



Co-located collaboration in interactive spaces for preliminary design

Alistair Jones

► To cite this version:

Alistair Jones. Co-located collaboration in interactive spaces for preliminary design. Other. Université de Technologie de Compiègne, 2013. English. NNT : 2013COMP2114 . tel-01067774

HAL Id: tel-01067774

<https://theses.hal.science/tel-01067774>

Submitted on 24 Sep 2014

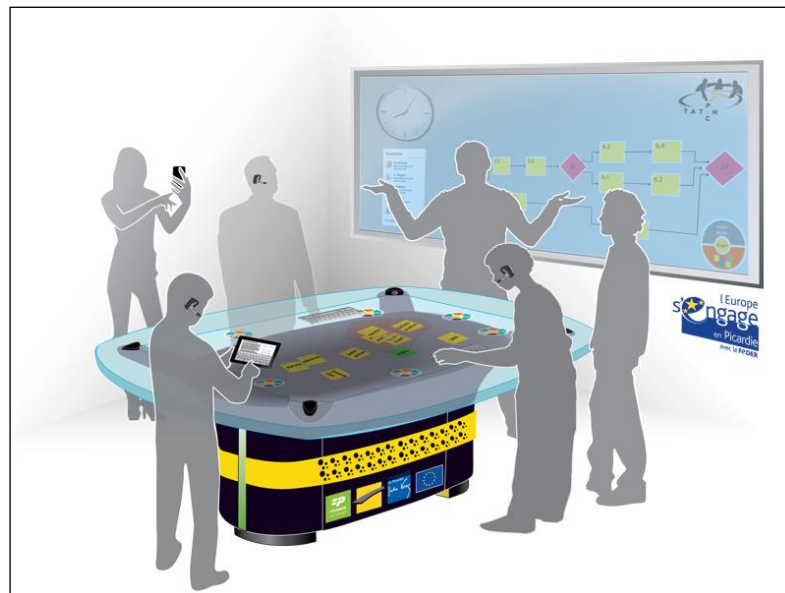
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

Par **Alistair JONES**

Co-located collaboration in interactive spaces for preliminary design

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 05 décembre 2013

Spécialité : Technologies de l'Information et des Systèmes

D2114

THÈSE PRÉSENTÉE POUR L'OBTENTION DU GRADE DE
DOCTEUR DE L'UTC

Spécialité : Technologie de l'Information et des Systèmes

CO-LOCATED COLLABORATION IN
INTERACTIVE SPACES FOR
PRELIMINARY DESIGN



Alistair JONES
alistair.jones@utc.fr

Co-located Collaboration in Interactive Spaces for Preliminary Design

Thèse soutenue le 5 décembre 2013 devant un jury composé de:

J.P BARTHÈS	Professeur Émérite Laboratoire Heudiasyc, UTC <i>Co-directeur de thèse</i>
J. EAGAN	Maître de Conférences Département INFRES, Télécom ParisTech
T. GIDEL	Maître de Conférences Laboratoire Costech, UTC
G. KASSEL	Professeur des universités Laboratoire MIS, UPJV
C. KOLSKI	Professeur des universités Laboratoire LAMIH, Université de Valenciennes <i>Rapporteur</i>
D. LENNE	Professeur des universités Laboratoire Heudiasyc, UTC <i>Co-directeur de thèse</i>
C. MOULIN	Maître de Conférences Laboratoire Heudiasyc, UTC
M. OCCELLO	Professeur des universités Laboratoire LCIS, Université de Grenoble <i>Rapporteur</i>

Thèse réalisée au Laboratoire Heudiasyc
UMR CNRS 7253
Université de Technologie de Compiègne

Sous la direction de Jean-Paul BARTHÈS jean-paul.barthes@hds.utc.fr
Dominique LENNE dominique.lenne@hds.utc.fr

Financement Allocation de recherche cofinancée par l'Union européenne et la région Picardie. L'Europe s'engage en Picardie avec le Fonds européen de développement régional (FEDER).



