Alors, un couplage est défini par une application de  $E \times E \times T$  dans F :

$$E \times E \times T \rightarrow F$$

Cette définition indique que la spécification de deux entités et d'une technique d'interaction détermine l'ensemble des fonctions obtenues par couplage de ces entités. La technique d'interaction sert à désigner les deux entités à coupler et à contraindre l'ensemble des fonctions cibles. Parmi ces entités, les entités mixtes et les ressources d'interaction sont au cœur de notre sujet.

---

## 1.2. ENTITÉS MIXTES ET RESSOURCES D'INTERACTION

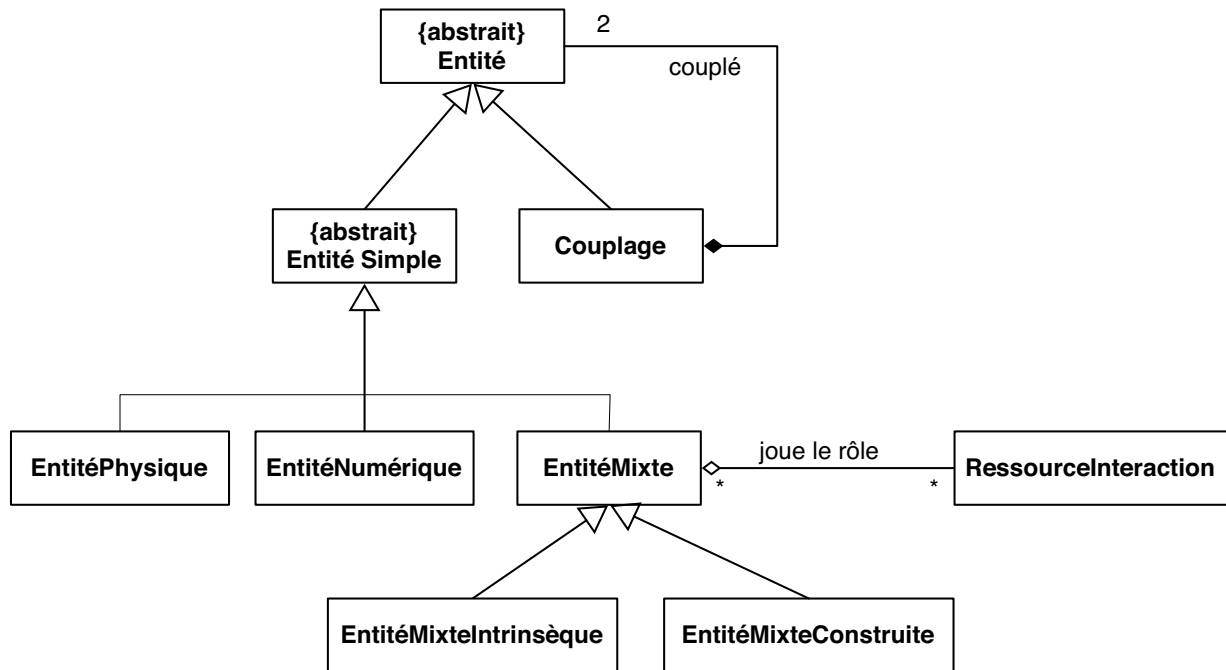
De nombreuses études et taxonomies traitent des entités mixtes sans pour autant observer que certaines entités sont mixtes par essence (comme le PDA) alors que d'autres ne deviennent mixtes que parce que l'utilisateur final en a effectué le couplage avec une entité numérique (ou avec une autre entité mixte). Par exemple, un clavier ne remplit sa fonction que s'il est couplé à une station de travail (elle-même entité mixte puisqu'elle résulte de couplages d'entités matérielles et numériques).

Ainsi, une entité mixte peut être :

- une entité mixte intrinsèque dès lors qu'elle remplit sa fonction sans nécessiter de couplages préliminaires de la part de l'utilisateur final (comme le PDA),
- une entité mixte construite lorsque l'utilisateur final doit effectuer un couplage pour qu'elle assure la fonction qui fait sa raison d'être (tel le clavier dont la raison d'être est de permettre la saisie de caractères).

Dans notre chapitre d'introduction, nous définissons une ressource d'interaction comme « un rôle (ou fonction) qu'un objet physique assure lorsqu'il sert de support aux échanges d'information entre un système informatique et un utilisateur ». Or, pour que l'échange d'information soit possible, un couplage doit exister entre cette entité physique et le monde numérique. Autrement dit, le rôle de ressource d'interaction ne peut être

assuré que par une entité mixte. Désormais, nous dirons qu'une entité mixte, qu'elle soit intrinsèque ou construite, est une ressource d'interaction (ou sert de ressource d'interaction) lorsque sa fonction est l'échange d'information entre un utilisateur et le système. Le diagramme de la Figure 1 résume nos définitions. Ce diagramme s'appuie sur le patron « composite » de Gamma et ses collègues [Gamma et al. 1995].

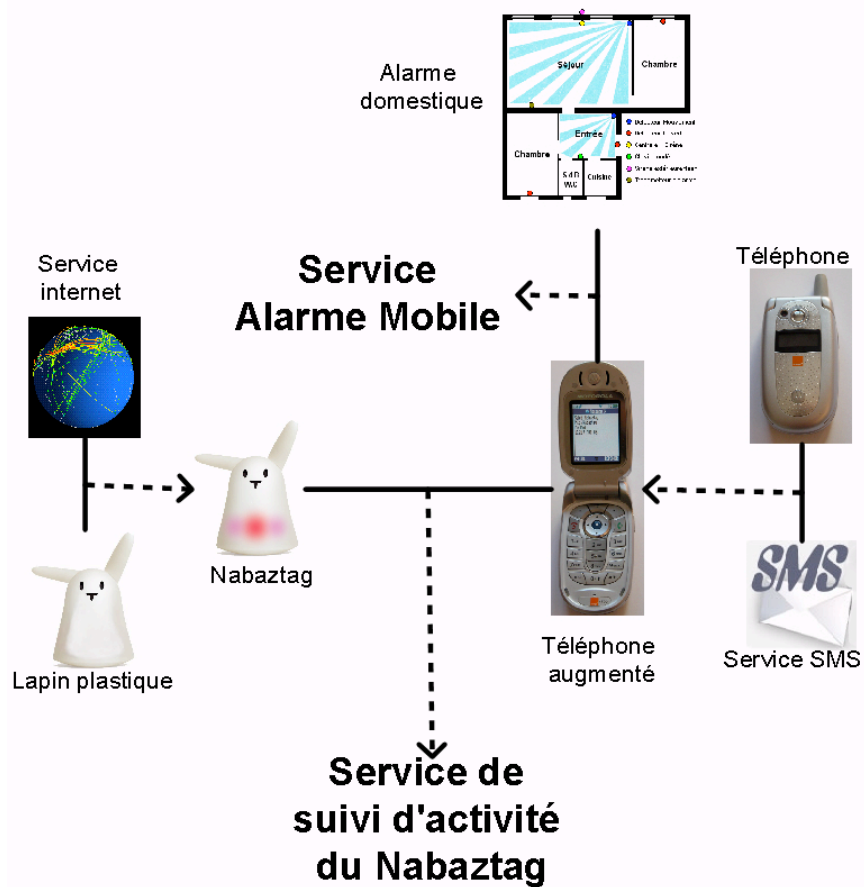


**Figure 1**  
**Relations entre les différentes classes d'entités, physiques, numériques ou mixtes et ressources d'interaction.**

Le couplage de deux entités de même type produit une entité de ce type ( $P+P=P$ ,  $N+N=N$ ,  $M+M=M$ ), et le couplage de deux entités de type différent produit une entité mixte ( $P+N=M$ ,  $M+P=M$ ,  $M+N=M$ ). Cette assertion s'exprime ainsi en OCL [Warmer et Kleppe 2003] :

```
context : Couplage
inv :
Let t1= oclType(entité1) in
Let t2= oclType (entité2) in
t1=t2 implies self->oclIsTypeOf(t1)
and t1<>t2 implies self->oclIsTypeOf(EntitéMixte)
```





**Figure 2**  
**Composition de couplages.**

Un couplage a lieu entre deux entités ; une entité peut résulter elle-même d'un couplage et peut également intervenir dans plusieurs couplages. Un trait plein représente un couplage. Une flèche en pointillés désigne l'entité produite par ce couplage.

### 1.3. COMPOSITION DE COUPLAGES

Notre définition indique que le couplage, en tant qu'action, porte sur deux entités où chaque entité peut être une entité élémentaire ou bien une entité composée, résultat d'un précédent couplage. Autrement dit, notre définition ne contraint pas la granularité des entités. De même, elle n'interdit pas à une entité d'intervenir dans plusieurs couplages. Par exemple (Figure 2), le couplage de l'objet en plastique du Chapitre I - le lapin, avec un service Internet, forme une nouvelle entité - le Nabaztag, une entité mixte. À son tour, le Nabaztag peut être couplé à un téléphone portable (entité mixte obtenue par couplage d'un téléphone et d'un service SMS) pour signaler l'arrivée de messages lorsque le propriétaire se trouve éloigné du lapin. Le téléphone portable peut également être couplé à une alarme domestique pour produire un service de surveillance mobile.

Comment caractériser et comparer cette diversité de couplages ?  
Les taxonomies actuelles apportent-elles des réponses à ces questions ?

## *2. Analyse des taxonomies de l'état de l'art*

En réponse à la prolifération de termes et de prototypes, des taxonomies émergent progressivement aussi bien du côté de l'informatique ambiante que du côté des interfaces tangibles.

En informatique ambiante, les taxonomies actuelles s'adressent au développeur de logiciels. À notre connaissance, aucune ne concerne directement l'IHM. Par exemple, Hightower et Borriello proposent une classification des systèmes de localisation. Les axes de cet espace incluent les techniques de perception employées, le type de localisation (symbolique ou physique, relative ou absolue), et un ensemble de critères (notamment la précision, couverture et facteur d'échelle) [Hightower et Boriello 2001]. Ce type de taxonomie offre au concepteur de logiciel un cadre systématique pour conduire des évaluations comparatives de solutions existantes et de là, construire les bons couplages entre des entités physiques (les capteurs notamment) et des entités numériques (infrastructure logicielle). Si ce choix a un impact sur l'utilisateur, celui-ci est indirect.

Pour les interfaces tangibles, plusieurs taxonomies ont été proposées sans toutefois étudier les implications logicielles : [Fitzmaurice et al. 1995], [Holmquist et al. 1999], [Calvillo-Gamez et al. 2003], [Koleva et al. 2003], [Fishkin 2004]. Dans ce qui suit, nous présentons la plus ancienne et la plus récente : l'espace de conception des briques de Fitzmaurice (1995) et l'espace de conception des interfaces tangibles de Fishkin (2004).

### **2.1. L'ESPACE DE CONCEPTION DES BRIQUES**

Le travail séminal sur le sujet revient à Fitzmaurice, Ishii et Buxton avec leur espace de classification des briques [Fitzmaurice et al. 1995]. En gros, les briques sont « de nouveaux dispositifs d'entrée qui peuvent être couplés étroitement (attachés) à des objets virtuels pour la manipulation ou l'expression d'actions (par exemple, pour définir la valeur de paramètres ou pour lancer des processus) ». Certaines dimensions de cette classification dépendent de la nature des entités impliquées dans le couplage (briques, table, entités virtuelles), alors que d'autres, présentées comme spécifiques aux briques, sont en vérité généralisables. Nous relevons notamment :

- L'affectation de fonctions : l'affectation de fonctions aux briques est-elle permanente, programmable ou facilement modifiable ? En nous appuyant sur notre définition, l'affectation de fonctions se traduit dans notre proposition TEC par le caractère immuable ou non de l'action de lier (et donc de délier), mais aussi de manière plus fine, par la muabilité ou non de l'ensemble des fonctions qu'il est possible d'obtenir par couplage. Par exemple,

l'action de coupler/re-coupler une (ou plusieurs) briques à un objet virtuel est toujours possible, mais l'ensemble F des fonctions cibles est, par conception, immuable. Inversement, dans ACCORD, l'assemblage d'entités conduit à un ensemble F de services potentiellement non borné [ACCORD].

- Le lien entre les couches physiques et virtuelles est-il étroit ou lâche ? Autrement dit, les états des entités physiques et numériques sont-ils parfaitement synchronisés ou non ? Dans TEC, nous ne retenons pas cet axe explicitement, mais nous traduisons cet aspect important de l'IHM sous forme de propriétés ergonomiques étudiées au Chapitre IV.
- Les représentations de l'interaction correspondent dans TEC aux techniques d'interaction. Les possibilités citées par Fitzmaurice sont généralisables avec les propriétés CARE [Coutaz et al. 1995].
- Les axes « nombre de briques utilisées en même temps », et « sensibilité spatiale » se généralisent dans TEC avec la notion de cardinalité de la relation de couplage : une brique couplée à un objet virtuel constitue le couplage d'une entité physique avec une entité numérique avec pour fonction cible, la rotation. Une seconde brique est couplée au même objet avec pour fonction cible la déformation.
- L'axe input&output décrit quelles propriétés sont captées ou retournées à l'utilisateur. TEC traduit cet axe par la notion de « rôles de ressource d'interaction » et la description de la fonction obtenue suite au couplage.

La taxonomie de Fitzmaurice, bien que centrée sur les IHM saisissables (graspables), est le premier effort marquant de structuration de l'espace de conception des interfaces tangibles. Récemment, Fishkin propose un espace de conception dont le pouvoir descriptif intègre celui des taxonomies proposées ces dernières années.

---

## **2.2. L'ESPACE DE CONCEPTION DES INTERFACES TANGIBLES (FISHKIN)**

En 2004, Fishkin propose une taxonomie pour l'analyse des interfaces tangibles [Fishkin 2004]. Elle concerne le couplage d'entité physique et d'entité numérique dans lequel l'entité physique sert de dispositif d'entrée pour manipuler l'entité numérique. Fishkin propose deux axes : l'incorporation et la métaphore.

L'incorporation traduit, du point de vue de l'utilisateur, les relations spatiales entre les points d'entrée et de sortie :

- L'incorporation est totale si le dispositif d'entrée sert également de dispositif de sortie. C'est l'exemple de TiltScreen [Rekimoto 1996] où l'objet PDA, soumis à des manipulations physiques (inclinaisons), reflète les changements d'état sur son écran.

- L'incorporation est à proximité lorsque la sortie a lieu à proximité du point d'entrée. Par exemple dans [Fitzmaurice et al. 1995], des Bricks sont manipulées pour modifier les positions et taille des fenêtres projetées. La sortie (l'affichage) est réalisée sous le dispositif d'entrée.
- L'incorporation est environnementale si la sortie a lieu autour de l'utilisateur. Les systèmes dont le retour est de type audio relèvent de l'incorporation environnementale. Le retour est séparé du dispositif d'entrée, mais se tient en un même lieu.
- L'incorporation distante va au-delà de l'incorporation environnementale puisque la sortie du système, dissociée de l'entrée, se situe en un autre lieu.

Nous notons ici que si nous considérons qu'une interaction entre un utilisateur et un système implique un flux d'information dans les deux sens (utilisateur vers système et système vers utilisateur), alors les entités mixtes ne permettant que des entrées (ou que des sorties) sont de fait classées comme entité mixtes construites. En effet, pour pouvoir être utilisables, elles devront être couplées pour qu'un feedback soit fourni (si l'entité ne permet que des entrées) et inversement (si l'entité ne permet que des sorties).

La métaphore qualifie la relation de ressemblance entre les entités physique et numérique couplées :

- L'absence de métaphore indique qu'il n'y a pas de corrélation entre les entités physique et numérique. Par exemple, la connexion de deux perles (Beads [Resnick et al. 1998]) modifiant leur apparence.
- La métaphore de nom correspond à la phrase : « un <X> du système est comme un <X> du monde réel ». Par exemple, les fenêtres du desktop sont analogues aux feuilles de papier sur un bureau physique.
- La métaphore de verbe est appliquée si « effectuer l'action <A> dans le système, c'est comme effectuer <A> dans le monde réel ». « Graspable Display » [Small et Ishii 1997] relève de la métaphore de verbe : lorsque l'utilisateur recule la tablette portable, la police de caractères diminue. Notons qu'avec ce type de métaphore, l'entité physique doit posséder suffisamment de degrés de liberté pour couvrir ceux de l'entité numérique. Cette notion de compatibilité a été étudiée par Card et ses collègues dans [Card et al. 1990]. Nous reviendrons sur cette propriété au Chapitre IV.
- La métaphore de nom et de verbe correspond à la fusion des deux métaphores précédentes : « faire l'action <A> sur un <X> du système, c'est comme faire <A> sur un <X> du monde réel ». Déposer une icône virtuelle dans la corbeille du desktop, c'est comme

mettre un fichier physique dans une vraie corbeille à papier.

- Avec la métaphore absolue (full metaphor), le système virtuel est le système physique. Contrairement aux métaphores de nom et de verbe, l'utilisateur n'a pas à comprendre le fonctionnement par analogie. Par exemple, dans « Illuminating Clay » [Piper et al. 2002], l'utilisateur dispose d'un bloc d'argile représentant un terrain. En remodelant l'argile, l'utilisateur change la forme du terrain. Le système en recalcule les caractéristiques qu'il projette sur l'argile.

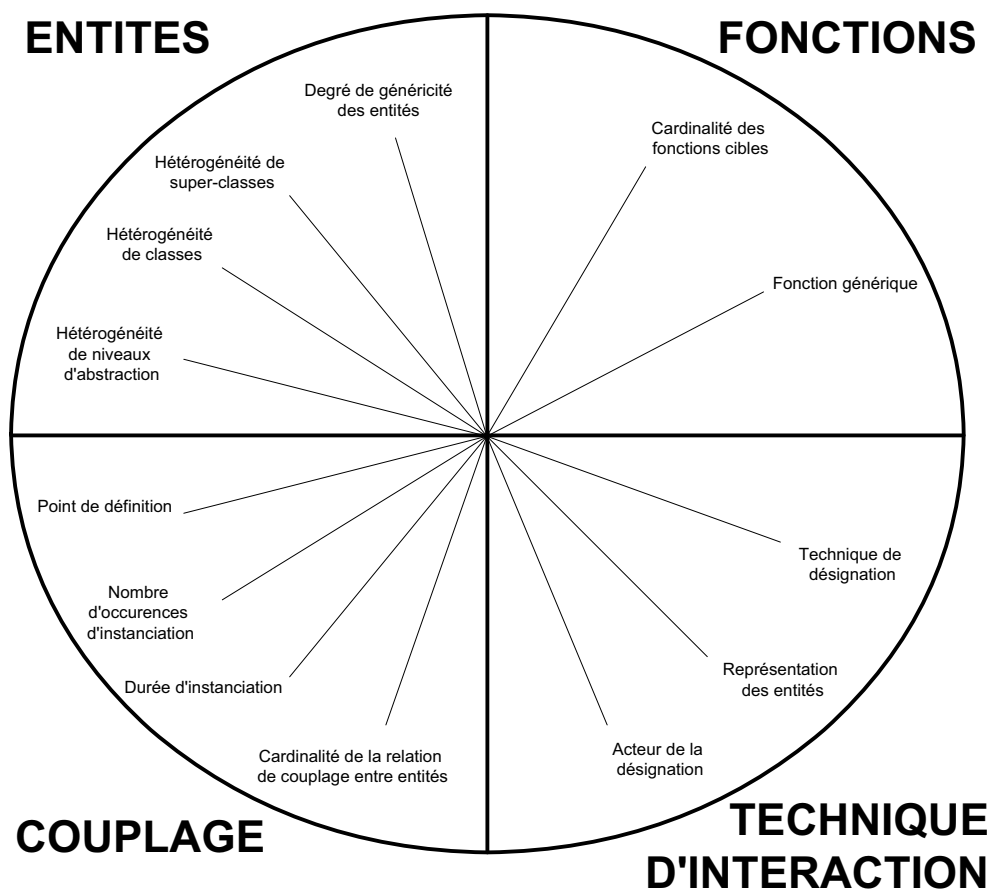
La taxonomie de Fishkin se concentre sur la perception qu'a l'utilisateur d'un couplage entre entités physique et numérique. Notons que la combinaison des deux axes, incorporation et métaphore, couvre la taxonomie d'Holmquist sur les interfaces tangibles [Holmquist et al. 1999]. Comme le montre la Figure 3, Holmquist distingue trois classes d'entités physiques :

- Le container est une entité physique qui peut être couplée à toute entité numérique dans le but de la transporter entre systèmes informatiques (ordinateurs, dispositifs d'affichage ou d'impression, etc.).
- Le token est une entité physique qui ressemble à l'entité numérique qu'il représente et fournit un point d'accès (une poignée) à cette entité numérique.
- Les outils (ou tools) sont des entités physiques représentant des fonctions (sélection, activation, déplacement, rotation, etc.).

Comme pour la taxonomie de Fitzmaurice, les propositions de Fishkin et d'Holmquist ne couvrent pas la mise en œuvre ni le cycle de vie des couplages, mais concernent plutôt la perception que l'utilisateur a des couplages. S'il est vrai qu'une taxonomie ne doit pas comporter trop d'axes d'analyse, il convient néanmoins qu'elle offre un bon cadre de raisonnement. Or la taxonomie de Fishkin, au-delà de l'absence de couverture des aspects logiciels et du cycle de vie du couplage, ne permet pas la mise en évidence de patrons de conception, aspect pourtant important sur la perception de l'utilisateur. TEC vise à combler ces lacunes.

Metaphor	None	Noun	Verb	Noun and Verb	Full
Embodiment					
Full			--Containers		
Nearby	Tools--			Tokens	
Env.		Tokens		Tokens	
Distant					

**Figure 3**  
 Couverture des trois classes d'entités physiques par la taxonomie de Fishkin, d'après [Fishkin 2004]



**Figure 4**  
 Vue globale des éléments taxonomiques de TEC selon les quadrants entités, fonctions, technique d'interaction et couplage.

### 3. TEC : une nouvelle taxonomie pour le couplage

Comme le montre la Figure 4, les axes d'analyse de TEC caractérisent les trois ensembles de définition du couplage (entités mises en jeu, techniques d'interaction, fonctions obtenues) et le couplage en tant qu'unité résultante. Nous les présentons maintenant en détail

#### 3.1. CARACTÉRISATION DES ENTITÉS

Comme nous l'indiquons dans le chapitre d'introduction, tout objet du monde réel peut intervenir dans un couplage. Par conséquent, il convient de s'interroger sur le niveau de généralité des entités mises en jeu, de même sur la capacité d'un couplage à admettre des entités de nature différente, c'est-à-dire son hétérogénéité.

##### Généricité des entités

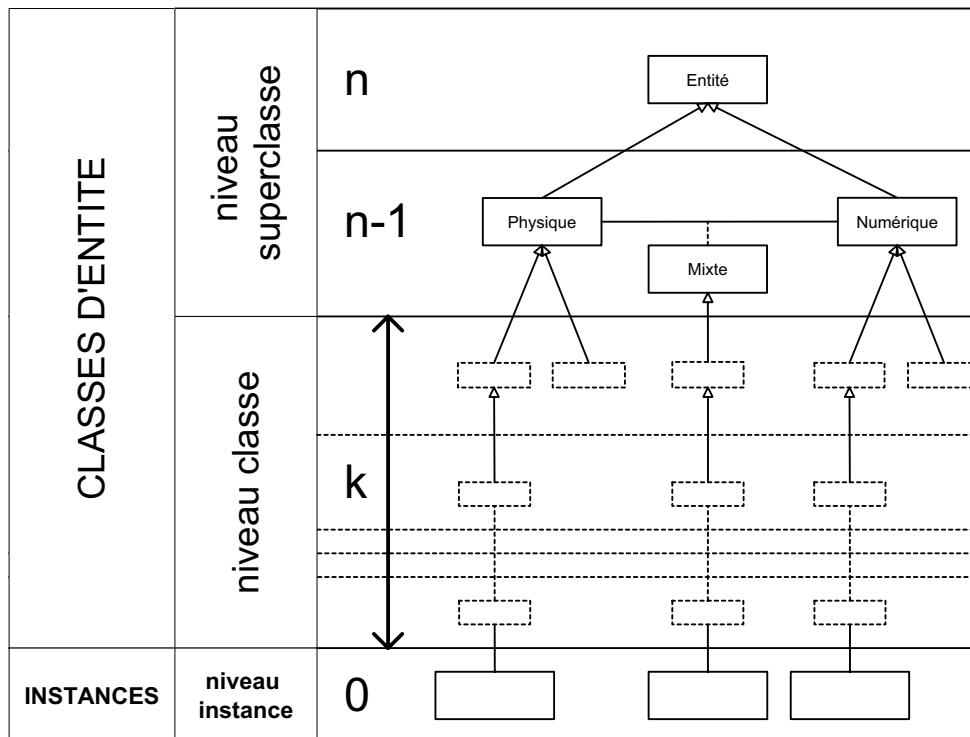
La généralité des entités susceptibles d'intervenir dans un couplage détermine la généralité de ce couplage que nous définissons par le couple d'entiers  $(x, y)$  tels que :

- $x, y \in \{0, k, n\}$  où  $x$  et  $y$  désignent le niveau de généralité de chacune des entités susceptibles d'être impliquées dans ce couplage,
- $n$  est le niveau maximal de généralité des entités,
- et  $k$  satisfait la relation d'ordre :  $0 < k < n-1$ .

Comme le montre la Figure 5, nous proposons, pour les entités, trois niveaux de généralité :

- aux deux extrêmes, feuille et sommet, les niveaux instance et superclasse auxquels nous faisons correspondre les entiers 0 et  $n$  respectivement : 0 pour exprimer l'absence de généralité ;  $n$  et  $n-1$  pour désigner le niveau de généralité maximal (le niveau Entité) et les sous-classes (Entités physiques, numériques, ou mixtes),
- au centre, le niveau classe constitué par une hiérarchie de classes intermédiaires de profondeur quelconque auquel correspond l'entier  $k$  ( $0 < k < n-1$ ).  $k$  exprime un pouvoir de généralité intermédiaire entre absence de généralité et absence de contrainte sur la nature des entités.

Une entité dite de niveau instance signifie que seul cet exemplaire peut intervenir dans un couplage donné. Par exemple, dans le MetaDesk [Ishii et Ullmer 1997], le couplage du phicon Dôme avec la carte du campus du MIT n'est possible qu'entre ces deux exemplaires. La généralité de ce couplage est donc  $(0,0)$ . Le niveau superclasse signifie que n'importe quelle entité, qu'elle soit physique, numérique ou mixte, peut intervenir dans un couplage donné ; le niveau classe désigne tout



**Figure 5**  
**Classification des entités selon leur niveau de généralité.**

autre niveau de généralité. Dans la Table Magique de Bérard [Bérard 2003], des jetons circulaires suivis par un système de vision par ordinateur permettent la manipulation d'entités numériques de bas niveau nommées patch. Ces patches, qui contiennent n'importe quelle entité numérique, sont du niveau superclasse n-1 alors que les jetons appartiennent au niveau k. La généralité de ce couplage est donc (k, n-1). Ou encore :

- la généralité du couplage de « Nabaztags » est (k, k),
- celle du « Nabaztag de Cathy » avec « celui de JérémY » est (0, 0),
- et la généralité du couplage du « Nabaztag de Nancy » avec « d'autres Nabaztag » est (0, k).

Comme pour la généralité, nous distinguons trois niveaux d'hétérogénéité : hétérogénéité de superclasses, de classes ou de niveaux.

### **Hétérogénéité de superclasses ou type de mixage**

Quel que soit le niveau de généralité d'un couplage, toute instantiation de couplage fait intervenir des instances d'entités qui, chacune, relèvent de l'une des trois superclasses : Physique, Numérique, Mixte. Si



les deux entités de ce couplage sont des instances de superclasses distinctes, alors ce couplage est hétérogène par les superclasses.

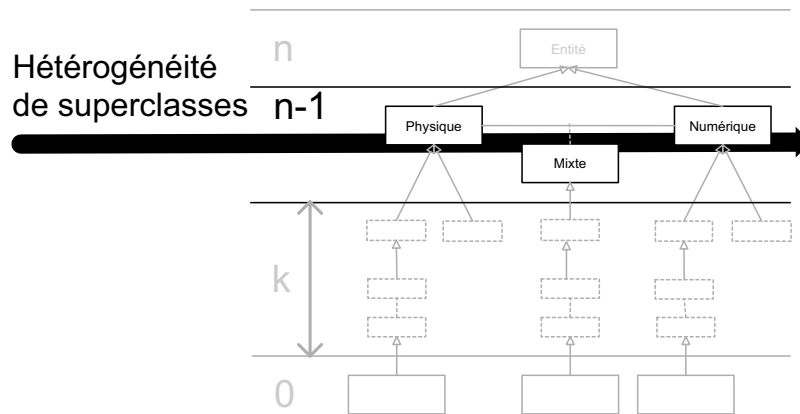


Figure 6  
Type de mixage ou hétérogénéité de superclasses.

Comme le montre la flèche horizontale de la Figure 6, par combinaison de deux superclasses parmi trois, nous obtenons six types de mixage : physique-physique (PP), physique-numérique (PN), physique-mixte (PM), mixte-mixte (MM), mixte-numérique (MN) et numérique-numérique (NN). Par exemple, le couplage du clavier et de la souris est un mixage de type MM. Dans le MetaDesk [Ishii et Ullmer 1997], le phicon du Dôme couplé à sa représentation numérique sur la carte du MIT est un mixage de type PN. Pour la Table Magique, le couplage de jetons avec des patches numériques est également un mixage PN.

Nous considérons qu'un mixage PP constitue une nouvelle entité unitaire P qui ne sera pas étudiée plus avant. La combinaison NN ne sera étudiée que comme conséquence de couplages (non pas comme couplage source) puisque nous nous intéressons avant tout aux couplages PN, PM, MM, MN en relation avec les entités mixtes et les ressources d'interaction.

Le raisonnement sur l'hétérogénéité par les superclasses (n-1) vaut également pour le niveau classe (k).

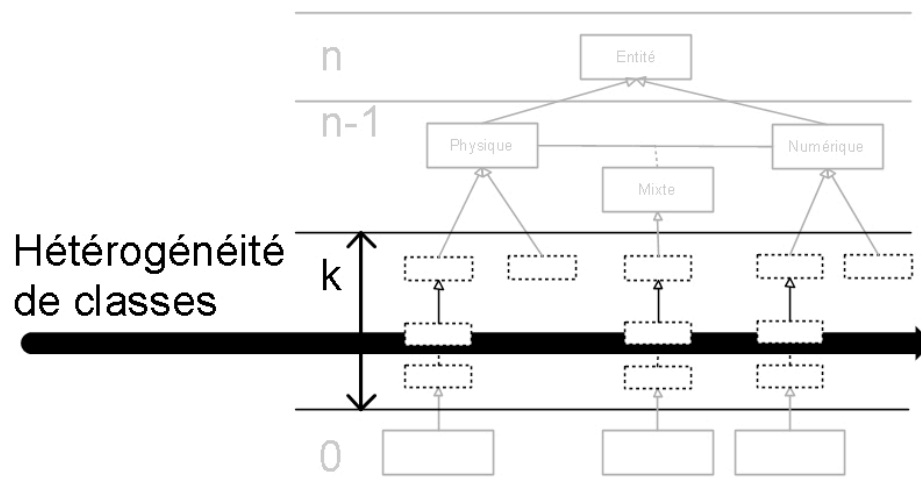
### Hétérogénéité de classes ou hétérogénéité de niveau k

L'hétérogénéité de classes (ou hétérogénéité de niveau k) se détermine à partir de la définition (spécification) des concepts (ou ontologie) du domaine considéré (Figure 7) : si les deux entités impliquées dans le couplage sont des instances de la même classe, on assiste à un couplage homogène par les classes. Inversement, si les deux

entités n'appartiennent pas à la même classe, il s'agit d'un couplage hétérogène par les classes (ou hétérogène de niveau  $k$ ).

Par exemple, dans la Table Magique [Bérard 2003], le couplage de jetons entre eux est homogène de niveau  $k$ . Inversement, si les PDA et les stations de travail sont modélisés comme deux classes distinctes, le couplage d'un PDA avec une station de travail est hétérogène de niveau  $k$ .

Notons que l'hétérogénéité de superclasses implique l'hétérogénéité de classes. Inversement, l'homogénéité de superclasses n'implique pas nécessairement l'homogénéité de classes : les couplages « PDA, stations de travail » sont homogènes au niveau superclasse (mixage MM), mais sont hétérogènes de niveau classe.



**Figure 7**  
**Hétérogénéité de classes.**

Comme le montrent la Figure 6 et la Figure 7, l'hétérogénéité de superclasses de même que l'hétérogénéité de classes, considèrent l'arbre de généricité horizontalement par niveau d'abstraction. L'hétérogénéité au niveau instance n'a pas de sens : par définition, les instances sont toutes uniques, donc hétérogènes. Aussi, nous proposons, comme dernier élément de classification, d'étudier l'hétérogénéité verticalement : c'est l'hétérogénéité de niveaux d'abstraction.

**Hétérogénéité de niveaux d'abstraction**

Un couplage est hétérogène par les niveaux d'abstraction lorsque le couple d'entiers  $(x, y)$  qui caractérise sa généricité est tel que  $x \neq y$ . Il est homogène par les niveaux d'abstraction si  $x = y$  (Figure 8).

Comme nous l'avons vu au paragraphe "Généricité des entités", la généricité du couplage jeton-patch est  $(k, n-1)$ . Le couplage jeton-patch est donc hétérogène par les niveaux d'abstraction.

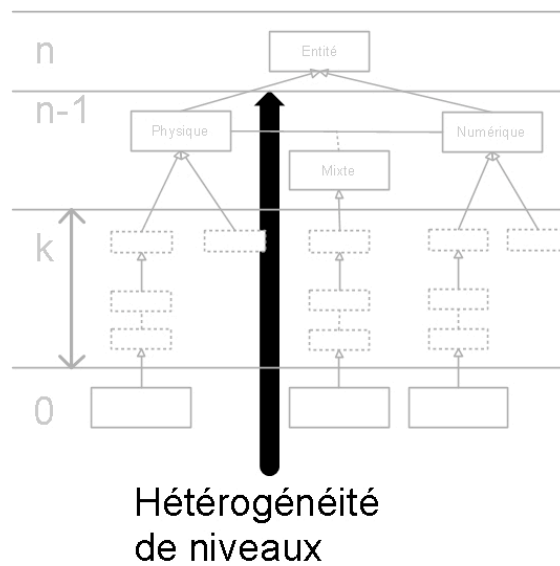


Figure 8  
Hétérogénéité de niveaux d'abstraction.

### 3.2. CARACTÉRISATION DES FONCTIONS

Nous cherchons ici à caractériser la richesse des fonctions susceptibles d'être obtenues par un couplage. Nous utilisons pour cela, la cardinalité de l'ensemble F et sa couverture.

#### Cardinalité de l'ensemble F

Considérons  $E1 \times E2 \rightarrow F$ . Alors :

- F désigne l'ensemble des fonctions accessibles par couplage d'entités de E1 et de E2 indépendamment des techniques d'interaction,
- $Card(F)$  est le nombre de fonctions dans F.

Deux cas retiennent notre attention :

- $Card(F) = 1$ . Le couplage est dit unifonctionnel,
- $Card(F) > 1$ . Le couplage est plurifonctionnel.

Le prototype Synchronous Gesture d'Hinckley [Hinckley 2003] offre un couplage plurifonctionnel de valeur 4. En entrechoquant deux tablettes PC, l'utilisateur peut obtenir quatre fonctions distinctes : l'annexion de l'affichage d'une des tablettes par l'autre, la création d'un espace collaboratif partagé, le partage de fichiers unilatéral ou bilatéral. L'ambiguïté est levée par la nature des chocs imprimés aux tablettes.

#### Couverture générique de F

La nature des fonctions de F dépend de la nature des entités dont on sait qu'elles sont fortement hétérogènes en informatique ambiante. En conséquence, en dresser une liste exhaustive n'est pas réaliste. Nous proposons néanmoins 4 fonctions génériques de base :

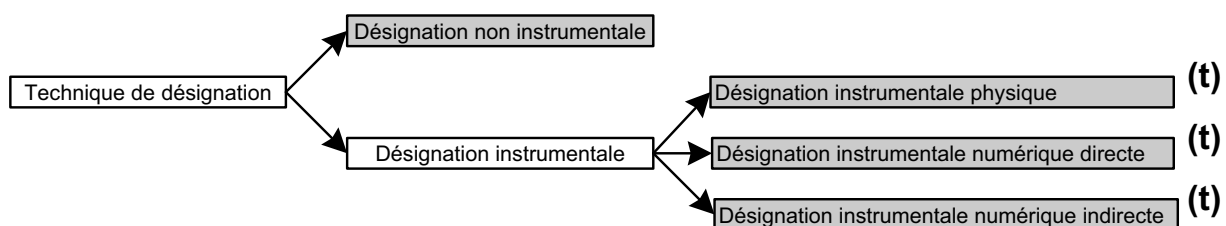
- contenir, représenter, manipuler (qui correspondent à la vision duale de l'approche centrée objet de Holmquist [Holmquist et al. 1999] sur les Tangible User Interfaces-TUI),
- omixer qui complète l'analyse de Holmquist en donnant à un acteur (utilisateur ou système) la possibilité de construire une entité mixte à partir d'une entité physique et d'une entité numérique. Cette nouvelle entité considérée comme un tout est une entité mixte construite au sens de 1.2. Elle traduit la possibilité qu'a l'utilisateur de façonner (voire programmer) son espace.

### 3.3. TECHNIQUE D'INTERACTION

Rappelons que la technique d'interaction sert d'une part, à désigner les entités à lier et d'autre part, à contraindre l'ensemble des fonctions cibles. La caractériser revient donc à caractériser la technique de désignation, de même, la représentation des entités et des fonctions cibles. Nous analysons ci-dessous les techniques de désignation et de représentation des entités.

#### Technique de désignation

La Figure 9 synthétise les différentes options de désignation. La désignation (d'une entité ou d'une fonction cible) peut se faire sans instrument ou bien avec instrument.



**Figure 9**

**Classement des techniques de désignation.**

(t) désigne l'utilisation possible d'un instrument tiers (c'est-à-dire distinct des entités à coupler).

La désignation non instrumentale correspond au cas où la désignation est vraiment directe, c'est-à-dire, sans artefact intermédiaire entre l'acteur et l'objet de la désignation. C'est le cas de l'interaction manuelle comme dans Synchronous Gesture [Hinckley 2003] où désigner une tablette, c'est la prendre dans la main (Figure 10 a). De même, désigner par la parole est un acte non instrumental. Un acte multimodal comme « coupler cette tablette à celle-là » est une désignation non instrumentale d'entités.

Nous distinguons trois formes de désignation instrumentale : la désignation instrumentale physique, et les désignations instrumentales numériques directe et indirecte.

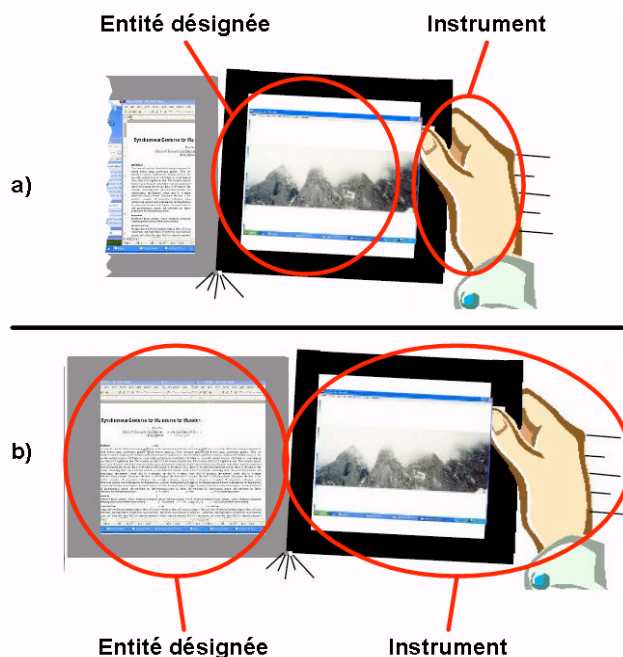


Figure 10

**Deux désignations dans Synchronous Gesture [Hinckley 2003].**

En a) une désignation non instrumentale, en b) une désignation instrumentale physique non tiers.

- La désignation instrumentale physique est réalisée au moyen de la composante physique d'une entité mixte jouant le rôle d'instrument. Synchronous Gesture illustre ce cas : la tablette qui est tenue dans la main (et qui est la composante physique de l'entité mixte tablette PC) sert d'instrument de désignation de la seconde tablette dès lors qu'elle est amenée en contact avec cette seconde tablette posée sur une table (Figure 10 b). Le symbole (t) de la Figure 9 dénote l'utilisation possible d'un instrument de désignation tiers, c'est-à-dire autre que l'une des entités à coupler. Par exemple, dans Synchronous Gesture, nous aurions un instrument de désignation physique tiers si un stylet était utilisé pour désigner les deux tablettes.
- La désignation instrumentale numérique directe ou désignation instrumentale numérique de niveau 1 est réalisée au moyen de la composante numérique d'une entité mixte jouant le rôle d'instrument. Ce n'est pas l'instrument physique qui désigne, mais sa représentation numérique. La désignation avec une souris relève de cette catégorie. Ce n'est pas la souris qui désigne, mais son curseur qui lui sert de représentation. Pour que cela soit possible, il faut un couplage entre une entité physique pouvant jouer le rôle d'instrument et une entité numérique avec, comme fonction cible, « représenter » en sorte que cette entité numérique représente l'instrument physique. C'est le cas entre la souris, entité physique qui tient dans la main et qui peut donc servir d'instru-

ment, et le curseur, entité numérique affichée à l'écran. Appliquée aux tablettes d'Hinckley, cette technique de désignation instrumentale numérique directe pourrait être la suivante : une tablette pourrait être désignée par le seul fait d'imprimer au curseur de la souris affiché sur son écran un rapide va-et-vient de quelques secondes. Effectuons le même geste avec l'autre main sur la seconde tablette. Ces deux gestes instrumentaux numériques directs et synchronisés auraient pour effet de désigner deux entités écran et de les coupler pour fournir un espace d'affichage unifié, à la manière du prototype de Hinckley. Dans cet exemple, le curseur est un instrument numérique tiers : il n'est pas l'une des entités à coupler. Inversement, si l'utilisateur se sert du curseur d'une souris pour désigner la représentation graphique d'un clavier avec lequel la souris doit être couplée, dans ce cas, le curseur de la souris est à la fois entité à coupler et instrument. Il n'est pas tiers au couplage.