**Collaboration co-localisée dans un
espace interactif pour la conception
préliminaire**

Résumé

La phase de conception préliminaire est déterminante lors de la réalisation d'un projet industriel. Elle exploite généralement des outils méthodologiques tels que le brainstorming, l'analyse causale et le chronogramme, qui permettent la collaboration entre des participants aux compétences et aux approches différentes. Ces activités se déroulent dans des salles de réunions traditionnelles, autour d'une table ou devant un tableau blanc, avec l'aide de nombreux papiers et Post-it, ce qui rend la préparation, l'exécution, et l'exploitation de ce processus particulièrement difficile. Jusqu'à présent, cette phase de conception préliminaire a résisté à la numérisation, notamment parce que l'addition d'un dispositif informatique au sein de ces activités perturbe la communication et la collaboration naturelles entre participants.

Au cours des dix dernières années, de nombreuses avancées technologiques ont été réalisées en ce qui concerne les dispositifs numériques tels que les tables et les tableaux interactifs, les smartphones et les tablettes tactiles. La similarité des configurations physiques de ces dispositifs avec les dispositifs plus traditionnels permet d'exploiter les capacités préexistantes des utilisateurs (l'habileté motrice, le raisonnement spatial, le langage parlé, etc.). Les chercheurs se basent sur l'hypothèse que ces nouveaux dispositifs, travaillant de concert au sein d'espaces interactifs, pourront augmenter la collaboration co-localisée pour les équipes de conception préliminaire.

L'objectif de cette thèse est, d'une part, d'étudier la conception d'un espace interactif pour la collaboration co-localisée durant la phase de conception préliminaire, et d'autre part, de proposer une architecture permettant de réunir les dispositifs hétérogènes et distribués composant cet espace.

La première contribution consiste en une présentation détaillée d'un espace interactif utilisant une configuration physique encore peu exploitée dans la littérature scientifique : une table et un tableau multi-tactiles de grandes dimensions. La conception de cet espace interactif a été basée sur des observations d'utilisateurs dans un contexte de conception préliminaire traditionnel et sur une revue de la littérature visant à identifier des principes de conception. Lors de la conception de cet espace, une attention particulière a été portée à l'interface utilisateur qui

s'étend sur des écrans partagés et qui maintient une séparation entre les activités d'un participant et les activités d'une équipe. Enfin, les évaluations, réalisées avec des groupes de cinq à six participants, démontrent une amélioration dans l'exploitation des outils méthodologiques sur supports numériques par rapport à une utilisation traditionnelle lors de la conception préliminaire.

La conception d'une infrastructure distribuée basée sur un système multi-agents constitue la deuxième contribution de cette thèse. Cette infrastructure parvient à rassembler de nombreuses plateformes et des dispositifs hétérogènes. Elle représente une solution intéressante pour les espaces interactifs, en particulier parce qu'elle tolère particulièrement bien la défaillance de réseau et permet un prototypage rapide des dispositifs.

Mots clés : espace interactif, environnement multi-surfaces, collaboration co-localisée, systèmes multi-agents, informatique distribuée, interaction multimodale

**Co-located collaboration in
interactive spaces for preliminary
design**

Abstract

The *preliminary design* phase occurs near the launch of an engineering project, normally after an initial requirements gathering phase. Through a series of meetings which gathers the key actors of a project, effective preliminary design involves discussion and decision-making punctuated by group creativity techniques. These activities are designed to explore the potential solutions of the problem, such as brainstorming or causal analysis, or to address the project itself, such as collaborative project planning. Such activities are usually conducted in traditional meeting rooms with pen and paper media, which requires significant time and effort to prepare, perform, and later render into a digitally exploitable format. These processes have resisted previous attempts of computer-supported solutions, because any additional instruments risk obstructing the natural collaboration and workflow that make these activities so beneficial.

Over the past decade, technologies such as interactive tabletops, interactive wall displays, speech recognition software, 3D motion sensing cameras, and handheld tablets and smartphones have experienced significant advances in maturity. Their form factors resemble the physical configuration of traditional pen-and-paper environments, while their “natural” input devices (based on multi-touch, gestures, voice, tangibles, etc.) allow them to leverage a user’s pre-existing verbal, spatial, social, motor and cognitive skills. Researchers hypothesize that having these devices working in concert inside *interactive spaces* could augment *collaboration* for *co-located* (i.e. physically present) groups of users.

There currently exist several interactive spaces in the literature, illustrating a wide range of potential hardware configurations and interaction techniques. The goal of this thesis is first to explore what qualities these interactive spaces should exhibit in their interaction design, particularly with regard to preliminary design activities, and second, to investigate how their heterogeneous and distributed computing devices can be unified into a flexible and extensible distributed computing architecture.

The first main contribution of this thesis is an extensive presentation of an interactive space, which at its core uses a configuration not yet fully explored in previous literature: a large multitouch tabletop and a large multitouch interactive

board display. The design of this interactive space is driven by observations of groups engaged in preliminary design activities in traditional environments and a literature review aimed at extracting user-centered design guidelines. Special consideration is given to the user interface as it extends across multiple shared displays, and maintains a separation of concerns regarding personal and group work. Finally, evaluations using groups of five and six users show that using such an interactive space, coupled with our proposed multi-display interaction techniques, leads to a more effective construction of the digital artifacts used in preliminary design.

The second main contribution of this thesis is a multi-agent infrastructure for the distributed computing environment which effectively accommodates a wide range of platforms and devices in concerted interaction. By using agent-oriented programming and by establishing a common content language for messaging, the infrastructure is especially tolerant of network faults and suitable for rapid prototyping of heterogeneous devices in the interactive space.

Keywords: interactive spaces, multi-surface environments, co-located collaboration, multi-agent systems, distributed computing, multimodal interaction.

Remerciements

First, I would like to thank my directors, Jean-Paul Barthès and Dominique Lenne, for their encouragement, support, humor, and patience, while I explored vast amount of theory and design, and for correcting so many grammar and spelling mistakes (in French and in English!). I would also like to thank the lead researchers on the TATIN-PIC project, Claude Moulin and Thierry Gidel, for all the hours of conversation and precious input they provided into my thesis. Additionally, if you are wondering why the interactive tabletop and interactive whiteboard that are featured in my research work appear to be so well done, I recommend speaking to Atman Kendira who was responsible for building these devices from scratch. He gave me considerable support throughout the entire thesis, and even created some graphics used in this dissertation, such as the one of the cover.

I would like to say thank you to Yvan Duhamel for not only his work on the tabletop and the whiteboard, but also for repairing my bike many times; to Andrea Guerra for his general inquisitiveness and also for all his invaluable assistance during the user evaluations with the tabletop; to Fouzia Harant for all of her administrative support and for her interminable cheer around the office; to Pierre-Yves, Margot, and Camille, my siblings of the same thesis advisors, for their support and friendship, which I will miss; to Guillaume, Justine, Gael and anyone else I had the privilege of sharing an office with; to the students who I worked with on the TATIN-PIC project for their interest and their investment; to James Eagan for his mentorship and his friendship; and to all the doctorants of the Heudiasyc and Costech laboritory for their camaraderie.

I'd like to thank the home front as well, for making the past three years so enjoyable. First of all, to Louise, for her charm and charisma, which soothed me when I needed reprieve and animated me when I needed encouragement; to William, my older brother, for being an older brother whom I'm lucky to have; to my family in the old continent who I had the chance to know better through these past few years in Europe, to the lifelong friends I have made in France, to all of you who corrected my French, shared a beer, or bought me a cigar, I'm thinking of all of you; and finally to my parents, to whom I dedicate this thesis, for all their love and generosity.