- La désignation instrumentale numérique indirecte ou désignation instrumentale de niveau 2 est effectuée au moyen d'une entité numérique jouant le rôle d'instrument. L'acteur agit par le biais d'un intermédiaire instrumental purement numérique. Pour que cela soit possible, il doit exister 2 couplages fournissant chacun la fonction représenter : un premier couplage « entité physique jouant le rôle d'instrument, entité numérique » comme dans la désignation numérique directe ; un second couplage de cette entité mixte avec une entité numérique jouant le rôle d'instrument. C'est le cas de la souris qui agit sur un curseur qui, à son tour, agit sur une entité cible. C'est la notion d'interaction instrumentale de Beaudouin-Lafon [Beaudouin-Lafon 2000]. Reprenons notre exemple de couplage d'écrans. Comme le montre la Figure 11, chaque écran est représenté par une icône. Le couplage des deux écrans s'effectue en amenant l'une des icônes au contact de la seconde : l'icône déplacée sert d'instrument numérique de désignation de la seconde. Parce que cette icône est à la fois instrument et entité à coupler, elle n'est pas instrument tiers. Nous verrons dans notre analyse de l'état de l'art, qu'il semble peu réaliste d'envisager une entité tiers à ce niveau.

Nous proposons ici une décomposition plus fine que celle de Beaudouin-Lafon. La Figure 9 montre les quatre premiers niveaux de désignation : sans indirection, avec indirection physique (tierce et non tierce en fonction des entités à coupler), avec indirection numérique (tierce et non tierce en fonction des entités à coupler), et avec double indirection numérique (tierce et non tierce). On peut envisager des indirections numériques de niveau supérieur à deux, mais ceci nous semble peu pertinent au regard des principes de la manipulation directe ou de la théorie de l'action [Norman 1986].

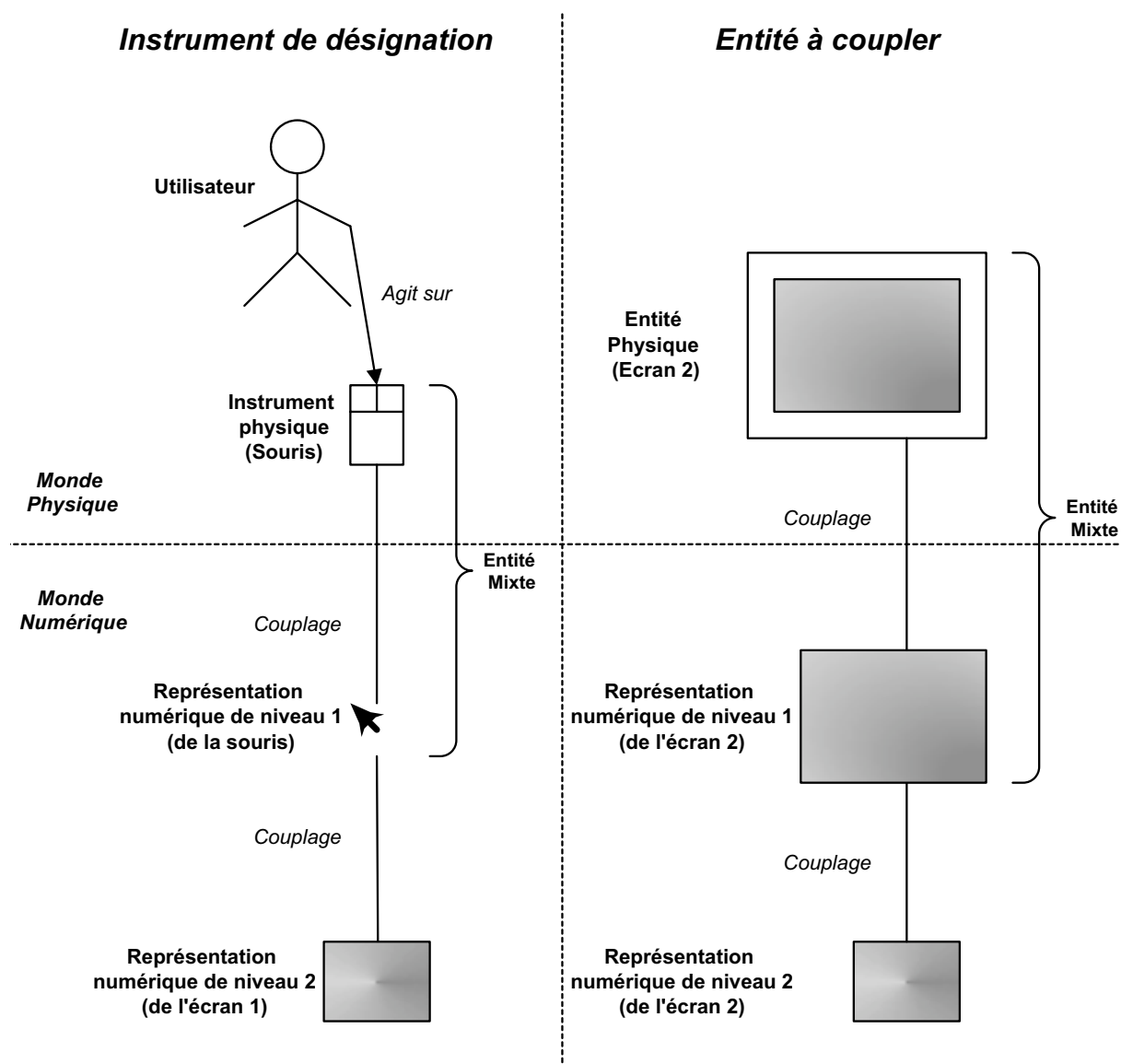


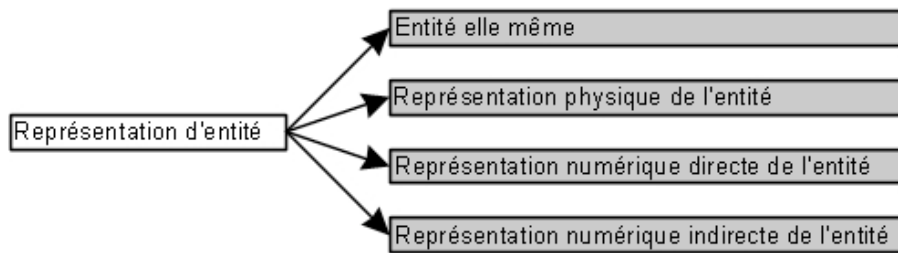
Figure 11

**Différentes formes de désignation instrumentale et de représentation de l'entité désignée.**

Illustration avec le couplage de deux écrans par action de la souris sur une icône graphique servant de représentant d'écran.

**Représentation des entités**

Comme le montre la Figure 12, la désignation (instrumentale ou non) d'une entité à coupler peut être appliquée à l'entité elle-même, ou bien à une représentation de cette entité. À son tour, la représentation d'une entité peut être physique ou numérique directe (dite de niveau 1) ou indirecte (dite de niveau 2). Ces différentes techniques de représentation d'entités sont illustrées dans la Figure 11 dans laquelle l'utilisateur cherche à coupler deux écrans en rapprochant au moyen de la souris, l'icône graphique représentant l'écran 1 vers l'icône graphique représentant le second écran.



**Figure 12**  
**Classement des techniques de représentations d'entité.**

Par exemple, dans *Synchronous Gesture* [Hinckley 2003], les tablettes n'ont pas de représentation. La désignation a lieu sur les tablettes elles-mêmes. Si, chaque tablette était représentée par un écran miniature en bois, alors ces phicons seraient des représentations physiques. Pour que cela soit possible, un couplage PP doit exister entre un écran miniature et une tablette avec la fonction « représenter ». Considérons la souris comme entité à coupler, non pas comme instrument de désignation. Alors, le curseur est une représentation numérique de niveau 1 de l'entité souris. Un couplage PN doit exister entre la souris et le curseur avec la fonction « représenter ». Si l'utilisateur désigne cette souris en choisissant son nom par le biais d'une liste déroulante dans un configurateur, ce nom est une représentation numérique de niveau 2 car on assiste à deux couplages de représentation : le nom représente l'entité mixte souris.

### **Acteur de la désignation et migrabilité**

L'acteur du couplage désigne l'entité responsable de l'acte de couplage. Lachenal [Lachenal 2004] désigne deux principaux acteurs impliqués en Interaction Homme-Machine : le système et l'utilisateur (final) de ce système. À ces deux acteurs, nous ajoutons le concepteur qui décide du « point de définition ».

Nous avons considéré la ré-instanciation, couplage et découplage, comme atomique. Il convient toutefois de distinguer la phase de couplage et de découplage, quand bien même dans les prototypes étudiés, l'acteur du couplage est très rarement différent de celui du découplage. Dans le prototype d'Hinckley [Hinckley 2003] où deux utilisateurs peuvent échanger leur carte de visite en choquant simultanément leur tablette, l'acteur du couplage est l'utilisateur, mais le découplage est réalisé par le système une fois les cartes de visite échangées. Alors que les trois acteurs peuvent mettre en œuvre un couplage, seuls l'utilisateur et le système peuvent le détruire puisqu'ils sont les seuls acteurs présents à l'exécution.

La migrabilité de tâche, qui désigne la capacité du système à prendre à sa charge une tâche utilisateur (et réciproquement) est une



propriété pertinente. L'acte de coupler correspond à l'accomplissement d'une tâche consistant en un ensemble de sous-tâches élémentaires : choisir l'instrument (si la technique d'interaction est instrumentale), désigner les deux entités à coupler ainsi que la fonction cible. Tout ou partie de ces sous-tâches peut dynamiquement migrer entre le système et l'utilisateur final. Nous parlons alors de migrabilité de la technique d'interaction.

---

### 3.4. CARACTÉRISATION DE L'INSTANCIATION DU COUPLAGE

Jusqu'ici, nous avons caractérisé les ensembles E1, E2, et T. Il convient maintenant d'analyser le couplage en tant qu'instance. Nous proposons comme éléments d'analyse le point de définition, le nombre d'occurrences d'instanciation, la durée d'instanciation et la cardinalité de la relation de couplage.

#### **Point de définition**

Un couplage peut être défini à l'exécution ou à la conception du système :

- Un point de définition à l'exécution signifie que l'utilisateur (ou le système) a la possibilité de choisir les entités à coupler ainsi que la fonction cible.
- Un point de définition à la conception implique que le choix des entités et de la fonction cible sont réalisés par le concepteur et donc embarqués dans le système. Dans ce cas, à l'exécution, l'utilisateur ne peut qu'exprimer son intention d'instancier le couplage.

#### **Nombre d'occurrences d'instanciation**

Comme son nom l'indique, le nombre d'occurrences d'instanciation exprime le nombre de fois qu'un même couplage peut être réalisé : 1 ou n fois. Lorsque le nombre d'occurrences d'instanciation est supérieur à 1 et que ces instanciations reviennent à l'utilisateur, alors celui-ci doit disposer de techniques d'interaction pour le détruire et le créer à nouveau. Dans le MetaDesk [Ishii et Ullmer 1997], comme pour la Table Magique [Bérard 2003], le nombre d'occurrences d'instanciation des couplages est n.

#### **Durée d'instanciation**

La durée d'instanciation d'un couplage désigne sa durée de vie. Il peut être :

- Intra-session : sa durée de vie est au plus celle d'une session d'application . Toutes les instances de couplage du MetaDesk [Ishii et Ullmer 1997] et de la Table Magique [Bérard 2003] disparaissent à l'arrêt de ces systèmes. Au redémarrage, l'utilisateur doit les recréer.
- Inter-sessions : le couplage persiste au-delà de la session. C'est le cas du Nabaztag : quand bien même, l'objet lapin en plastique est débranché, le couplage avec des services numériques est automatiquement restauré dès la remise sous tension.

**Cardinalité de la relation de couplage entre entités**

La cardinalité de la relation de couplage entre entités d'un système est un couple d'entiers noté  $u-v$  tel que :

- $u$  est le nombre maximal d'entités de  $E1$  intervenant dans des instances de couplage avec des entités  $E2$ ,
- $v$  est le nombre maximal d'entités de  $E2$  intervenant dans des instances de couplage avec  $E1$ .

Ce critère est valué de la même façon que la cardinalité des relations dans les diagrammes de classes UML [Breu et al. 1997]. En UML, les cardinalités d'une relation précisent le nombre (minimum et maximum) d'instances d'une classe (entités) pouvant participer simultanément dans cette relation.

Par exemple, en informatique conventionnelle, un clavier ne peut être couplé au maximum qu'avec une souris et réciproquement. La cardinalité de la relation de couplage est donc 1-1. Pour la Table Magique, plusieurs entités jetons peuvent être couplées à une même entité table. La cardinalité de la relation de couplage jetons-table est donc  $n-1$ .

#### *4. Synthèse*

---

Nous venons d'introduire les éléments nécessaires à une analyse approfondie de l'état de l'art : une définition formelle de la notion de couplage dans laquelle interviennent trois ensembles (entités, techniques d'interaction, fonctions résultats) qui servent à leur tour de base aux éléments de classification de TEC. Nous sommes maintenant en mesure d'étudier l'état de l'art en matière de couplage. C'est le sujet du chapitre suivant.



---

*Chapitre III      Couplage de ressources  
d'interaction et son état de  
l'art selon TEC*

---

---

## *Avant Propos*

---

Notre analyse de l'état de l'art ne prétend pas être exhaustive, mais propose un panorama représentatif des solutions actuelles pour lesquelles nous ne retenons que les couplages les plus significatifs. L'analyse qui suit s'appuie sur TEC en utilisant comme guide directeur l'hétérogénéité de superclasses (section 1) et l'homogénéité de superclasses (section 2). Chacune de ces sections est structurée en groupes d'exemples que TEC identifient comme proches. Chaque groupe se termine par un tableau synthétique. Certaines colonnes sont en italique, elle indique que le couplage décrit dans la colonne précédente requiert que le couplage en italique soit présent.

---

## *1. Hétérogénéité de superclasses : mixage PN et MN*

---

---

### **1.1. COUPLAGE PN : METADESK, MCRPD ET DATATILES**

Dans Tangible Bits du MetaDesk [Ishii et Ullmer 1997], Ullmer et Ishii étudient le couplage de « bits » avec des objets physiques. Geospace, qui s'appuie sur le metaDesk, utilise les phicons du Dôme du MIT et du bâtiment du MediaLab dont la Figure 1 illustre l'utilisation. La carte géographique inclut une représentation numérique du Dôme et du MediaLab. Le Dôme physique (entité de genericité instances) représente l'objet numérique Dôme (entité également de genericité instances). Cette fonction « représenter » résulte du couplage phicon « Dôme-entité numérique Dôme » établi à la conception. Son point de définition est donc « conception », et le nombre d'occurrences d'instanciation est le nombre de redémarrages du système. À l'exécution, en posant le phicon du Dôme sur le metaDesk, l'utilisateur crée un couplage phicon-carte qui fournit les fonctions de translation et de rotation de la carte centrée sur le Dôme. Le point de définition de ce couplage est donc « exécution ». Concernant la technique d'interaction, l'utilisateur manipule les entités elles-mêmes (c'est-à-dire le phicon et la carte numérique, non pas des représentants de phicon et de carte numérique). La désignation du phicon est non instrumentale (puisque l'utilisateur le prend dans la main) et celle de la carte est une désignation instrumentale physique (au moyen du phicon). Ce couplage peut être instancié autant de fois que souhaité, mais ne perdure pas au-delà d'une session (sa durée d'instanciation est intra-session).

En complément à ces travaux, Ullmer et Ishii [Ullmer and Ishii 2000] proposent un cadre architectural, MCRpd pour Model Control

Representation (physical et digital) fondé sur le model MVC [Goldberg et Robson 1984] dans lequel la vue est partagée entre les représentations physique et numérique. Ce modèle est de fait, exclusivement applicable aux couplages dont la fonction est « représenter ».



**Figure 1**  
**Manipulation du phicon représentant le Dôme du MediaLab, sur le MetaDesk.**

Les DataTiles de Rekimoto [Rekimoto et al. 2001] sont, du point de vue de TEC, similaires au MetaDesk, à la fonction près. Chaque tuile physique (Figure 2) dénote un service numérique. Par exemple, une tuile dont la surface est gravée pour représenter les contours d'un pays, permet d'obtenir la météo de ce pays. En déposant la tuile sur une surface dédiée, la météo s'affiche sur la gravure.

La synthèse de ces travaux est présentée dans le Tableau 1.



**Figure 2**  
**A gauche, différentes tuiles pré-imprimées ou gravées. A droite, les mêmes tuiles en action sur la surface dédiée.**

Référence	[Ishii and Ullmer 1997]	[Ullmer and Ishii 2000]	[Rekimoto et al. 2001]
Entité 1	phicon	phicon	tuile
Entité 2	concept	concept	représentation
Hétérogénéité de superclasses	PN	PN	PN
Niveau de généralité de E1	0	0	0
Niveau de généralité de E2	0	0	0
Hétérogénéité de classes	Oui	Oui	Oui
Hétérogénéité de niveaux	Non	Non	Non
Card(F)	1	1	1
Fonction générique	<i>représenter</i>	<i>représenter</i>	<i>représenter</i>
Point de définition	<i>Conception</i>	<i>Conception</i>	<i>Conception</i>
Nombre d'occurrences d'instanciation	n	n	0
Durée d'instanciation	Inter-session	Inter-session	Inter-session
Cardinalité de la relation de couplage	1-1	1-1	1-1
Instrument de désignation 1	aucun	aucun	aucun
Niveau désignation E1	entité	entité	entité
Acteur	utilisateur	utilisateur	utilisateur
Instrument de désignation 2	aucun	aucun	aucun
Niveau désignation E2	entité	entité	entité
Acteur	système	système	système

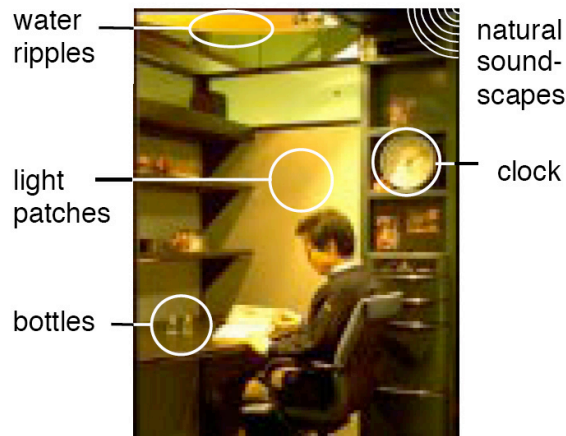
**Tableau 1**  
**Taxonomie appliquée au MetaDesk, MCRpd et aux DataTiles.**  
 Couplages PN.

Notons que ces couplages PN produisent des entités mixtes construites. Leur mise en œuvre à l'exécution nécessite toujours la présence d'une entité mixte tierce (la table du Geospace, un écran équipé pour les DataTiles).

## 1.2. COUPLAGE PN : AMBIENTROOM ET LE CUBE DE CONTRÔLE TÉLÉVISION

AmbientRoom [Wisneski et al. 1998] a pour objectif de rendre perceptibles des informations sur l'environnement. Un affichage au plafond représente, grâce à la projection d'ondulation d'eau (water ripples, voir la Figure 3), l'activité d'une personne distante alors que l'ouverture d'une bouteille produit un bruit de trafic routier pour représenter la charge d'un ordinateur. Du point de vue de l'utilisateur, l'objet physique représente l'objet numérique, mais contrairement aux prototypes précédents, l'entité physique n'a pas nécessairement de ressemblance avec

l'entité numérique qu'elle représente. Le fait que l'entité physique « donne accès à l'entité numérique » indique que la fonction générique correspondante est « représenter ». Les instances d'entités physiques et numériques sont couplées à la conception et instanciées au démarrage du système. Aucune technique d'interaction de couplage n'est nécessaire à l'exécution hormis le démarrage du système.



**Figure 3**  
**Ambient Room.**

Au plafond des effets d'eau représentent l'activité d'une personne distante.

De façon identique, dans le prototype de [Block et al. 2004], le cube physique (instance) est couplé à un cube virtuel (instance) sur un écran de télévision. Ce couplage est prédéfini à la conception. Ici aussi, aucune technique d'interaction de couplage n'est nécessaire à l'exécution hormis le démarrage du système. Une chaîne de télévision est associée à chaque face du cube virtuel que l'utilisateur fait pivoter en manipulant le cube physique. Si l'utilisateur repose le cube, la chaîne présente sur la face avant du cube virtuel est sélectionnée et affichée en plein écran. La Figure 4 illustre l'utilisation du cube.

Le Tableau 2 montre la synthèse de ces premiers systèmes. Nous étudions ensuite les solutions dont le degré de généricité des entités est supérieur au niveau instance.





**Figure 4**  
**Couplage du cube physique et du cube virtuel.**  
 Le cube physique manipulé par l'utilisateur provoque la rotation du cube virtuel présentant une chaîne de télévision sur chaque face.

Référence	[Wisneski et al. 1998]	[Block et al. 2004]
Entité 1	objet	cube R
Entité 2	source d'info	cube V
Hétérogénéité de superclasses	PN	PN
Degré de généralité de E1	0	0
Degré de généralité de E2	0	0
Hétérogénéité de classes	Oui	Oui
Hétérogénéité de niveaux	Non	Non
Card(F)	1	1
Fonction générique	<i>représenter</i>	<i>représenter</i>
Point de définition	<i>Conception</i>	<i>Conception</i>
Nombre d'occurrences d'instanciation	1	1
Durée d'instanciation	Inter-session	Inter-session
Cardinalité de la relation de couplage	1-1	1-1
Instrument de désignation 1		
Niveau désignation E1		
Acteur	concepteur	concepteur
Instrument de désignation 2		
Niveau désignation E2		
Acteur	concepteur	concepteur

**Tableau 2**  
**Taxonomie appliquée à AmbientRoom et un cube pour contrôler la télévision.**  
 Couplages PN.

**1.3. 1 COUPLAGE MN :  
MEDIABLOCKS,  
SENTIENT DATA  
ACCESS ET  
AUGMENTED  
SURFACES**

Dans MediaBlocks [Ullmer et al. 1998], tout comme dans Sentient Data Access [Fitzmaurice et al. 2003] et Augmented Surfaces [Rekimoto et Saitoh 1999], des entités mixtes peuvent « contenir » des entités numériques. Dans MediaBlocks, des cubes en bois (entités mixtes dont la généralité est de niveau classes) servent de conteneur (Figure 6). Dans Sentient Data Access, des PDA (entités mixtes de généralité de niveau classes) remplissent cette fonction alors que dans Augmented Surfaces, ce sont des entités physiques taguées (entités mixtes dont la généralité est de niveau superclasses).

La Figure 5a illustre les différentes possibilités des Augmented Surfaces. À l'aide de son ordinateur portable et de sa souris intégrée, l'utilisateur peut prendre un objet numérique affiché sur l'écran du portable et le déplacer. Arrivé aux bords de l'écran, l'objet numérique apparaît sur la table et l'utilisateur peut continuer la manipulation sans rupture. Cette technique, nommée hyperdragging est concrétisée par un anchored cursor, segment de droite projeté sur la table entre la semelle du portable et l'objet numérique migrateur. On assiste ici à un couplage d'objet numérique à un objet mixte (la table ou une cassette vidéo) offrant la fonction « contenir ». La technique de désignation est de type instrumentale numérique (curseur souris sur le portable, dessin de l'anchored cursor sur la table augmentée - et/ou sur un autre objet mixte posé sur la table, lui-même couplé à cette table). L'entité déplacée est représentée de manière numérique directe (de niveau 1) désignée par le curseur de la souris (représentation numérique directe). L'entité destination est l'entité elle-même, désignée par l'autre entité numérique manipulée par le curseur. Il s'agit ici d'une représentation numérique de niveau 2 : la chaîne de désignation est formée par l'utilisateur-la souris-le curseur-la représentation de l'entité (ici la donnée, donc non tiers). La Figure 5b illustre cet usage.

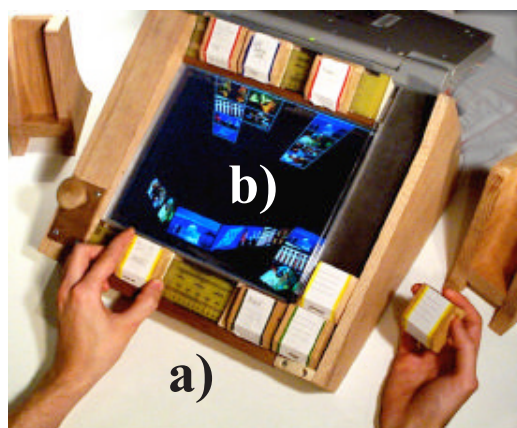
Ici, les objets qui servent de conteneurs sont de « vrais » objets physiques identifiés par des marqueurs visuels qui deviennent des entités mixtes (entités mixtes construites) par leur couplage avec la table augmentée (entité mixte). Un conteneur peut recevoir plusieurs objets numériques (la cardinalité de la relation de couplage est donc 1-n). Dans leur description, les auteurs insistent sur le fait que ces conteneurs (sur la Figure 5c, une cassette vidéo) ont un rôle en dehors de l'utilisation que nous venons d'exposer. Il s'agit donc de vraies entités mixtes construites. Pour les deux autres prototypes, la technique d'interaction est plus directe puisque la désignation est non instrumentale : l'utilisateur désigne directement les blocs (MediaBlocks Figure 6) et les PDA (entités mixtes intrinsèques dans Sentient Data Access (Figure 7)). Les entités numériques à coupler sont désignées par des représentations physiques : un élément du media sequencer et des codes-barres. En l'absence de

précision dans les publications des auteurs, il semble raisonnable de considérer que la durée d'instanciation est de type inter-sessions et que la cardinalité de la relation de couplage est 1-n. Le nombre d'occurrences d'instanciation est n puisque l'utilisateur peut coupler et découpler à loisir. Le Tableau 3 résume ces trois prototypes.



**Figure 5**  
**Vue d'ensemble du prototype Augmented Surfaces.**

Les utilisateurs disposent d'une table et d'un mur augmentés et de leurs ordinateurs portables (a). Grâce à la technique de l'hyperdragging l'utilisateur manipule de l'information numérique hors de l'écran des portables (b). L'anchored cursor rend visible cette manipulation. Un objet physique de la vie courante peut servir de conteneur pour de l'information numérique(c)..



**Figure 6**  
**Blocks et Media Sequencer.**

Le block présent dans le target pad (a) du media sequencer est couplé aux vidéos associées (b)..



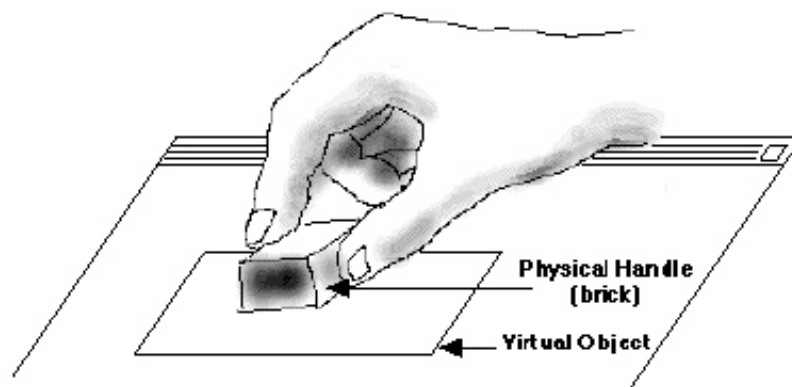
**Figure 7**  
**Utilisation du PDA dans Sentient Data Access.**  
 Un croquis est équipé d'un code-barres (a) qui est lu par un PDA (b) qui contient dès lors une version numérique du croquis pouvant être déposée sur un mur interactif (c).

Référence	[Rekimoto et Saitoh 1999]		[Ullmer et al. 1998]		[Fitzmaurice et al. 2003]
Entité 1	objet	<i>objet</i>	block	<i>block</i>	PDA
Entité 2	donnée	<i>table</i>	média	<i>sequencer</i>	donnée
Hétérogénéité de superclasses	MN	<i>PM</i>	MN	<i>PM</i>	MN
Degré de généralité de E1	n	<i>n</i>	k	<i>k</i>	k
Degré de généralité de E2	k	<i>1</i>	k	<i>1</i>	k
Hétérogénéité de classes	Oui	<i>Oui</i>	Oui	<i>Oui</i>	Oui
Hétérogénéité de niveaux	Non	<i>Oui</i>	Non	<i>Oui</i>	Non
Card(F)	1	<i>1</i>	1	<i>1</i>	1
Fonction générique	contenir	<i>mixer</i>	contenir	<i>mixer</i>	contenir
Point de définition	Exécution	<i>Exécution</i>	Exécution	<i>Exécution</i>	Exécution
Nombre d'occurrences d'instanciation	n	<i>n</i>	n	<i>n</i>	n
Durée d'instanciation	Inter-session	<i>Inter-session</i>	Inter-session	<i>Inter-session</i>	Inter-session
Cardinalité de la relation de couplage	1-n	<i>n-1</i>	1-n	<i>m-1</i>	1-n
Instrument de désignation 1	rep num 2	<i>aucun</i>	aucun	<i>aucun</i>	aucun
Niveau désignation E1	entité	<i>entité</i>	entité	<i>entité</i>	entité
Acteur	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>	utilisateur
Instrument de désignation 2	rep num	<i>physique</i>	physique tiers	<i>physique</i>	physique tiers
Niveau désignation E2	entité	<i>entité</i>	rep phys	<i>physique</i>	rep phys
Acteur	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>	utilisateur

**Tableau 3**  
**Taxonomie appliquée aux MediaBlocks, Sentient Data Access et Augmented Surfaces.**  
 Couplages PN et MN.

#### 1.4. COUPLAGE MN : BRICKS, TABLE MAGIQUE ET SENSETABLE

Les travaux fondateurs de Fitzmaurice [Fitzmaurice et al. 1995] (les bricks) ont inspiré la Table Magique [Bérard 2003] et la Sensetable [Patten et al. 2001]. Les bricks (entités de généricité classes) ont trois degrés de liberté : deux pour la position et un pour la rotation. Avant d'utiliser une brick comme entité mixte, celle-ci doit être couplée à la table augmentée, la faisant passer du statut d'entité physique à celui d'entité mixte construite. La technique consiste, pour l'utilisateur, à déposer la (ou les) brick(s) directement sur la table. Alors, une brick peut être couplée à une fenêtre (Figure 8) ; la rotation appliquée à la brick est répercutée sur la fenêtre. La technique d'interaction de couplage consiste à déposer une brick sur une entité numérique ; retirer la brick la découple de l'entité numérique. Les bricks jouent le rôle de dispositif d'entrée générique pouvant être couplé avec n'importe quelle entité numérique (entité de généricité superclasse). Le couplage de briques (entités de généricité classes) avec des entités numériques de généricité superclasses est donc hétérogène de niveaux.



**Figure 8**  
**Une brick servant pour la manipulation d'un objet virtuel.**

La Table Magique est similaire dans son fonctionnement à la différence que les jetons (entités mixtes construites de généricité classes) ne possèdent que deux degrés de liberté. La forme ronde des jetons et leur couleur uniforme empêchent la détection de la rotation. Contrairement à Fitzmaurice, la Table Magique est capable de réaliser le suivi d'un nombre important de jetons par son système de vision par ordinateur (Figure 9). De façon analogue aux bricks, les jetons peuvent être couplés à des entités numériques, les patches, qui peuvent contenir n'importe quelle information numérique (niveau de généricité superclasses).

En ce qui concerne les degrés de liberté fournis, la Sensetable qui est capable de détecter la rotation des pucks s'apparente aux bricks. En

substance, les trois prototypes que nous venons de décrire sont similaires, mais si la technique de couplage est identique, la technique de découplage dans la Sensetable diffère. Dans les autres prototypes, le simple fait de retirer une brick ou un jeton provoque le découplage alors qu'un puck reste couplé à son entité lorsqu'il est retiré. La durée d'instanciation est donc inter-sessions. La technique d'interaction de découplage consiste à secouer le puck.

Le Tableau 4 présente la synthèse de ces trois derniers travaux.



**Figure 9**  
**Utilisation simultanée de quatre jetons sur la Table Magique.**

---

Référence	[Fitzmaurice et al. 1995]		[Bérard 2003]		[Patten et al. 2001]	
Entité 1	brick	<i>brick</i>	jeton	<i>jeton</i>	puck	<i>puck</i>
Entité 2	donnée	<i>table</i>	patch	<i>table augmentée</i>	donnée	<i>senstable</i>
Hétérogénéité de superclasses	MN	<i>PM</i>	MN	<i>PM</i>	MN	<i>PM</i>
Degré de généralité de E1	k	<i>k</i>	k	<i>k</i>	k	<i>k</i>
Degré de généralité de E2	n	<i>1</i>	n	<i>1</i>	n	<i>1</i>
Hétérogénéité de classes	Oui	<i>Oui</i>	Oui	<i>Oui</i>	Oui	<i>Oui</i>
Hétérogénéité de niveaux	Oui	<i>Oui</i>	Oui	<i>Oui</i>	Oui	<i>Oui</i>
Card(F)	1	<i>1</i>	1	<i>1</i>	1	<i>1</i>
Fonction générique	manipuler	<i>mixer</i>	manipuler	<i>mixer</i>	manipuler	<i>mixer</i>
Point de définition	Exécution	<i>Exécution</i>	Exécution	<i>Exécution</i>	Exécution	<i>Exécution</i>
Nombre d'occurrences d'instanciation	n	<i>n</i>	n	<i>n</i>	n	<i>n</i>
Durée d'instanciation	Session	<i>Inter-session</i>	Session	<i>Inter-session</i>	Inter-session	<i>Inter-session</i>
Cardinalité de la relation de couplage	2-1	<i>n-1</i>	2-1	<i>n-1</i>	1-1	<i>n-1</i>
Instrument de désignation 1	aucun	<i>aucun</i>	aucun	<i>aucun</i>	aucun	<i>aucun</i>
Niveau désignation E1	entité	<i>entité</i>	entité	<i>entité</i>	entité	<i>entité</i>
Acteur	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>
Instrument de désignation 2	physique	<i>physique</i>	physique	<i>physique</i>	physique	<i>physique</i>
Niveau désignation E2	entité	<i>entité</i>	entité	<i>entité</i>	entité	<i>entité</i>
Acteur	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>	utilisateur	<i>utilisateur</i>

**Tableau 4**  
**Taxonomie appliquée aux Bricks, à la Table Magique et à Sensetable.**  
 Couplages PN.

## *2. Homogénéité de superclasses : PP et MM*

### **2.1. COUPLAGE PP : GEOSPACE, UN VRAI COUPLAGE PHYSIQUE**

Dans Geospace [Ishii et Ullmer 1997] (précédemment décrit), chaque phicon peut imprimer une rotation à la carte affichée. L'un des problèmes identifiés est que lorsque deux phicons sont couplés au metaDesk, l'utilisateur peut leur imprimer des rotations contraires. La réponse originale qu'apportent les auteurs consiste en un réel couplage physique qui n'est pas indispensable mais qui, en son absence, peut conduire à des incohérences (distorsion de la carte). Les deux phicons sont joints par une tige physique leur interdisant des rotations indépendantes. La Figure 10 illustre ce couplage. Cet exemple montre de manière originale un couplage PP. Ce couplage a, de plus, des impacts système puisqu'il réduit les degrés de liberté du couple. Un couplage voisin est mis en œuvre dans les ordinateurs portables actuels où le TouchPad est physiquement couplé au clavier. Ceci n'interdit pas de façon logicielle de découpler le clavier de la « souris » du TouchPad. C'est une façon mécanique de produire une nouvelle entité par couplage PP. Le Tableau 5 présente la synthèse selon TEC.



**Figure 10**  
**Couplage physique de deux phicons dans le MetaDesk.**



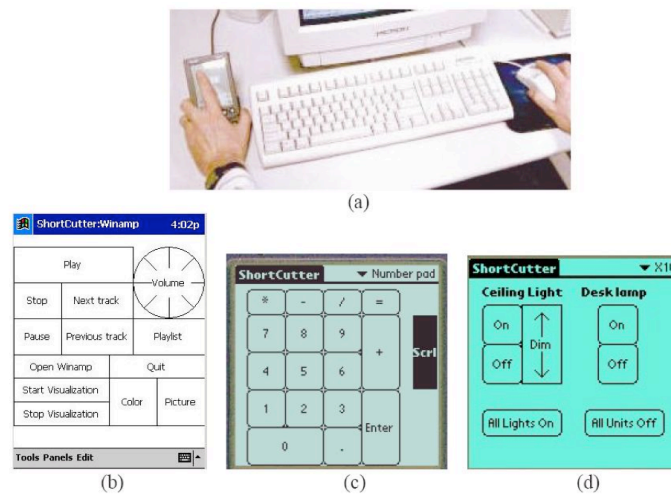
Référence	[Ishii and Ullmer 1997]
Entité 1	phicon
Entité 2	phicon
Hétérogénéité de superclasses	PP
Degré de généralité de E1	0
Degré de généralité de E2	0
Hétérogénéité de classes	Non
Hétérogénéité de niveaux	Non
Card(F)	1
Fonction générique	
Point de définition	Conception
Nombre d'occurrences d'instanciation	1
Durée d'instanciation	Inter-session
Cardinalité de la relation de couplage	1-1
Instrument de désignation 1	
Niveau désignation E1	
Acteur	concepteur
Instrument de désignation 2	
Niveau désignation E2	

**Tableau 5**  
**Taxonomie appliquée au cas particulier du couplage physique de deux phicons dans le Geospace.**

## 2.2. COUPLAGE MM : PEEBLES, DYNAMO, PROGRAMMABLE BEADS, SMART-ITS FRIENDS, ICON

Pebbles [Myers 2002], Dynamo [Izadi et al. 2003], les Programmable Beads [Resnick et al. 1998], les Smart-Its Friends [Holmquist et al. 2001] et Icon [Dragicevic 2004] ont en commun la capacité de permettre des couplages homogènes MM.

Dans Pebbles, le choix des entités est restreint aux PDA (entités mixtes intrinsèques dont la généralité est de niveau classes). Ces entités permettent de piloter une station de travail, des applications interactives ou servir de calculatrice « externe » comme le montre la Figure 11. Dans l'exemple, la technique de couplage consiste à déposer manuellement le PDA sur sa base (qui sert de représentation physique de la station de travail) ; le retirer entraîne le découplage.

**Figure 11****Couplage dans Pebbles.**

Un PDA à gauche d'un clavier (a) rend aisé l'utilisation de l'application Shortcutter sur un PocketPC (b) ou un Palm (c)(d) pour contrôler les applications d'un PC.

Dynamo [Izadi et al. 2003] permet le couplage dynamique de claviers et de souris. Le système est équipé de points d'interaction (interaction points). Ces points d'interaction sont des concentrateurs USB représentés de manière numérique (Figure 12). La technique d'interaction pour coupler un clavier et une souris consiste à connecter le clavier à un point d'interaction. Alors, le clavier prend le statut d'entité mixte construite qui est couplée à la souris connectée au même point d'interaction. Les fonctions résultantes sont celles des systèmes conventionnels : la souris fournit le focus au clavier et ce dernier enrichit de modificateurs (ctrl, alt, ...) les événements souris.



**Figure 12**

**Les points d'interaction dans Dynamo.**

En bas à gauche et à droite, sont représentés deux points d'interaction auxquels sont connectés des dispositifs USB.

Les perles visent un objectif pédagogique et ludique [Resnick et al. 1998]. En composant des perles augmentées de diodes électroluminescentes et d'un microprocesseur (entités mixtes intrinsèques), un enfant peut programmer des motifs. Par exemple, une lumière qui circule le long du collier (Figure 13). La technique d'interaction de couplage et de découplage s'apparente à la fabrication des colliers de perles. La technique d'assemblage impose qu'une perle soit connectée à deux autres perles au plus.

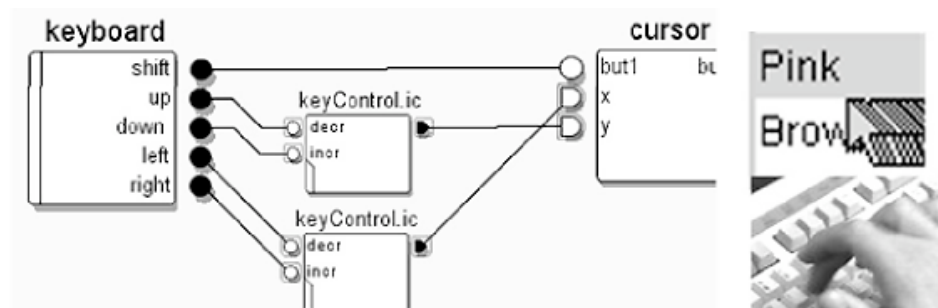
Dans ces trois systèmes (Peebles, Dynamo, Programmable Beads), le couplage repose sur des connexions physiques. De fait, les couplages peuvent être compris par le système à son redémarrage. La durée d'instanciation est donc vraisemblablement de type inter-sessions.



**Figure 13**  
**Couplage de Programmable Beads pour la confection d'un collier.**

Parmi les applications des Smart-Its Friends [Holmquist et al. 2001], on relève le couplage de deux Smart-Its en les tenant dans les mains jointes et en les agitant. La fonction obtenue est la mesure des distances séparant les entités couplées qui peut servir d'information contextuelle. Par exemple, en couplant sa carte de crédit et son porte-monnaie, on est sûr de ne pas partir en voyage sans l'un ou l'autre : si la distance entre les deux entités franchit le seuil admis, le système en avertit l'utilisateur.

Dragicevic propose Icon [Dragicevic 2004], une boîte à outils logicielle qui permet, selon un modèle flux de données, de construire des applications interactives configurables en entrée. À la phase de conception, une configuration en entrée correspond à un ou plusieurs couplages utilisés pour piloter une application interactive. La technique d'interaction utilisée est un configurateur logiciel dans lequel des composants sont connectés à l'aide d'un dispositif de pointage pour créer une configuration en entrée. La Figure 14 montre une partie de l'interface graphique. Bien que cette étape ne soit pas explicitement décrite dans les publications de l'auteur, une configuration peut être dynamiquement chargée (mais reste immuable en cours de session).