Related Publications

- [Joiron 13b] Joiron, Céline, Frédéric Fürst, Gilles Kassel, Alistair Jones, Jean-Paul Barthès, Claude Moulin, and Dominique Lenne. 2013. “An Ontology for Designing a Collaborative Platform Involving a Multi-touch and Multi-user Interactive Table.” In Proceedings of the 5th International Conference of Knowledge Engineering and Ontology Development (KOED '13). pp. 156–163. Vilamoura, Portugal
- [Joiron 13a] Joiron, Céline, Frédéric Fürst, Gilles Kassel, Alistair Jones, Jean-Paul Barthès, Claude Moulin, and Dominique Lenne. 2013. “Une Ontologie Pour La Conception D’une Plateforme Collaborative Exploitant Une Table Multi-tactile et Multi-modale.” In 24èmes Journées Francophones d’Ingénierie Des Connaissances (IC’2013). Lille, France.
- [Guerra 13] Guerra, Andrea, Thierry Gidel, Atman Kendira, Enrico Vezzetti, and Alistair Jones. 2013. “Co-evolution of Design Tactics and CSCWD Systems: Methodological Circulation and the TATIN-PIC Platform.” In 19th International Conference on Engineering Design - ICED13.
- [Kendira 13] Kendira, Atman, Alistair Jones, Thierry Gidel, and Dominique Lenne. 2013. “Verbal and Nonverbal Communication for Evaluating Interactive Spaces.” In International Conference on Human-Computer Interaction (Work-in-Progress), CHI EA’13, pp. 319-324.
- [Jones 12a] Jones, Alistair, Atman Kendira, Thierry Gidel, Claude Moulin, Dominique Lenne, and Jean-Paul Barthès. 2012. “Personal Assistant Agents and Multi-agent Middleware for CSCW.” In Proceedings of the 16th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 72–79. Wuhan,

- [Jones 12b] Jones, Alistair, Atman Kendira, Thierry Gidel, Claude Moulin, Dominique Lenne, Jean-Paul Barthès, and Andrea Luigi Guerra. 2012. "Evaluating Collaboration in Table-centric Interactive Spaces." In *Proceedings of the AVI Workshop on Designing Collaborative Interactive Spaces (DCIS 2012)*. Capri, Italy.
- [Jones 12c] Jones, Alistair, Atman Kendira, Claude Moulin, Jean-Paul Barthès, Dominique Lenne, and Thierry Gidel. 2012. "Vocal Interaction in Collocated Cooperative Design." In *IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing (IEEE ICCI*CC)*, Kyoto, Japan, pp. 246–252.
- [Barthès 12] Barthès, Jean-Paul, Alistair Jones, Atman Kendira, Dominique Lenne, Claude Moulin, and Thierry Gidel. 2012. "Ambiguity in Multimodal Interaction with Multi-touch Multi-user Graphics Tables." In *Cooperative Design, Visualization, and Engineering*, edited by Yuhua Luo, vol. 7467 pp. 1–13. *Lecture Notes in Computer Science*. Osaka, Japan.
- [Moulin 11] Moulin, Claude, Alistair Jones, Jean-Paul Barthès, and Dominique Lenne. 2011. "Preliminary Design on Multi-touch Surfaces Managed by Multi-agent System." *International Journal of Energy, Information and Communications*, vol. 2, no. 4.
- [Jones 11a] Jones, Alistair, Atman Kendira, Dominique Lenne, Thierry Gidel, and Claude Moulin. 2011. "The TATIN-PIC Project: A Multi-modal Collaborative Work Environment for Preliminary Design." In *Proceedings of the 15th International Conference On Computer Supported Cooperative Work in Design (CSCWD)*, 2011, pp. 154–161. Lausanne, Switzerland. (Best Student Paper)
- [Kendiria 11] Kendira, Atman, Thierry Gidel, Alistair Jones, Dominique Lenne, Jean-Paul Barthès, and Claude Moulin. 2011. "Conducting Preliminary Design Around an Interactive Tabletop." In *Proceedings of the 18th International Conference on Engineering Design (ICED'11)*, vol. 2 pp. 366–376. København, Copenhagen. (Reviewers' Favorite)
- [Jones 11b] Jones, Alistair, Dominique Lenne, Atman Kendira, Thierry Gidel, Claude Moulin, and Jean-Paul Barthès. 2011. "Apprentissage Autour D'une Table Interactive." In *Actes de l'Atelier "IHM Avancées Pour L'apprentissage" - EIAH 2011*, edited by Audrey Serna and Sébastien George, pp. 36–44. Mons, Belgium.
- [Kendiria 10] Kendira, Atman, Alistair Jones, Guillaume Lehoux, Thierry Gidel, Stephanie Buisine, and Dominique Lenne. 2010. "Project TATIN: Creativity and Collaboration During a Preliminary Product Design Session Using an Interactive Tabletop Surface." In *Proceedings of IDMME-Virtual Concept 2010*, pp. 158–164. Bordeaux, France.

Contents

Résumé	v
Abstract	ix
Remerciements	xi
Related Publications	xiii
Contents	xv
List of Figures	xxi
List of Tables	xxix
Introduction	1
Part 1: Analysis	9
1 Observation	11
1.1 Activity Theory	11
1.1.1 Subjects, objects and tool-mediated action	12
1.1.2 Community, division of labor and rules	13
1.1.3 Hierarchy of activity, actions, and operations	13
1.1.4 Common objectification	14
1.1.5 Shifting the focus from community to groupware	15
1.1.6 Kaptelinin et al.'s activity theory checklist	16
1.1.7 Conclusion	18
1.2 Observation of Causal Analysis Method	19
1.2.1 General Overview	19
1.2.2 Means & Ends	21

1.2.3	Environment	23
1.2.4	Learning/cognition/articulation	25
1.2.5	Development	27
1.2.6	Summary	28
1.3	Observation of Project Planning using the “Brown Paper” method .	29
1.3.1	General Overview	29
1.3.2	Means & Ends	32
1.3.3	Environment	36
1.3.4	Learning/cognition/articulation	39
1.3.5	Development	40
1.3.6	Summary	40
1.4	Conclusion	41
2	Related work in interactive spaces	43
2.1	State-of-the-art of multitouch and multimodal gestures	44
2.1.1	Designing multitouch gesture sets	45
2.1.1.1	Norms and conflicts in gesture sets	46
2.1.1.2	Recognition vs. recall in Natural User Interfaces . .	47
2.1.1.3	User-defined gesture sets	48
2.1.2	Design principles for multitouch gestures	51
2.1.2.1	Direct manipulation:	51
2.1.2.2	Physics-based gestures:	51
2.1.2.3	“Big” and “small” gestures:	52
2.1.2.4	Bi-manual manipulation:	52
2.1.2.5	A design caveat: <i>prior experience</i>	53
2.1.3	Extending direct manipulation through instrumental inter- action	53
2.1.3.1	Asynchronous bimanual interaction	54
2.1.3.2	Stylus-based interaction	54
2.1.3.3	Tangible User Interfaces	56
2.1.3.4	Mixed-reality interfaces	57
2.1.3.5	Voice and touch-based multimodal gestures	58
2.1.3.6	Reflections on extending direct manipulation . . .	59
2.2	State-of-the-art of multi-user surfaces	59
2.2.1	Support for collaboration	59
2.2.1.1	Interpretation	60
2.2.1.2	Coordination	63
2.2.1.3	Communication	66
2.2.2	Design considerations for collaboration on multi-user surfaces	68
2.2.2.1	Orientation	69
2.2.2.2	Space	71

2.2.2.3	User identification	72
2.2.2.4	Clutter	73
2.2.2.5	Feedthrough and group awareness	74
2.3	Design consideration for established workflows	75
2.3.1	Organizational roles	76
2.3.2	Access to personal documents and information	76
2.3.3	Output and distribution	77
2.3.4	Pre-existing workflows during an activity	77
2.3.5	Providing transitions between activities	78
2.4	State-of-the-art of augmented physical environments	78
2.4.1	Related work in interactive spaces	79
2.4.2	Shareability in interactive spaces	82
2.5	Conclusions	84