**Figure 14**  
**Configuration ICon pour le couplage d'un clavier avec une souris pour la fonction "Modifier".**

Le Tableau 6 caractérise les couplages permis par ces systèmes.

Référence	[Myers 2002]	[Izadi et al. 2003]			[Resnick et al. 1998]	[Holmquist et al. 2001]	[Dragicevic 2004]
Entité 1	PDA	clavier	<i>clavier</i>	<i>souris</i>	perle	smart it	disp d'entrée
Entité 2	PC	souris	<i>interaction point</i>	<i>interaction point</i>	perle	smart it	disp d'entrée
Hétérogénéité de superclasses	MM	MM	<i>PM</i>	<i>PM</i>	MM	MM	MM
Degré de généralité de E1	k	k	<i>k</i>	<i>k</i>	k	k	k
Degré de généralité de E2	k	k	<i>k</i>	<i>k</i>	k	k	k
Hétérogénéité de classe	Oui	Oui	<i>Oui</i>	<i>Oui</i>	Non	Non	Non
Hétérogénéité de niveau	Non	Non	<i>Non</i>	<i>Non</i>	Non	Non	Non
Card(F)	n	1	<i>1</i>	<i>1</i>	1	n	n
Fonction générique	manipuler	coopérer	<i>mixer</i>	<i>mixer</i>	coopérer	coopérer	coopérer
Point de définition	Exécution	Exécution	<i>Exécution</i>	<i>Exécution</i>	Exécution	Exécution	Exécution
Nombre d'occurrence d'instanciation	n	n	<i>n</i>	<i>?</i>	n	n	n
Durée d'instanciation	Inter-session	Inter-session	<i>Inter-session</i>	<i>Inter-session</i>	Inter-session	???	Session
Cardinalité de la relation de couplage	1-1	1-1	<i>1-1</i>	<i>1-1</i>	2-2	1-1	n-n
Instrument de désignation 1	aucun	aucun	<i>aucun</i>	<i>aucun</i>	aucun	aucun	
Niveau désignation E1	entité	entité	<i>entité</i>	<i>entité</i>	entité	entité	
Acteur	utilisateur	utilisateur	<i>utilisateur</i>	<i>utilisateur</i>	utilisateur	utilisateur	concepteur
Instrument de désignation 2	physique	physique	<i>physique</i>	<i>physique</i>	physique	physique	
Niveau désignation E2	rep phys	rep phys	<i>entité</i>	<i>entité</i>	entité	entité	
Acteur	utilisateur	utilisateur	<i>utilisateur</i>	<i>utilisateur</i>	utilisateur	utilisateur	concepteur

**Tableau 6**  
Synthèse des travaux de Resnick, Holmquist, Dragicevic, et Myers.

### 2.3. COUPLAGE MM : TRIANGLES, BUILDING BLOCKS, ET LA TABLE MAGIQUE

Le couplage de triangles augmentés [Gorbet et al. 1998] (entités mixtes intrinsèques) correspond à la composition d'une histoire. Chaque triangle est une pièce de plastique en forme de triangle équilatéral. Les bords sont équipés de connecteurs magnétiques, conducteurs d'électricité, permettant la communication entre les microprocesseurs embarqués. Dans le prototype de construction d'histoire, un personnage de l'histoire « Cendrillon » est associé à chaque triangle. En formant dynamiquement

une mosaïque de triangles (comme on agencerait les pièces d'un puzzle), l'utilisateur crée une histoire (Figure 15). Le couplage se fait en joignant les bords de deux triangles. Cette action est facilitée par les connecteurs magnétiques qui aident le positionnement et maintiennent le couplage. Un triangle peut donc être couplé à trois autres. Comme nous l'avons indiqué précédemment, puisque le couplage s'appuie en partie sur une connexion physique, la durée de d'instanciation est inter-sessions (le couplage peut être reconstruit automatiquement au redémarrage du système sans intervention de l'utilisateur).



**Figure 15**  
**Composition de Triangles pour former une histoire.**

---

Dans un esprit ludique et pratique, Anderson et ses collègues [Anderson et al. 2000] fondent leurs travaux de modelage 3D sur le couplage de blocs (entités mixtes intrinsèques) ressemblant à des pièces de Lego™. Ces pièces sont dotées de connecteurs et d'un microprocesseur. Le modelage 3D se fait par assemblage des pièces donc par couplage. En raison de leur forme, un bloc peut être couplé à six autres blocs au plus. Les blocs couplés sont capables par eux-mêmes de déduire la topologie de la construction globale (Figure 16). Du point de vue de TEC, ce prototype est en tout point semblable au précédent (à l'ensemble F près, et à la cardinalité des relations de couplage près).

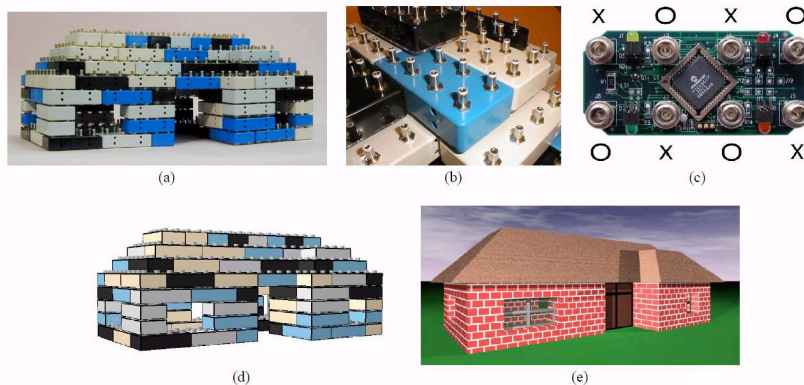


Figure 16

### Couplage de BuildingBlocks.

Les Blocs (b)(c) sont assemblés (a) pour former une construction. Les Blocs fournissent une représentation topographique (d). Une construction virtuelle en est déduite(e).

Le couplage de jetons de la Table Magique [Bérard 2003] est également comparable (aux entités et fonctions près) aux deux prototypes précédents. Pour obtenir la fonction de sélection par couplage, il suffit de mettre en contact deux jetons (entités mixtes construites). L'autre possibilité est le couplage pour la fonction de rotation/mise à l'échelle réalisée en couplant deux jetons à un même patch (Figure 17). Le découplage s'obtient en masquant l'un des deux jetons ou les deux à la fois (le suivi des jetons étant réalisé par vision par ordinateur). En réalité, le masquage de jetons détruit le couplage constructeur entre le jeton physique et la table augmenté et de là, le couplage du jeton et du patch. La particularité de ce prototype par rapport aux deux précédents réside dans les deux fonctions activables par couplage de jetons. Par contre, un jeton n'est couplable qu'avec un seul autre et puisque les jetons ne sont pas identifiables de façon unique, la durée d'instanciation est intra-session. Ces quatre derniers prototypes sont synthétisés dans le Tableau 7.

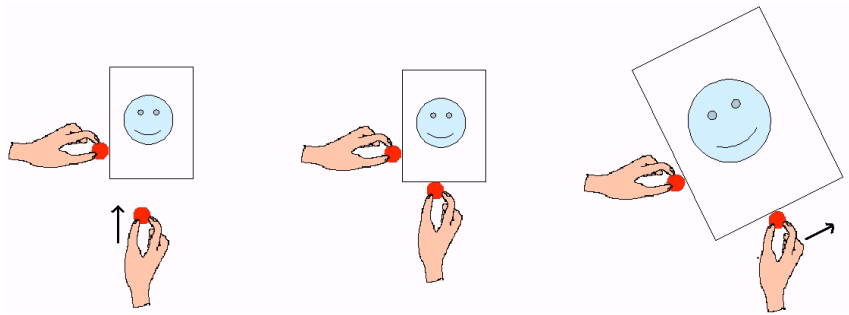


Figure 17

**Couplage de jetons dans la Table Magique.**

Deux jetons couplés dans la Table Magique pour la fonction de rotation et mise à l'échelle d'un

Référence	[Gorbet et al. 1998]	[Anderson et al. 2000]	[Bérard 2003]
Entité 1	triangle	block	jeton
Entité 2	triangle	block	jeton
Hétérogénéité de superclasses	MM	MM	MM
Degré de généralité de E1	k	k	k
Degré de généralité de E2	k	k	k
Hétérogénéité de classes	Non	Non	Non
Hétérogénéité de niveaux	Non	Non	Non
Card(F)	1	1	2
Fonction générique	coopérer	coopérer	coopérer
Point de définition	Exécution	Exécution	Exécution
Nombre d'occurrences d'instanciation	n	n	n
Durée d'instanciation	Inter-session	Inter-session	Session
Cardinalité de la relation de couplage	3-3	6-6	1-1
Instrument de désignation 1	aucun	aucun	aucun
Niveau désignation E1	entité	entité	entité
Acteur	utilisateur	utilisateur	utilisateur
Instrument de désignation 2	physique	physique	physique
Niveau désignation E2	entité	entité	entité
Acteur	utilisateur	utilisateur	utilisateur

Tableau 7

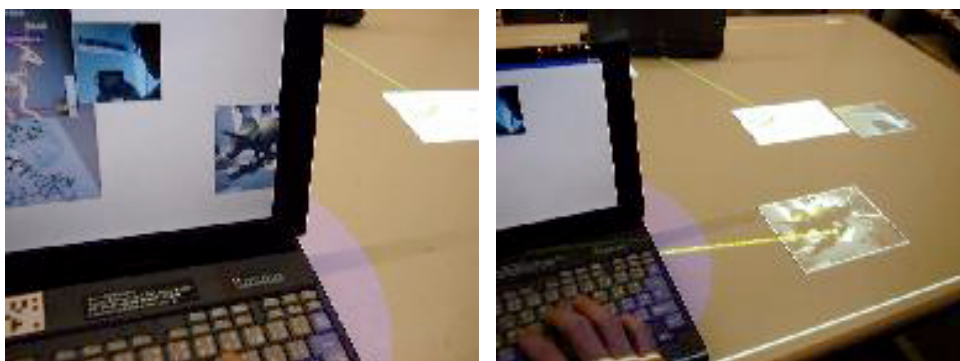
**Synthèse des travaux de Gorbet, Anderson et Bérard.**



---

**2.4. COUPLAGE MM :  
AUGMENTED  
SURFACES,  
SYNCHRONOUS  
GESTURE,  
CONNECTABLES,  
DATATILES**

De nombreuses solutions de l'état de l'art concernent le couplage de surfaces. Dans *Augmented Surfaces* [Rekimoto et Saitoh 1999], lorsque l'utilisateur dépose un ordinateur portable sur une table augmentée (deux entités mixtes intrinsèques), le système de vision par ordinateur reconnaît un tag et en déduit l'identité de l'ordinateur, sa position et son orientation. Grâce à ce couplage, la surface d'affichage de l'ordinateur portable est étendue par la table (Figure 18). Sur cette même figure (à droite), on distingue un deuxième ordinateur portable couplé également à la table, ce qui permet l'échange d'objets numériques entre les deux portables. En l'absence de précisions, nous donnons comme cardinalité de la relation de couplage 2-1. En raison de la technique d'interaction employée, la durée d'instanciation est de type inter-sessions (en relançant le système, le couplage est facilement rétabli).



**Figure 18**

**Couplage dans Augmented Surfaces.**

La surface d'affichage du portable est étendue par la table.

---

Dans *Synchronous Gesture*, Hinckley [Hinckley 2003] propose, comme Rekimoto, des fonctions d'extension de surface d'affichage. Cependant, la technique d'interaction est plus riche puisque qu'elle est déclinée pour proposer à l'utilisateur quatre fonctions. Par exemple, en choquant deux tablettes mobiles (entité mixtes intrinsèques) tenues par deux utilisateurs, un échange réciproque de cartes de visite est mis en œuvre ou, si une seule des tablettes est mobile et choque une autre tablette non tenue par un utilisateur, la première tablette annexe la surface d'affichage de la seconde (Figure 19). Hinckley propose également de coupler plusieurs tablettes en cascade. La cardinalité de la relation de couplage est donc 4-4 bien que le couplage en cascade pose des problèmes de « rebonds » entre tablettes. Ici la technique d'interaction, qui repose sur des chocs, ne permet pas de recréer automatiquement le couplage. En l'absence de précisions, nous considérons donc une durée d'instanciation de type intra-session.

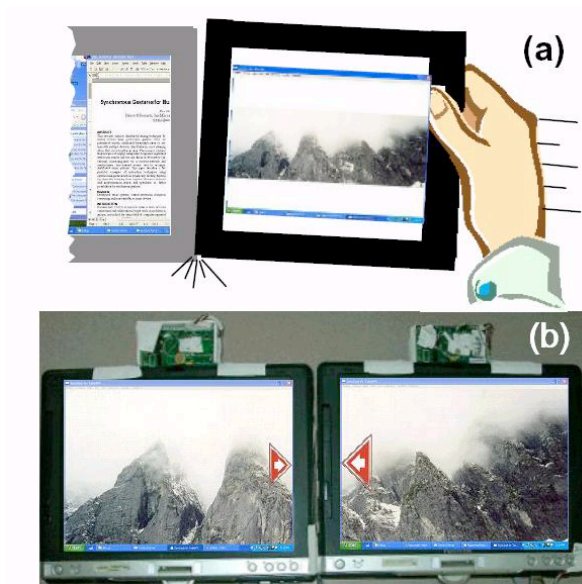


Figure 19

**Couplage de tablettes dans Synchronous Gesture.**

En choquant une tablette sur une autre (a), la surface de la seconde est annexée par la première (b).



Figure 20

**Couplage de deux Connectables.**

Les Connectables [Tandler et al. 2001] ont un fonctionnement similaire aux tablettes d'Hinckley. Moins mobiles, elles n'autorisent qu'un couplage des écrans par le bord haut (Figure 20) idéal pour une collaboration. L'ensemble des fonctions est restreint à la fusion des affichages initiaux pour ne former qu'un seul espace d'affichage. Le couplage est donc limité à deux Connectables (cardinalité de la relation de couplage 1-1). Ce sont les seuls points de divergence avec Synchronous Gesture.

Contrairement à ces prototypes, les Datatiles [Rekimoto et al. 2001] relèvent bien du couplage de surface mais pas uniquement pour des fonctions d'affichage. Les tuiles transparentes sont posées sur un écran à

cristaux liquides (couplage constructeur des entités DataTiles mixtes que nous avons déjà évoqué). Deux techniques de couplage sont possibles pour que les tuiles collaborent. La première consiste à poser deux tuiles côte à côte. La Figure 21 à gauche montre le couplage d'une tuile de gestion du temps (la forme circulaire servant de guide pour le stylet) avec une tuile d'affichage d'une vidéo. À droite, la même tuile de gestion du temps avec une tuile affichant une carte météo. Deux tuiles peuvent aussi être couplées par un geste d'une tuile vers l'autre (Figure 22a et c) et découplées par un geste symbolisant la séparation des deux tuiles (Figure 22 b). Si la cardinalité de la relation de couplage pouvait être réduite à 4-4 en utilisant seul le positionnement spatial, la technique de liaison autorise le couplage d'un nombre arbitraire de tuiles. Le Tableau 8 présente la synthèse de ces trois travaux.

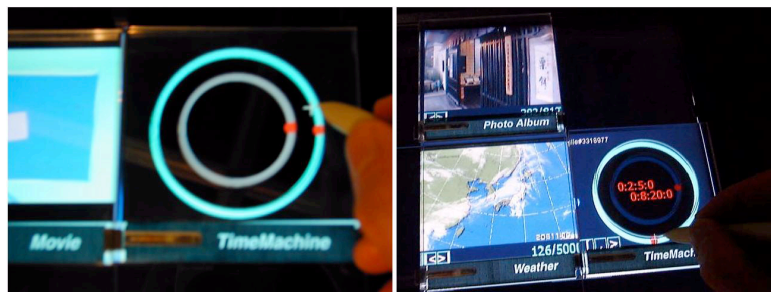


Figure 21

**Couplage de deux DataTiles.**

A gauche une tuile de gestion du temps couplée à une tuile d'affichage vidéo. A droite, une tuile de gestion du temps couplée à la tuile météo.

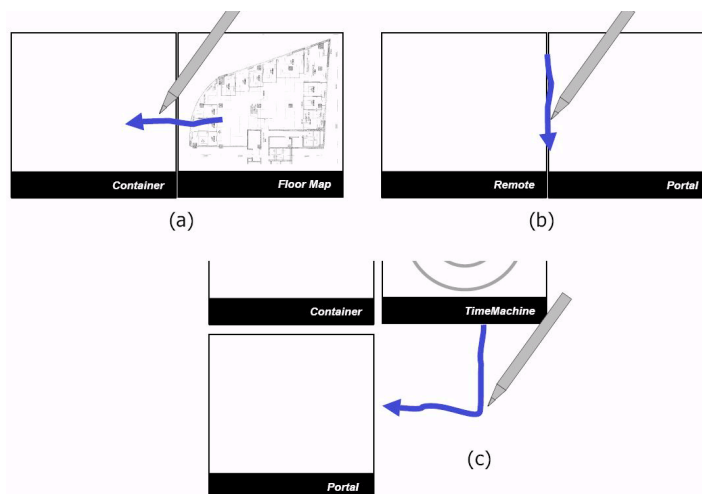


Figure 22

**Couplage de DataTiles par un geste.**

Deux tuiles peuvent être couplées par un geste d'une tuile vers l'autre (a) (c), et découplées par un geste symbolisant la séparation des deux tuiles (b).

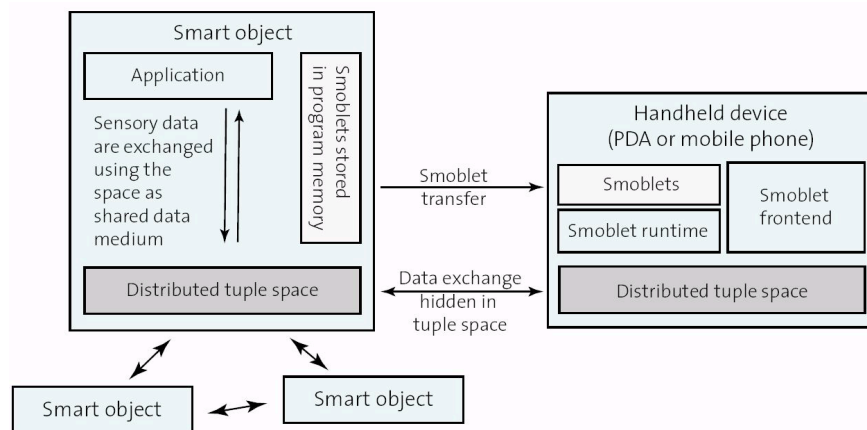
Référence	[Rekimoto and Saitoh 1999]	[Hinckley 2003]	[Tandler et al. 2001]	[Rekimoto et al. 2001]	
Entité 1	portable	tablette	connectable	tuile	<i>tuile</i>
Entité 2	table	tablette	connectable	tuile	<i>table augmentée</i>
Hétérogénéité de superclasses	MM	MM	MM	MM	<i>PM</i>
Degré de généralité de E1	k	k	k	k	<i>k</i>
Degré de généralité de E2	k	k	k	k	<i>1</i>
Hétérogénéité de classes	Oui	Non	Non	Non	<i>Oui</i>
Hétérogénéité de niveaux	Non	Non	Non	Non	<i>Oui</i>
Card(F)	1	4	1	n	<i>1</i>
Fonction générique	coopérer	coopérer	coopérer	coopérer	<i>mixer</i>
Point de définition	Exécution	Exécution	Exécution	Exécution	<i>Exécution</i>
Nombre d'occurrences d'instanciation	n	n	n	n	<i>n</i>
Durée d'instanciation	Inter-session	Session	Inter-session	Inter-session	<i>Inter-session</i>
Cardinalité de la relation de couplage	2-1	4-4	1-1	n-n	<i>n-1</i>
Instrument de désignation 1	aucun	aucun	aucun	aucun	<i>aucun</i>
Niveau désignation E1	entité	entité	entité	entité	<i>entité</i>
Acteur	utilisateur	utilisateur	utilisateur	utilisateur	<i>utilisateur</i>
Instrument de désignation 2	physique	physique	physique	physique	<i>physique</i>
Niveau désignation E2	entité	entité	entité	entité	<i>entité</i>
Acteur	utilisateur	utilisateur	utilisateur	utilisateur	<i>utilisateur</i>

**Tableau 8**  
**Synthèse des travaux de Rekimoto, Hinckley, Tandler.**

## 2.5. COUPLAGE MM : SMOBLLET ET GAS

Smoblet [Siegemund et Krauer 2004] et GAS [Christopoulou et Kameas 2004] offrent principalement des solutions logicielles et théoriques pour la mise en œuvre du couplage. Un Smoblet permet à n'importe quel objet « intelligent » de la vie courante d'accéder dynamiquement aux ressources (mémoire, de calcul,...) des dispositifs mobiles

d'un utilisateur situé à proximité [Siegemund et Krauer 2004]. L'idée sous-jacente est que les objets de la vie courante peuvent emprunter à d'autres entités, grâce à des couplages, les ressources dont ils ont besoin. Une structure de données réparties facilite la coopération entre plusieurs objets intelligents et les dispositifs mobiles d'un utilisateur. Sur la Figure 23, deux types de couplage sont possibles : entre objets intelligents (entités mixtes de niveau classes) ou entre des objets intelligents et des dispositifs portés (niveau classes). En l'absence de précisions, nous considérons puisqu'il s'agit d'une solution principalement logicielle, que la durée d'instanciation est de type intra-session. La description du couplage laisse penser qu'aucune technique d'interaction n'est prévue puisque les objets intelligents se découvrent et se couplent automatiquement (interaction implicite).



**Figure 23**  
**Couplage et coopération de Smart Objects.**  
A gauche; entre un Smart Object et un PDA à droite.

Christopoulou et Kameas [Christopoulou et Kameas 2004] s'intéressent à la transformation des objets de la vie courante en artefact (en ajoutant des capacités de capture, communication et de calcul). Comme des composants logiciels, les artefacts exposent leurs propriétés, capacités et services par le biais d'interfaces et sont couplables dynamiquement. La Figure 24 montre le couplage d'une eLamp et d'un eBook : l'ouverture du livre provoque l'allumage de la lampe. Inversement, la fermeture du livre fait éteindre la lampe. L'approche est intéressante, car les auteurs proposent une ontologie dans laquelle apparaît notamment le concept de compatibilité que nous reprendrons dans la suite de ce manuscrit. Ces contributions, dont une partie est théorique, présentent l'avantage d'être applicables à n'importe quel type d'entité.

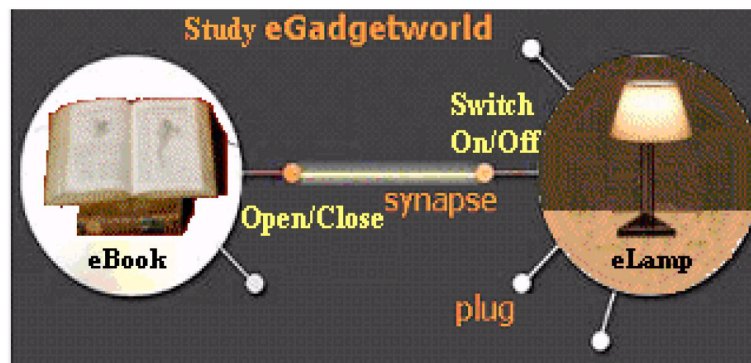


Figure 24  
Couplage d'un eBook avec une eLamp.

Ces deux derniers exemples, résumés dans le Tableau 9, achèvent notre présentation de l'état de l'art dont nous présentons la synthèse dans la section qui suit.

Référence	[Siegmund et Krauer 2004]	[Christopoulou et Kameas 2004]
Entité 1	smart obj	artefact
Entité 2	smart obj	artefact
Hétérogénéité de superclasses	MM	MM
Degré de généralité de E1	n	n
Degré de généralité de E2	n	n
Hétérogénéité de classes	Non	Non
Hétérogénéité de niveaux	Non	Non
Card(F)	n	n
Fonction générique	coopérer	coopérer
Point de définition	Exécution	Exécution
Nombre d'occurrences d'instanciation	n	n
Durée d'instanciation	???	???
Cardinalité de la relation de couplage	n-n	n-n
Instrument de désignation 1	rep num	aucun
Niveau désignation E1	rep num	rep num
Acteur	utilisateur	système
Instrument de désignation 2	rep num	aucun
Niveau désignation E2	rep num	rep num
Acteur	utilisateur	système

Tableau 9  
Synthèse des travaux de Siegmund et Krisopoulou.

### 3. Analyse synthétique

Nous venons de présenter en détail des exemples de l'état de l'art selon les dimensions taxonomiques de TEC. De ces exemples et de bien d'autres prototypes non inclus dans notre discussion, nous sommes maintenant en mesure de dégager des tendances, des patrons, voire des généralisations sur le couplage en informatique ambiante. Nous les présentons selon les quadrants de TEC : entités, fonctions, couplage et technique d'interaction.

#### 3.1. ENTITÉS : GÉNÉRICITÉ ET HÉTÉROGÉNÉITÉ

TEC, on le rappelle, caractérise les deux entités susceptibles d'intervenir dans un couplage par les niveaux de généralité et d'hétérogénéité que ce couplage autorise :

- Pour la généralité : niveau instance (noté 0), niveau classes (noté k), et niveau superclasses (noté n).
- Pour l'hétérogénéité, on distingue : (a) l'hétérogénéité par les superclasses (de valeur PN, PM, NM) - et inversement l'homogénéité par les superclasses (de valeur PP, MM, NN) ; (b) l'hétérogénéité/ homogénéité par les classes (dite de niveau k) et (c) l'hétérogénéité par les niveaux d'abstraction (lorsque les deux entités ont des niveaux de généralité distincts).

Les couplages homogènes NN et PP ne sont pas au centre de notre étude que nous voulons centrée sur la fusion des mondes numérique et physique. Nous retenons désormais les couplages dans lesquels interviennent des composantes numériques et physiques, c'est-à-dire les mixages de type PN, PM, MN et MM.

Les couplages PN désignent la construction d'entités mixtes à partir d'objets physiques. Dans les exemples de l'état de l'art actuel,

- L'entité physique P et l'entité numérique N sont étroitement liées par la fonction « représenter ». Il en résulte que le P ressemble au N qu'il représente (métaphore de nom et/ou de verbe, dans la terminologie de Fishkin [Fishkin 2004]).
- L'entité mixte résultante n'existe pas pour elle-même, mais intervient dans d'autres couplages (cas de la souris, des tuiles des DataTiles [Rekimoto et al. 2001], des phicons du metaDesk [Ishii et Ullmer 1997]).

En complément de l'état de l'art, TEC permet d'envisager de nouvelles pistes, par exemple, un couplage PN de généralité (n, n) dont le point de définition serait « exécution ». Ce couplage signifierait que l'utilisateur aurait la capacité de fabriquer des entités mixtes à partir de

n'importe quel objet physique en l'associant à n'importe quelle entité numérique. La (ou les) technique(s) d'interaction correspondante(s) pour réaliser ce type de couplage relève(nt) du end-user development, un domaine qui prend un nouvel essor depuis le end-user programming [Sutcliffe et al. 2004].

Pour les couplages PM, soit (a) l'entité P est couplée à la composante P de l'entité M, soit (b) l'entité P est couplée à la composante N de l'entité M. Le choix entre les options (a) et (b) dépend de la cardinalité de la relation de couplage des composantes P et N de M.

- Dans le cas (a), on assiste à une configuration P-P-N. Dans l'exemple de la Table Magique, un jeton P posé sur la table M (P-N) est couplé à la table selon la configuration P-P-N.
- Pour le cas (b), on obtient une configuration P-N-P qui indique que deux entités physiques partagent la même entité numérique (sans nécessairement fournir les mêmes fonctions). Si les deux couplages fournissaient la fonction « représenter », alors la configuration P-N-P correspondrait au principe bien connu de la représentation multiple (ici, physique) d'un même concept (numérique). Nous n'avons pas relevé d'exemple de ce type de configuration dans l'état de l'art.

Pour les couplages MN, soit (a) l'entité N est couplée à la composante N de l'entité M donnant la configuration N-N-P, soit (b) l'entité N est couplée à la composante P de l'entité M selon la configuration N-P-N.

- L'état de l'art contient plusieurs exemples relevant du cas (a), fournissant les fonctions génériques « contenir » ou « manipuler » selon une généricité (k, n) : l'entité mixte (de généricité k) permet de contenir ou de manipuler n'importe quelle entité numérique (qui est donc de généricité n). Dès lors, contrairement aux couplages PN, la composante P de M n'a pas (ou ne doit pas avoir) de ressemblance avec l'entité numérique à laquelle M est couplée, mais a une ressemblance avec la fonction résultat du couplage. Si l'on reprend la terminologie de Fishkin, la composante P de M doit obéir à une métaphore de verbe plutôt qu'à une métaphore de nom, voire à l'absence totale de métaphore.
- L'assemblage de type N-P-N peut s'interpréter comme le partage d'un objet physique par plusieurs entités numériques (sans nécessairement fournir les mêmes fonctions). Dans les Augmented Tables, le transport par hyperdragging d'un objet numérique sur la table peut se voir comme un couplage N-P-N entre la table N-P et l'entité N.



Le raisonnement précédent s'applique également aux couplages MM, mais avec une combinatoire plus élevée : les entités mixtes peuvent se coupler par les P (donnant une configuration N-P-P-N), par les N (configuration P-N-N-P), ou par un mélange de P et de N (P-N-P-N). Ces aspects seront discutés en détail au chapitre suivant. L'état de l'art révèle des exemples de couplage par les P (les DataTiles [Rekimoto et al. 2001] ou les Triangles [Gorbet et al. 1998]) ou des couplages par les N (comme les Smoblets [Siegemund et Krauer 2004] et GAS [Christopoulou et Kameas 2004]), mais nous n'avons pas relevé d'exemple de configuration P-N-P-N. Les couplages MM relevés dans la littérature sont généralement homogènes de généricité (k, k) et ont pour vocation la coopération. Il est alors indispensable que les entités puissent se découvrir et communiquer en partageant des caractéristiques communes portées par une classe particulière. Nous verrons en 3.3. que le couplage par les P peut contraindre la cardinalité des relations de couplage.

Comme le montre l'ensemble des tableaux récapitulatifs de ce chapitre, la plupart des prototypes MN sont hétérogènes par les niveaux d'abstraction puisque la généricité des couplages possible est (k, n) : la généricité des entités mixtes est de niveau classes et celle des entités numériques de niveau superclasses. Cette différence s'explique par le fait que les entités mixtes reposent sur une base matérielle nécessairement spécifique. Le constat est général sur notre taxonomie où l'on vérifie que les entités mixtes sont presque exclusivement de niveau classes. Les rares exceptions sont des solutions théoriques et logicielles utilisables en principe avec n'importe quelle entité physique [Siegemund et Krauer 2004] [Christopoulou et Kameas 2004] ou des solutions simples comme l'estampillage d'entités physiques [Rekimoto et Saitoh 1999].

---

### **3.2. L'ENSEMBLE DES FONCTIONS CIBLES : CARDINALITÉ ET GÉNÉRICITÉ**

Les tableaux de ce chapitre montrent qu'une majorité de prototypes n'offre que des couplages unifonctionnels. Autrement dit, la technique d'interaction n'a pas le rôle de désignation de la fonction cible. Le couplage pourrait être réduit à ExE  $\rightarrow$  F,  $\text{card}(F) = 1$ . Cette unicité s'explique en partie par le caractère expérimental des travaux dont le but est le plus souvent la démonstration de concepts. La Table Magique de Bérard et les tablettes d'Hinckley font exception, deux prototypes visant à faciliter la coopération.

Nous avançons comme explication à la plurifonctionnalité de la fonction de coopération que cette dernière peut être déclinée de différentes façons. De plus, suite au couplage, la fonction de coopération peut être portée par l'entité mixte ou par l'entité numérique. Dans les prototypes de l'état de l'art, les entités numériques sont généralement des

données, non pas des fonctions ou des programmes. Si les fonctions sont portées par les entités mixtes, le couplage de deux entités mixtes offre a priori un potentiel fonctionnel plus riche.

Concernant la couverture fonctionnelle de F, aucun prototype ne couvre toutes les fonctions génériques de base et notamment mixer.

---

### 3.3. COUPLAGE ET INSTANCIATION

TEC caractérise l'instanciation du couplage en termes de point de définition (à la conception et/ou à l'exécution), de nombre d'occurrences d'instanciation (de 1 à autant de fois voulues), de durée de vie d'une instanciation (intra- ou inter-sessions), et de cardinalité de la relation de couplage mesurant le partage d'entités entre plusieurs couplages.

Nos tableaux récapitulatifs montrent que tous les couplages fournissant la fonction « représenter » ont comme point de définition la conception. En l'état, l'utilisateur ne peut donc pas créer de nouvelles façons de représenter les choses. Par contre, le nombre d'occurrences d'instanciation à l'exécution n'est généralement pas limité (ce qui suppose toutefois l'existence d'une technique d'interaction pour commander le découplage).

Concernant la durée de vie d'une instanciation, tous les cas sont représentés. Notons qu'un couplage par les P milite en faveur d'une instanciation inter-sessions alors qu'un couplage par les N est plutôt intra-session. Par exemple, on peut raisonnablement supposer (bien que non précisé dans les articles publiés) que les colliers de Beads (perles) [Resnick et al. 1998] sont des couplages dont la durée de vie est inter-sessions. Deux jetons de la Table Magique [Bérard 2003] couplés à la même entité numérique interviennent dans un couplage à durée de vie intra-session. Inversement, ces mêmes jetons, s'ils restent posés sur la table entre deux sessions restent couplés chacun à la table (car il s'agit d'un couplage par les P). On observe aussi le cas de la SenseTable [Patten et al. 2001] où la durée de vie du couplage d'un puck et d'une donnée est inter-sessions.

La cardinalité de la relation de couplage peut dépendre étroitement de la forme des composantes P des entités et de la technique d'interaction. Par exemple, on observe une cardinalité 3 pour les Triangles (qui ont trois côtés) et une cardinalité 4 pour les tablettes rectangulaires de Hinckley. La contrainte vient de la technique d'interaction de commande du couplage qui s'appuie sur la géométrie et la connexité des composantes P. Dans le cas des DataTiles [Rekimoto et al. 2001], la cardinalité de la relation de couplage peut être supérieure à 4 grâce à la technique d'interaction qui fait intervenir un instrument tiers : l'utilisateur peut coupler deux tuiles non contiguës en les désignant au moyen d'un stylet. Nous verrons au chapitre 4 traitant des propriétés ergonomiques l'impact de la cardinalité de la relation de couplage sur la mise en œuvre logicielle.

---

### 3.4. TECHNIQUES D'INTERACTION

Dans TEC, nous caractérisons une technique d'interaction par trois composantes : la technique de désignation des entités, le mode de représentation des entités et l'acteur responsable de la commande du couplage (utilisateur ou système).

Les tableaux récapitulatifs révèlent l'émergence d'un nouveau patron de technique de désignation : lorsque la première entité est désignée de manière non instrumentale, elle sert d'instrument pour désigner la seconde entité. Ainsi, avec l'informatique ambiante, l'interaction passe du monde numérique au monde physique en accord avec l'axe « Intégration avec l'environnement physique » de Lyytinen et Yoo [Lyytinen et Yoo 2002] présenté dans le chapitre d'introduction. Dans le même esprit, on trouve peu d'exemples de représentations numériques qui ajouteraient un niveau d'indirection.

De ce fait, les techniques d'interaction s'ancrent dans le monde physique avec quelques formes émergentes inspirées des usages et actions dans le monde physique : positionnement d'une entité sur la seconde pour les solutions à base de surfaces ([Bérard 2003], [Rekimoto et al. 2001], [Fitzmaurice et al. 1995], [Patten et al. 2001]), gestes synchronisés ([Holmquist et al. 2001], [Hinckley 2003], [Patten et al. 2001] pour le découplage), connexité physique ([Resnick et al. 1998], [Gorbet et al. 1998], [Ullmer et al. 1998], [Tandler et al. 2001], [Patten et al. 2001], ou mise en contact éphémère [Bérard 2003]).

Dans l'état de l'art actuel, toutes ces techniques d'interaction sont destinées à l'utilisateur. À l'exception des couplages dont le point de définition est à la conception, l'instanciation des couplages revient à l'utilisateur. Avec la notion de migrabilité de tâche que nous présentons au chapitre suivant, nous verrons comment la commande du couplage peut également être prise en charge par le système.

## 4. Synthèse

---

L'analyse de l'état de l'art en matière de couplage de ressources d'interaction révèle une grande diversité d'approches sans toutefois couvrir toutes les options envisageables selon TEC. En particulier, nous relevons l'absence de représentations multiples physiques d'un même concept numérique, l'absence de configurations P-N-P-N, et surtout l'absence de couplage PN de généralité (n, n) et de point de définition exécution, autrement dit, l'impossibilité pour les utilisateurs actuels de façonner l'espace ambiant à partir de n'importe quelles entités des mondes physique et numérique. Or, dans la vision de l'informatique ambiante, le

comportement opportuniste des utilisateurs, de même, la reconfiguration dynamique des systèmes prévalent. Il convient donc d'étudier la dynamique des couplages et leurs liens avec l'utilisabilité selon un cadre systématique. Ces aspects font l'objet du chapitre 4 suivant.

Le constat est identique concernant les techniques d'interaction. Par exemple, secouer un Puck de la SenseTable [Patten et al. 2001] provoque son découplage alors que secouer deux entités Smart-Its [Holmquist et al. 2001] ensemble réalise leur couplage. Au chapitre 5, nous proposons un formalisme pour raisonner sur les techniques d'interaction de manière unifiée.



---

*Chapitre IV*

*Cadre d'analyse pour la  
dynamique des couplages de  
ressources d'interaction*

---

## *Avant-propos*

---

Au Chapitre I, nous avons introduit la notion de couplage avec l'exemple de l'utilisation conjointe de la fourchette et du couteau. Pour illustrer la nécessité de considérer la dynamique des couplages et leur utilisabilité, nous reprenons à notre compte le « dîner des philosophes » utilisé en cours de système comme paradigme du partage de ressources critiques :

- Cinq philosophes sont assis autour d'une table ronde.
- Chaque philosophe passe sa vie alternativement à penser et à manger. Au centre de la table, un immense plat de spaghetti,
- La table ne comporte que 5 assiettes et 5 fourchettes et tout le monde sait que les philosophes utilisent deux fourchettes pour manger un plat de spaghetti.
- Une fourchette est placée entre chaque philosophe et comme chaque philosophe est bien élevé, il utilise uniquement les fourchettes situées à sa droite et à sa gauche. Un philosophe ne prend pas une fourchette déjà détenue par un autre philosophe.
- Quand il a fini de manger, le philosophe repose les deux fourchettes utilisées.

Cet exemple appelle les remarques suivantes. Un philosophe ne peut manger qu'en couplant deux fourchettes. Les fourchettes sont sans cesse couplées et découplées. Une fourchette ne peut pas être couplée avec n'importe quelle autre. Il se peut aussi qu'une fourchette ne soit pas libre à un instant donné si un autre philosophe la détient. Pour constater qu'une fourchette n'est pas libre, un philosophe doit pouvoir observer ses voisins (il ne leur est pas possible de dîner dans le noir).

Dans ce chapitre, nous proposons un cycle de vie régissant l'évolution dans le temps d'un couplage et étudions certains aspects de la synchronisation de plusieurs cycles de vie (section 1). Nous effectuons ensuite une analyse de ce cycle de vie à l'aide de propriétés ergonomiques (section 2). En explorant plus avant des exemples, il apparaît que la mise en œuvre d'un nouveau couplage à l'exécution devrait dans certains cas donner lieu à la génération d'autres couplages. Dans l'exemple des philosophes, lorsque deux fourchettes sont couplées, on peut considérer qu'elles sont alors couplées automatiquement à l'assiette du philosophe qui les détient. Aussi, à l'aide de notre travail sur la mécanique liée à un couplage, nous fournissons un mécanisme qui permet de raisonner sur la génération dynamique de couplage par effet de bord (section 3).

Dans ce chapitre, nous nous intéressons exclusivement aux entités mixtes (construites ou intrinsèques) jouant le rôle de ressource d'interaction. Nous les étudions sous l'angle du couplage dynamique,

c'est-à-dire en se référant à TEC, aux couplages dont le point de définition est l'exécution. Aussi, dans ce chapitre lorsque nous parlons par exemple de la ressource d'interaction « souris 1 », nous désignons en réalité, l'entité mixte (construite ou intrinsèque) composée d'une souris physique couplée à son curseur numérique.

## 1. Cycle de vie d'un couplage

Dans le cadre que nous venons de fixer, l'utilisateur doit disposer à tout instant des « indices nécessaires et suffisants » pour juger de l'existence ou de la possibilité de couplage entre deux ressources d'interaction. Cette exigence tient de la nécessité pour l'utilisateur de construire, au cours du processus d'interaction, une représentation mentale en conformité avec l'état interne du système [Norman 1986]. Cette exigence précisée, nous posons que le couplage de ressources d'interaction passe par une suite d'états observables et contrôlables par l'utilisateur. Tout état se définit au moyen de critères. Dans ce qui suit, nous rappelons notre définition de la notion de couplage, puis nous introduisons les critères caractéristiques d'un état de couplage et les conditions de transition entre états. Ainsi, un couplage obéit à un cycle de vie.

### 1.1. COUPLAGE DE RESSOURCES D'INTERACTION : RAPPEL DE LA DÉFINITION

Le couplage de deux entités est l'acte de lier ces entités de manière à ce qu'elles opèrent conjointement pour offrir un nouvel ensemble de fonctions qu'elles ne pourraient offrir individuellement. C'est aussi le résultat de cette action, c'est-à-dire une nouvelle entité composée des deux entités liées.