Part 2: Proposition 87

3 TATIN-PIC Information Landscape & Interaction Techniques 89

3.1	Information landscape for project-oriented and user-oriented activities	89
3.1.1	Workflow and community in interactive spaces for preliminary design	91
3.1.2	Environmental concepts of meeting rooms and interactive spaces	94
3.1.2.1	Categories of devices in meeting rooms and interactive spaces	94
3.1.2.2	Device configurations and the principle of shareability	96
3.1.3	Multi-surface layered information landscape	101
3.1.3.1	Participant-oriented layer	101
3.1.3.2	Project-oriented layer	101
3.1.3.3	Emphasizing the difference between layers	104
3.1.3.4	Using project-oriented layers for complementary tasks in the activities	104
3.1.4	Conclusion	105
3.2	User interaction techniques	105
3.2.1	Interaction techniques for grouping	105
3.2.2	Interaction techniques for transferring information across surfaces	108
3.2.2.1	Portals	110
3.2.2.2	Exporting and importing between applications	111
3.2.3	User-specific interaction on the tabletop	113

3.2.4	Vocal and multimodal interaction techniques	114
3.3	Conclusion	115
4	Multi-agent infrastructure for interactive spaces	119
4.1	Software agents and societies	120
4.1.1	Agent-oriented programming and multi-agent systems	120
4.1.2	The Gaia methodology for the design of multi-agent systems	122
4.2	Requirements summary of interactive spaces	123
4.2.1	Single-device architecture for groupware	123
4.2.2	Distributed architecture for groupware	124
4.2.3	Design objectives for interactive space infrastructure	127
4.2.4	Using multi-agent systems and the Gaia methodology to meet the infrastructure requirements	128
4.3	TATIN-PIC multi-agent infrastructure: Application of Gaia method- ology	130
4.3.1	Roles and Environment	130
4.3.1.1	Gaia's role schema	130
4.3.1.2	TATIN-PIC environmental model	131
4.3.1.3	TATIN-PIC role schemas	133
4.3.1.4	Discussion	136
4.3.2	Interaction model and organizational rules	138
4.3.2.1	Gaia's interaction model and organizational rules .	138
4.3.2.2	TATIN-PIC's organizational rules	139
4.3.2.3	TATIN-PIC interaction model	140
4.3.3	Organizational Structure	142
4.3.4	Agent model	144
4.3.5	Service model	145
4.4	Conclusion	147
Part 3:	Implementation & Evaluation	149
5	Implementation	151
5.1	TATIN-PIC hardware	152
5.1.1	Interactive tabletop	152
5.1.2	Interactive board	154
5.1.3	Voice Boxes	155
5.2	TATIN-PIC user interface and applications	157
5.2.1	Login screen	157
5.2.2	TouchSuite: applications for the TATIN-PIC interactive space	157
5.2.2.1	BrainTouch : brainstorming application	159

5.2.2.2	ProjectTouch: project planning application	159
5.2.2.3	CauseTouch: causal analysis application	164
5.2.3	Personal tools	166
5.2.4	Personal assistant agent and multimodal interaction	166
5.2.5	Tablets and smartphones	170
5.3	TATIN-PIC infrastructure	171
5.3.1	Multi-agent systems	172
5.3.1.1	JADE Agents	172
5.3.1.2	OMAS agents	173
5.3.2	Inter-platform communication	173
5.3.2.1	Agent Communication Language	174
5.3.2.2	Agent Content Language	175
5.3.2.3	Message passing example	177
5.4	Conclusion	180
6	Evaluation	181
6.1	User Evaluation	181
6.1.1	Discussion of evaluation objectives	182
6.1.2	Brainstorming Evaluation	183
6.1.2.1	Participants	184
6.1.2.2	Environment and Apparatus	184
6.1.2.3	Experimental Design	186
6.1.2.4	Activity	187
6.1.2.5	Data gathering	187
6.1.2.6	Results and Analysis	190
6.1.2.7	Discussion	194
6.1.3	Project planning	197
6.1.3.1	Participants	198
6.1.3.2	Environment and Apparatus	198
6.1.3.3	Experimental Design	199
6.1.3.4	Activity	200
6.1.3.5	Data gathering	203
6.1.3.6	Results	204
6.1.3.7	Discussion	211
6.1.4	Causal analysis	213
6.1.4.1	Context	213
6.1.4.2	Procedure	213
6.1.4.3	Results & Discussion	214
6.1.5	Conclusion	216
6.1.5.1	Objective 3: Group Dynamics	216
6.1.5.2	Objective 2: Preliminary design performance	217