### 1.2. UN ÉTAT : TROIS CRITÈRES DE VÉRITÉ

Soient  $R$  l'ensemble des ressources d'interaction (c'est-à-dire des entités jouant le rôle de ressource d'interaction),  $F$  l'ensemble des fonctions de couplage et  $C$  l'ensemble des couplages. Par rapport à notre définition du couplage présentée au Chapitre II, nous nous affranchissons ici de  $T$  (l'ensemble des techniques d'interaction) qui ne nous intéresse pas dans la définition du cycle de vie.

Soient  $r_1$  et  $r_2$  deux ressources d'interaction,  $F$  l'ensemble des fonctions obtenues par le couplage  $c$  de  $r_1$  avec  $r_2$ , noté  $(r_1, c, r_2)$  ; avec :

$$r_1, r_2 \in R, F \subset F, c \in C$$

Un état de  $(r_1, c, r_2)$ , est défini par l'ensemble des trois critères de vérité suivants :



- $r_1$  est (couplée/non couplée) avec  $r_2$  pour  $c$ ,
- $r_1$  est (verrouillée/non verrouillée) pour  $r_2$  et pour  $c$ ,
- $r_1$  est (couplable/non couplable) à  $r_2$  pour  $c$ , ou bien  $r_1$  est (découplable/non découplable) de  $r_2$  pour  $c$ .

Ou, si l'on utilise les prédicats équivalents :

- *Couplée* ( $r_1, c, r_2$ ) est vrai si et seulement si  $F \neq \emptyset$ . Si  $F = \emptyset$ , *Couplée* ( $r_1, c, r_2$ ) est faux et *NonCouplée* ( $r_1, c, r_2$ ) est vrai.
- *Verrouillée* ( $r_1, c, r_2$ ) est vrai si l'état de  $r_1$  ne permet pas de changer l'état de ( $r_1, c, r_2$ ). Ce prédicat traduit l'indisponibilité de  $r_1$  pour entrer ou sortir du couplage  $c$  avec  $r_2$ . Par exemple, un utilisateur ne veut pas joindre à l'écran public, l'écran privé de son PDA utilisé à des fins personnelles. L'état de ( $r_1, c, r_2$ ) est contraint au statu quo jusqu'à ce que *Verrouillée* ( $r_1, c, r_2$ ) devienne faux ou, de manière duale *NonVerrouillée* ( $r_1, c, r_2$ ) vaut vrai.
- *Couplable* ( $r_1, c, r_2$ ) est une expression de prédicats autres que *Couplée* ( $r_1, c, r_2$ ) et *Verrouillée* ( $r_1, c, r_2$ ). Elle regroupe toutes les conditions nécessaires à la réalisation de ( $r_1, c, r_2$ ) autres que les conditions sur l'existence de  $F$  ou sur le verrouillage de  $r_1$  vis-à-vis de  $r_2$  dans  $c$ . Par exemple, les compatibilités de forme, de rôle et de cardinalité de la relation de couplage, peuvent être modélisées par *Couplable*. De manière symétrique, *Découplable* exprime toutes les conditions, autres que les conditions sur l'existence de  $F$  ou sur le verrouillage, nécessaires à la réalisation de la destruction de ( $r_1, c, r_2$ ).

---

### 1.3. TRANSITIONS

Les transitions entre états correspondent à des événements dont la prise en compte par le système modifie la valeur de l'un des prédicats de vérité. Ces événements traduisent soit la demande de couplage/découplage (c'est-à-dire la demande de lancement ou de suspension des fonctions de  $F$ ), soit le verrouillage/déverrouillage de ressource, soit leur couplabilité/découplabilité.

---

### 1.4. CYCLE DE VIE

L'automate de la Figure 1 modélise le cycle de vie de ( $r_1, c, r_2$ ). Le cycle de vie comprend deux sous-automates, l'un formé par les états 1, 2, 3, 4 pour lesquels *Couplée* ( $r_1, c, r_2$ ) est vrai, l'autre par les états 5, 6, 7, 8 pour lesquels *Couplée* ( $r_1, c, r_2$ ) est faux, les états 4 et 6 servant de passerelle entre les deux sous-automates. L'état 4 correspond à la situation où toutes les conditions sont réunies pour que ( $r_1, c, r_2$ ) soit réalisé. Il ne

manque qu'un événement de demande de couplage pour entrer dans l'état 6.

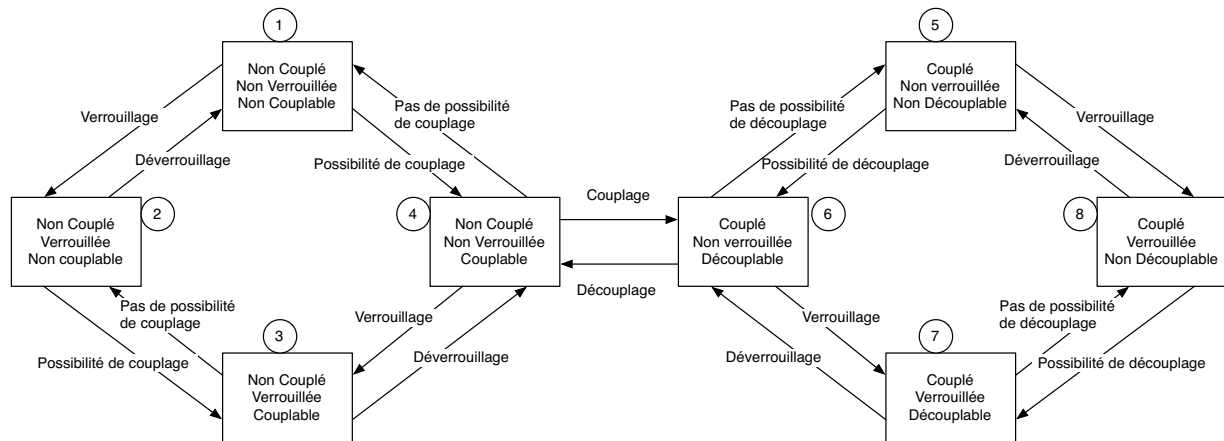


Figure 1  
Automate du cycle de vie du couplage  $(r_1, c, r_2)$ .

## 1.5. ILLUSTRATIONS

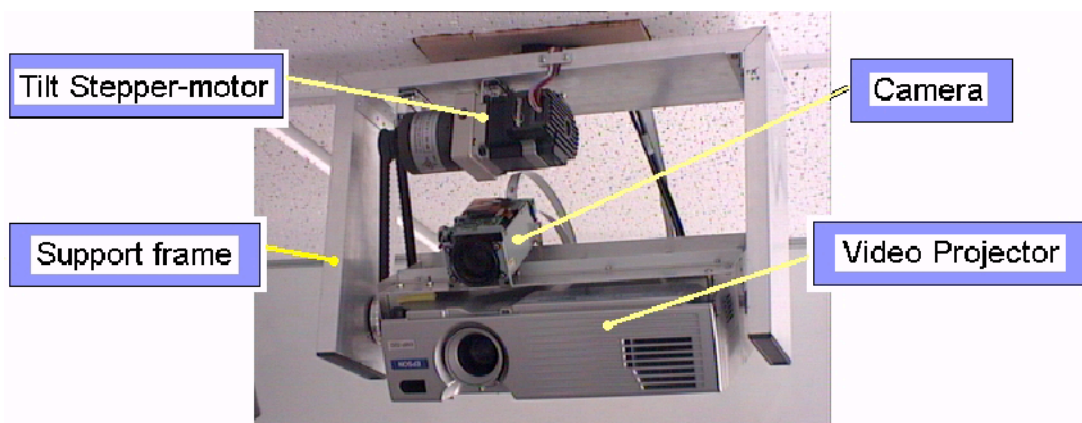
### La Table Magique

À titre d'illustration, déroulons l'automate pour le cas de deux jetons  $j_1$  et  $j_2$  de la Table Magique [Bérard 2003]. A l'initialisation du système,  $(j_1, c, j_2)$  est dans l'état 4 :  $j_1$  n'est ni couplé ni verrouillé et couplable vis-à-vis de  $j_2$  puisqu'il est sur la table, de même couleur, de même forme et de même taille que  $j_2$  (le suivi des jetons s'appuie sur un système de vision par ordinateur fondé sur un modèle de couleur). La mise en contact de  $j_1$  et  $j_2$  signifie, pour le système, une demande de couplage.  $(j_1, c, j_2)$  entre dans l'état 6 et la fonction « sélectionner un patch » est disponible. Si l'utilisateur sélectionne un patch, puis masque  $j_2$  (ou  $j_1$ ) de la main,  $(j_1, c, j_2)$  revient en 4. Mais comme  $j_1$  est maintenant attaché à un patch (couplage jeton-patch),  $j_1$  est verrouillé pour  $(j_1, c, j_2)$ , et  $(j_1, c, j_2)$  passe de l'état 4 à l'état 3.  $(j_1, c, j_2)$  reviendra à l'état 4 dès que  $j_1$  sera détaché du patch, c'est-à-dire dès qu'il sera masqué par la main. S'il est verrouillé pour  $(j_1, c, j_2)$ ,  $j_1$  n'est pas verrouillé pour  $(j_1, c', j_2)$  où  $c'$  donne accès aux fonctions de rotation, de zoom et de destruction de patch.  $(j_1, c', j_2)$  est dans l'état 4. Il passe en 6 dès que l'utilisateur pose  $j_2$  à l'intérieur du patch auquel  $j_1$  est attaché.

### SCP et PDS

Dans [Borkowski et al. 2004] nous trouvons la description d'un système basé sur une caméra et un projecteur orientable (Steerable Camera-Projector, SCP) qui permet de distribuer dynamiquement l'affichage d'interfaces graphiques sur les surfaces planes d'une pièce et

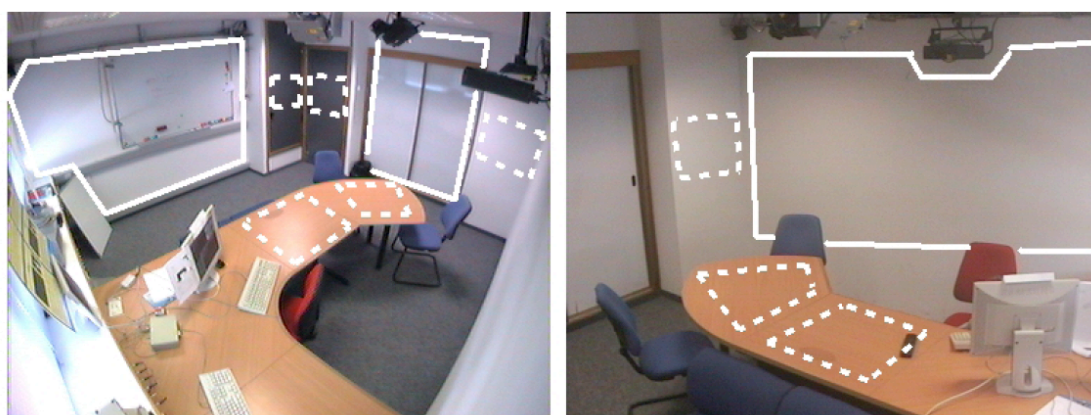
également d'interagir avec ces interfaces. La caméra et le projecteur (deux entités mixtes naturelles) sont couplés (Figure 2), aussi bien physiquement que logiciellement car la caméra a pour rôle d'observer ce qui se passe sur la surface sur laquelle le projecteur projette l'interface. Un système similaire dans l'esprit a également été proposé par Pinhanez où le moteur actionne un miroir plutôt que le couple vidéo-projecteur/caméra réduisant le champ d'action du dispositif [Pinhanez 2001].



**Figure 2**  
Le SCP ; le couplage d'une caméra et d'un vidéo projecteur actionnés par un moteur.

Durant une première phase de calibration, le système, aidé d'autres caméras, détecte les surfaces planes de la pièce. D'autres surfaces peuvent également être ajoutées. En traits pleins et pointillés sur la Figure 3, les surfaces exploitables.

Le système se compose également d'une surface portable qui peut



**Figure 3**  
Les surfaces planes de l'environnement du SCP.

être reconnue par le SCP et sur laquelle une interface peut être projetée. Il s'agit du PDS (Portable Display System). Une surface claire, avec des bords clairement délimités (Figure 4 à gauche) suffit au système de vision pour la reconnaître, déterminer sa position par rapport au SCP, et effectuer des corrections d'image pour que l'utilisateur perçoive la projection correctement (si le PDS n'est pas perpendiculaire à l'axe de projection). Une fois cette étape franchie, l'utilisateur peut interagir au doigt sur l'interface graphique (Figure 4 à droite).



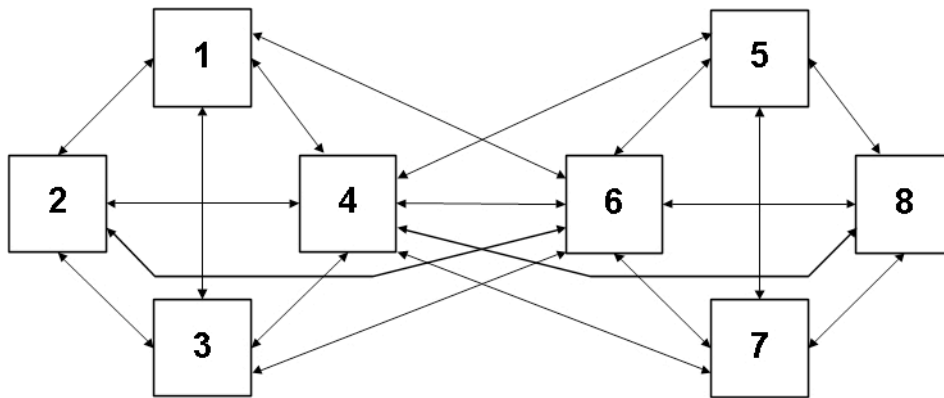
**Figure 4**

**Le Portable Display.**

Une surface portable à gauche constituée d'un morceau de carton blanc avec les bords noirs. A droite, la même surface sur laquelle est projetée une interface avec laquelle l'utilisateur interagit.

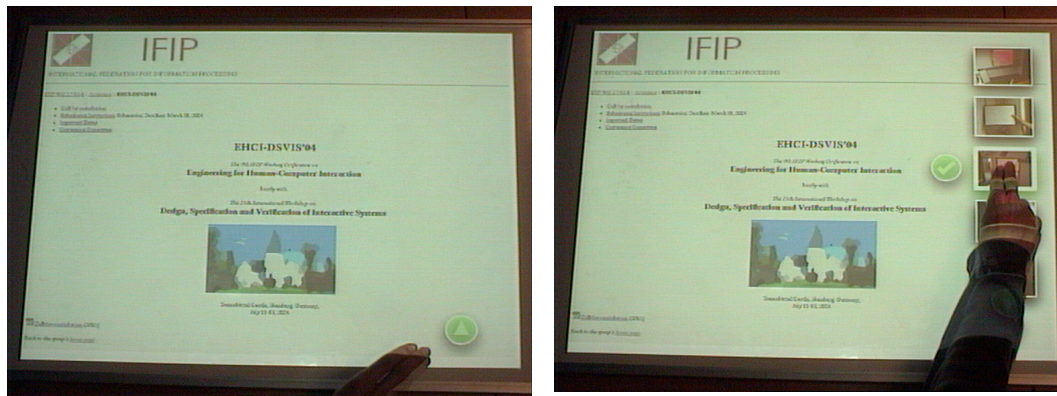
Comme nous l'avons dit, ce système permet la migration d'interfaces graphiques entre différentes surfaces. Pour ce faire, trois techniques d'interaction sont proposées : l'interaction à base de menus, l'interaction à distance et l'utilisation du PDS. Nous reprendrons ces trois techniques dans le chapitre suivant sur les techniques d'interaction. Nous les considérons ici uniquement sous l'angle des couplages physique-numérique (surface-interface graphique) mis en œuvre et reprenons une partie de l'analyse réalisée par Borkowski [Borkowski 2006].

Borkowski propose une adaptation de l'automate en réalisant une partie de la fermeture transitive de l'automate. La Figure 5 montre cette modification. Elle est le résultat de l'union des sous automates 1,2,3,4,6 et 4,5,6,7,8 sur lesquels on réalise la fermeture transitive. Nous nommons cette modification « demi-fermeture transitive ». L'état 4 devient pour le sous-automate 1,2,3,4 une passerelle directe vers les états 5,6,7,8. L'état 6 devient pour le sous-automate 5,6,7,8 une passerelle directe vers les états 1,2,3,4. Nous conservons cette hypothèse pour la présentation des travaux de Borkowski et la discutons ensuite.



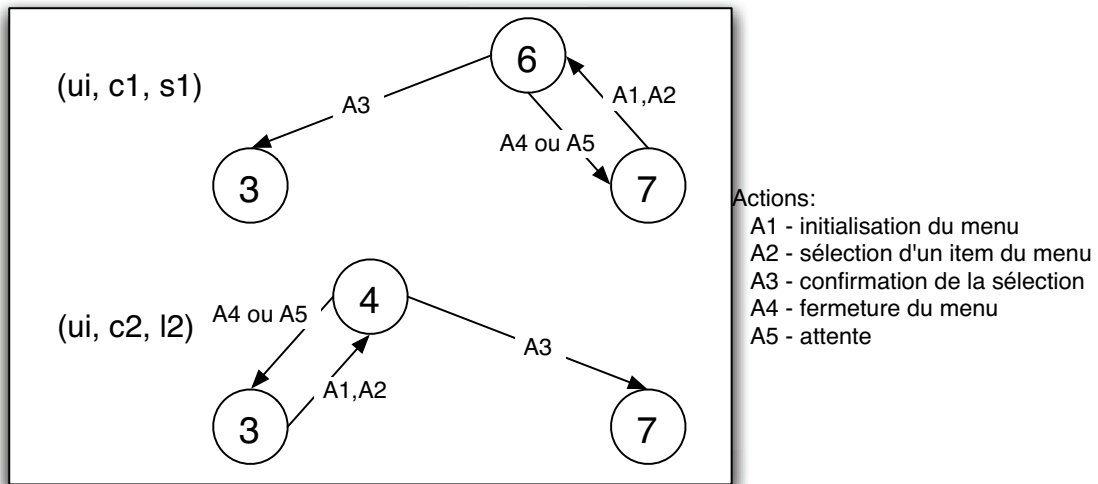
**Figure 5**  
**Automate résultant de la « demi-fermeture transitive ».**  
L'état 4 est la passerelle vers les états 5,6,7,8 et l'état 6 la passerelle vers les états 1,2,3,4.

La technique à base de menus s'inspire des techniques utilisées dans les systèmes d'exploitation conventionnels. Sur la Figure 6 à gauche, l'utilisateur clique sur le bouton vert, ce qui a pour effet d'ouvrir une liste (à droite) des surfaces disponibles. En cliquant sur la représentation numérique de niveau 2 d'une surface, l'utilisateur exprime sa volonté de découpler l'interface de la surface courante pour la coupler avec la nouvelle surface qu'il vient de sélectionner. Pour éviter une sélection accidentelle, l'utilisateur doit ensuite confirmer à l'aide d'un bouton sa volonté de migration.



**Figure 6**  
**Interaction avec une liste de surfaces.**

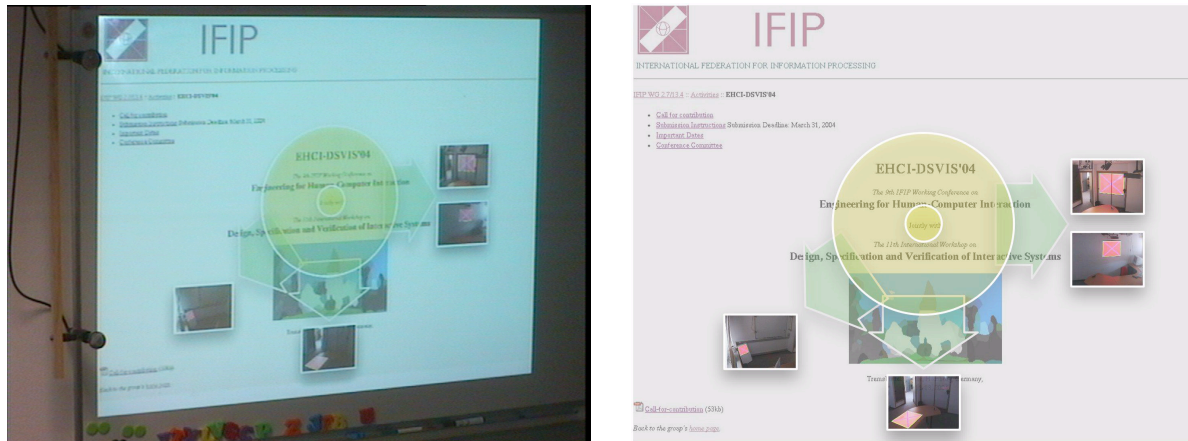
Deux couplages nous intéressent ici :  $(ui, c_1, s_1)$  et  $(ui, c_2, s_2)$  dans lesquels  $ui$  est l'interface graphique,  $s_1$  la surface originale, et  $s_2$  la surface destination. Nous modélisons, à l'aide de deux automates (Figure 7), l'évolution des deux couplages au cours de la technique d'interaction.



**Figure 7**  
Automates des couplages pour la technique à l'aide de menus d'après [Borkowski 2006]

Au départ, l'interface est couplée avec la surface  $s_1$  et, puisque  $ui$  ne peut pas être couplée à une autre surface, le prédicat verrouillé est vrai pour le découplage. Par contre,  $c_1$  est découplable. Pour le deuxième couplage,  $ui$  est découplée de  $s_2$ , verrouillée puisque actuellement couplée avec une autre surface, mais reste couplable. L'initialisation du menu et la sélection d'une surface de destination provoquent le déverrouillage pour le découplage de  $c_1$  et le déverrouillage pour le couplage de  $c_2$  provoquant le passage de  $c_1$  dans l'état 6 et  $c_2$  dans l'état 3. Si le menu est fermé ou qu'aucune action n'intervient dans un certain laps de temps,  $c_1$  retourne à l'état initial 7 et  $c_2$  à 3. Cependant, si la confirmation a lieu,  $ui$  est découplée de  $s_1$  et couplée à  $s_2$ . Parce que le menu n'apparaît plus après la migration,  $ui$  est verrouillée pour le découplage dans  $c_2$  et le couplage dans  $c_1$ . Par rapport à l'automate que nous avons présenté, le passage par l'état 4 pour  $c_1$  et 6 pour  $c_2$  ne sont pas considérés. Nous discutons de ce point après la présentation des deux autres techniques.

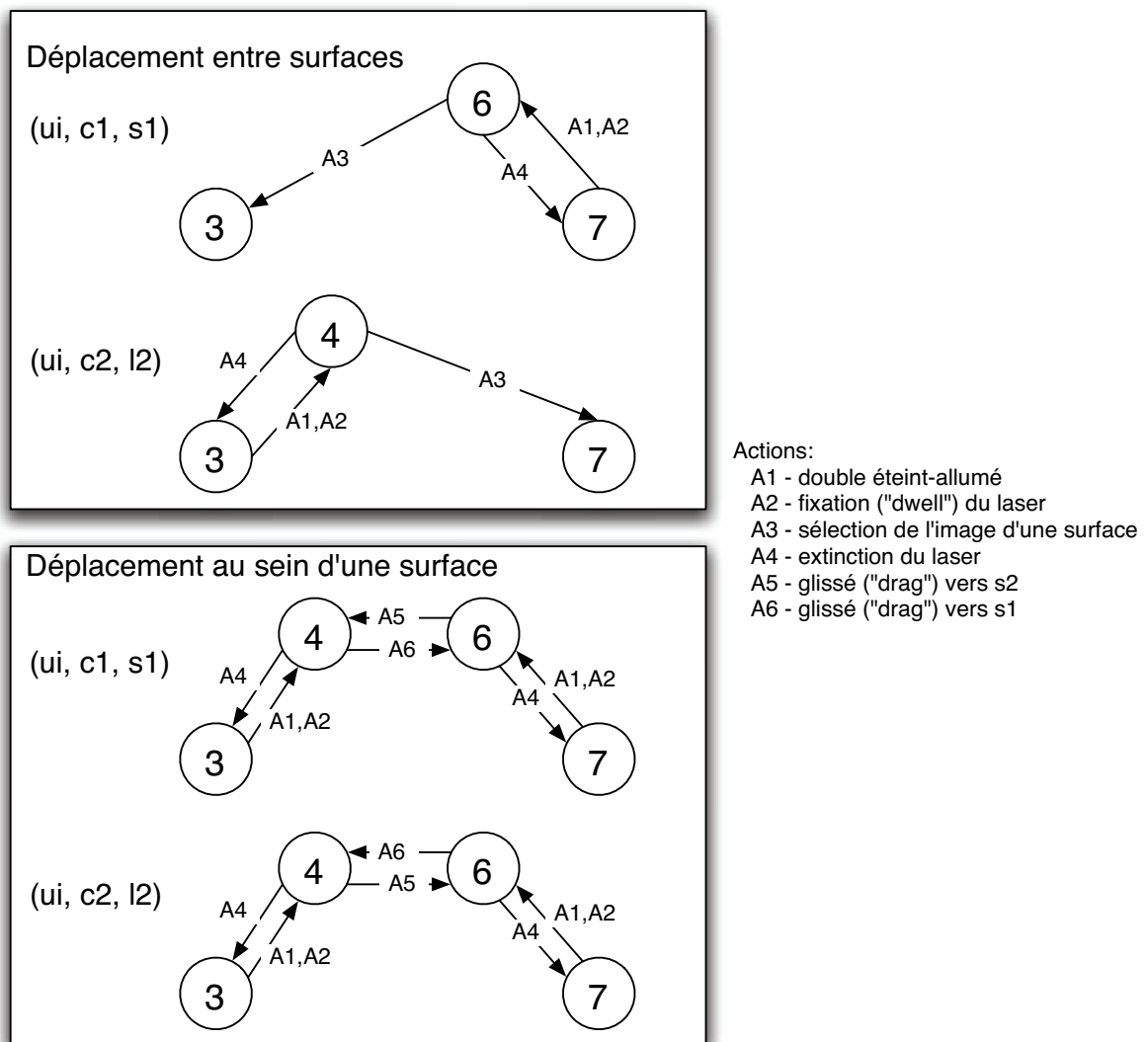
**L'interaction à distance** est la deuxième technique proposée. Elle s'appuie sur l'utilisation d'un laser qui agit sur l'IHM présentée en Figure 8. En réalisant l'équivalent d'un double-clic avec le laser, l'utilisateur fait apparaître une interface de contrôle (les disques jaunes, les flèches et les images des surfaces). Cette interface est transparente et se superpose à l'IHM métier. Le plus petit disque est centré sur la position du point du laser. En déplaçant le laser vers une flèche ou sur une image représentant une surface, l'utilisateur exprime sa volonté de faire migrer l'interface sur la surface ainsi désignée.



**Figure 8**  
**Interface graphique pour l'interaction au laser.**

Cette technique est modélisée par les automates de la Figure 9. Les états initiaux sont identiques à ceux de la technique précédente. L'interface graphique  $ui$  est couplée et verrouillée pour la surface  $s_1$  et non couplée et verrouillée pour  $s_2$ . Lorsque le laser est éteint et rallumé deux fois, l'IHM de contrôle est activée et la fixation du laser dans un élément de cette IHM provoque le déverrouillage des deux couplages. L'extinction du laser provoque le retour dans les états initiaux alors que la sélection de l'image d'une surface provoque le découplage d' $ui$  d'avec  $s_1$  et son couplage avec  $s_2$ . Dans le même temps, comme l'interface de contrôle n'apparaît plus après la migration, les deux couplages sont automatiquement re-verrouillés. Borkowski propose également à l'aide de la même interface de contrôle le déplacement de l'interface graphique  $ui$  au sein de  $s_1$ . En effet, pour Borkowski, cette opération relève d'un découplage et d'un recouplage avec  $s_1$ .

**La technique utilisant le PDS** est la troisième et dernière technique proposée. L'idée est d'utiliser le PDS comme surface intermédiaire pour le transport d'IHM entre surfaces. Ainsi, la Figure 10 montre les trois automates mis en jeu pour cette technique. Pour fonctionner, le PDS a besoin que le système de suivi par vision par ordinateur soit démarré. C'est pourquoi, dans l'état initial,  $s_2$  est non couplable (non suivi par le système) et verrouillé. L'initialisation du menu et la sélection de l'image correspondant au PDS sur  $s_1$  provoquent, comme pour les autres techniques, le déverrouillage. Nous rappelons que la caméra couplée au système de projection observe cette zone de projection. Aussi, afin de devenir couplable, le PDS doit être amené sur  $s_1$ . A partir de ce moment-là, si la migration d' $ui$  vers le PDS est confirmée, le découplage d'avec  $s_1$  est réalisé simultanément au couplage avec le PDS. Le PDS est ensuite



**Figure 9**  
Automates des couplages pour la technique d'interaction à distance [Borkowski 2006]

transporté jusqu'à  $s_2$ . Lorsque l'utilisateur appuie sur le bouton « freeze » projeté sur le PDS,  $ui$  est alors découplée du PDS et couplée avec  $s_2$  car le système de suivi du PDS est stoppé. Borkowski propose une variante dans laquelle le système de suivi est démarré en permanence ce qui simplifie un peu les automates.

Après avoir présenté ces trois exemples, nous les analysons dans la section qui suit.



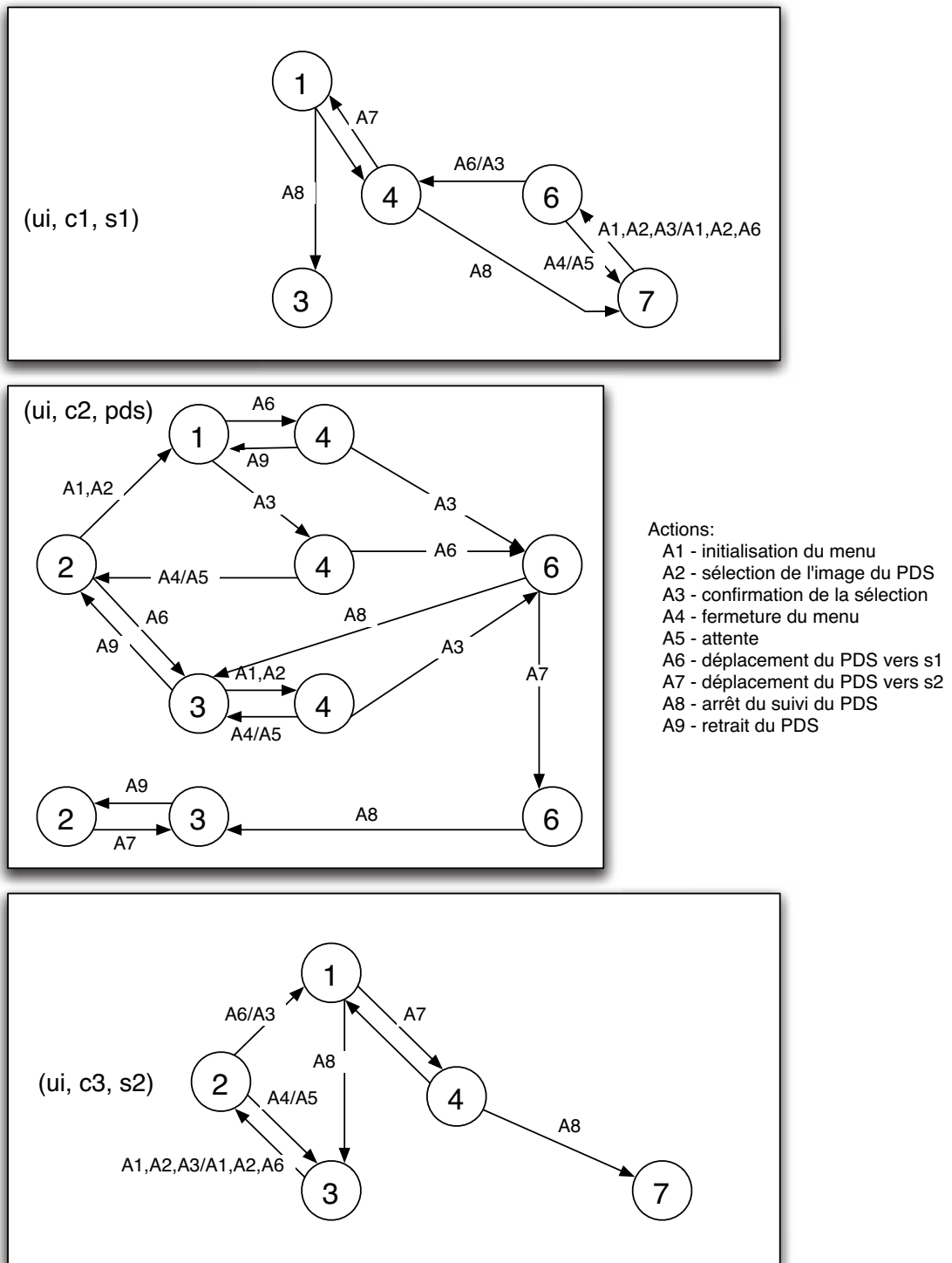


Figure 10  
 Automates des couplages pour la technique avec PDS d'après [Borkowski 2006].

---

## 1.6. ANALYSE

### Nombre d'états couverts

Le nombre d'états couverts par un couplage est comparable à la notion de trajectoire d'interaction [Grahm et Cockton 1996]. En l'occurrence, c'est le nombre de transitions à franchir entre l'état initial et l'état final qui donne une indication sur la « longueur » de la technique d'interaction. Nous associons également le nombre d'actions à effectuer pour franchir ces transitions. A titre d'exemple, la première technique à base de menus est très rapide (deux transitions pour trois actions au maximum) contrairement à l'utilisation du PDS (sept états et sept transitions). Les techniques rapides peuvent sembler intéressantes. Toutefois, les techniques couvrant un grand nombre d'états ont un pouvoir d'expression supérieur puisque l'utilisateur dispose de plusieurs états intermédiaires pour gérer au mieux le couplage (découplage). Notons tout de même qu'une profusion d'états, de transitions et d'actions peut également être source de confusion pour l'utilisateur qui doit mémoriser l'ensemble des actions à effectuer pour coupler/découpler.

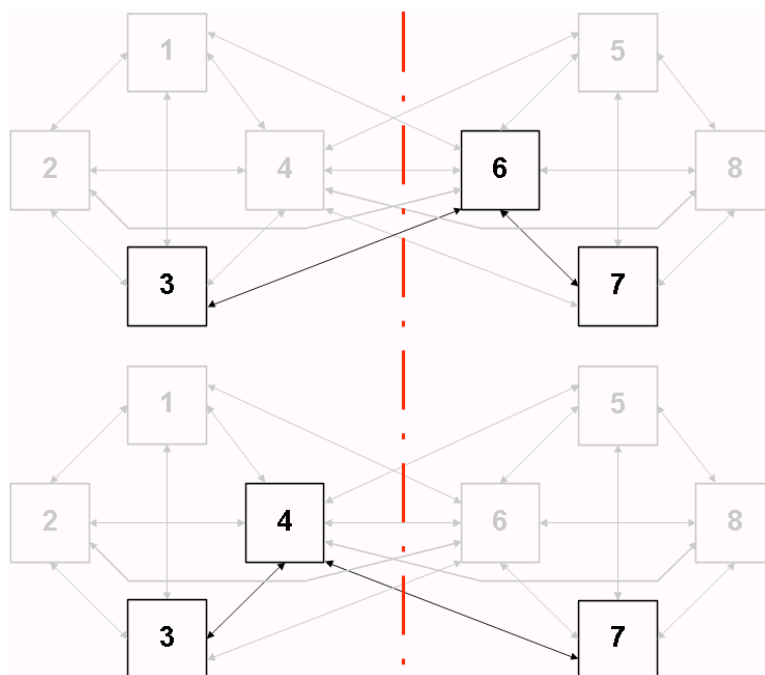
### Actions impactant plusieurs automates

Au sein d'une technique d'interaction, certaines actions ne concernent qu'un automate alors que d'autres en impactent deux. Dans la première technique, l'initialisation du menu (A1) et la sélection d'un item (A2) provoquent le passage de l'état 7 à l'état 6 pour le couplage  $c_1$  et de l'état 3 à l'état 4 pour le couplage  $c_2$ . A l'inverse, dans la troisième technique l'action A9 (« retrait du PDS ») n'a une incidence que sur un seul automate.

Nous constatons que les actions qui impactent deux cycles simultanément, ont la particularité d'être liées à des transitions. Dans l'exemple de la première technique que nous citons, les deux actions (A1 et A2) sont liées pour les deux cycles à la transition de déverrouillage. Dans ce même exemple, l'action qui suit, A3, provoque deux transitions puisqu'elle correspond au découplage et verrouillage pour  $c_1$  et au couplage et verrouillage pour  $c_2$ . Cette constatation nous amène à nous interroger sur les ressemblances entre les automates.

### Automates symétriques

La technique de déplacement entre surfaces présente une caractéristique intéressante. Sur le plan graphique, la progression entre états suit une symétrie par rapport à un axe vertical sur notre représentation de l'automate (Figure 11). Cette symétrie révèle la correspondance entre les états 1 et 5, 2 et 8, 3 et 7, 4 et 6.



**Figure 11**  
Symétrie graphique des automates de la technique de déplacement entre surfaces.

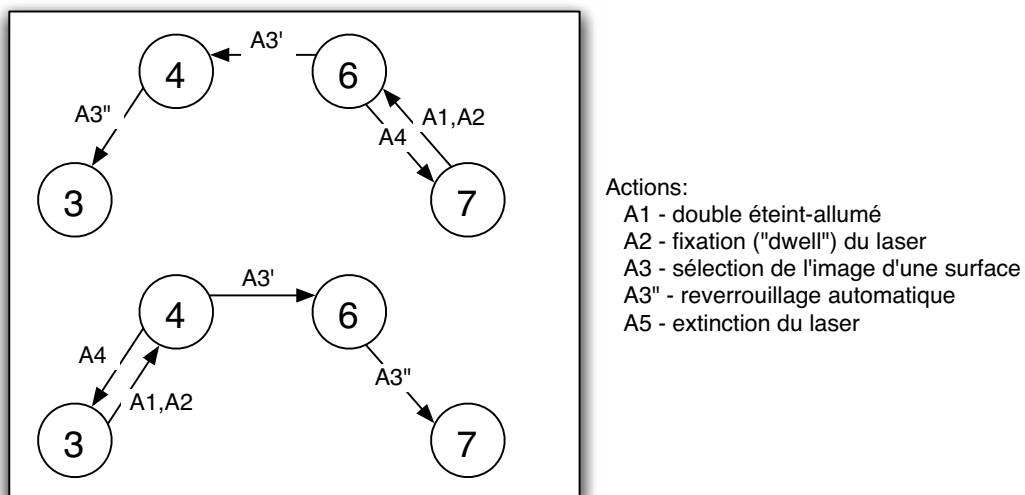
Pour étudier plus précisément cette symétrie, nous reprenons l'exemple de la technique de déplacement entre surfaces mais laissons de côté la demi-fermeture transitive proposée par Borkowski (Figure 5) et reprenons l'automate que nous proposons en Figure 1. La technique de déplacement entre surfaces est alors décrite comme présentée en Figure 12. Cette nouvelle description implique la décomposition de l'action A3 en deux actions A3' et A3'' puisque deux transitions sont maintenant franchies. A3' correspond à l'action de l'utilisateur qui sélectionne l'image d'une surface. Lorsque nous avons décrit cette technique nous indiquions « Dans le même temps [que l'action A3'], comme l'interface de contrôle n'apparaît plus après la migration, les deux couplages sont automatiquement re-verrouillés ». Nous traduisons cette phrase par l'action A3''. Puisque l'utilisateur n'a effectué qu'une seule action (A3), A3'' revient au seul autre acteur pouvant intervenir à l'exécution : le système.

Dans ces conditions, nous introduisons les définitions suivantes :

- Le couple des états a et b noté  $\langle a, b \rangle$  est symétrique si et seulement si :
  - Si couplé est vrai pour a alors non-couplé est vrai pour b et réciproquement,
  - a et b ont le même état de verrouillage : si a est verrouillé alors b l'est aussi ; si a est non verrouillé alors b l'est aussi.

- Si a est couplable alors b est découplable ; si a est non couplable alors b est non découplable.
- Une transition est franchie lors de la réalisation d'une ou plusieurs actions. A chaque action est associé l'acteur en charge de sa réalisation.
- Deux automates sont dits symétriques si et seulement si
  - la structure de ces deux automates est identique : nombre d'états, transitions entre ces états et actions/acteurs déclenchant ces transitions,
  - la permutation entre les numéros des états respecte la symétrie deux à deux de ces deux états.

La Figure 13 montre la structure commune des deux nouveaux automates de la Figure 12 et la permutation des états respectant la symétrie.



**Figure 12**  
Nouveaux automates pour la technique de déplacement entre surfaces.

Nous pourrions envisager d'appliquer la définition de la symétrie à l'automate proposé par Borkowski (Figure 5). Cependant nous préférons conserver l'automate original qui présente l'avantage de n'utiliser que des transitions élémentaires : couplage/découplage, verrouillage/déverrouillage, couplabilité/découplabilité. L'automate de Borkowski fait apparaître de nouvelles transitions résultant d'une combinaison de la transition de couplage/découplage avec les autres transitions. Par exemple la transition 4-7 correspond à un couplage et un verrouillage ; la transition 6-3 à un découplage et un verrouillage. Cette composition de transitions

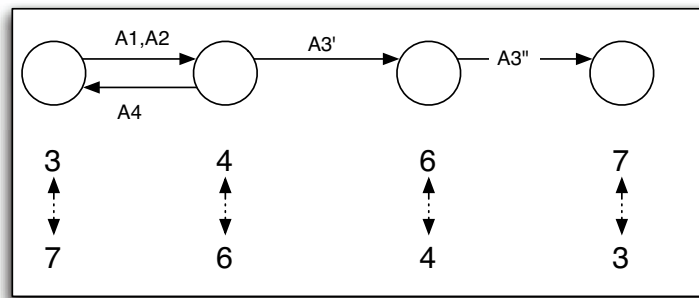


Figure 13

**Structure commune à deux automates.**

En haut la structure commune des deux automates. En dessous les numéros des états pour les deux automates avec leur permutation respectant la symétrie : 6 et 4, 7 et 3.

pourrait amener à des erreurs dans l'évaluation de la symétrie. Par exemple, si les actions A1 et A2 provoquent la transition de l'état 6 à l'état 3. En réalité, dans l'automate original, deux transitions sont franchies : découplage et verrouillage. Aucune indication ne nous permet d'associer l'une des deux actions à l'une des deux transitions. Aussi, si les actions A1 et A2 impactent un deuxième automate pour la transition 4-7, il semblerait raisonnable de considérer les deux automates comme symétriques. Cependant, il est possible qu'en utilisant notre automate d'origine, on constate que A1 provoque le découplage dans le premier automate et le verrouillage dans le second. Dans ce cas, les automates ne sont pas symétriques, ce qui entre en contradiction avec la symétrie constatée avec l'automate « demi-fermé ». Aussi, comme nous le proposons en début de paragraphe, nous conservons l'automate original qui ne manipule que des transitions élémentaires ; les transitions sont décorées des actions qui permettent de les franchir, ces actions étant elles-mêmes décorées de l'acteur en charge de sa réalisation.

De façon générale, l'automate que nous proposons permet d'étudier les solutions existantes, mais peut s'utiliser comme aide à la conception. A des fins d'analyse, l'automate peut servir de métrique de la complexité d'un couplage, mais aussi représenter le pouvoir d'expression des systèmes étudiés : possibilité de verrouiller ou non, couplable ou non.

Du point de vue de l'utilisateur, il convient de s'interroger si cette symétrie peut rendre la technique d'interaction plus facilement compréhensible car les deux couplages ont un comportement similaire. Nous associons aux actions, l'acteur qui les a en charge. Il est intéressant d'étudier comment une action peut migrer d'un acteur à l'autre. Enfin, on constate que certaines transitions possèdent une transition inverse alors que d'autres non. De façon globale, un automate est-il réversible ? Sur ces

interrogations, nous poursuivons en appliquant des propriétés ergonomiques du domaine de l'Interaction Homme-Machine au problème du couplage.

## *2. Propriétés ergonomiques d'un couplage*

---

Nous analysons ci-dessous l'utilisabilité du couplage selon les deux grandes dimensions, souplesse et robustesse de l'interaction introduites dans [Grahm et Cockton 1996]. D'autres cadres d'analyse comme les règles de Scapin et Bastien [Scapin et Bastien 1993] ou de Bellotti [Bellotti et al. 2002] sont applicables.

---

### **2.1. SOUPLESSE DE L'INTERACTION**

Atteignabilité : « Capacité du système à permettre à l'utilisateur de naviguer dans l'ensemble des états observables du système ». Appliquée à notre problème, l'atteignabilité signifie que l'utilisateur doit pouvoir atteindre les états (couplé, verrouillé, couplable) et leur inverse (découplé, déverrouillé, découplable). Il convient donc que l'utilisateur dispose de techniques d'interaction pour atteindre chacun de ces états. Par exemple, pour le verrouillage, Hinckley [Hinckley 2003] propose de masquer avec la main la tablette dont on ne veut pas partager le contenu. Nous constatons que tous les prototypes de l'état de l'art ne proposent pas le verrouillage ou la couplabilité. Dans les exemples de Borkowski que nous avons détaillés précédemment, certains états ne sont pas atteignables. Cependant, la définition est restreinte aux états observables, ce qui réduit le nombre d'états dans lequel l'utilisateur peut naviguer. A ces états, on peut ajouter ceux qui sont accessibles par des transitions réalisées par le système. Dans l'exemple de la Figure 12, l'état 3 est atteint par une transition de verrouillage initiée par le système.

Non-préemption : « Le prochain but souhaité par l'utilisateur est directement atteignable ». Pour notre problème, on retiendra que les états souhaités par l'utilisateur doivent être accessibles à tout instant et ceci par des trajectoires d'interaction optimisées. C'est le cas des ConnecTables [Tandler et al. 2001] qui peuvent toujours être couplées ou découplées en un seul geste. Encore une fois, l'étude des transitions permises par le système peut permettre la réalisation de cette propriété. Dans l'exemple de la Figure 12, l'état 7 est « directement » atteignable par l'utilisateur. Cependant, la mise en place de transitions réalisées automatiquement par le système réduit de fait l'ensemble des états atteignables ou augmente la trajectoire vers certains états. Dans l'exemple de la Figure 12, la transition

réalisée par le système de l'état 6 vers l'état 7 allonge la trajectoire pour aller de l'état 4 à l'état 5. L'action A3" provoque la transition de l'état 4 vers l'état 6 qui est immédiatement suivie par une transition vers l'état 7. L'utilisateur devrait alors franchir deux transitions pour accéder à l'état 5.

Interaction multifilaire : « Capacité du système à permettre la réalisation de front de plusieurs tâches ». Dans le contexte de notre étude, cette propriété traduit la capacité du système, comme la Table Magique [Bérard 2003], à traiter plusieurs couplages simultanément ; ou bien, comme cela peut se produire en interaction proximale, plusieurs utilisateurs demandent en même temps le couplage d'entités avec une même entité cible. Soit le système gère l'accès concurrent et l'exprime, soit les conflits sont résolus de manière sociale.

Migrabilité de tâche : « Capacité de délégation dynamique de tâches entre le système et l'utilisateur ou entre utilisateurs. C'est un changement dynamique de l'acteur(s) responsable(s) de l'accomplissement de la tâche ». Au regard du couplage, cette propriété s'interprète comme la possibilité de déléguer une partie du couplage au système. Par exemple, les claviers et souris d'une même plate-forme sont automatiquement couplés dès le lancement du système. La migration de la réalisation d'une transition dans l'automate peut, comme nous l'avons dit concernant la non-préemption, réduire dans certains cas la trajectoire entre deux états.

Adaptabilité : « Personnalisation du système sur intervention explicite de l'utilisateur ». Aucun exemple de l'état de l'art sur le couplage de ressources d'interaction, n'illustre cette propriété. Couplage/découplage, verrouillage/déverrouillage, etc., sont conventionnels et immuables. La propriété de migrabilité que nous venons d'étudier est une piste pour l'adaptabilité. En effet, la migration de la réalisation d'une transition d'un acteur à l'autre est une forme d'adaptabilité.

---

## **2.2. ROBUSTESSE DE L'INTERACTION**

Observabilité : « Capacité du système à rendre perceptible l'état pertinent du système ». Cette propriété, appliquée à l'automate de couplage, impose de rendre observable l'ensemble des états de l'automate. En particulier, nous voyons dans l'analyse de l'état de l'art que les états « couplable » et « non couplable » ne sont généralement pas observables. Par exemple dans la Table Magique [Bérard 2003], le couplage d'un jeton avec un patch n'est pas observable. L'observabilité peut d'ailleurs être déclinée pour d'autres propriétés. Par exemple, l'observabilité appliquée au prédicat « couplable » peut renforcer la propriété de prévisibilité. Borkowski [Borkowski 2006] fait remarquer que l'observabilité

s'applique également aux transitions ce qui permet à l'utilisateur d'anticiper au mieux les résultats d'une action.

Honnêteté : « Capacité du système à rendre observable l'état du système sous une forme conforme à cet état et qui engendre une interprétation correcte de la part de l'utilisateur ». Autrement dit, l'état du couplage doit non seulement être observable à tout instant, mais ce rendu doit être compris correctement par l'utilisateur. Dans I-AM, dont on trouvera en annexe une description technique [Lachenal 2004], l'utilisateur peut disposer de plusieurs souris et claviers couplables et découplables à volonté. Mais l'évaluation de l'état actuel des couplages peut poser problème : l'utilisateur est amené à se demander quel clavier est couplé avec quelle souris. Leur proximité spatiale sur la surface de travail ne suffit pas à lever l'ambiguïté.

Curabilité : « Capacité pour l'utilisateur de corriger une situation non désirée ». Dans l'automate du couplage, cette propriété se traduit par la possibilité de découpler et de déverrouiller à tout instant. De façon globale, la propriété de curabilité imposerait une réversibilité totale de toutes les transitions des automates en particulier celles réalisées par le système.

La curabilité impose également une vigilance toute particulière sur les conséquences du découplage. Par exemple, considérons un écran  $e_1$  couplé à une souris et à un écran  $e_2$  en sorte que le curseur de la souris couvre l'ensemble des deux écrans. Si l'on découple  $e_1$  et  $e_2$ , la souris reste-t-elle couplée à  $e_1$  ou à  $e_2$  ?

Prévisibilité : « Capacité pour l'utilisateur de prévoir, pour un état donné, l'effet d'une action ». Ici, l'utilisateur doit anticiper les conséquences d'un couplage ou d'un découplage. Chez Hinckley, selon le geste, le couplage de deux tablettes ne fournit pas le même résultat et ce résultat, pour un utilisateur néophyte, n'est pas prévisible. Ou encore, soient deux souris couplées  $s_1$  et  $s_2$ , et deux claviers couplés  $c_1$  et  $c_2$ . Que se passe-t-il si l'utilisateur couple  $s_1$  à  $c_1$  ? Le couplage est-il transitif ? Si tel est le cas,  $s_1$  devient également couplé à  $c_2$  ; de même,  $s_2$  est couplé à  $c_1$  et  $c_2$ . Ou encore, si  $s_1$  est couplé à un écran  $e_1$ , et  $s_2$  est couplée à un écran  $e_2$ , quels sont les effets de bord pour les souris suite au couplage de  $e_1$  et  $e_2$  ? Pour l'utilisateur, ces conséquences sont-elles prévisibles ?

Après nous être focalisés sur les mécanismes liés à un couplage, nous étudions maintenant comment de nouveaux couplages peuvent être automatiquement générés par effet de bord.



### 3. Génération de couplages

Nous proposons deux notations pour la génération de couplages : l'une sous forme de graphes, faciles à lire ; l'autre sous forme algébrique plus concise pour exprimer les conséquences d'un couplage de ressources sur l'ensemble des couplages existants. Nous ne nous occuperons que de la création de nouveaux couplages. Il va de soi qu'il sera nécessaire d'étudier également la suppression de couplages pour couvrir totalement le problème. Nous ne considérerons pas la suppression dans cette thèse mais montrerons où la suppression pourrait intervenir. Nous prenons un petit exemple tiré de la métaphore du bureau qui nous est familière comme base à notre réflexion.

#### 3.1. EXEMPLE DE BASE

En termes de couplage, la métaphore du bureau se traduit de la manière suivante : Un écran  $\text{écran}_1$  est couplé à une souris  $\text{souris}_1$  ( $\text{écran}_1, c_1, \text{souris}_1$ ). Cet écran est également couplé à un clavier  $\text{clavier}_1$  ( $\text{écran}_1, c_2, \text{clavier}_1$ ). Enfin, le clavier et la souris sont couplés ( $\text{clavier}_1, c_3, \text{souris}_1$ ) de telle manière que le clavier offre la fonction « modifiers » à la souris et que cette dernière fournit la fonction « focus » au clavier.

Étendons maintenant l'exemple : un second écran  $\text{écran}_2$  et une seconde souris  $\text{souris}_2$  sont couplés ( $\text{écran}_2, c_4, \text{souris}_2$ ) pour la fonction de désignation. On nomme configuration initiale l'ensemble des couplages  $\{c_1, c_2, c_3, c_4\}$  (Figure 15). Supposons que l'utilisateur de cette configuration initiale demande le couplage de  $\text{écran}_1$  et  $\text{écran}_2$  en sorte que  $\text{écran}_1$  étende la surface d'affichage d' $\text{écran}_2$  et réciproquement ( $\text{écran}_1, c_5, \text{écran}_2$ ). On dit que  $c_5$  est un couplage initiateur. Ce dernier va donner lieu à l'examen d'un certain nombre de couplages que nous appelons couplages évalués. Parmi les couplages évalués, les couplages retenus sont nommés couplages conséquents.

#### 3.2. NOTATION GRAPHIQUE

La représentation des couplages est réalisée sous forme de graphe non orienté, dont les sommets désignent les ressources d'interaction et les arêtes les couplages entre deux ressources. L'ensemble des sommets est noté  $S$ , l'ensemble des arêtes  $A$ . Les arêtes portent également le nom du couplage qu'elles représentent ( $c_1, c_2, \dots, c_n$ ) sous la forme d'étiquettes. Un couplage initiateur est dénoté par le symbole « \* », les couplages conséquents par « = » et les couplages évalués par « ? ». Les couplages initiateurs, conséquents et évalués sont représentés en trait pointillé pour mettre en évidence leur caractère particulier, voire transitoire.

La Figure 14 correspond à un cas de génération de notre exemple de base que nous traiterons en section 3.4. Les couplages  $c_1, c_2, c_3$  et  $c_4$  constituent la configuration initiale à laquelle s'ajoutent  $c_5$ , le couplage

initiateur (celui demandé par l'utilisateur), les couplages évalués  $c'_6$ ,  $c'_4$  et les deux couplages conséquents  $c_6$  et  $c_7$ .

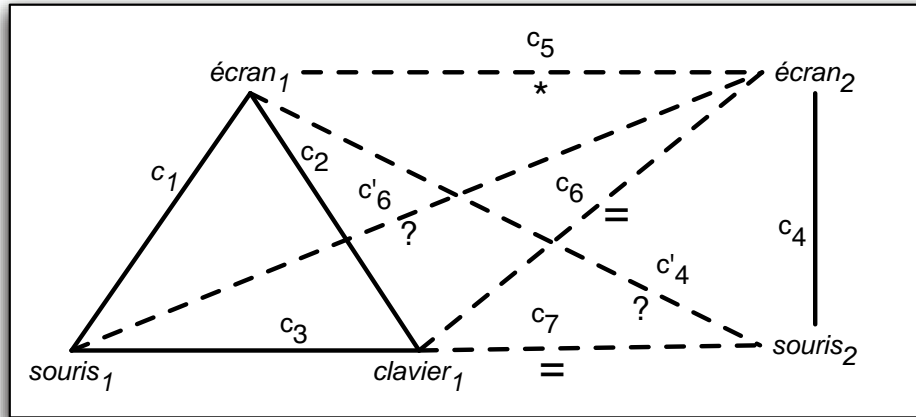


Figure 14  
 Notation graphique appliquée au cas 2 de l'exemple.

Nous présentons l'algorithme de génération puis l'appliquons à notre exemple de base.

### 3.3. ALGORITHME DE GÉNÉRATION

Soient les prédicats :

- $r1(c)$  (resp.  $r2(c)$ ) renvoie la première (resp. la seconde) ressource d'interaction de  $c$ .
- $F(r_1, r_2)$  retourne la fonction obtenue par le couplage de  $r_1$  à  $r_2$ .
- $F(c)$  retourne la fonction du couplage.
- $Compatible(f_1, f_2, f_3)$  renvoie vrai si les fonctions  $f_1$  et  $f_2$  autorisent la génération de  $f_3$ . Cette compatibilité est décidée par conception. Elle est équivalente à la couplabilité de deux ressources introduite dans la section 1.2. Ici, le critère de compatibilité peut dépendre de la suppression de couplages non traitée dans ce manuscrit.

La génération des couplages conséquents suit le schéma suivant :

1. Pour chaque couplage initiateur, on considère les chemins de longueur 2 contenant  $r1(c)$  et  $r2(c)$ .
2. Les chemins obtenus sont de type  $r1(c)-r2(c)-s$  ou  $s-r1(c)-r2(c)$
3. Si  $s-r1(c)-r2(c)$  et  $Compatible(F(c), F(r1(c),s), F(r2(c),s))$  alors ajouter l'arête  $r2(c)-s$

Si  $r1(c)-r2(c)-s$  et  $Compatible(F(c), F(r2(c),s), F(r1(c),s))$  alors  
ajouter l'arête  $r1(c)-s$

Ou

```

Pour tout couplage initiateur c
  Construire l'ensemble Sc des sommets tel que
     $s \in Sc \Leftrightarrow s \in \text{chemin} \wedge \text{longueur}(\text{chemin}) = 2$ 
     $\wedge r1(c) \in \text{chemin} \wedge r2(c) \in \text{chemin}$ 

  pour tout s de Sc
    si  $s \neq r1(c)$  et  $s \neq r2(c)$ 
      si arête(s, r1(c))  $\in A$ 
        si compatible(F(c), F(s, r1(c)), F(s, r2(c)))
           $A = A \cup \text{new arête}(r2(c), s)$ 
        sinon
          si compatible(F(c), F(s, r2(c)), F(s, r1(c)))
             $A = A \cup \text{new arête}(r1(c), s)$ 

```

De façon générale, on étudie chaque arête permettant la fermeture transitive des chemins de longueur 2 contenant  $r1(c)$  et  $r2(c)$ . Afin d'obtenir une génération complète, chaque nouvelle arête créée (nouveau couplage) prend le statut de couplage initiateur, donnant lieu à une nouvelle génération.

### 3.4. APPLICATION À L'EXEMPLE

La configuration initiale de notre exemple de base est représentée en Figure 15. D'une part,  $\text{écran}_1$  est couplé avec  $\text{souris}_1$  et  $\text{clavier}_1$  ;  $\text{clavier}_1$  et  $\text{souris}_1$  sont également couplés. D'autre part,  $\text{écran}_2$  et  $\text{souris}_2$  sont couplés.

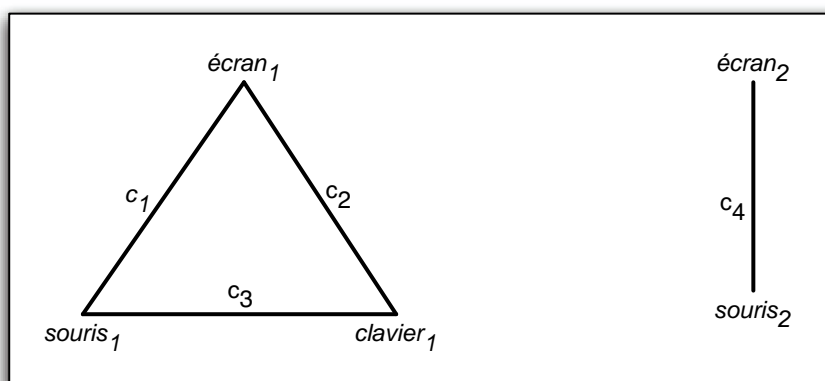


Figure 15  
Condifiguration initiale.

La réalisation de  $c_5$  (couplage d' $\text{écran}_1$  avec  $\text{écran}_2$ ) appelle plusieurs comportements possibles (la liste n'étant pas exhaustive) :

1. Aucun couplage supplémentaire n'est créé,
2. L'utilisateur conserve les possibilités de saisie de texte dans son espace d'affichage,
3. Tout dispositif d'entrée doit pouvoir agir sur n'importe quel écran.

Ici bien d'autres couplages peuvent être générés et nous en verrons un exemple dans la suite. La totalité des couplages que l'on peut générer est représentée par le graphe complet formé sur la base de l'union des deux graphes initiaux. Nous développons les points 2 et 3 qui nécessitent la génération de nouveaux couplages.

### Application à l'exemple, cas 2

Dans le cas 2, le système doit générer les couplages ( $écran_2, c_6, clavier_1$ ) pour que la saisie de texte puisse être effectuée sur l' $écran_2$  mais également ( $clavier_1, c_7, souris_2$ ) pour que l'utilisateur puisse désigner les interacteurs qui doivent recevoir les événements clavier. On appelle cet ensemble de couplages configuration terminale puisque présentant les « couplages terminaux » à la fin de la phase de génération du système. La Figure 16 montre la configuration terminale correspondant au cas 2. En pratique, l'utilisateur peut manipuler de l'information au moyen de  $souris_1$  et de  $clavier_1$  sur  $écran_1$ . Il peut faire migrer une fenêtre d' $clavier_1$  vers  $écran_2$  sans pour autant que  $souris_1$  puisse « sortir » de  $écran_1$ . À partir du moment où un interacteur se retrouve sur l' $écran_2$  (même s'il avait le focus fourni par  $souris_1$ ) alors il est nécessaire de redonner le focus avec  $souris_2$  pour saisir du texte. Le comportement est symétrique au regard d' $écran_2, souris_2$  et  $clavier_1$ .

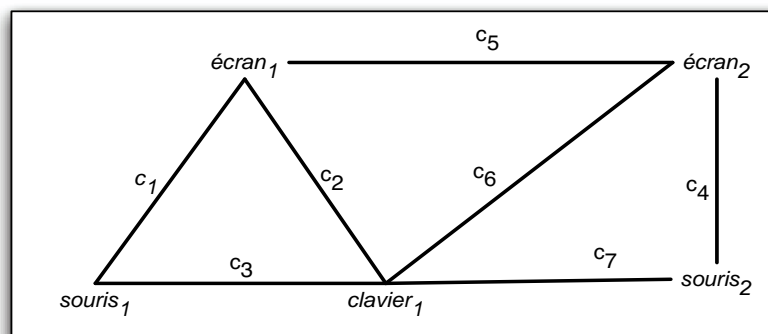
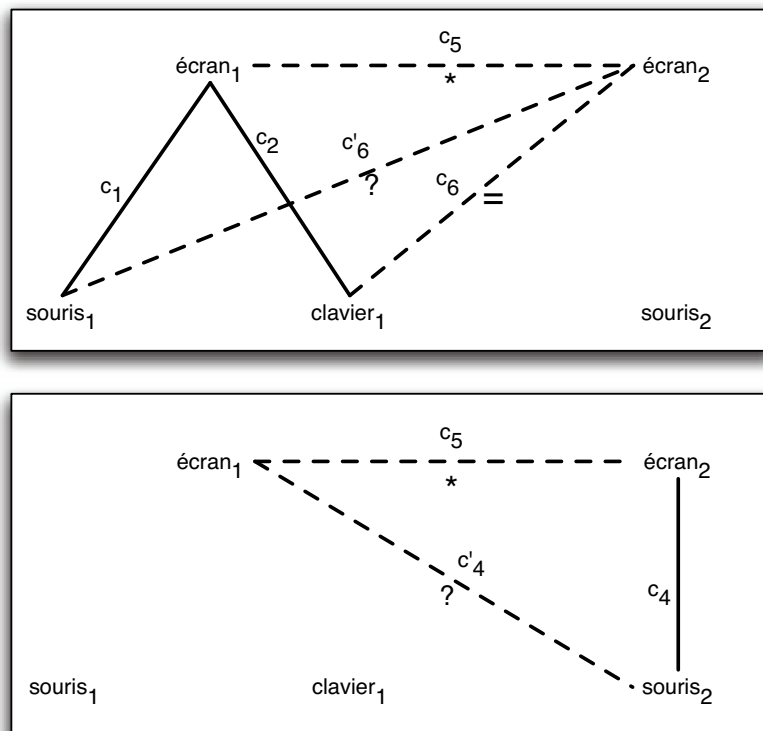


Figure 16  
Configuration terminale du cas 2.

Nous regardons maintenant comment aboutir à la configuration terminale.

La Figure 17 montre la première génération où les chemins pris en compte sont  $c_1-c_5$  et  $c_2-c_5$  et les couplages évalués  $c'_6$  et  $c_6$  ainsi que le

chemin  $c_5-c_4$  pour évaluer  $c'_4$ . De ces évaluations, seul  $c_6$  sera conservé car, par conception, les fonctions de  $c_2$  et  $c_5$  sont compatibles pour la production de la fonction de  $c_6$ . En effet,  $c_5$  étend la surface d'affichage d'*écran<sub>1</sub>* sur *écran<sub>2</sub>*. Aussi, puisque que l'on conserve les capacités de saisie de texte, *clavier<sub>1</sub>* doit être couplé avec tous les écrans étendant *écran<sub>1</sub>*.  $c_6$  constitue le couplage initiateur de la génération suivante.



**Figure 17**  
 Couplages évalués pour les chemins de type s-écran<sub>1</sub>-écran<sub>2</sub> en haut, écran<sub>1</sub>-écran<sub>2</sub>-s en bas.

La Figure 18 montre la deuxième étape de génération pour laquelle  $c_6$  est maintenant considéré comme un couplage initiateur et duquel est déduit le couplage  $c_7$ . Si un clavier est couplé à un écran, mais pas à une souris, il peut certes envoyer des événements mais ne peut saisir du texte n'importe où à l'écran.  $c_6$  et  $c_4$  sont compatibles pour la génération de  $c_7$ . Nous rappelons que la configuration terminale est présentée en Figure 19.

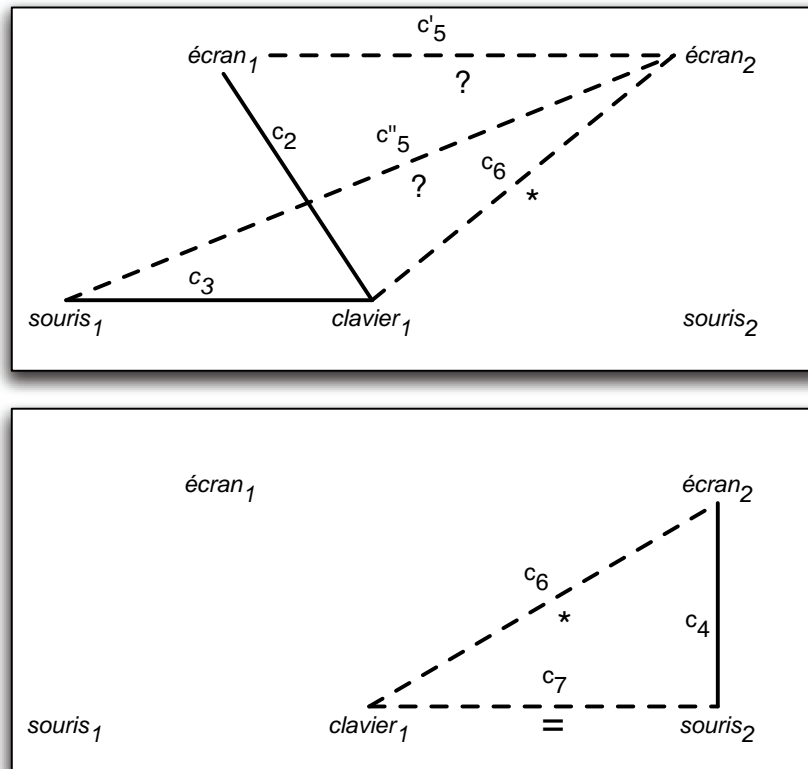


Figure 18  
Couplages évalués pour les chemins de type s-clavier<sub>1</sub>-écran<sub>2</sub> en haut, clavier<sub>1</sub>-écran<sub>2</sub>-s en bas.

### Application à l'exemple, cas 3

La Figure 19 présente la configuration terminale obtenue pour le cas 3. Les couplages (*souris*<sub>1</sub>, *c*<sub>7</sub>, écran<sub>2</sub>) et (*clavier*<sub>1</sub>, *c*<sub>6</sub>, écran<sub>2</sub>) sont générés : puisque l'affichage d'écran<sub>1</sub> est étendu sur écran<sub>2</sub>, il est « légitime » que *souris*<sub>1</sub> et *clavier*<sub>1</sub> accèdent à l'information de la même manière que sur écran<sub>1</sub>. Pour les mêmes raisons, (*souris*<sub>2</sub>, *c*<sub>8</sub>, écran<sub>1</sub>) est généré. Le comportement des interfaces graphiques est moins compliqué que pour le cas 2. En effet, pour chaque dispositif d'entrée, écran<sub>1</sub> et écran<sub>2</sub> sont considérés comme formant un seul écran. De ce fait, chaque dispositif d'entrée a la même fonction quel que soit l'écran.

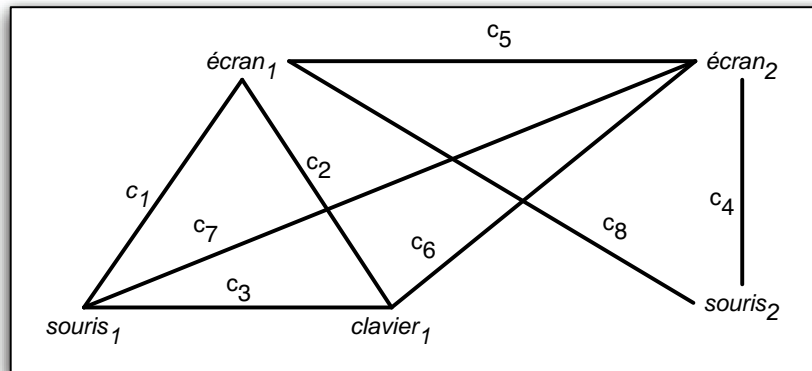


Figure 19  
Configuration terminale pour le cas 3.

La Figure 20 montre les couplages évalués et conséquents pour le cas 3 de notre exemple. Tous les couplages évalués dans la première génération sont acceptés. La deuxième évaluation (avec  $c_6$ ,  $c_7$  et  $c_8$  en couplage initiateur) ne produit rien.

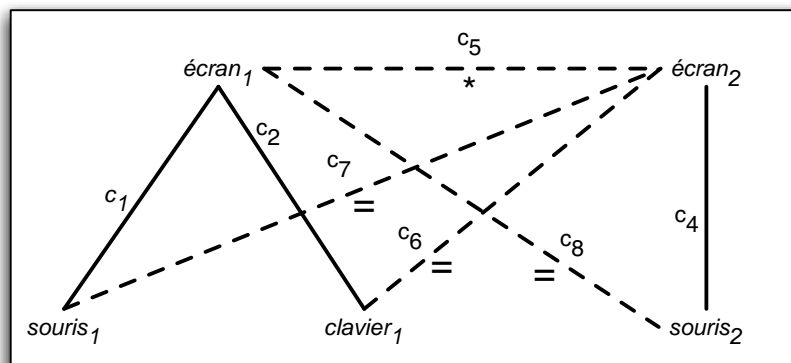


Figure 20  
Couplages évalués pour le couplage initiateur  $c_5$ .

Les solutions des Figure 16 et Figure 19 posent évidemment le problème de gestion de focus multiples sur une application ou de désambiguïsation de focus puisque deux focus peuvent être définis simultanément. Nous ne traitons pas plus avant ce type d'implications.

La notation graphique présente l'inconvénient d'être lourde à mettre en œuvre pour représenter le processus de génération. La notation mathématique qui suit permet d'exprimer le processus de manière concise.

### 3.5. FORMALISATION MATHÉMATIQUE

Nous proposons une notation mathématique supportant le même algorithme de génération que la notation graphique. Nous en faisons une description avant de l'appliquer aux mêmes cas de génération que la notation graphique.

#### Description

La formalisation est régie par deux opérateurs « \* » appelé opérateur de génération et « + », l'opérateur d'union. L'état du système en termes de couplage est noté  $[c_1 + c_2 + \dots + c_n]$  représentant l'union de  $c_1$ ,  $c_2$ , ... et  $c_n$ . L'opérateur « \* » est distributif sur « + » et la priorité relative de ces deux opérateurs est également respectée.

L'ajout d'un couplage se déroule en deux phases. Dans un premier temps, le couplage initiateur est créé et ajouté à la liste des couplages composant le système, puis on évalue si d'autres couplages doivent être déduits.

La première phase est prise en charge par la règle d'ajout :

$$c_p + [c_1 + c_2 + \dots + c_n] \text{ qui produit } [c_p + c_1 + c_2 + \dots + c_n]$$

La deuxième phase est réalisée à l'aide de la règle de génération :

$$c_p * [c_p + c_1 + c_2 + \dots + c_n] = c_p * c_p + c_p * c_1 + c_p * c_2 + \dots + c_p * c_n$$

La résolution de  $c_p * c_p + c_p * c_1 + c_p * c_2 + \dots + c_p * c_n$  nécessite d'évaluer un par un les termes  $c_p * c_i$ . On peut d'ores et déjà noter que, par convention, le terme  $c_p * c_p$  ne peut rien produire. Afin de mieux comprendre l'évaluation d'un terme, on procède à la réécriture de chaque couplage sous la forme simplifiée  $(R, R, F)$  :

$$c_p * c_i = (r_{p1}, r_{p2}, f_p) * (r_{i1}, r_{i2}, f_i)$$

Deux cas sont alors possibles : les deux couplages sont soit transitifs, soit intransitifs :

$$c_p \text{ et } c_i \text{ sont transitifs} \Leftrightarrow (r_{p1}=r_{i1} \vee r_{p1}=r_{i2} \vee r_{p2}=r_{i1} \vee r_{p2}=r_{i2}) \\ \wedge \text{Compatible}(f_p, f_i, f) \wedge c_p \neq c_i$$

$c_p$  et  $c_i$  sont intransitifs sinon.

Si  $c_1$  et  $c_2$  sont transitifs alors :

$$c_p * c_i = (r_{p1}, r_{p2}, f_p) * (r_{i1}, r_{i2}, f_i) = c_{res} \text{ avec} \\ \text{Compatible}(f_p, f_i, f) \\ c_{res} = (r_{p2}, r_{i2}, f) \text{ si } r_{p1}=r_{i1}$$



$$\begin{aligned}
 c_{res} &= (r_{p2}, r_{i1}, f) \text{ si } r_{p1} = r_{i2} \\
 c_{res} &= (r_{p1}, r_{i2}, f) \text{ si } r_{p2} = r_{i1} \\
 c_{res} &= (r_{p1}, r_{i1}, f) \text{ si } r_{p2} = r_{i2} \\
 \text{Transitif}(c_p, c_i) &= \text{vrai}
 \end{aligned}$$

Si  $c_1$  et  $c_2$  sont intransitifs alors :

$$\begin{aligned}
 c_p * c_i &= (r_{p1}, r_{p2}, f_p) * (r_{i1}, r_{i2}, f_i) = \emptyset \\
 \text{Transitif}(c_p, c_i) &= \text{faux}
 \end{aligned}$$

On voit apparaître la correspondance avec la notation sous forme de graphes. Un nouveau couplage  $c_{res}$  n'est créé que si  $c_p$  et  $c_i$  sont transitifs, c'est-à-dire s'ils possèdent une ressource en commun. Cette condition correspond aux chemins de longueurs 2 contenant  $r1(c_p)$  et  $r2(c_p)$ .

Dans le cas de deux couplages transitifs  $c_p$  et  $c_i$ , la fonction du couplage  $c_{res}$  produit se doit d'être compatible avec  $F(c_p)$  et  $F(c_i)$ , tout comme dans la notation graphique. La règle de production telle que l'on vient de la décrire permet de générer de nouveaux couplages. La résolution complète de cette règle suppose que chaque couplage généré donne lieu à la ré-application de la règle de génération. Autrement dit, chaque couplage  $c_{res}$  donne lieu à l'application de la règle d'ajout puis celle de génération. L'algorithme suivant génère la liste complète des couplages à partir de la liste des couplages initiaux et de l'ajout d'un couplage initial. Le langage utilisé est proche de Java avec lequel des libertés d'écriture ont été prises (par exemple sur la mise à jour de la taille des tableaux).

```

Function []Couplage regleGeneration(
    //couplage initial donnant lieu à la génération
    Couplage couplageInitial,
    //ensemble des couplages effectifs
    Couplage []couplagesEffectifs){

    //ajout du couplageInitial dans les couplagesEffectifs
    Couplage []couplages = {couplagesEffectifs, couplageInitial } ;

    //initialisation de la liste des couplage sur lesquels doit
    //être appliquée la règle de génération
    Couplage []aAnalyser = { couplageInitial } ;

    //parcours de la liste des couplages candidats pour la
    //génération
    For (int i=0 ; i< aAnalyser.length ; i++){
        //récupération du ième candidat
        Couplage c1 = aAnalyser[i] ;

        //parcours des couplages effectifs
        For (int j=0 ; j<couplagesEffectifs.length ; j++){
            //récupération du jème couplage effectif
            Couplage c2 = couplagesEffectifs [j] ;

            //test de transitivité des deux couplages
            If (transitifs(c1, c2)){
                //génération du nouveau couplage
                Couplage gen = composition(c1, c2) ;

                //ajout dans la liste des couplages effectifs
                couplagesEffectifs [couplagesEffectifs.length]= gen ;

                //le couplage généré doit subir la génération
                aAnalyser [aAnalyser.length]= gen ;
            }
        }
    }
    Return couplagesEffectifs;
}

```

**Application à  
l'exemple, cas 2**

Configuration initiale :  $[c_1 + c_2 + c_3 + c_4]$

Couplage initiateur :  $c_5$

Règle d'ajout :

$$c_5 + [c_1 + c_2 + c_3 + c_4] = [c_5 + c_1 + c_2 + c_3 + c_4]$$

Règle de génération :

$$\begin{aligned} c_5 * [c_5 + c_1 + c_2 + c_3 + c_4] &= c_5 * c_5 + c_5 * c_1 + c_5 * c_2 + c_5 * c_3 + c_5 * c_4 \\ &= \emptyset + c_5 * c_1 + c_5 * c_2 + c_5 * c_3 + c_5 * c_4 \\ &= \emptyset + \quad + c_6 + \emptyset + \emptyset \end{aligned}$$

Règle d'ajout :

$$c_6 + [c_5 + c_1 + c_2 + c_3 + c_4] = [c_6 + c_5 + c_1 + c_2 + c_3 + c_4]$$

Règle de génération :

$$\begin{aligned} c_6 * [c_6 + c_5 + c_1 + c_2 + c_3 + c_4] \\ &= c_6 * c_6 + c_6 * c_5 + c_6 * c_1 + c_6 * c_2 + c_6 * c_3 + c_6 * c_4 \\ &= \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + c_7 \end{aligned}$$

Règle d'ajout :

$$c_7 + [c_6 + c_5 + c_1 + c_2 + c_3 + c_4] = [c_7 + c_6 + c_5 + c_1 + c_2 + c_3 + c_4]$$

Règle de génération :

$$\begin{aligned} c_7 * [c_7 + c_6 + c_5 + c_1 + c_2 + c_3 + c_4] \\ &= c_7 * c_6 + c_7 * c_5 + c_7 * c_1 + c_7 * c_2 + c_7 * c_3 + c_7 * c_4 \\ &= \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset \end{aligned}$$

Configuration terminale :  $[c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7]$

### Application à l'exemple, cas 3

Configuration initiale :  $[c_1 + c_2 + c_3 + c_4]$

Couplage initiateur :  $c_5$

Règle d'ajout :

$$c_5 + [c_1 + c_2 + c_3 + c_4] = [c_5 + c_1 + c_2 + c_3 + c_4]$$

Règle de génération :

$$\begin{aligned} c_5 * [c_5 + c_1 + c_2 + c_3 + c_4] &= c_5 * c_5 + c_5 * c_1 + c_5 * c_2 + c_5 * c_3 + c_5 * c_4 \\ &= \emptyset + c_6 + c_7 + \emptyset + c_8 \end{aligned}$$

Règle d'ajout :

$$c_6 + c_7 + c_8 [c_5 + c_1 + c_2 + c_3 + c_4] = [c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8]$$

Règle de génération :

$$(c_6 + c_7 + c_8) * [c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8]$$

Pour plus de lisibilité, on scinde le calcul en trois

1.  $c_6 * c_1 + c_6 * c_2 + c_6 * c_3 + c_6 * c_4 + c_6 * c_5 + c_6 * c_6 + c_6 * c_7 + c_6 * c_8$
2.  $+ c_7 * c_1 + c_7 * c_2 + c_7 * c_3 + c_7 * c_4 + c_7 * c_5 + c_7 * c_6 + c_7 * c_7 + c_7 * c_8$
3.  $+ c_8 * c_1 + c_8 * c_2 + c_8 * c_3 + c_8 * c_4 + c_8 * c_5 + c_8 * c_6 + c_8 * c_7 + c_8 * c_8$

1.  $\emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset$
2.  $+ \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset$
3.  $+ \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset$

Configuration terminale :  $[c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8]$

Afin d'alléger l'écriture, la règle de génération peut ne pas comporter tous les couplages du système en partie droite, mais uniquement les couplages transitifs. En ce sens, l'écriture des couplages sous la forme  $(r_1, r_2, f_1)$  facilite l'élimination des couplages non transitifs. Ce premier filtre permet de n'évaluer que les couplages compatibles.

Nous avons vu que la transitivité entre couplages s'appuie sur la notion de compatibilité de fonctions. Cette compatibilité est établie à la conception du système. Jusqu'ici, nous avons raisonné de manière théorique sans nous attarder sur la nature des fonctions de couplage et leur effet sur la compatibilité. Nous pensons que les fonctions génériques et les heuristiques ont un rôle à jouer.

### 3.6. LES FONCTIONS ET LA COMPATIBILITÉ

La génération de nouveaux couplages repose sur la notion de compatibilité qui peut être évaluée au cas par cas. Nous proposons ici quatre pistes pour la déterminer a priori considérant les entités construites, la cardinalité de la relation de couplage, les fonctions génériques et la notion de nouvelle fonction.

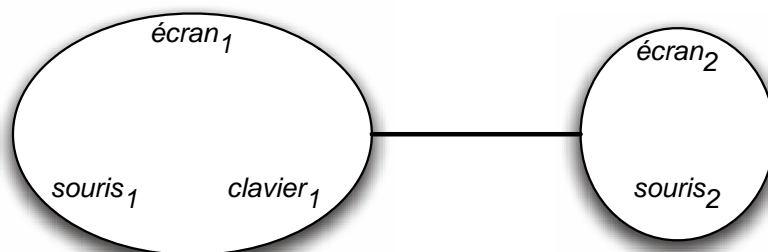
#### Entités construites

L'idée ici est de réduire le nombre de nœuds dans les graphes de génération de couplages comme ceux des Figure 14, Figure 17, Figure 18 et Figure 20. La réduction du nombre des nœuds réduit de fait le nombre de couplages à évaluer. La réduction que nous proposons repose sur la construction d'une nouvelle entité à partir de plusieurs par couplage

comme nous l'introduisons au chapitre 2 avec les entités mixtes construites.

Nous partons du constat qu'une entité remplit une fonction et que par ailleurs nous avons défini le couplage comme permettant également la réalisation d'une fonction. Une entité construite par couplage peut donc être vue de façon unitaire. C'est d'ailleurs le cas des ressources d'interactions que nous connaissons classiquement. Une souris est le résultat d'un certain nombre de couplages physiques et logiciels : deux boutons, une roulette et un tracker optique ont été couplés physiquement pour ne former qu'un seul et unique objet. De façon logicielle, les signaux électriques sont fusionnés ainsi que les interruptions logicielles pour ne générer qu'un seul et unique événement souris.

Dans ces conditions, l'exemple présenté en section 3.1 peut être simplifié en considérant l'*écran<sub>1</sub>*, la *souris<sub>1</sub>* et le *clavier<sub>1</sub>* comme une seule et unique entité. De la même façon, comme le montre la Figure 21, l'*écran<sub>2</sub>* et la *souris<sub>2</sub>* peuvent être vus comme une seule entité. Ainsi, le couplage  $c_5$  entre les deux écrans symboliserait la volonté de coupler la totalité des deux entités construites. De ce fait, on aboutirait à une solution que nous n'avons pas envisagée dans notre exemple et que nous explicitons avec la Figure 22. Ici encore, nous ne pouvons être certains de notre génération car il serait envisageable que la *souris<sub>1</sub>* et la *souris<sub>2</sub>* soient couplées pour ne manipuler qu'un seul curseur.



**Figure 21**  
Écran<sub>1</sub>, souris<sub>1</sub> et clavier<sub>1</sub> vus comme une seule entité ; écran<sub>2</sub> et souris<sub>2</sub> également.

Nous pouvons également considérer que les deux écrans forment une entité construite comme le présente le haut de la Figure 23. La génération pourrait raisonnablement produire le même résultat que celui présenté au bas de la Figure 23.

Encore une fois, nous constatons qu'aucune solution idéale n'émerge. Cependant la construction d'entités est une piste pour la

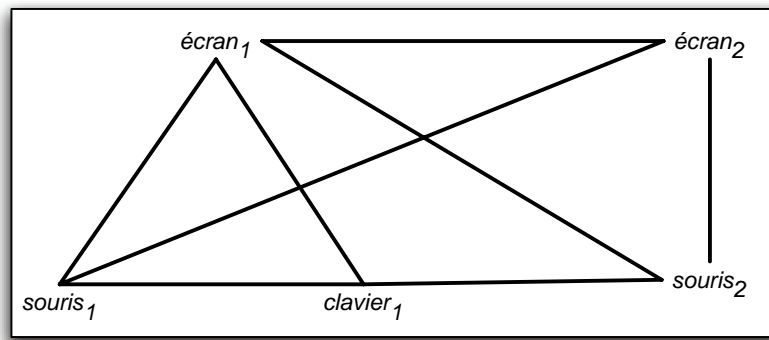


Figure 22  
Génération complète de couplages.

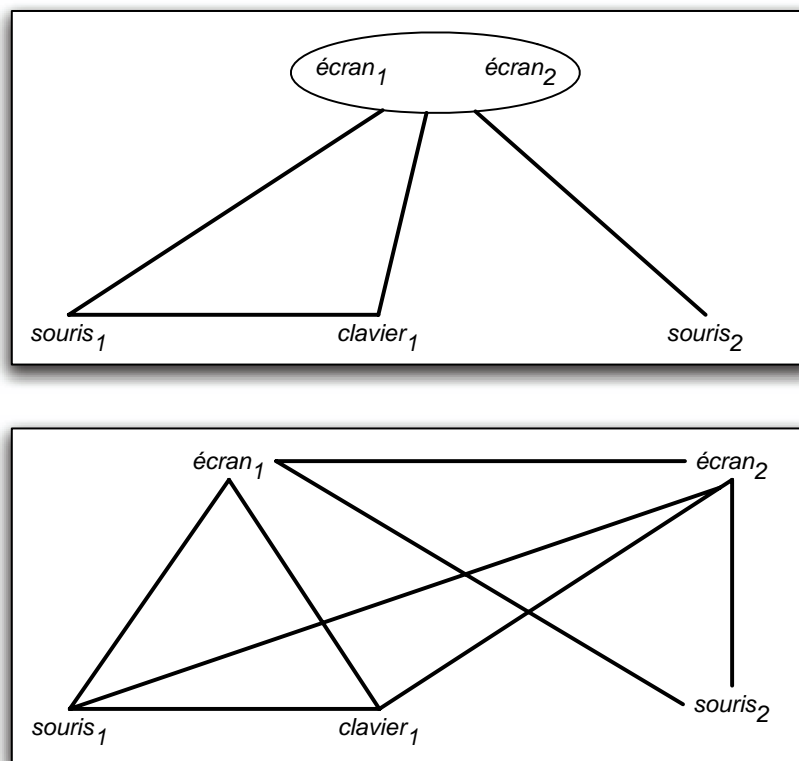


Figure 23  
En haut, l'écran<sub>1</sub> et l'écran<sub>2</sub> sont considérés comme une seule entité. En bas, les couplages terminaux.

génération des couplages. D'autres critères, comme l'usage social des entités, peuvent s'appliquer. Dans sa thèse, Lachenal [Lachenal 2004] propose le caractère privé, semi-privé et public comme critères de base. Pour notre problème, la Figure 24 montre une génération résultant de la définition de l'écran<sub>1</sub>, la souris<sub>2</sub> et le clavier<sub>1</sub> comme étant privés et l'écran<sub>2</sub> et la souris<sub>1</sub> publics ou semi-privés. Les ressources souris<sub>1</sub> et

*clavier<sub>1</sub>* sont autorisées à être couplées avec l'*écran<sub>2</sub>* (public) alors que la *souris<sub>2</sub>* étant publique n'est pas autorisée à être couplée avec l'écran 1.

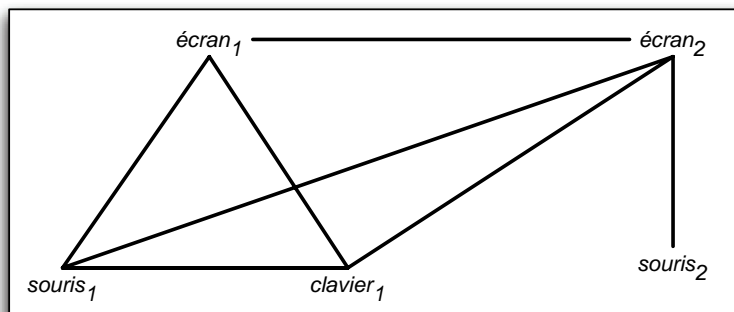


Figure 24  
Configuration terminale en considérant l'écran<sub>1</sub>, souris<sub>1</sub>, clavier<sub>1</sub> comme privés, l'écran<sub>2</sub> et la souris<sub>2</sub> comme publics.

### Cardinalité de la relation de couplage entre entités

Nous reprenons ce critère de notre taxonomie TEC et le proposons ici comme critère permettant de limiter le nombre de nouveaux couplages à évaluer. Pour rappel, la cardinalité de la relation est le nombre maximal d'entités intervenant dans des instances de couplage. Dans notre exemple, en déterminant qu'un clavier ne peut être couplé qu'à une seule souris au maximum, puisque *souris<sub>1</sub>* et *clavier<sub>1</sub>* sont déjà couplés, alors *clavier<sub>1</sub>* et *souris<sub>2</sub>* ne peuvent être couplés. Le maintien du nombre maximal de couplages peut aussi dépendre de la suppression de couplages. Nous venons d'indiquer que *clavier<sub>1</sub>* et *souris<sub>2</sub>* ne peuvent pas être couplés en raison de cette cardinalité. La suppression du couplage entre *clavier<sub>1</sub>* et *souris<sub>1</sub>* pourrait rendre possible la réalisation du couplage *clavier<sub>1</sub>* et *souris<sub>2</sub>*. En nous référant au paragraphe précédent, sur les entités construites, nous proposons également comme solution la construction d'une entité souris<sub>3</sub> résultat du couplage de *souris<sub>1</sub>* et *souris<sub>2</sub>* dans lequel, un seul curseur serait manipulé par deux souris physiques. Dans les systèmes d'exploitation courants, le branchement de deux souris USB sur un même système provoque ce type de couplage. En utilisant souris<sub>3</sub>, on peut alors réaliser son couplage avec *clavier<sub>1</sub>* sans violer la cardinalité.

### Fonctions génériques et heuristiques

Dans la métaphore du bureau, les fonctions copier-coller-undo-redo sont génériques. Par analogie, il devrait exister des fonctions de couplage génériques. Nous pensons à la notion de fusion et son inverse, la fission. Le couplage de deux ressources est une fonction de fusion si les services fournis par chacune de ces ressources prises séparément est conservé et si chacune de ces ressources peut être utilisée indifféremment (équivalence fonctionnelle au sens de CARE [Coutaz et al. 1995]).

Nous analysons les fonctions suivant leur domaine de départ et leur domaine image. Par exemple, Synchronous Gesture [Hinckley 2003] offre un exemple de fusion entre écrans : quand un écran annexe un second écran, l'utilisateur obtient un espace d'affichage unique représenté par deux écrans. Le domaine de départ de la fonction finale est un espace d'affichage. Le domaine image est l'union des domaines images des fonctions d'affichage de chaque écran. En d'autres termes, les écrans se partagent le même domaine de départ.

Un pointeur souris manipulable par deux souris est un autre exemple de fusion : les deux souris sont associées au même pointeur. Dans les paragraphes précédents, nous évoquons la création d'une nouvelle entité, *souris<sub>3</sub>*, résultat du couplage de *souris<sub>1</sub>* et *souris<sub>2</sub>* qui pilote alors le même curseur. Avant le couplage, *souris<sub>1</sub>* et *souris<sub>2</sub>* ont un comportement similaire. Leur domaine de départ est un repère  $x_1, y_1$  (respectivement  $x_2, y_2$ ) porté par la surface sur laquelle la souris évolue. Le domaine image est un repère  $x, y$  (respectivement  $x', y'$ ) qui est porté par l'écran sur lequel le curseur évolue. Le couplage donnant lieu à *souris<sub>3</sub>* a pour domaine de départ l'union de  $x_1, y_1$  et  $x_2, y_2$  alors que le domaine image est soit  $x, y$  (si le curseur de *souris<sub>2</sub>* n'est pas conservé), soit  $x', y'$ . En tout état de cause, les deux souris physiques se partagent le même curseur. L'utilisateur peut employer l'une ou l'autre indifféremment pour désigner un point de l'écran.

L'analyse de l'état de l'art et l'expérience pratique tendent à montrer que la fonction de fusion est compatible avec toute autre fonction de couplage. L'expérience conduira à l'émergence d'heuristiques et de bons principes. Pour l'instant, le domaine se construit et se contente de proposer des techniques d'interaction.

### **La notion de nouvelle fonction**

Notre définition du couplage (Chapitre I) repose sur la notion de nouvelle fonction. Nous constatons dans les exemples que nous avons utilisés pour illustrer nos propos deux types de nouvelles fonctions. Dans les paragraphes précédents, nous proposons de coupler deux souris pour la manipulation d'un seul curseur. La fonction du couplage est sensiblement la même que celle des entités non couplées à la modification près des domaines de départ et image.

Dans le cas du couplage d'un clavier avec une souris, la fonction obtenue n'est pas une modification des domaines des fonctions. Avant couplage, la souris désigne des points  $x, y$  à l'écran et le clavier permet la saisie de caractères. Par couplage, la fonction produite est la fonction de focus. Cette fonction n'étant pas remplie par le clavier seul ou la souris seule.



## 4. Synthèse

---

En partant du constat de la nécessité de définir des cadres d'analyse pour le couplage, nous proposons un cycle de vie régissant les différents états d'un couplage. Ces états sont décrits à l'aide des prédicats : couplé/découplé, verrouillé/déverrouillé, couplable/découplable. En appliquant ce cadre aux exemples de l'état de l'art, nous avons mis en évidence certains éléments importants. Ainsi le nombre d'états couverts par un prototype montre le pouvoir d'expression en termes de couplage du dit prototype mais également sa complexité. Les actions réalisées par l'utilisateur et la symétrie ou non des impacts de ses actions sont également pertinentes.

L'analyse des propriétés de souplesse et de robustesse de l'interaction appliquées à l'automate du couplage de ressources d'interaction, montre, avant toute implémentation, qu'il s'agit d'un problème plus complexe qu'il n'y paraît. Il montre aussi que l'état de l'art s'est peu préoccupé du problème ou s'y est attaché de manière plutôt exploratoire que rationnelle. Dans la section qui suit, nous formalisons le couplage de ressources d'interaction en vue de répondre aux questions laissées sans réponse, notamment sur la transitivité du couplage.

Après avoir étudié la vie des couplages, nous envisageons la génération de nouveau couplage. Sans donner de règle figée, nous proposons une grammaire supportant la génération de nouveaux couplages. Cette génération décrite sur la base de fermeture transitive, s'appuie sur des heuristiques que le concepteur devra définir et pour lesquelles à terme, nous espérons qu'un certain nombre d'entre elles feront consensus.

Après avoir étudié comment vivent les couplages, nous nous intéressons aux techniques d'interaction que l'utilisateur va mettre en œuvre pour initier ces couplages. Nous nous intéressons particulièrement à la notion de désignation.

---

*Chapitre V*

*Techniques d'interaction pour  
le couplage de ressources  
d'interaction*

---

## 1. Introduction

---

Selon Foley, une technique d'interaction est une manière d'utiliser des dispositifs d'entrée pour saisir de l'information (« interaction techniques are ways to use input devices to enter information into the computer ») [Foley et al. 1990]. Selon Bowman, une technique d'interaction est « une méthode selon laquelle l'utilisateur réalise une tâche sur un ordinateur via l'interface utilisateur » [Bowman et Hodges 1999]. Ici, nous nous intéressons aux techniques d'interaction mise à la disposition de l'utilisateur pour spécifier un couplage de ressources d'interaction. La sémantique de la phrase de couplage correspond à la fonction de couplage.

En se référant à la définition du couplage que nous avons donné dans le chapitre II, une technique d'interaction doit être une action grâce à laquelle un ou plusieurs utilisateurs désignent deux ressources d'interaction à coupler et si cela est nécessaire, la fonction que doit produire le couplage. Nous étudions donc les deux notions de désignation de ressources d'interaction et de technique d'interaction monofonctionnelle et plurifonctionnelle.

---

## 2. Notation

---

Dans les techniques précédemment citées (chapitre III), certaines comme Synchronous Gesture [Hinckley 2003] ou les ConnectTables [Tandler et al. 2001] sont très ancrées dans le physique, d'autres comme le Click'n'Couple (détaillée en section “Une nouvelle technique d'interaction : Click'n'Couple”) ou un configurateur [Barralon et al. 2004] ou Figure 22, sont résolument logicielles. Enfin comme dans les DataTiles, l'utilisateur manie un instrument pour désigner les ressources d'interaction. Afin de classifier ces techniques sur le plan de la désignation des ressources d'interaction, nous nous basons sur une notation graphique proposée par Renevier [Renevier 2004] concernant les systèmes mixtes. Cette notation offre l'avantage de modéliser aussi bien des entités physiques que des entités numériques.

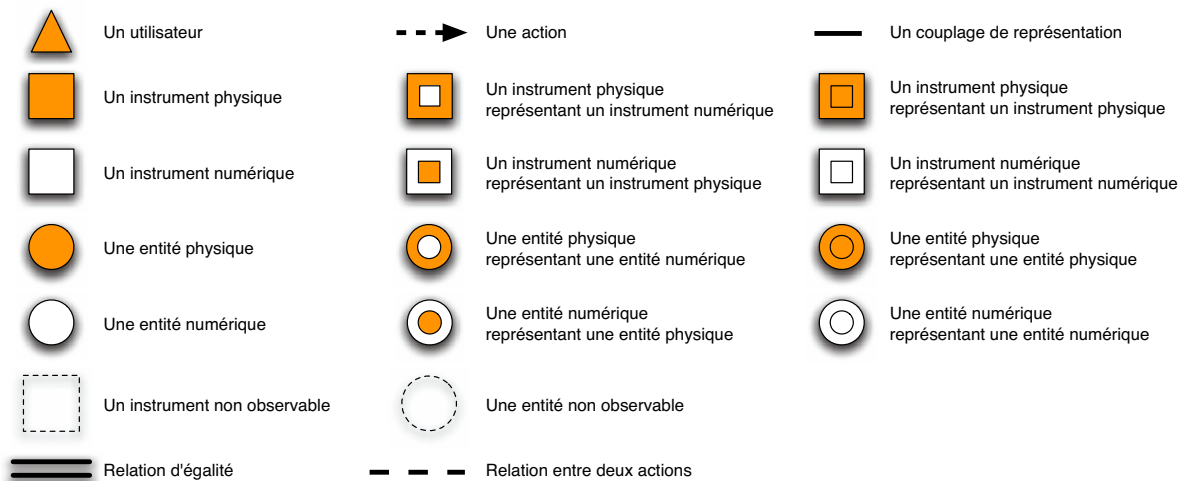
---

### 2.1. LES ÉLÉMENTS DE LA NOTATION

La Figure 1 montre les symboles que nous retenons pour notre notation :

- Un rond symbolise un instrument, un carré une entité (autre qu'un instrument),

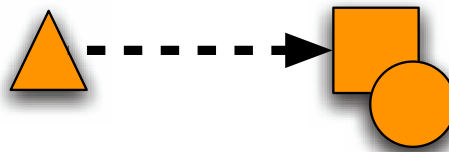
- Une forme pleine (remplie en orange) symbolise indique le caractère physique alors qu'une forme vide symbolise le caractère numérique,
- Une forme incluse dans une autre indique une représentation,
- Les relations de représentation sont symbolisées par un trait plein,
- Les actions sont symbolisées par une flèche en pointillé,
- L'utilisateur est représenté par un triangle,
- Une forme entourée par un trait pointillé est non observable,
- On montre la duplication d'une entité en reliant par un trait double les deux instances de cette entité.



**Figure 1**  
Récapitulatif des types d'entités et des relations.

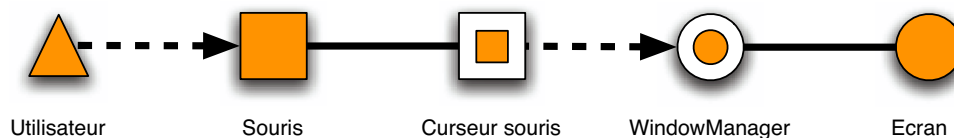
Par rapport à TEC, les instruments et les objets sont tous des entités. Il est ici nécessaire de faire la distinction entre les entités instruments de désignation et les ressources d'interaction objets de la désignation. Nous conservons donc le terme « instrument » de Renevier et remplaçons le terme objet par le terme générique « entité ».

La colocalisation de deux symboles indique que les entités forment un tout. Dans la Figure 13 a), la ressource à coupler est également l'instrument ; c'est le cas lorsque la ressource à coupler embarque avec elle des capteurs qui la transforment en instrument.



**Figure 2**  
**Colocalisation de symboles.**  
L'entité désignée est également l'instrument de désignation.

En guise d'illustration, la Figure 3 montre l'exemple d'une désignation en informatique conventionnel. Un utilisateur manipule le curseur souris (instrument numérique représentant un instrument physique) grâce à une souris (instrument physique) et va agir sur le window manager (représentation numérique) d'un écran (ressource à coupler physique). On note que les relations d'action n'interviennent qu'entre des entités (instrument ou objet) de même classe (physique ou numérique). Par ailleurs, un utilisateur ne peut agir que sur un instrument physique.



**Figure 3**  
**Exemple d'interaction entre une souris manipulée par un utilisateur et un écran**

## 2.2. GRAMMAIRE D'UNE DÉSIGNATION

Nous proposons une grammaire pour la construction d'une chaîne de désignation. Il ne nous est pas possible de proposer une grammaire simple couvrant deux désignations et les relations entre ces désignations. En effet, le graphe formé n'est pas un arbre. Chaque désignation est donc exprimée à l'aide de la grammaire BNF suivante :

$$\begin{aligned}
 V_T &= \{i_p, i_n, i_{prp}, i_{prn}, i_{nrn}, i_{nrp}, e_p, e_n, e_{prp}, e_{prn}, e_{nrn}, e_{nrp}, \text{user}, \rightarrow, -\} \\
 V_N &= \{ \langle \text{TechniqueDesignation} \rangle, \langle \text{ChaineDesignation} \rangle, \langle \text{Chaine}_{ipp} \rangle, \langle \text{Chaine}_{epp} \rangle, \langle \text{Chaine}_{ipn} \rangle, \\
 &\langle \text{Chaine}_{enp} \rangle, \langle \text{Chaine}_{epn} \rangle, \langle \text{Chaine}_{epn} \rangle, \langle \text{Chaine}_{ipn} \rangle, \langle \text{Chaine}_{enn} \rangle, \langle \text{Chaine}_{iprp} \rangle, \langle \text{Chaine}_{inrn} \rangle, \\
 &\langle \text{Chaine}_{inrn} \rangle, \langle \text{Chaine}_{eprp} \rangle, \langle \text{Chaine}_{eprp} \rangle, \langle \text{Chaine}_{enrn} \rangle \} \\
 \\
 \langle \text{TechniqueDesignation} \rangle &::= \text{user} \rightarrow \langle \text{ChaineDesignation} \rangle \\
 \langle \text{ChaineDesignation} \rangle &::= i_p e_p \\
 &\quad | \langle \text{Chaine}_{ipp} \rangle \rightarrow \langle \text{Chaine}_{epp} \rangle \\
 &\quad | \langle \text{Chaine}_{ipn} \rangle \rightarrow \langle \text{Chaine}_{enp} \rangle \\
 &\quad | \langle \text{Chaine}_{ipp} \rangle \rightarrow \langle \text{Chaine}_{epn} \rangle \\
 &\quad | \langle \text{Chaine}_{ipn} \rangle \rightarrow \langle \text{Chaine}_{enn} \rangle \\
 \\
 \langle \text{Chaine}_{ipp} \rangle &::= i_p \langle \text{Chaine}_{iprp} \rangle \\
 \langle \text{Chaine}_{epp} \rangle &::= \langle \text{Chaine}_{eprp} \rangle e_p \\
 \langle \text{Chaine}_{ipn} \rangle &::= i_p \langle \text{Chaine}_{iprp} \rangle - i_{nrp} \langle \text{Chaine}_{inrn} \rangle \\
 \langle \text{Chaine}_{enp} \rangle &::= \langle \text{Chaine}_{enrn} \rangle e_{nrp} - \langle \text{Chaine}_{eprp} \rangle e_p \\
 \langle \text{Chaine}_{epn} \rangle &::= \langle \text{Chaine}_{eprp} \rangle e_{prn} - \langle \text{Chaine}_{enrn} \rangle e_n \\
 \langle \text{Chaine}_{enn} \rangle &::= \langle \text{Chaine}_{enrn} \rangle e_p \\
 \\
 \langle \text{Chaine}_{iprp} \rangle &::= \varepsilon \mid i_{prp} \langle \text{Chaine}_{iprp} \rangle \\
 \langle \text{Chaine}_{inrn} \rangle &::= \varepsilon \mid - i_{nrn} \langle \text{Chaine}_{inrn} \rangle \\
 \langle \text{Chaine}_{eprp} \rangle &::= \varepsilon \mid e_{prp} - \langle \text{Chaine}_{eprp} \rangle \\
 \langle \text{Chaine}_{enrn} \rangle &::= \varepsilon \mid e_{nrn} - \langle \text{Chaine}_{enrn} \rangle
 \end{aligned}$$

Nous détaillons maintenant certaines parties de cette grammaire pour voir quelles désignations elle autorise.

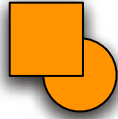
### Les éléments terminaux

$$V_T = \{i_p, i_n, i_{prp}, i_{prn}, i_{nrn}, i_{nrp}, e_p, e_n, e_{prp}, e_{prn}, e_{nrn}, e_{nrp}, \text{user}, \rightarrow, -\}$$

Les éléments terminaux sont les équivalents des symboles des entités présentés en Figure 1. La première lettre indique s'il s'agit d'un instrument (i) ou d'une entité (e) excepté pour l'utilisateur (user). Les lettres en indice expriment le caractère de l'entité : « n » pour numérique, « p » pour physique et « r » pour représentation. Ainsi par exemple, « iprp » représente un instrument physique représentant un instrument physique soit avec la notation graphique un carré orange inclus dans un carré orange ; « en » est une entité numérique soit un carré vide. Afin de ne pas surcharger la grammaire, nous n'avons pas fait apparaître le caractère observable. Puisqu'il est possible de décorer n'importe quel entité nous noterons une entité non observable : « NO(entité) ». Par exemple, un

instrument numérique représentant un instrument numérique (inrn) non observable est noté NO(inrn).

L'équivalent de la colocalisation de la notation graphique est la « concaténation » de deux terminaux. Par exemple « ipep » est l'équivalent de la notation de la Figure 4 dans laquelle l'entité physique à coupler est également l'instrument physique de la désignation.



**Figure 4**  
L'entité physique à désigner joue également le rôle d'instrument de désignation..

La notation que nous donnons ne permet en l'état que la description de la structure d'une désignation. Le nommage des éléments terminaux est absent. C'est pourquoi nous proposons d'ajouter le nom de l'entité concernée en exposant s'il s'agit d'une entité, un nom arbitraire s'il s'agit d'une action. Par exemple, l'écran physique « écran1 » sera noté, s'il est l'entité désignée : eécran1, une action *ia*1.

### Technique de désignation et chaîne de désignation

```
<TechniqueDesignation> ::= user → <ChaîneDesignation>  
<ChaîneDesignation> ::= ipep  
| <Chaîneipp> → <Chaîneepp>  
| <Chaîneipn> → <Chaîneenp>  
| <Chaîneipp> → <Chaîneepn>  
| <Chaîneipn> → <Chaîneenn>
```

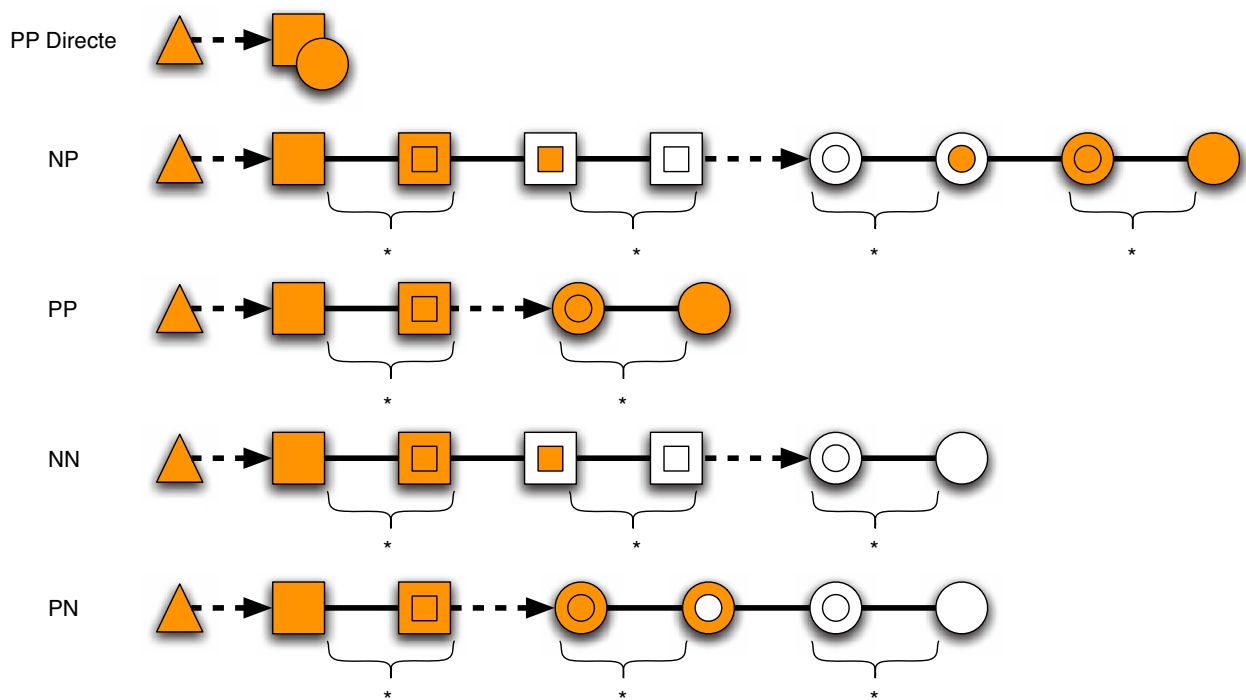
Une désignation débute nécessairement par un utilisateur qui va faire une action pour désigner. Nous avons fondé notre grammaire à partir d'un ensemble de contraintes que nous énonçons maintenant :

1. Un utilisateur agit nécessairement dans le monde physique,
2. La frontière entre le monde physique et le monde numérique n'est franchie que par des couplages de représentation (jamais par des actions).

En prenant en compte ces contraintes, une désignation (<ChaîneDesignation>) peut être de cinq formes différentes : trois permettant la

désignation d'entités physiques et deux d'entités numériques. La Figure 5 représente ces cinq types de désignation ainsi nommées : Physique-Physique direct, Numérique-Physique, Physique-Physique, Numérique-Numérique, Physique-Numérique.

La règle initiale pour le nommage est la suivante. Nous donnons les classes des entités suivant et précédent une action (sans tenir compte de l'utilisateur). La deuxième désignation de la Figure 5 devrait se nommer PNNP. Or, compte tenu de la contrainte 1 énoncée précédemment, le premier P est inutile. La contrainte 2 nous indique quant à elle que l'on trouve systématiquement la même classe d'entités de par et d'autre, d'une flèche d'action. Donc les lettres deux et trois du nom sont identiques. Au final, seules les deux dernières lettres sont nécessaires. Un cas particulier est concédé pour le premier cas, très proche dans l'idée de la désignation PP mais pour laquelle, il n'y a pas réellement d'action entre instrument et entité désignée puisque l'entité est l'instrument. Cette désignation est nommée PP directe.



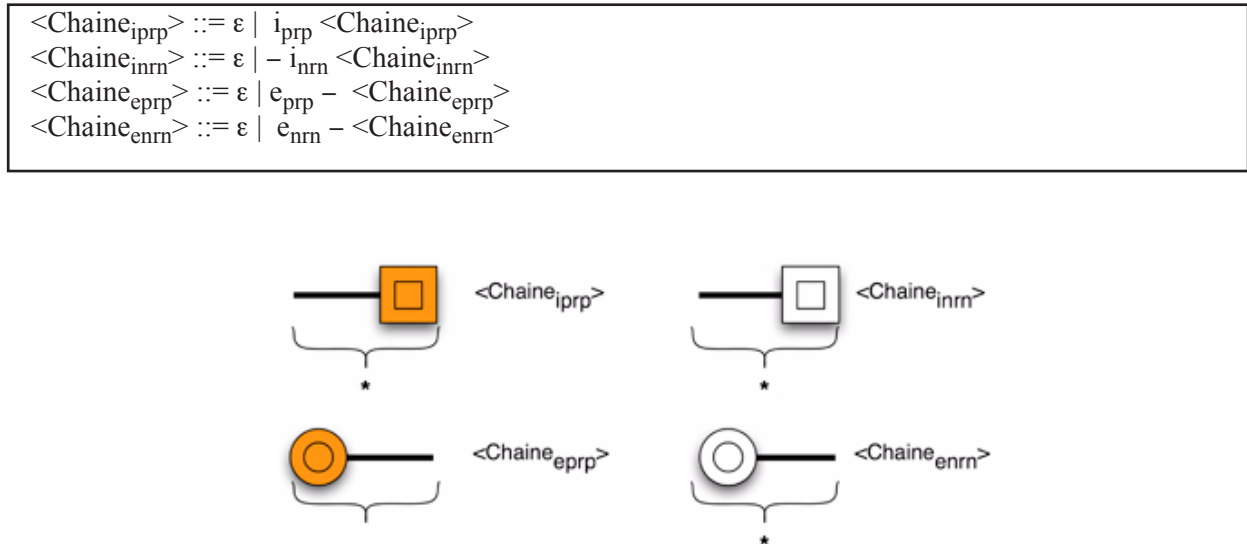
**Figure 5**  
Cinq types de désignations : Physique-Physique direct, Numérique-Physique, Physique-Physique, Numérique-Numérique, Physique-Numérique.

**Chaîne de représentation**

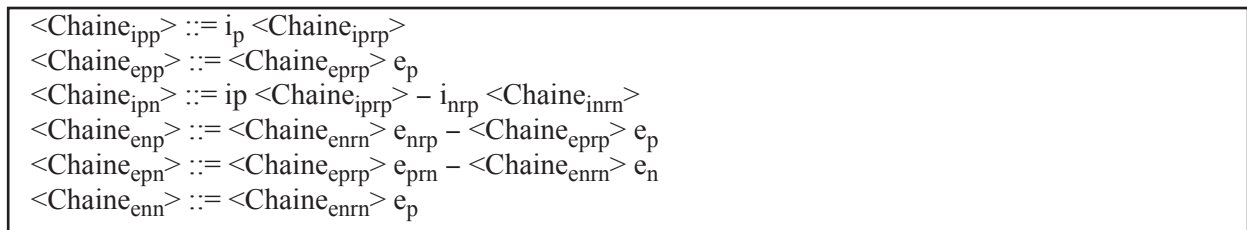
Les symboles « \* » indiquent que le fragment de chaîne peut être présent entre 0 et plusieurs fois. Les quatre chaînes de représentation sont



présentées en Figure 6 avec les noms des non-terminaux qu'ils représentent :  $\langle \text{Chaîne}_{\text{iprp}} \rangle$ ,  $\langle \text{Chaîne}_{\text{inrn}} \rangle$ ,  $\langle \text{Chaîne}_{\text{eprp}} \rangle$ ,  $\langle \text{Chaîne}_{\text{enrn}} \rangle$



**Figure 6**  
Chaînes de représentations : iprp, inrn, eprp, enrn.



A partir des éléments que nous avons décrits précédemment, les descriptions ci-dessus construisent les chaînes d'instrument et d'entité.

## Deux désignations

Une technique d'interaction étant composée de deux désignations, la notation introduit la notion de relation entre actions. Comme le montre la Figure 1, une relation est symbolisée par un trait en pointillé et nous y associons une décoration. Nous ne donnons ici un ensemble exhaustif des décorations possibles. Cependant, nous proposons :

- « la colocalisation » pour exprimer que deux actions ont lieu au même endroit et de manières plus générales n'importe quelle relation spatiale,
- « le parallélisme » et de manières plus générales les relations temporelles d'Allen [Allen et al. 1994],
- les propriétés CARE [Coutaz et al. 1995].

L'égalité est quant à elle, représentée par un trait double entre deux entités. Dans la notation graphique, l'égalité pourra avantageusement être remplacée par la fusion des deux entités. Certes, cette possibilité implique que la grammaire n'est plus respectée en l'état, mais elle permet de simplifier la compréhension. Nous verrons également que puisque les entités sont nommées lors de la représentation textuelle (en exposant), l'égalité n'a pas besoin d'être exprimée. Néanmoins, si cela s'avère nécessaire, nous la noterons ainsi avec le symbole « = » traditionnel. Les autres relations (colocalisation, parallélisme, etc .) sont notées par leur nom et utilisées comme opérateur binaire par exemple :

$e_p^{\text{ecran1}} = e_p^{\text{ecran2}}$	pour l'égalité
$e_p^{\text{ecran1}} \text{ coloc } e_p^{\text{ecran2}}$	pour la colocalisation de deux entités
$a1 \text{ parallèle } a2$	pour le parallélisme de deux actions

Après avoir présenté la notation graphique et la grammaire permettant de générer les chaînes de techniques d'interaction, nous l'illustrons avec des exemples dont certains sont issus de l'état de l'art.

### 2.3. ILLUSTRATION

#### Interaction vocale

La Figure 7 montre la description d'une technique basée sur la parole. L'utilisateur prononce la phrase : « Couple l'écran1 avec l'écran2 ». Dans la description, le micro est l'instrument sur lequel l'utilisateur agit avec sa voix. Ce micro est couplé avec un système de reconnaissance de la parole non observable. Ce dernier agit sur des représentations numériques non observables des deux surfaces à coupler. Ainsi, l'utilisateur ne pourra au mieux qu'observer les résultats du couplage pour évaluer s'il est conforme à son désir initial.

L'équivalent de la notation de la Figure 7 sous forme textuelle est la suivante :

$user^{\text{utilisateur}} \rightarrow i_p^{\text{micro}} - NO(i_{nrp}^{\text{reco}}) \rightarrow e_{nrp}^{\text{surfaceaffichage1}} - e_p^{\text{ecran1}}$
$user^{\text{utilisateur}} \rightarrow i_p^{\text{micro}} - NO(i_{nrp}^{\text{reco}}) \rightarrow e_{nrp}^{\text{surfaceaffichage2}} - e_p^{\text{ecran2}}$

Nous pouvons également ajouter les trois relations d'égalité mais puisque les entités sont nommées cela devient superflu.

Nous présentons en Figure 8 une forme condensée de la représentation graphique de cet exemple. Cette représentation ne correspond pas à la grammaire que nous proposons mais qui simplifie la compréhension.

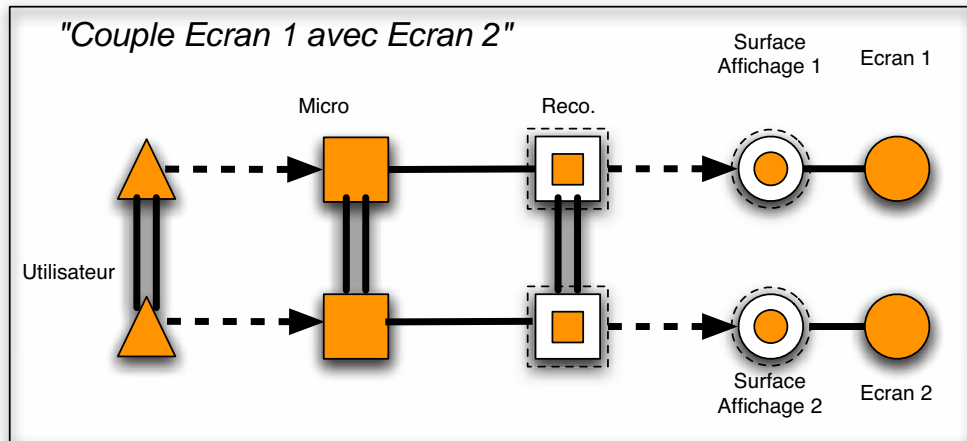


Figure 7  
Description graphique d'une technique avec modalité orale.

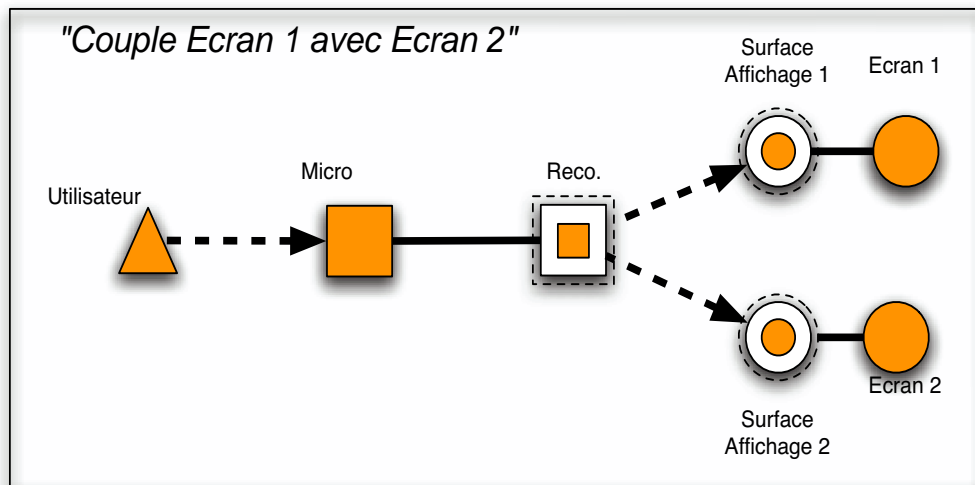


Figure 8  
Forme condensée de la notation graphique.

La Figure 9 est une variante de la technique précédente. La phrase prononcée par l'utilisateur est « couple cet écran avec écran2 » et l'associe à un geste de la main pour toucher l'écran. Nous avons donc mis en relation les actions entre la main et écran1, et le système de reconnaissance et la représentation d'écran1. Cette relation est décorée de la propriété de redondance tirée de CARE [Coutaz et al. 1995] entre l'action a1 et l'action a2. Sous forme textuelle, nous notons :

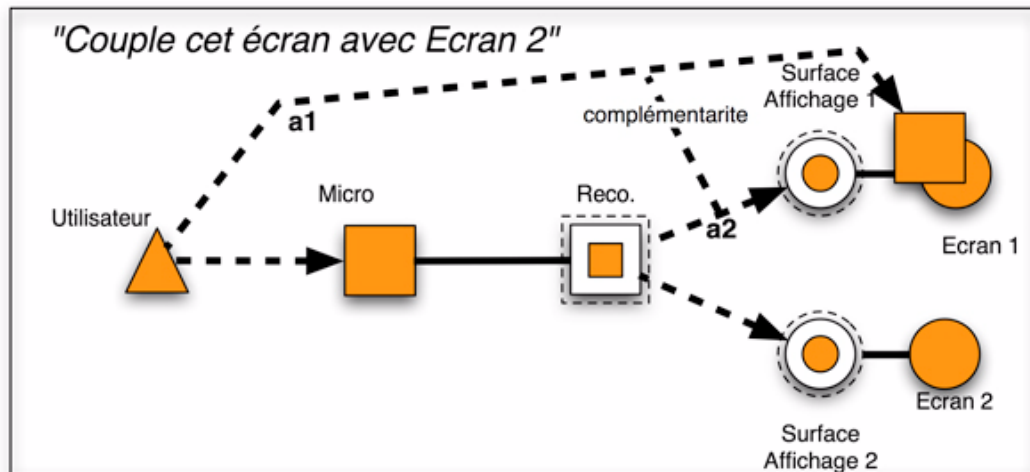
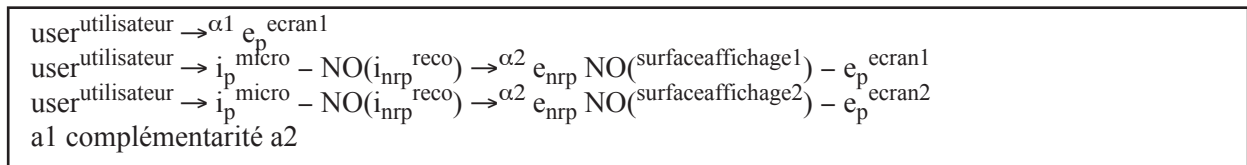
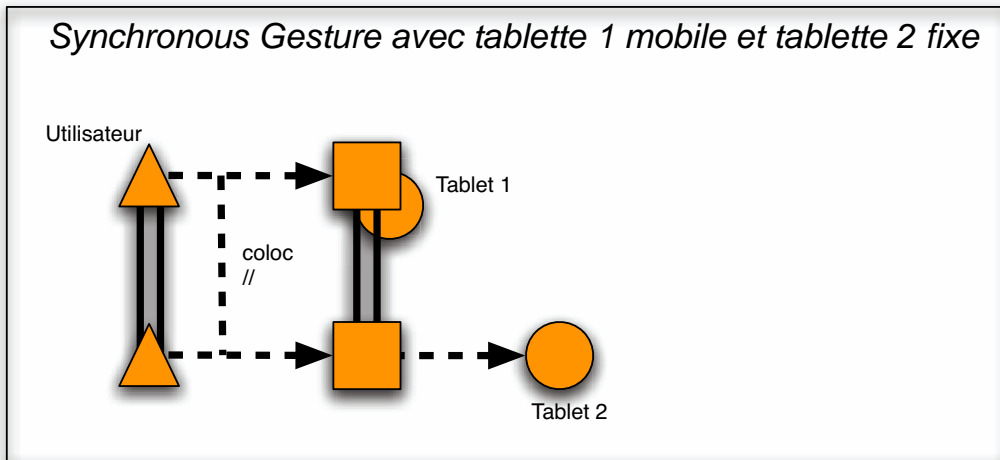


Figure 9  
Description graphique d'une technique avec modalité orale comportant une complémentarité.

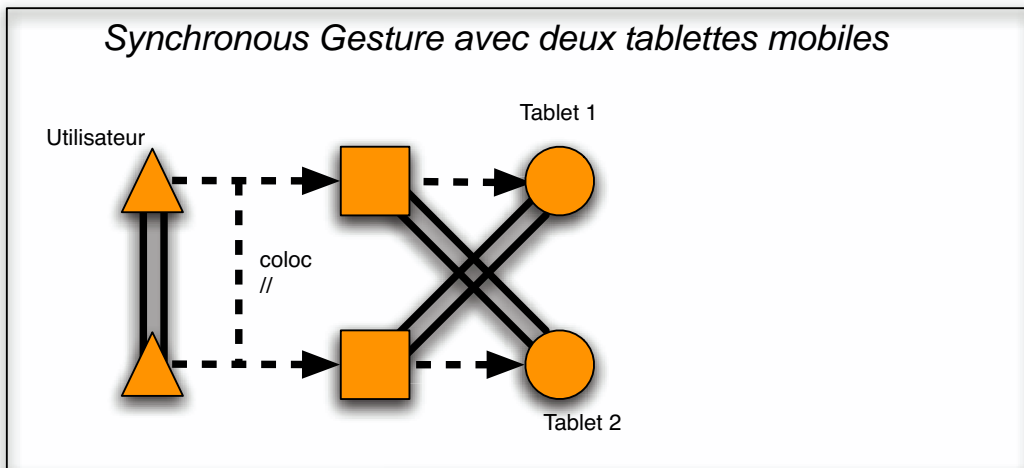
### Synchronous Gesture

La Figure 10 est relative à l'une des variantes de technique d'interaction dans Synchronous Gesture [Hinckley 2003]. Dans cette technique, l'utilisateur tient dans sa main une tablette et s'en sert pour venir choquer une seconde. La désignation de la première tablette est de type PP direct puisque c'est le capteur de pression sur la première tablette qui sert à la désigner. La seconde tablette est désignée avec un instrument intermédiaire à savoir la première tablette. Les deux actions sont alors naturellement colocalisées et réalisées en parallèle.



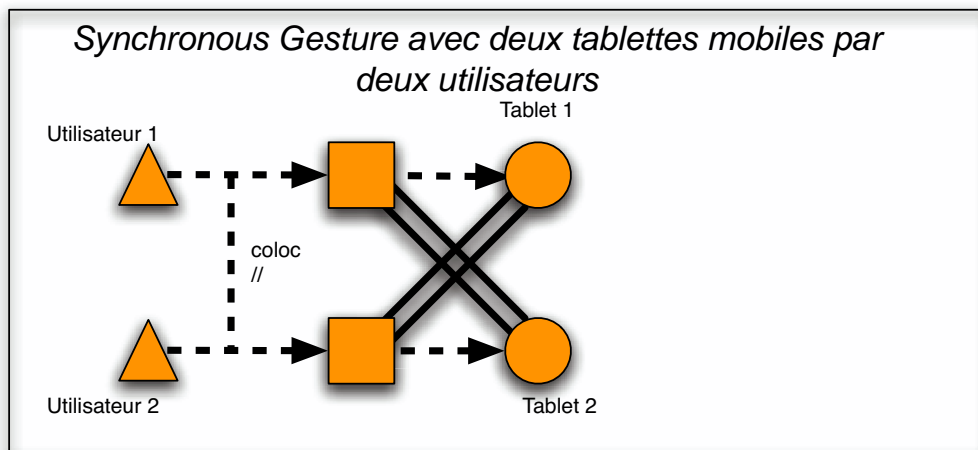
**Figure 10**  
Description graphique du couplage dans Synchronous Gesture avec une tablette mobile et une autre fixe.

La Figure 11 montre une deuxième variante de technique de Synchronous Gesture dans laquelle les deux tablettes sont mobiles et se désignent mutuellement. Ici encore, les actions sont colocalisées et parallèles.



**Figure 11**  
Description graphique du couplage dans Synchronous Gesture avec deux tablettes mobiles.

Enfin, la Figure 12 montre la dernière variante de technique quasiment identique à la précédente mais variant sur le fait que deux utilisateurs distincts sont présents.

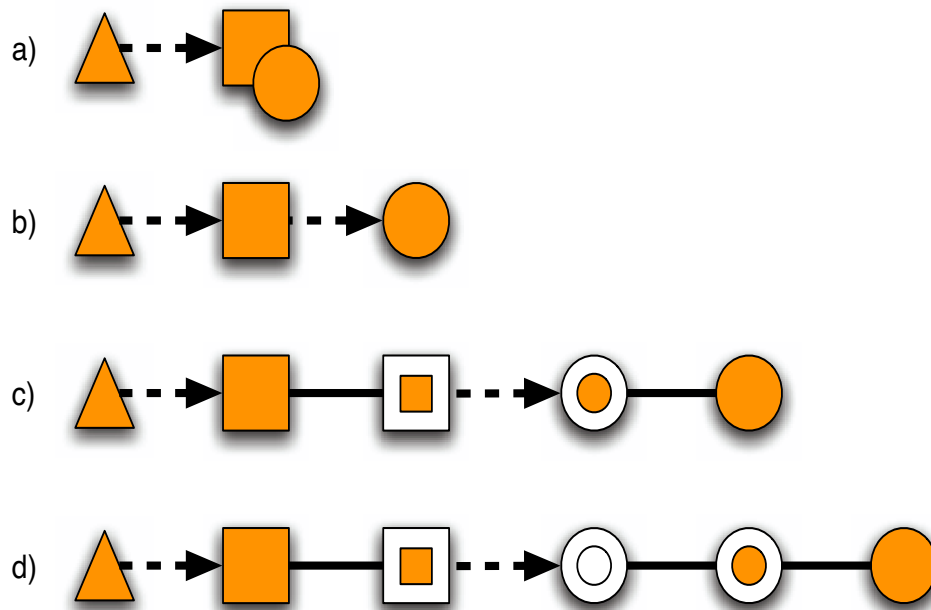


**Figure 12**  
Description graphique du couplage dans Synchronous Gesture avec deux tablettes mobiles tenues par deux utilisateurs.

Nous voyons d'après la grammaire qu'une infinité de désignations peut être construite en particulier à cause des chaînes de représentation pouvant générer un nombre illimité de symboles. Cependant en analysant l'état de l'art, nous constatons qu'un certain nombre d'entre elles sont récurrentes. Nous en discutons maintenant.

#### 2.4. CLASSES DE DÉSIGNATION

L'analyse de l'état de l'art fait ressortir quatre grandes classes de désignation (Figure 13) que nous détaillons en ajoutant une analyse a priori des avantages et inconvénients de chaque classe.



**Figure 13**  
**Quatre grandes classes de désignation..**

a) L'utilisateur agit directement sur la ressource d'interaction à coupler qui prend alors également le statut d'instrument. Les tablettes d'Hinckley [Hinckley 2003] font clairement partie de cette catégorie dans la technique où la désignation est réalisée par ces capteurs de pression qui détecte qu'un utilisateur tient la tablette dans ses mains (première désignation de la Figure 10). Cette classe de technique présente l'avantage que la charge cognitive pour l'utilisateur quant à la désignation des ressources est quasi-nulle. Cependant, ce genre de technique n'est applicable que pour des dispositifs mobiles et équipés de capteurs.

b) L'utilisateur, à l'aide d'un instrument désigne les ressources qu'il désire coupler dans le monde physique. Dans les DataTiles [Rekimoto et al. 2001], l'utilisateur se sert d'un stylet pour désigner les deux tuiles à coupler. Ici, l'action se situe toujours dans le monde physique, que se soit pour l'instrument ou la ressource à coupler. Cette physicalité présente les mêmes intérêts que la classe de technique précédente au niveau cognitif et rend plus souple son utilisation avec des dispositifs moins mobiles ou une interaction à distance. L'ajout de l'instrument par rapport à la précédente classe de technique crée un niveau d'indirection entre l'utilisateur et la ressource à coupler.

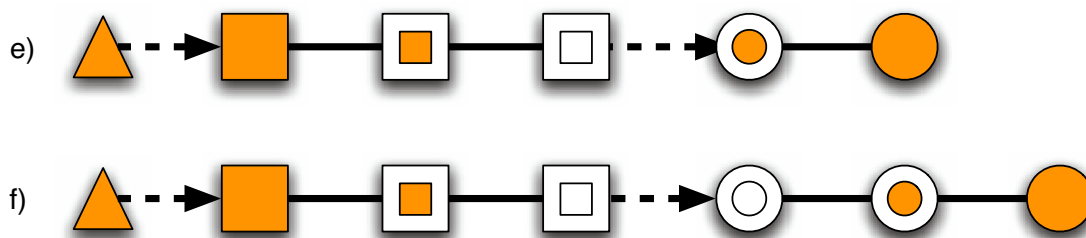
c) L'utilisateur manipule un instrument numérique grâce à un instrument physique. Ensuite l'instrument numérique agit sur un autre objet numérique représentant la ressource à coupler. La technique Click'n'Couple [Barralon et al. 2005] (décrite dans le chapitre IV) réalisée

avec une souris en est l'exemple parfait. La souris manipule le curseur qui agit sur le window manager de la surface à coupler. Dans cette classe de technique, l'utilisateur doit faire l'association entre l'instrument physique et sa représentation, la ressource d'interaction et sa représentation et le fait que l'interaction se passe entre les deux représentations. Si cette classe semble plus lourde cognitivement pour l'utilisateur, elle est néanmoins très flexible quand au type de ressources utilisées se présente comme une bonne alternative lorsque les contraintes (matérielles par exemple) empêchent l'utilisation des techniques a) et b).

d) Dans cette dernière classe, un niveau supplémentaire est ajouté par rapport aux techniques c) puisque la désignation de la ressource se fait par une représentation numérique de la représentation de la ressource. Les configureurs, comme celui présenté en Annexe ou dans [Barralon et al. 2005] incarnent cette classe de technique. C'est par le biais d'une représentation du window manager que la désignation est réalisée. Classe de technique la plus lourde cognitivement, c'est également celle qui offre un contexte d'utilisation le plus vaste. On peut citer par exemple le couplage à distance de ressources d'interaction ou la manipulation d'un grand nombre de ressources.

Les quatre classes de techniques ci-dessus ont toutes une implémentation dans des prototypes de recherche. Nous constatons également que les chaînes de représentation ( $\langle \text{Chaîne}_{\text{iprp}} \rangle$ ,  $\langle \text{Chaîne}_{\text{inrm}} \rangle$ ,  $\langle \text{Chaîne}_{\text{eprp}} \rangle$ ,  $\langle \text{Chaîne}_{\text{enrm}} \rangle$ ) génèrent au maximum une entité. En se basant sur ce constat, l'utilisation de la notation permet de mettre en évidence deux autres classes de techniques. La Figure 14 représente ces deux classes que nous ne pouvons donc pas illustrer. Les classes e) et f) ont été construits respectivement à partir des classes c) et d) en ajoutant un instrument numérique représentant un instrument numérique. L'intérêt d'un tel ajout semble limité. Ceci est conforté par le fait qu'aucun exemple de l'état de correspond à ces classes de désignation. Il est intéressant de voir que notre modèle est génératif et nous ne rejetons donc pas d'emblée les désignations issues de cette génération.





**Figure 14**  
Classes de techniques d'interaction non présentes dans les prototypes de recherche.

Nous donnons maintenant un exemple complet de désignation, c'est-à-dire comportant deux désignations avant de proposer une nouvelle technique d'interaction pour le couplage de surfaces nommée Click'n'Couple.

## 2.5. COUPLAGE DANS I-AM

### I-AM (Interaction Abstract Machine)

I-AM [Lachenal 2004] est une machine du même niveau d'abstraction que les gestionnaires graphiques de base comme Xwindow. Ces gestionnaires gèrent les ressources d'interaction d'une station de travail dont ils assurent le partage entre les applications clientes. Comme ces gestionnaires, I-AM est un intergiciel orienté Interaction Homme-Machine, mais I-AM en étend la couverture fonctionnelle pour tenir compte des requis de l'informatique ambiante. En effet, I-AM permet la construction dynamique d'espaces interactifs dont les surfaces et les instruments sont gérés par des stations de travail distinctes exécutant des systèmes d'exploitation éventuellement différents (MacOS, Windows), et donne l'illusion, au programmeur comme à l'utilisateur, que ces ressources sont gérées de manière uniforme par une seule station de travail. Plus spécifiquement, I-AM :

1. Masque l'hétérogénéité du matériel et des systèmes d'exploitation sous-jacents au bon niveau d'abstraction. Cette fonction est assurée en projetant les services d'I-AM dans les termes des systèmes d'exploitation sous-jacents.
2. Intègre les services de découverte dynamique de ressources d'interaction grâce à l'infrastructure de contexteurs [Rey 2005].
3. Modélise les relations spatiales entre les ressources d'interaction (et en déduit une topologie des surfaces). Ce service est assuré grâce aux informations fournies par l'infrastructure de contexteurs.

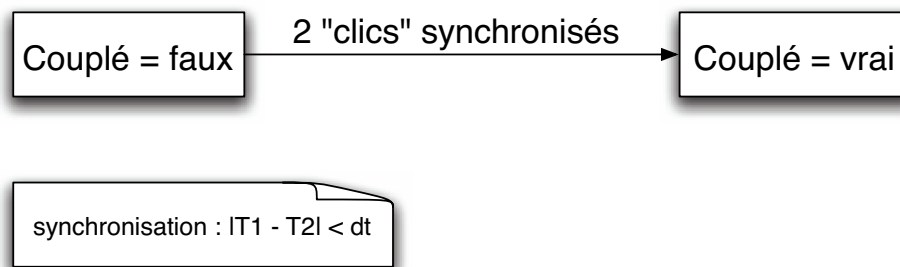
4. Permet la distribution et la migration des IHM graphiques au niveau du pixel. Ce service est assuré par une fonction de projection (une transformation affine) entre les espaces numériques et la topologie des surfaces.

Une description détaillée d'I-AM est présentée dans Lachenal [Lachenal 2004]. Notre contribution à cette infrastructure est détaillée en Annexe.

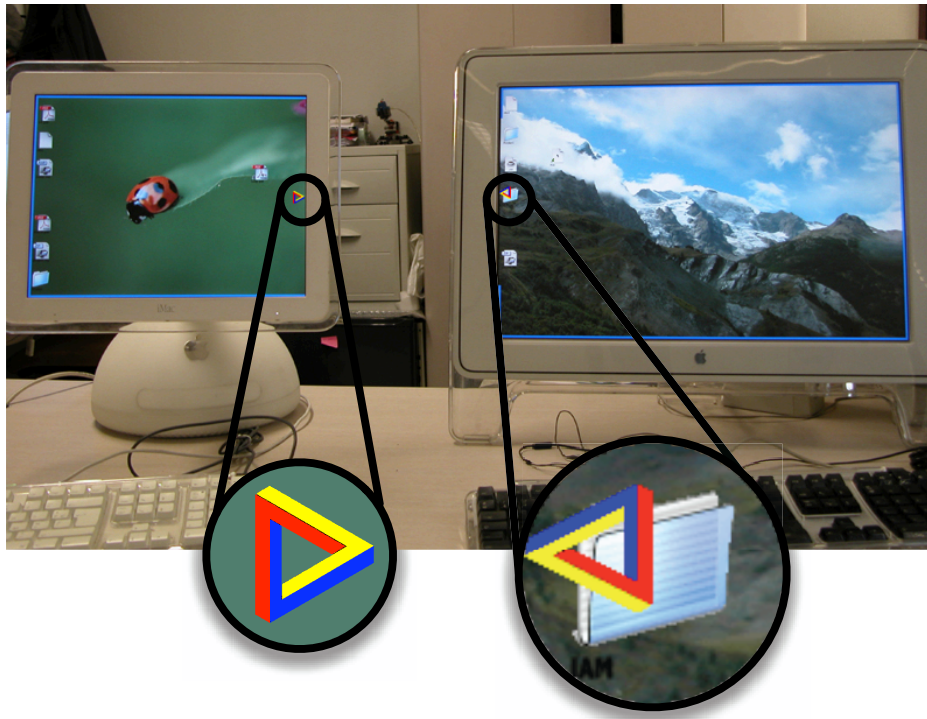
**Une nouvelle  
technique  
d'interaction :  
Click'n'Couple**

Click'n'Couple est une technique d'interaction pour le couplage de surfaces d'affichage inspiré des travaux SyncTap de Rekimoto [Rekimoto et al. 2003a].

Cette technique, décrite dans [Barralon et al. 2004] et illustrée en Figure 15, est basée sur la détection d'événements souris synchronisés. En utilisant simultanément une souris sur chacune des deux surfaces à coupler, l'utilisateur réalise deux « clics » synchronisés. L'effet immédiat est l'ouverture d'une passerelle entre les deux espaces interactifs. Le halo délimitant les frontières de l'espace interactif présente maintenant un passage entre les deux surfaces. De plus, un feedback apparaît sur les deux surfaces (Figure 16) à la manière des tablettes de Hinckley [Hinckley 2003] pour symboliser la liaison. En agissant sur ce feedback, il est possible de casser les liaisons, donc de découpler.

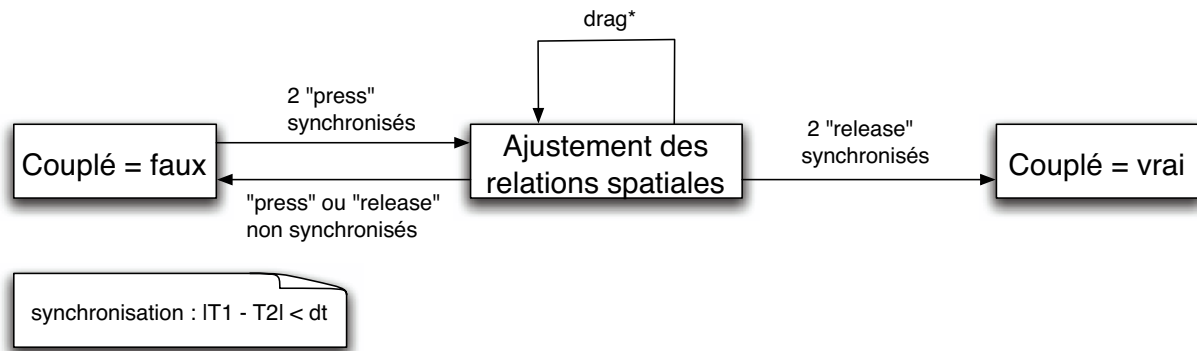


**Figure 15**  
Technique Click'n'Couple basé sur deux clics synchronisés pour effectuer le couplage.



**Figure 16**  
Feedback de couplage en deux écrans.

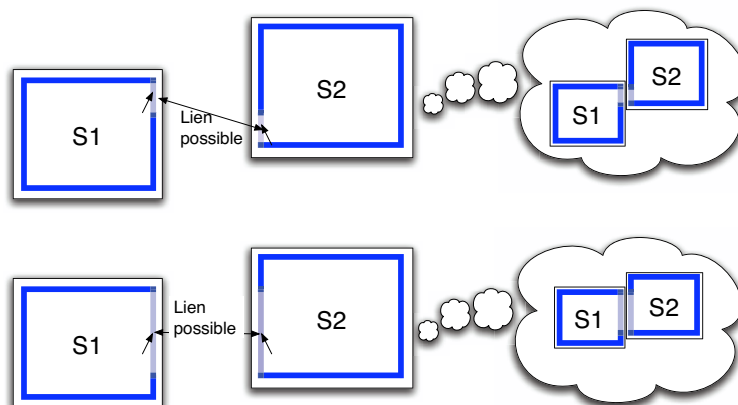
Une variante de cette technique permet à l'utilisateur d'anticiper le résultat du couplage (propriété de prédictibilité énoncée dans le chapitre 4). L'automate relatif à cette variante se trouve en Figure 17. Comme le montre la Figure 18 en partie gauche, l'utilisateur, une souris dans chaque main, amène les curseurs des souris dans le halo des bords à joindre. En appuyant simultanément sur les boutons des souris, les bords de jointure pâlisent, montrant le passage qui s'ouvrira entre les deux écrans au relâché des boutons (Figure 18 partie droite). La position du curseur dans le halo désigne de manière directe et dynamique un point de liaison de la surface. En maintenant les boutons enfoncés, jouant sur les positions respectives des points de liaison, l'utilisateur peut ajuster la taille souhaitée de la passerelle et donc la forme du futur espace unifié. La Figure 18 illustre cette pratique avec en partie gauche la l'état des souris-écran-passerelle en cours et à droite l'interprétation de l'espace unifié qu'en fait le système. La passerelle est opérationnelle au relâchement simultané des deux boutons. Le halo s'ouvre au niveau de la passerelle pour traduire la continuité de l'espace d'affichage formé par le couplage des deux surfaces comme avec la technique précédente.



**Figure 17**

**Technique Click'n'Couple permettant l'ajustement de la taille des passerelles.**

Deux "press" synchronisés permettent de passer dans le mode ajustement. Les « drag » des souris ajustent des passerelles. Les « release » synchronisés réalisent le couplage.



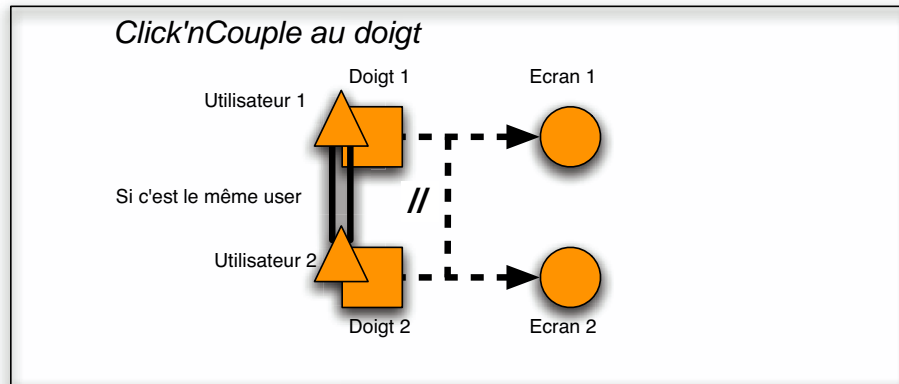
**Figure 18**

**Ajustement dynamique des points de liaison entre surfaces (partie gauche) et évaluation a priori de l'espace unifié potentiel (partie droite).**

Pour le découplage, une icône (Figure 16) apparaît 2 secondes lorsque le curseur d'une souris passe par la passerelle et que cette souris n'est pas couplée à un interacteur. Un « press » sur l'icône permet de savoir si les surfaces sont découplables et ce que deviendront les frontières (les halos) si le découplage est effectué. Si le « release » a lieu sur l'icône alors que les surfaces sont découplables, le découplage est effectif.

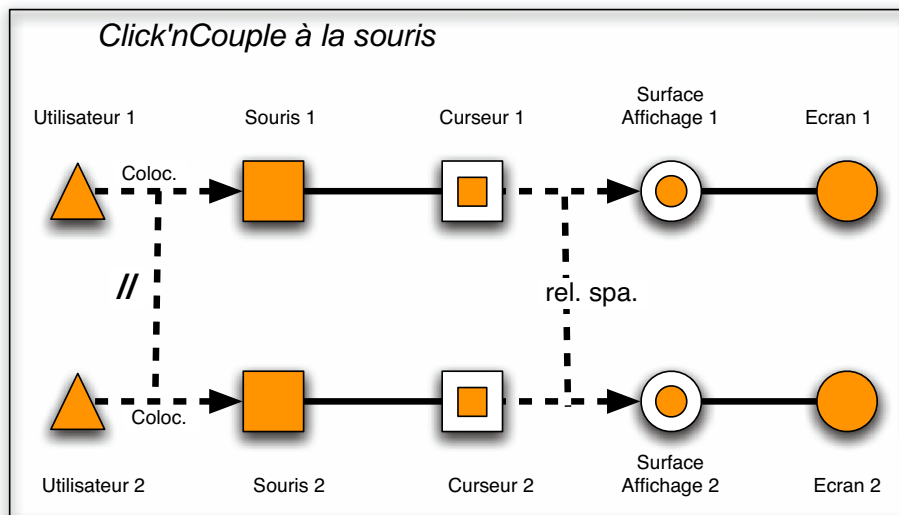
La Figure 19 montre la technique Click'n'Couple réalisée au doigt dans laquelle un utilisateur « clique » directement sur les écrans. Nous remarquons la relation d'égalité entre les deux symboles «utilisateur ».

Une relation de parallélisme temporel est aussi indiquée entre les deux actions sur les écrans.



**Figure 19**  
Description de la désignation de la technique Click'n'Couple réalisée au doigt.

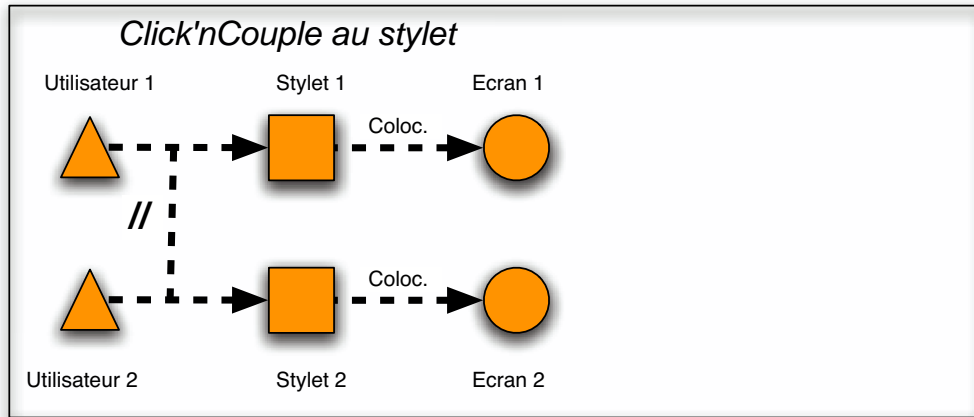
La Figure 20 représente les deux désignations à l'aide de la souris. Celles-ci correspondent à celle que nous avons présentée en Figure 1. La relation de temporalité, le parallélisme est déporté vers les actions entre l'utilisateur et la souris.



**Figure 20**  
Description de la désignation de la technique Click'n'Couple réalisée à l'aide de deux souris.

La Figure 21 montre la dernière déclinaison de la technique Click'n'Couple qui est réalisée à l'aide de stylets. Dans la description graphique, les actions peuvent elles aussi être décorées. Ainsi nous avons

mis en évidence dans la Figure 21 la colocalisation du stylet1 et écran1 ainsi que stylet2 avec écran2.



**Figure 21**  
Description de la désignation de la technique Click'n'Couple réalisée à l'aide de deux stylets.

### *3. Aspect fonctionnel d'une technique d'interaction*

#### **3.1. TECHNIQUE MONOFONCTIONNELLE / PLURIFONCTIONNELLE**

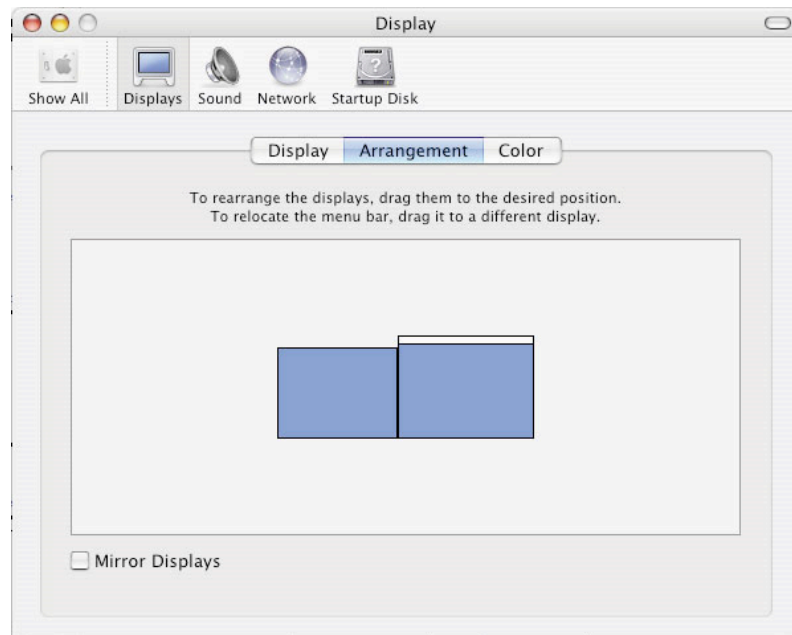
Parmi les prototypes de recherches de nouvelles techniques d'interactions sont apparues. Dans DataTiles, l'utilisateur agence des tuiles transparentes sur un écran à cristaux liquides. Le couplage de deux tuiles se fait en traçant à l'aide d'un stylet particulier, un trait entre deux tuiles adjacentes. Dans Synchronous Gesture deux tablettes équipées d'accéléromètres sont choquées l'une contre l'autre pour échanger des informations. En choquant une seule tablette sur une autre « fixe », la première étend son affichage sur la seconde. Tandler avec les Connect-Tables propose de coupler des surfaces en mettant en contact leur haut pour entamer une collaboration. Enfin, Rekimoto [Rekimoto et al. 2003a] présente SyncTap comme technique pour le couplage de dispositifs mobiles.

Ce bref récapitulatif montre que deux catégories de techniques d'interaction sont identifiables. Certaines, dites mono-fonctionnelles, permettent la mise en œuvre d'une seule fonction de couplage. On leur oppose les techniques plurifonctionnelles. Ici nous faisons directement le lien avec TEC dans laquelle nous avons mis en évidence le critère de

cardinalité de l'ensemble des fonctions. Les techniques dites plurifonctionnelles correspondent alors aux systèmes dont la cardinalité de l'ensemble des fonctions est supérieur à un. A ceci, nous ajoutons la notion de technique dérivée. Par exemple, nous avons vu dans les Figure 10, Figure 11, et Figure 12 que les trois techniques décrites s'appuient sur le même ensemble de symboles pour la désignation. Seules les relations entre ces entités sont différentes.

Pour les techniques monofonctionnelles, nous notons par exemple, la connexion de deux ConnectTables permet systématiquement l'échange des contenus des deux surfaces. De même, dans DataTiles, le couplage (par un trait) de deux tuiles produit la connexion des données de sorties de l'une des tuiles avec les données d'entrées de l'autre tuile. Du point de vue utilisateur, ces techniques offrent l'avantage d'être faciles à utiliser et par définition prévisibles.

Les techniques plurifonctionnelles s'illustrent bien avec l'exemple avec Synchronous Gesture. Cette technique est dérivée pour obtenir trois fonctions différentes : l'annexion de surface, le partage réciproque d'information, le partage bilatéral et le partage unilatéral. Une technique permettant plusieurs fonctions semble plus intéressante qu'une monofonctionnelle car un seul apprentissage couvre plusieurs utilisations. Cependant, une vigilance s'impose car des erreurs sont alors possibles, par exemple entre choquer deux tablettes ensemble et n'en choquer qu'une avec l'autre. Dernier exemple de technique plurifonctionnelle, le configurateur de surfaces fournit avec les systèmes d'exploitation actuels (Figure 22). Cette technique autorise deux fonctions : par défaut l'extension de la surface principale sur la surface secondaire ou le choix de répliquer l'affichage sur la surface secondaire. La désignation des entités est la même, seul le choix (par checkbox) de la fonction diffère.



**Figure 22**  
**Configurateur de surfaces sous Mac Os X pour les fonctions « extension de la surface d'affichage » ou « mirroring ».**

En synthèse, certaines techniques sont dédiées à une fonction, faciles à apprendre et dont le risque d'erreur est moindre, d'autres, peut-être plus difficiles d'utilisation, présentent l'avantage de produire plusieurs fonctions. En faisant le lien avec les fonctions génériques que nous avons évoqué, nous traitons maintenant des techniques génériques présentant un lien certain avec les techniques plurifonctionnelles.

### 3.2. TECHNIQUES GÉNÉRIQUES

Par définition, une technique générique porte sur une fonction générique ou sur des ressources d'interaction génériques. Une ressource d'interaction générique est par exemple, un dispositif de pointage, et sera instancié en souris ou stylet.

Dans [Holmquist et al. 2001] de petits dispositifs physiques, équipés d'accéléromètres, se connectent lorsqu'ils détectent qu'ils sont agités ensemble. Cette technique générique peut être énoncée ainsi : « Agiter deux ressources d'interaction pour les coupler ». Dans ce même article, deux dispositifs sont utilisés pour fabriquer une alarme lorsque qu'un enfant est trop éloigné de ses parents. Enfants et parents portent des dispositifs qui ont été couplés, indiquant que c'est leur distance relative qui déclenche l'alarme.

De façon similaire, si l'on reprend l'idée de Synchronous Gesture, on construit les techniques suivantes :



1. « Choquer deux ressources d'interaction pour les coupler »,
2. « Choquer deux ressources d'interaction pour échanger leur fonction » (qui hérite de 1),
3. « Choquer deux surfaces pour échanger leur affichage » (qui hérite de 2),
4. « Choquer deux souris pour permuter l'affectation des pointeurs » (qui hérite également de 2).

Cette décomposition apporte un bénéfice certain du point utilisateur, pour lequel l'apprentissage peut-être réduit. Cependant, le premier niveau que nous proposons n'apporte que peu d'intérêt sinon de restreindre cette technique uniquement aux dispositifs mobiles. Pour le développeur, la capitalisation est également indéniable. Dans le cas de Synchronous Gesture, la mise en œuvre du module détectant les gestes synchronisés serait réutilisé, qu'il s'agisse de surfaces ou d'instruments.

#### 4. Synthèse

---

L'analyse de l'état de l'art (chapitre III) fait ressortir un manque de cohérence dans les techniques d'interaction pour le couplage. Aussi nous proposons une notation graphique reposant sur une grammaire pour décrire de manière uniforme les désignations au sein d'une technique d'interaction. Deux types d'entités interviennent dans notre notation : les instruments et l'entité désignée ainsi que les entités intermédiaires intervenant dans un couplage de représentation. Une désignation est représentée par une chaîne débutant par l'utilisateur et se terminant par l'entité désignée. Nous proposons également de mettre en relation, à l'aide de décorations (temporelle, spatiale, ...), les chaînes de désignation.

L'application de notre notation à l'état de l'art fait ressortir plusieurs classes de techniques d'interaction dont chacune présente des avantages et des inconvénients. En comparant les techniques nous avons également mis en évidence la notion de technique plurifonctionnelle/plurifonctionnelle et de technique générique.

---

*Chapitre VI*    *Conclusion*

Mark Weiser, comme la plupart des scénarios fondés sur l'informatique ambiante, insiste sur la puissance qui devrait résulter de l'interaction entre une multitude d'entités mixtes. Pour que cette richesse soit mise à profit de manière utile et sensée, nous pensons qu'il convient de réfléchir à l'un des fondements sous-jacents à toute interaction : le couplage et notamment, pour ce qui concerne l'informatique ambiante, le couplage de ressources d'interaction. Que signifie-t-il dans ce contexte ? Comment l'utilisateur peut-il l'appréhender et le maîtriser ? Qu'en est-il pour le concepteur d'IHM ? Cette thèse a tenté de répondre à ces questions en adoptant une approche analytique ascendante à partir d'un état de l'art foisonnant. Nous résumons ici notre contribution qui appelle à son tour de nouvelles perspectives de recherche.

## *1. Synthèse de la contribution*

---

Notre contribution comprend 6 volets : (a) une définition formelle de la notion de couplage dont nous identifions les composantes clés (entité, fonction et technique d'interaction) et de là, les éléments structurants de notre deuxième contribution (b) TEC, une nouvelle taxonomie. Cette vision descriptive du domaine est complétée par une analyse du comportement dynamique des couplages avec d'une part, (c) la définition d'un automate générique fondé sur des prédicats dénotant notamment la couplabilité et le verrouillage d'entités, et d'autre part les liens de conséquences entre couplages. Le quatrième volet de l'étude est (d) la mise à disposition d'une notation permettant de raisonner de manière abstraite sur les techniques d'interaction que nous illustrons avec (e) Ambient Desktop réalisé au-dessus de I-AM. I-AM, au développement duquel nous avons également contribué (f), est un middleware qui, par couplage dynamique de machines hétérogènes, permet une interaction multi-surfaces, multi-pointeurs, multi-claviers comme si ces ressources d'interaction étaient gérées par une seule machine [Lachenal 2004].

---

### **1.1. DÉFINITION DU COUPLAGE ET TEC : TAXONOMY FOR ENTITY COUPLING**

Les études récentes concernant le façonnage de l'espace sont pour l'instant principalement centrées sur les salles de réunions [Johanson et al 2002][Streitz et al. 1999] ou encore sur les services liés à la localisation (location-aware systems) et l'informatique portée (wearable computing). De nombreux exemples, comme les ConnecTables, s'appuient sur le couplage dynamique d'écrans pour la création d'espaces de travail partagés [Hinckley 2003]. Des travaux tentent d'organiser les problèmes liés au couplage par le biais de taxonomies [Fitzmaurice et al. 1995], [Calvillo-Gamez et al. 2003], [Koleva et al. 2003], [Fishkin 2004], mais

celles-ci sont plutôt centrées sur la fusion des mondes que sur le couplage qui sert de fondement technique à cette fusion. Aussi, nous proposons une définition formelle du couplage et TEC (Taxonomy For Entity Coupling), une nouvelle taxonomie.

**Définition** Nous rappelons qu'un couplage est une application de  $E \times E \times T$  dans  $F$  où

- $E$ , un ensemble d'entités physiques  $P$ , numériques  $N$  ou mixtes  $M$ ,
- $T$ , un ensemble de techniques d'interaction,
- $F$ , un ensemble d'ensembles de fonctions.

Cette définition indique que la spécification de deux entités et d'une technique d'interaction détermine l'ensemble des fonctions obtenues par couplage de ces entités. La technique d'interaction sert à désigner les deux entités à coupler et à contraindre l'ensemble des fonctions cibles.

Parmi les entités, nous identifions les entités physiques, numériques et mixtes. Nous proposons de distinguer les entités mixtes intrinsèques - qui ne nécessitent pas la mise en œuvre d'un couplage préalable, par exemple un PDA - et les entités mixtes construites - qui nécessitent un couplage préalable, par exemple un clavier. Notre définition est représentée au moyen d'un diagramme UML et de contraintes OCL qui expriment la composition d'entités par couplage. Parmi ces entités, nous nous intéressons plus particulièrement aux entités mixtes jouant le rôle de ressource d'interaction.

**TEC** TEC s'appuie sur notre définition du couplage puisqu'elle en reprend les éléments comme axes principaux : entité, fonction, technique d'interaction et couplage proprement dit. Chacun de ces axes est affiné en un ensemble de critères.

De l'application de TEC à l'état de l'art, il ressort une grande diversité d'approches sans toutefois couvrir toutes les options que TEC permet d'envisager. En ce sens, TEC est un modèle génératif. Nous notons également que le caractère générique des couplages est sous-représenté. En effet, peu de solutions permettent de coupler n'importe quelle entité avec n'importe quelle autre. Le caractère dynamique des couplages est également peu abordé comme si l'existence de couplage entre entités passait de manière instantanée de vrai à faux. Pour cette raison, nous proposons un cycle de vie qui rend explicites les étapes clés de l'existence des couplages. Si certains patrons sont identifiables grâce à TEC (par exemple qu'une partie des couplages PN sont liés à la fonction de représentation), on relève également un manque de cohérence entre tous les prototypes concernant les techniques d'interaction, notamment. C'est

pourquoi, nous proposons un formalisme pour raisonner sur les techniques d'interaction de manière unifiée.

---

## **1.2. CADRE D'ANALYSE POUR LA DYNAMIQUE DES COUPLAGES**

Notre cadre d'analyse s'appuie tout d'abord sur un automate régissant le cycle de vie du couplage  $c$  de deux ressources d'interaction  $r_1$  et  $r_2$  noté  $(r_1, c, r_2)$ . Dans cet automate, les états sont décrits à l'aide de trois critères de vérité : couplé/non couplé, verrouillé/non verrouillé, couplable/non couplable. Deux ressources sont couplées si et seulement si l'ensemble des fonctions produit par ce couplage n'est pas vide. Deux ressources sont verrouillées dans un couplage si l'état de l'une des deux ressources ne permet pas de changer l'état du couplage. Le prédicat de couplabilité exprime l'ensemble des autres conditions (autres que verrouillé) à réunir pour qu'un couplage ou un découplage ait lieu. Les transitions nécessaires à la modification de la valeur des trois critères de vérité complètent la structure du cycle de vie que nous proposons.

En appliquant ce cycle de vie à des exemples de l'état de l'art, nous avons mis en évidence certaines propriétés pertinentes comme : le nombre d'états couverts, les actions impactant plusieurs automates, et les automates symétriques. Nous avons également étudié la corrélation des propriétés ergonomiques de souplesse et de robustesse [Grahm et Cockton 1996] avec notre automate. Certaines de ces propriétés telles que l'observabilité et la réversibilité jouent un rôle prépondérant dans le couplage de ressources d'interaction.

En partant du constat que le couplage opportuniste de ressources d'interaction peut engendrer des couplages conséquents, nous proposons un formalisme et des algorithmes détaillant le processus de génération. L'un des éléments clé est la notion de compatibilité. Nous proposons quatre autres pistes pour faciliter la phase de génération : la prise en compte d'entités construites, la cardinalité de la relation de couplage des entités, les fonctions génériques et la notion de nouvelle fonction.

---

## **1.3. TECHNIQUE D'INTERACTION**

Une technique d'interaction pour le couplage se doit de remplir deux rôles : la désignation des deux entités à coupler et la spécification de la fonction. Nous nous sommes principalement intéressés à la phase de désignation des entités car généralement, une technique est dédiée à une fonction. Afin d'analyser les techniques existantes et d'être génératif, nous proposons une notation graphique pour représenter comment se décompose une désignation en termes d'entités (physiques ou numériques) intervenant dans la désignation. Après avoir spécifié la grammaire des désignations, nous avons décrit des désignations de l'état de l'art et découvert des classes de techniques. Notre grammaire nous a permis d'être génératif en montrant qu'un certain nombre de classes n'avait pas de représentant dans les prototypes de l'état de l'art et identifié

des techniques dites plurifonctionnelles. Nous avons également décrit une nouvelle technique provenant de notre mise en œuvre logicielle.

---

#### 1.4. IMPLÉMENTATION

Notre implémentation repose sur I-AM [Lachenal 2004] décrit en Annexe. Il s'agit d'un prototype nommé Ambient Desktop dont la description complète se trouve dans [Barralon et al. 2005]. Trois techniques d'interaction sont réalisées : Click'n'Couple décrite au chapitre V, un configurateur et une technique avec des capteurs matériels décrits en Annexe. Ce prototype est un bureau (Desktop) permettant la manipulation de fichier entre plusieurs machines. Le couplage des écrans est réalisé entre autres par notre technique Click'n'Couple. Basée sur SyncTap [Rekimoto et al. 2003a], elle fonctionne en détectant deux événements « clics » produits par deux instruments de pointage (souris, stylet, doigt suivant la technologie employée) sur les bords de deux écrans. D'après les positions respectives des deux événements, le système en déduit une passerelle entre les deux écrans permettant l'extension de la surface d'affichage des deux écrans et la migration de fichiers d'un bureau à l'autre.

---

## 2. *Limites et perspectives*

Nous présentons maintenant, pour chacune de nos contributions, les limites de nos travaux et les perspectives possibles suivies d'une synthèse.

---

#### 2.1. DÉFINITION DU COUPLAGE ET TEC

La limitation principale que nous voyons à notre définition du couplage est sa limitation à deux entités. L'envie spontanée est de proposer le couplage de  $n$  entités simultanément. Nous envisageons d'explorer cette piste et ses impacts sur le reste de nos travaux : TEC, cadre d'analyse pour la dynamique des couplages et les techniques d'interaction. Toutefois, notre proposition de composition permet de résoudre, au moins sur le plan conceptuel, le cas du couplage à  $n$  entités.

Concernant TEC, que nous avons appliquée à un sous-ensemble des solutions de l'état de l'art, il serait intéressant d'en poursuivre l'application en vue d'identifier de nouveaux patrons, voire d'extraire des heuristiques de conception.

---

#### 2.2. CADRE D'ANALYSE DU COUPLAGE

Dans la définition de notre automate, le critère de vérité « Couplable » désigne « un ensemble de prédicats » que nous ne définissons pas, mais pour lequel nous donnons des exemples au fil du discours. Il conviendrait d'approfondir ces éléments qui peuvent de plus nous être utiles pour

traduire le critère de compatibilité utile à la génération de couplages conséquents. Au regard des dépendances entre cycles de vie, nous avons noté que certaines actions de l'utilisateur pouvaient avoir un impact sur deux cycles de vie. Il existe certainement d'autres types de dépendance et de synchronisation entre cycles de vie. Leur identification permettrait de mieux appréhender la mécanique régissant les couplages. Nous avons appliqué les propriétés IFIP [Grahm et Cockton 1996] à notre automate et avons mis en évidence que certaines sont de premier ordre. L'utilisation d'autres propriétés comme celles de Bastien/Scapin [Scapin et Bastien 1993] ou de Nielsen [Nielsen 1989] pourrait se révéler fructueuse.

Sur le plan de la génération, en l'absence d'exemples concrets dans la littérature, nous ne pouvons pas valider complètement notre proposition et ne possédons surtout pas d'heuristique fiable. L'une des perspectives communes avec notre proposition sur les techniques d'interaction est la mise en place d'expérimentation utilisateur. L'idée serait, pour la génération, d'envisager pour quelques couplages que l'utilisateur mettra en œuvre, une multitude de couplages générés possibles et d'évaluer ceux que l'utilisateur trouve pertinents. L'autre approche est d'observer les couplages qu'un utilisateur (ou des utilisateurs) met en œuvre régulièrement et tenter d'identifier des séquences de couplages que le système pourrait réaliser automatiquement. Avec une étude suffisamment large, des heuristiques pourraient apparaître. Enfin, dans notre algorithme de génération, nous indiquons explicitement que la suppression de couplage n'a pas été prise en compte. Le problème lié à la suppression, est qu'en l'introduisant dans notre algorithme, nous ne garantissons pas que ce dernier converge. Il sera indispensable de résoudre ce problème pour obtenir une solution réellement utilisable.

---

### **2.3. TECHNIQUE D'INTERACTION**

La notation que nous proposons couvre les entités physiques et numériques, qu'il s'agisse de l'instrument de désignation ou de l'entité désignée, et des entités impliquées dans les couplages de représentation. Nous avons choisi de mettre en relation les actions et les décorer de propriétés tel le parallélisme, la colocalisation ou les propriétés CARE. Nous pouvons envisager l'extension de cette décoration à d'autres types de relations. Nous avons également choisi de faire apparaître la propriété d'observabilité (par le biais d'un entourage en pointillés), d'autres propriétés sont certainement intéressantes à faire apparaître, par exemple, la propriété de curabilité appliquée aux actions. De manière générale, nous pourrions appliquer au problème de la désignation, l'ensemble des propriétés vues au chapitre IV.

Ces extensions ne pourront résulter que de l'utilisation régulière de notre notation. Cette utilisation nous permettra de plus de valider que notre

grammaire permet bien de couvrir « toutes » les désignations de l'état de l'art et voir émerger plus de patrons que ceux que nous avons déjà identifiés. Dans l'objectif d'utiliser de façon systématique notre notation, nous proposons le développement d'un logiciel autorisant sa manipulation, la vérification automatique de la validité des chaînes construites par rapport à la grammaire. Ce logiciel participera à la dissémination de la notation en la rendant accessible et sa manipulation plus aisée à d'autres chercheurs.

En se basant sur notre implémentation, nous envisageons la mise en place d'expérimentations utilisateur ayant pour but la création de métriques associées à notre notation. Par exemple, on peut supposer que, plus la chaîne de désignation est longue, plus l'effort cognitif de l'utilisateur est élevé. Ces expérimentations pourraient également aboutir à la création de nouvelles techniques d'interaction.

---

## 2.4. IMPLÉMENTATION

Du point de vue de l'implémentation, en vue des expérimentations utilisateur, nous nous attacherons à étendre la couverture des ressources d'interaction pris en charge par notre système, l'intégration de l'automate du cycle de vie et la prise en compte des couplages PN en tant que vrai couplage. En effet, dans notre implémentation, le curseur d'une souris et la souris physique ne sont pas vus comme un couplage. Nous nous sommes restreints au couplage MM (souris-curseur/souris-curseur).

Le deuxième axe de développement est l'intégration du moteur de génération que nous proposons au chapitre IV. Nous mettons ce développement en second car il implique d'avoir réalisé les points précédents et que notre système couvre un grand nombre de couplages possibles pour pouvoir être génératif.

De manière transversale, il nous semble important d'approfondir la notion de fonction. Nous avons déjà effectué quelques ébauches de proposition dans ce manuscrit, mais une étude plus poussée sur cette notion spécifique ouvrirait des pistes, par exemple, sur la corrélation d'un type de désignation avec un type de fonction ou la découverte d'heuristiques sur la génération de couplage. De façon globale, il semble également intéressant d'intégrer la dimension sociale dans notre étude avec ses impacts sur la génération, sur les techniques d'interaction, mais aussi en tant que prédicat intervenant dans le critère de couplabilité du cycle de vie.



### *3. Synthèse*

---

Ces travaux portent sur l'étude du couplage de ressources d'interaction en informatique ambiante. Coupler deux ressources, c'est l'action de les lier de manière à ce qu'elles opèrent conjointement pour fournir de nouvelles fonctions. Certes, le couplage n'est pas une notion nouvelle puisqu'elle existe en informatique conventionnelle, mais le passage à l'informatique ambiante appelle de nouvelles questions. En réponse, nous proposons une taxonomie qui permet de comprendre l'espace problème, la description du cycle de vie d'un couplage, l'identification des relations entre couplages en vue d'établir une théorie de la composition des couplages, enfin une notation permettant de décrire et d'analyser les techniques d'interaction nécessaires à l'utilisateur pour contrôler le couplage de ressources d'interaction. Nous avons conçu et réalisé une plate-forme logicielle pour tester à terme nos propositions théoriques, voire proposer de nouvelles techniques d'interaction.

Enfin, nous pouvons ici proposer le sujet d'une nouvelle thèse : le « découplage de ressources d'interaction en informatique ambiante », un sujet à part entière à nos yeux.

# *Annexe*

---

---

## 1. *I-AM (Interaction Abstract Machine)*

---

### 1.1. PRINCIPE

I-AM est une machine du même niveau d'abstraction que les gestionnaires graphiques de base comme X window. Ces gestionnaires gèrent les ressources d'interaction d'une station de travail dont ils assurent le partage entre les applications clientes. Comme ces gestionnaires, I-AM est un intergiciel orienté Interaction Homme-Machine, mais I-AM en étend la couverture fonctionnelle pour tenir compte des requis de l'informatique ambiante. En effet, I-AM permet la construction dynamique d'espaces interactifs dont les surfaces et les instruments sont gérés par des stations de travail distinctes exécutant des systèmes d'exploitation éventuellement différents (MacOS, Windows), et donne l'illusion, au programmeur comme à l'utilisateur, que ces ressources sont gérées de manière uniforme par une seule station de travail. Plus spécifiquement, I-AM :

1. Masque l'hétérogénéité du matériel et des systèmes d'exploitation sous-jacents au bon niveau d'abstraction. Cette fonction est assurée en projetant les services d'I-AM dans les termes des systèmes d'exploitation sous-jacents.
2. Intègre les services de découverte dynamique de ressources d'interaction grâce à l'infrastructure de contexteurs [Rey 2005].
3. Modélise les relations spatiales entre les ressources d'interaction (et en déduit une topologie des surfaces). Ce service est assuré grâce aux informations fournies par l'infrastructure de contexteurs.
4. Permet la distribution et la migration des IHM graphiques au niveau du pixel. Ce service est assuré par une fonction de projection (une transformation affine) entre les espaces numériques et la topologie des surfaces.

---

### 1.2. I-AM DESCRIPTION

I-AM se base sur le principe que le programmeur d'application dispose d'une vue unifiée de l'espace d'affichage et crée les interacteurs dans cet espace unifié. I-AM se charge ensuite de réaliser l'association entre cet espace logique et l'espace physique composé à l'exécution par différents écrans. La Figure 1 montre le principe de fonctionnement d'I-AM. Le programmeur exploite des interacteurs logiques dans un espace logique. Le système effectue la correspondance avec les écrans physiques à l'aide d'interacteurs effectifs. La topologie des écrans est déduite par l'exploitation de contexteurs (cf. section suivante) (points verts) fonctionnant sur chaque plateforme et de liaisons (point jaunes) définies à l'exécution entre les écrans.

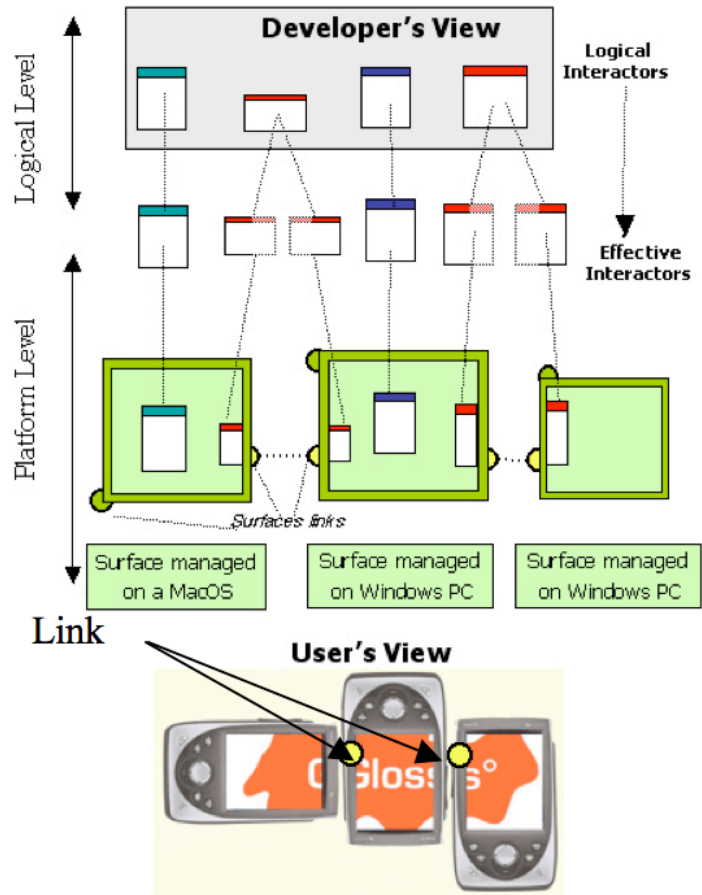


Figure 1  
Principe d'I-AM.

Du point de vue du programmeur, la création d'une fenêtre de base se déroule comme classiquement en programmation objet :

```
// Mon programme est une IAMApp
IAMApp myiamapp = new IAMApp ();

// Creation de mywindow, et mise à jour de la taille et de la
position
IAMWindow mywindow = new IAMWindow();
mywindow.setCenterLocation (300, 300);
mywindow.setSize (300, 200);

// Insert du logo GLOSS
...
// Demande à I-AM de gérer mywindow
myiamapp.addInteractor (mywindow);
```

---

### 1.3. LES CONTEXTEURS

Les contexteurs forment une infrastructure répartie de capture du contexte d'interaction. Ils constituent les travaux de thèse de Gaëtan Rey [Rey 2005]. Cette infrastructure s'auto construit dynamiquement par assemblage d'agents logiciels, les contexteurs, qui, ensemble, fournissent les données contextuelles requises par des processus clients.

Les informations fournies par les contexteurs sont :

- l'arrivée et le départ des entités physiques reliées à chaque machine jouant le rôle de surface,
- la résolution de chaque écran en pixels, sa taille en cm, la taille de chacun de ses bords ainsi que le type (TFT/CRT) de l'écran,
- la détection des relations spatiales entre les surfaces et notamment la proximité de deux surfaces.

---

## 2. *Ambient-Desktop*

Ambient-Desktop offre à l'utilisateur des techniques d'interaction pour configurer à façon son espace de travail. La (re)configuration s'effectue par couplage de surfaces, avec pour conséquence, le couplage de bureaux (au sens desktop des stations de travail usuelles).

Dans cet objectif, nous avons développé une application particulière d'I-AM pour représenter le bureau avec les interacteurs nécessaires, entre autres les icônes pour représenter les fichiers.

Ambient-Desktop se doit de garantir l'observabilité du couplage (l'utilisateur doit pouvoir en évaluer l'état), la curabilité (un couplage non désiré doit pouvoir être défait), la prévisibilité (l'utilisateur doit pouvoir prédire l'effet d'un couplage) et l'honnêteté (l'état observable d'un couplage doit être conforme à l'état interne et doit être compris correctement par l'utilisateur). Dans l'état actuel d'avancement, nous appliquons ces propriétés uniquement au couplage de bureaux (couplages d'écrans) et la gestion des fichiers qui en découle

## 2.1. COUPLAGE D'ÉCRANS

La Figure 2 montre deux plates-formes dont les surfaces sont couplables mais pas encore couplées : des halos bleus entourant chaque écran expriment cet état. Ces halos indiquent que chaque écran est un domaine fermé : sur chaque surface, l'espace d'interaction est « fermé », restreint à cette surface. Le fait que les halos sont de couleur identique indique que les surfaces sont couplables.



**Figure 2**  
**Un espace d'interaction dont les deux surfaces sont couplables.**

Dans l'état d'avancement actuel du développement, la fonction obtenue par couplage de surface est l'extension de la surface d'affichage. Dans l'exemple de la Figure 2, toutes les surfaces disponibles sont couplables. Mais on peut imaginer d'autres fonctions de couplage, par exemple la fonction « extension de la surface d'affichage par classe de dispositifs ». Dans ce cas, tous les écrans couplables de stations de travail auraient par exemple des bordures bleues, ceux des PDA couplables auraient des bordures rouges alors que les murs interactifs couplables auraient des bordures vertes.

Ambient-Desktop offre trois techniques d'interaction de couplage de surfaces : (1) Click'n'Couple, (2) Couplage par proximité, (3) Couplage via un configurateur.

### Click'n'Couple

Puisque nous avons déjà abordé la description de notre technique Click'n'Couple dans le chapitre V, nous ne montrons ici que quelques limites.

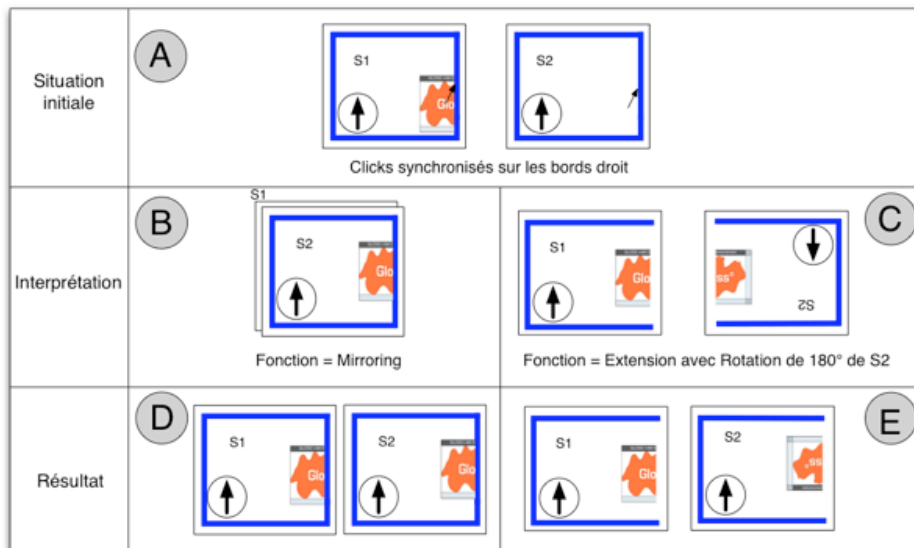


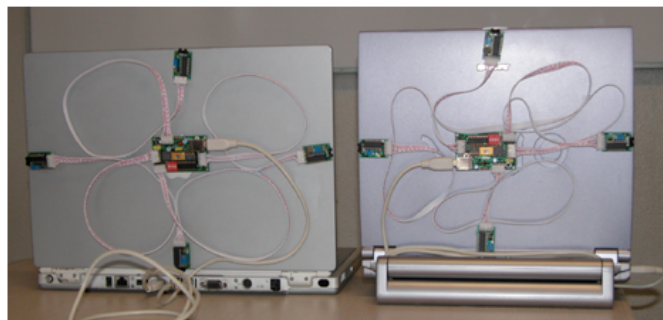
Figure 3  
Problèmes d'interprétation avec le Click'n'Couple.

I-AM permettant la rotation des surfaces, la technique du Click'n'Couple peut conduire à des situations ambiguës. On considère deux surfaces S1 et S2, côte à côte comme le montre la Figure 3A. Deux clics sur les bords droits de deux surfaces peuvent engendrer deux interprétations possibles : la fonction « mirroring » (Figure 3B) ou en « extension de la surface d'affichage » (Figure 3C) avec rotation d'une des deux surfaces. Sur la Figure 3, les flèches à l'intérieur des écrans indiquent leur orientation relative. Ainsi en B, on constate une rotation de 180° de la surface S2 par rapport à S1. Suivant la fonction retenue par le système (mirroring ou extension), l'affichage sur S1 et S2 sera différent. Le cas D représente l'application de la fonction mirroring de S1 sur S2 ; le cas E de la fonction d'extension avec rotation. On constate dans ce dernier cas que l'affichage est erroné par rapport à l'agencement spatial physique des écrans. On rencontre ici les limites de la technique Click'n'Couple si l'on n'associe pas une fonction à une technique.

### Couplage par proximité de surfaces

La deuxième technique possible est l'utilisation des détecteurs de proximité développés dans notre équipe de recherche et qui équipent le dos des écrans (Figure 4). Pour coupler, l'utilisateur doit rapprocher physiquement les deux écrans et aligner deux modules satellites. Le couplage des surfaces équipées est immédiatement réalisé. La prise en compte ou non des bords physiques des écrans est à configurer. Contrai-

rement à la technique Click'n'Couple. La position physique des capteurs ainsi que leur champ d'action n'autorise que la création de mosaïques de surfaces comme les DataTiles [Rekimoto et al. 2001] ; deux surfaces ne peuvent se superposer. Le découplage s'effectue simplement par l'éloignement des deux surfaces. On se place ici clairement dans l'interaction proximale [Rekimoto et al. 2003b] dont peuvent se passer le Click'n'Couple et le configurateur. En effet, ces deux techniques permettent de simuler la proximité sans pour autant que les surfaces soient physiquement en contact. Nous noterons cependant qu'il est peu envisageable de réaliser un Click'n'Couple si les surfaces sont trop éloignées.

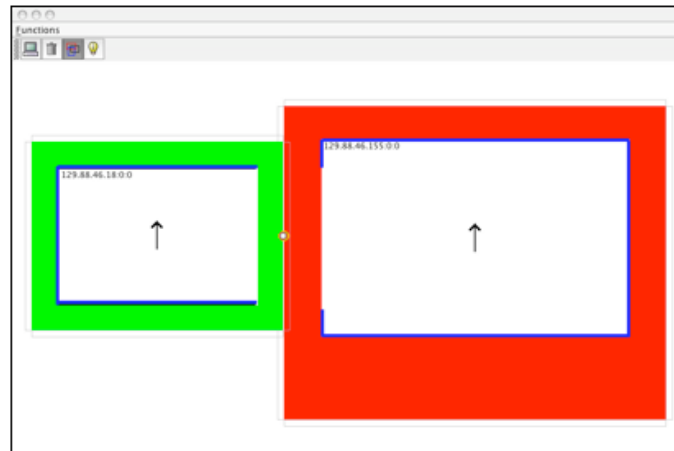


**Figure 4**  
Portables équipés de notre capteur de proximité.

---

**Configurateur** Cette dernière technique utilise une IHM semblable, sur certains points, à ceux que l'on trouve pour l'ajout d'un écran sur la plupart des systèmes d'exploitation.





**Figure 5**  
**Configurateur avec deux surfaces, leurs bords physiques et une liaison.**

Le configurateur dialogue avec I-AM pour détecter l'arrivée et le départ de surfaces. Tout comme les configurateurs actuels, la manipulation directe des écrans permet de gérer leur arrangement. Nous avons ajouté les fonctionnalités suivantes :

- Rotation des écrans. Notre système pouvant être utilisé en situation mobile, les écrans mis en jeu pour le couplage de surfaces peuvent ne pas être orientés de la même façon, comme les ConnectTables [Tandler et al. 2001]. Ainsi les flèches au centre de chaque écran permettent de visualiser les orientations relatives des écrans.
- Visualisation des bords physiques. Lachenal [Lachenal 2004] et Baudisch [Baudisch et al. 2004] illustrent les problèmes liés à la présence de bords physiques aux écrans. Notre configurateur prend en compte cette information pour rendre la représentation de la topologie relative des surfaces aussi réaliste que possible (Figure 5).
- Visualisation des frontières des surfaces. La représentation des surfaces inclut une mise à jour en temps réel de la représentation des frontières sur chaque surface. Ainsi, si une liaison est créée et deux surfaces couplées, l'utilisateur peut anticiper la création des frontières. Les conséquences du couplage deviennent plus « prévisibles ».
- Création automatique des liaisons associées à l'arrangement en cours. Le configurateur s'appuie sur la simulation de la proximité des surfaces. La zone d'influence autour des bords des écrans est matérialisée et la création automatique des liaisons n'a lieu que si deux surfaces sont suffisamment proches dans le configurateur.

- Création automatique de liaisons prédéfinies. Lors de la phase de développement, nous avons constaté que nous créions systématiquement toujours le même ensemble de liaisons. Par exemple au bureau avec son portable et sa station de travail, le configurateur en mode « daemon » peut être paramétré de façon à recréer deux liaisons données lorsque les deux surfaces sont présentes.

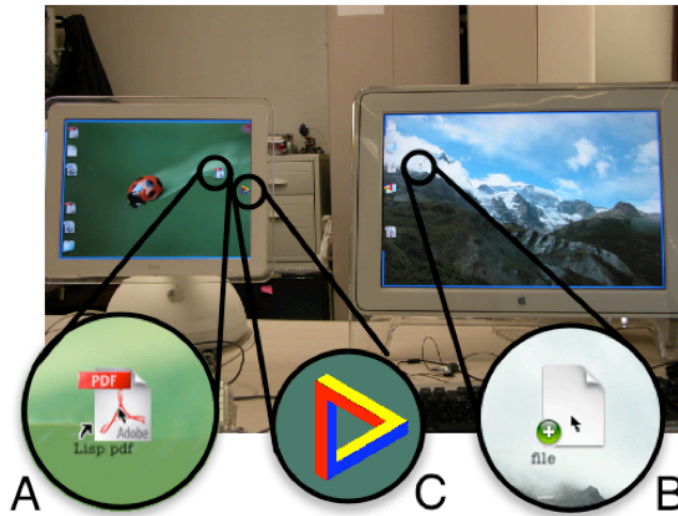
De façon globale, notre système utilise l'infrastructure des contexteurs avec deux capteurs logiciels (Click'n'Couple et le configurateur) et un capteur matériel (capteur de proximité). Cette phase de couplage de surfaces est exploitée par Ambient-Desktop pour fournir des fonctions de gestion de fichiers.

---

## 2.2. GESTION DE FICHIERS

Le couplage de deux surfaces crée un espace d'interaction unifié comme si les deux surfaces étaient gérées par une seule machine. Or si dans le cas usuel d'ajout de surfaces un seul système de gestion de fichiers (SGF) est concerné, notre système permet d'en faire cohabiter plusieurs. Comme le montre la Figure 6, deux bureaux se partagent le même espace unifié. Par manipulation directe des icônes, plusieurs fonctions sont possibles :

- Réorganisation d'icônes. Au sein d'une même surface, les icônes peuvent être réarrangées spatialement, comme avec n'importe quel bureau traditionnel, mais potentiellement par plusieurs souris simultanément.
- Migration d'icônes. La migration d'une icône d'une surface à l'autre provoque l'apparition (sur l'icône) d'une flèche de raccourci (Figure 6). Celle-ci indique que le fichier représenté par l'icône ne se trouve pas sur le SGF relatif à la surface d'affichage.
- Copie de fichiers. Lors de la migration d'icônes, si l'utilisateur maintient enfoncée la touche shift du clavier couplée avec la souris (qui sert pour la manipulation), alors la migration se transforme en copie de fichiers. Un signe « + » apparaît sur l'icône pour symboliser la copie (Figure 6B).



**Figure 6**  
**Surfaces couplées.**

A) Migration d'une icône. B) Copie d'un fichier. C) Feedback de couplage.

Ces fonctions sont offertes par les systèmes traditionnels. Elles sont ici étendues à l'interaction multisurface, multi-instrument et rendues possible par Ambient-Desktop.

### 3. Synthèse

Ambient-Desktop se présente comme une solution permettant, grâce aux capteurs de proximité de réaliser simplement et rapidement le couplage de deux surfaces. Nous proposons également deux solutions alternatives : Click'n'Couple et un configurateur. Ces trois techniques permettent l'utilisation d'Ambient-Desktop en situation mobile autorisant la création opportuniste d'espaces d'interaction par couplage. Au sein de ces espaces, l'utilisateur peut, par exemple, réaliser des copies de fichiers ou la manipulation d'applications dont les IHM graphiques peuvent migrer ou être distribuées.

# *Bibliographie*

---



- 
- [ACCORD]                   Projet RNTL, Assemblage de Composants par Contrats en environnement Ouvert et Réparti, disponible à : <http://www.infres.enst.fr/projects/accord/>
- [Allen et al. 1994]       Allen, J.F., Ferguson, G. Actions and Events in Temporal Logic, *Journal of Logic and Computation*, 1994.
- [Anderson et al. 2000]   Anderson, D., Frakel, J.L., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryal, K., Sullivan, E., Yedidia, J.S. Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modeling, in *Proc. Of SIGGRAPH 2000*, New Orleans, 2000, pp. 393-402.
- [Ballagas et al. 2003]   Ballagas, R., Meredith, R., Stone, M., Borchers, J. iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments, in *Proc. Of CHI 2003*, Ft. Lauderdale, Florida, 2003, pp. 537-544.
- [Barralon et al. 2004]   Barralon, N., Lachenal, C., Coutaz, J. Couplage de ressources d'interaction. In *Proceedings of IHM'04, International Conference Proceedings Series*, Namur, Belgium, 2004, ACM Press, pp.13-20.
- [Barralon et al. 2005]   Barralon, N., Nguyen, V.T., Rey, G. Techniques de couplage de bureaux : Ambient-Desktop comme illustration. In *Proceedings UbiMob'05*, Grenoble, 31 mai-3 juin 2005, pp. 193-200.
- [Baudisch et al. 2004]   Baudisch, P., Cutrell, E., Hinckley, K., and Gruen, R. Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays. In *CHI 2004 Extended Abstracts (short paper)*, Vienna, 2004, pp. 1379-1382.
- [Beaudouin-Lafon 2000]   Beaudouin-Lafon, M., Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces, *Avril 2000, CHI 2000 Proceedings*, p.446-453.
- [Bellotti et al. 2002]   Bellotti, V., Back, M., Edwards, K., Grinter, R., Henderson, A., Lopes, C. Making Sense of Sensing Systems: Five Questions for Designers and Researchers, in *Proc. of CHI'02*, New York, 2002, pp. 415-422.
- [Bérard 2003]            F. Bérard, The Magic Table: Computer-Vision Based Augmentation of a Whiteboard for Creative Meetings, in *IEEE workshop on Projector-Camera Systems*, Nice, France, 2003.
- [Block et al. 2004]       Block, F., Schmidt, A., Villar, N., Gellersen, H.W. Towards a Playful User Interface for Home Entertainment Systems, in *Proc. of EUSAI'04*, Eindhoven, Netherlands, 2004, pp. 207-217.
- [Borkowski et al. 2004]   Borkowski, S., Letessier, J., Crowley, J.L. Spatial Control of Interactive Surfaces in an Augmented Environment, *Engineering Human Computer Interaction and Interactive Systems, Joint Working Conferences EHCI-DSVIS 2004*, jul 2004, pp.228-244
- [Borkowski 2006]         Borkowski, S. Interfaces pilotables pour l'interaction homme machine dans des environnements interactifs, *these de l'Institut National Polytechnique de Grenoble*, juin 2006.

- 
- [Bowman et Hodges 1999] Bowman, D.A., Hodges, L.F. Formalizing the Design, Evaluation, and Application of Interaction Techniques for Immersive Virtual Environments. *Journal of Visual Languages and Computing*, 10, 1999, pp.37-53.
- [Breu et al. 1997] Breu, R., Hinkel, U., Hofmann, C., Klein, C., Paech, B., Rumpe, B., Thurner, V., Towards a Formalization of the Unified Modeling, *Lecture Notes in Computer Science*, 1997.
- [Calvillo-Gamez et al. 2003] Calvillo-Gamez, E.H., Leland, N., Shaer, O., Jacob R. The TAC Paradigm: Unified Conceptual Framework to Represent Tangible User Interfaces, in *Proc. of CLIHC'03*, 2003.
- [Card et al. 1990] Card, S.K., Mackinlay, J.D., Robertson, G. The design space of input devices, in *Proceedings of the SIGCHI conference on Human factors in computing*, Seattle, Washington, United States, 1990, pp.117 - 124
- [Christopoulou et Kameas 2004] Christopoulou, E., Kameas, A. Using Ontologies to Address Key Issues in Ubiquitous Computing Systems, in *Proc. of EUSAI'04*, Eindhoven, Netherlands, 2004, pp. 13-24.
- [Coutaz et al. 1995] Coutaz, J., Nigay, L., Salber, D., Blandford, A. May, J., Young, R. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE properties, *Proceedings of the INTERACT'95 conference*, S. A. Arnesen & D. Gilmore Eds., Chapman&Hall Publ., Lillehammer, Norway, Juin 1995, pp. 115-120
- [Dragicevic 2004] Dragicevic, P. Un modèle de configurations d'entrée pour des systèmes interactifs multi-dispositifs hautement configurables, thèse de l'Université de Nantes, 2004, 271 pp .
- [Fishkin 2004] Fishkin, K.P. A taxonomy for and analysis of tangible interfaces, in *Personal and Ubiquitous Computing archive Volume 8 , Issue 5, September 2004*, pp. 347 - 358
- [Fitzmaurice et al. 1995] Fitzmaurice, G., Ishii, H., Buxton, W. Bricks: Laying the foundations for graspable user interfaces, in *Proc. of CHI'95*, New York, 1995, pp. 432-449.
- [Fitzmaurice et al. 2003] Fitzmaurice, G., Kan, A., Buxton, W., Kurtenbach, G., Balakrishnan, R., Sentient Data Access via a Diverse Society of Devices, *ACM Queue: Tomorrow's Computing Today*, Volume 1, Number 8, 2003, pp. 52-62.
- [Foley et al. 1990] Foley, J.D., Van Dam, A., Feiner, S.K., Hughes, J.F. *Computer Graphics Principles and Practice*. Second Edition, Addison-Wesley Publishing Company, 1990.
- [Gamma et al. 1995] Gamma, E., R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Addison-Wesley*, 1995.
- [Goldberg et Robson 1984] Goldberg, A., Robson, D. *Smalltalk-80 : The Interactive Programming Environment*; Addison-Wesley Publ., 1984.

- 
- [Gorbet et al. 1998] Gorbet, G.M., Orth, M., Ishii, H. Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography, in Proc. of CHI'98, Los Angeles, 1998, pp. 49-56.
- [Grahm et Cockton 1996] Grahm, Ch., Cockton, G. Design Principles for Interactive Software. Chapman & Hall, London, 1996.
- [Hightower et Boriello 2001] Hightower, J., Borriello, G. Location Systems for Ubiquitous Computing. IEEE Computer, 34(8), 2001, pp. 57-66.
- [Hinckley 2003] K. Hinckley, Synchronous gestures for multiple persons and computers, in Proc. of UIST'03, Vancouver, Canada, 2003, pp. 149-158.
- [Holmquist et al. 1999] Holmquist, L.E., Redström, J., Ljungstrand, P. Token-Based Access to Digital Information, in Proc. of HUC, Karlsruhe, Germany, 1999, pp. 234-245.
- [Holmquist et al. 2001] Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.W. Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, in Proc. of Ubicomp 2001, Atlanta, Georgia, 2001, pp. 116-221.
- [Ishii et Ullmer 1997] Ishii, H., Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, in Proc. of CHI'97, Atlanta, Georgia, 1997, pp.234-241.
- [Izadi et al. 2003] Izadi, S., Brignull, H., Rodden, T., Rogers, Y., Underwood, M. Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media, in Proc. of UIST'03, Vancouver, Canada, 2003, pp. 159-168.
- [Johanson et al 2002] Johanson, B. Fox, A. Winograd, T. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing Magazine 1(2), April-June 2002
- [Koleva et al. 2003] Koleva, B., Benford, S., Ng, K.H., Rodden, T. A Framework for Tangible User Interfaces, in Workshop in Mobile HCI'03, Udine, Italy, 2003.
- [Lachenal 2004] Lachenal, C. Modèle et Infrastructure Logicielle pour l'Interaction multi-instrument, multi-surface. Thèse de l'Université Joseph Fourier, décembre 2004.
- [Lecolinet 2003] Lecolinet, E. A molecular architecture for creating advanced GUI. Proceedings of the 16th annual ACM symposium on User interface software and technology. Vancouver, Canada. pp. 135-144.
- [Lyytinen et Yoo 2002] Lyytinen, K. and Yoo, Y. Issues and Challenges in Ubiquitous Computing. Communications of the ACM, 45(12), 2002, pp. 62-65.
- [Myers 2002] Myers, B. Mobile Devices for Control, in Proc. of the 4th International Symposium on Mobile Human-Computer Interaction, Springer-Verlag, 2002, pp. 1-8.
- [Nabaztag] Violet, the smart Object Company, Disponible à : <http://www.violet.net/> et <http://www.nabaztag.com/>.
-



- 
- [Nielsen 1989] Nielsen, J. Usability engineering at a discount, in Salvendy, G., and Smith, M.J. (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*, Elsevier Science Publishers, Amsterdam, 1989,394-401.
- [Norman 1986] Norman, D., "User Central System Design", Lawrence Erlbaum, 1986.
- [Patten et al. 2001] Patten, J., Ishii, H., Hines, J., Pangaro, G. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces, in *Proc of CHI'01*, Seattle, Washington, 2001, pp. 253-260.
- [Pinhanez 2001] Pinhanez C., The everywhere displays projector: A device to create ubiquitous graphical interface, in *Proceedings of Ubiquitous Computing 2001 Conference*,September 2001.
- [Piper et al. 2002] Piper, B., Ratti, C., Ishii, H. Illuminating Clay: A Tangible Interface with potential GRASS applications, GRASS User's Conference Trento, Italy, 11-13 September 2002.
- [Rekimoto 1996] Rekimoto, J. Tilting operations for small screen interfaces, in *Proceedings of the 9th annual ACM symposium on user interface software and technology (UIST'96)*, Seattle, Washington, November 1996, pp 167-168.
- [Rekimoto et Saitoh 1999] Rekimoto, J., Saitoh, M. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments, in *Proc. of CHI'99*, Pittsburg, Pennsylvania, 1999, pp. 378-385.
- [Rekimoto et al. 2001] Rekimoto, J., Ullmer, B., Oba, H. DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions, in *Proc. of CHI'01*, Seattle, Washington, 2001, pp. 269-276.
- [Rekimoto et al. 2003a] Rekimoto, J., Ayatsuka, Y., Kohno, M. SyncTap: An Interaction Technique for Mobile Networking, in *Proc. of MOBILE HCI'03*, Udine, Italy, 2003, pp.104-115.
- [Rekimoto et al. 2003b] J. Rekimoto, Y. Ayatsuka, M. Kohno, H., Oba, Proximal Interactions: A Direct Manipulation Technique for Wireless Networking, in *Proc. of Interact'03*, Zürich, 2003, pp.511-518.
- [Renevier 2004] Renevier, P. *Systèmes Mixtes Collaboratifs sur Supports Mobiles : Conception et Réalisation*. Thèse de l'Université Joseph Fourier, Juin 2004.
- [Resnick et al. 1998] M. Resnick, F. Martin, R. Berg, R. Borovoy, V. Colella, K. Kramer, B. Silverman, *Digital Manipulatives: New Toys to Think With*, in *Proc. of CHI'98*, Los Angeles, 1998, pp. 281-287.
- [Rey 2005] Rey, G. *Contexte en Interaction Homme-Machine : le contexteu*, thèse de l'Université Joseph Fourier, 1er aout 2005, 186 pages
- [Scapin et Bastien 1993] Bastien, J.M.C., Scapin, D.L. *Critères Ergonomiques pour l'Évaluation d'Interfaces Utilisateurs*, Technical report 1993. INRIA. 1993.

- 
- [Siegemund et Krauer 2004] Siegemund, F., Krauer, T. Integrating Handhelds into Environments of Cooperating Smart Everyday Objects, in Proc. of EUSAI'04, Eindhoven, Netherlands, 2004, pp. 160-171.
- [Small et Ishii 1997] Small, D., Ishii, H. Design of spatially aware graspable displays. In: Extended abstracts of the CHI'97 conference on human factors in computing systems, Atlanta, Georgia, March 1997, pp 367-368.
- [Streitz et al. 1999] Streitz, N., Geibler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., Steinmetz, R. i-Land: An interactive Landscape for Creativity and Innovation. CHI 1999, pp. 120-127.
- [Sutcliffe et al. 2004] Sutcliffe, A., Mehandjiev & N. End-User Development. Communication of the ACM, special Issue on End-User Development, ACM publ., sept. 2004.
- [Tandler et al. 2001] Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., Steinmetz, R. ConnecTables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces, in Proc. of UIST'01, Orlando, Florida, 2001, pp. 11-20.
- [Thevenin et al. 1999] Thevenin, D., Calvary, G., Coutaz, J. La Plasticité en interaction Homme-Machine, RJCiHM2000, île de Berder, 3-5 Mai 2000.
- [Ullmer et al. 1998] Ullmer, B., Ishii, H., Glas, D. mediaBlocks: Physical Containers, Transports, and Controls for Online Media, in Proc. of SIGGRAPH'98, Orlando, Florida, 1998, ACM Press, pp. 379-386.
- [Ullmer and Ishii 2000] Ullmer, B., Ishii, H. Emerging frameworks for tangible user interfaces, IBM Systems Journal, Vol. 39, Nos 3&4, 2000, pp. 915-932.
- [Warmer et Kleppe 2003] Warmer, J.B., Kleppe, A.G. The object constraint language, Addison Wesley, 2003.
- [Wisneski et al. 1998] Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P. Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information, in Proc. of CoBuild'98, 1998, pp. 22-32.
- [Weiser 1991] Weiser, M. The computer for the 21st century. Scientific American, September 1991, pp. 94-104.