6.1.5.3	Objective 1: General usability	217
6.2	Multi-agent infrastructure: comparison with related work	218
6.2.1	iRos middleware	219
6.2.2	BEACH	220
6.2.3	Shared substance	221
6.2.4	ZOIL	222
6.2.5	Service-oriented infrastructures	223
6.2.6	Discussion	224
Conclusion and future work		227
 Appendix		 233
A Preliminary Design Project Ontology		235
B Multi-level grouping technique		237
C Models of Gaia analysis		243
C.1	Role schemas	243
C.2	Interaction Models	252
C.3	Service Model	258
D Supplementary materials from evaluations		261
 Bibliography		 271

List of Figures

0.0.1	The preliminary design curve (also known as the MacLeamy curve) was proposed by Patrick MacLeamy to illustrate the importance of shifting effort earlier in the timeline of the project [MacLeamy 04].	2
0.0.2	A six-person team engaged in brainstorming at an IDEO make-a-thon event. (source: http://www.flickr.com/photos/ideopostcards/)	3
0.0.3	The CSCW matrix, originally proposed by Johansen et al [Johansen 88], provides a broad, but effective, categorization of groupware system.	4
1.1.1	Basic model for tool-mediated action in AT.	12
1.1.2	Socially mediated activity in AT according to Engeström's model. Adapted from [Kharrufa 10] and from [Kuutti 95].	13
1.1.3	Hierarchical structure of activity, with actions and operations. Adapted from [Kharrufa 10].	14
1.1.4	Pragmatic, goal-directed, action cycle (left) and epistemic, exploratory action cycle (right), adapted from [Fjeld 02].	16
1.1.5	Döweling et al's [Döweling 12] revised model of an activity specifically tailored for the analysis of groupware.	17
1.2.1	A cause-and-effect diagram as presented to the participants of the observation.	20
1.2.2	Hierarchical task decomposition for causal analysis.	22
1.2.3	The table and whiteboard environment after preparation.	23
1.2.4	The scribe creating and managing a bank of Post-its on the tabletop.	24
1.2.5	The role of the moderator can naturally and temporarily rotate among participants.	26
1.3.1	An example of the brown paper model for two phases of project.	30
1.3.2	An example of a task Post-it in the brown paper method with the calculations from the PERT analysis displayed underneath (left) with the legend used denote the significance of the values (right).	31

1.3.3	The procedure given to the participants instructing how to perform PERT analysis.	31
1.3.4	Hierarchical task decomposition for project planning activity. . . .	34
1.3.5	Preparation of brown paper. From left to right, by row: participant divides brown paper along x-axis to denote project phases (task 2.3); two participants generate task Post-it notes (task 2.4); participant places tasks into the flowchart (task 2.5); participant generates PERT Post-its (task 2.6); participants place PERT Post-its under their corresponding task (task 2.7).	35
1.3.6	Use of the whiteboard for the task of building the organigram and performing task identification (1.2 & 1.3), the PC workstation with overhead display (5.2), and the tabletop during the construction of the revised organigram (5.1). These represent two-person collaborative efforts which involved interpretive tasks.	38
2.1.1	Progression of user interfaces with regards to their strategies for learning. Adapted from [Mathis 11].	48
2.1.2	An example of a conflict between gesture sets: Wobbrock et al. [Wobbrock 09] proposed copy gesture (left) and Frisch et al. [Frisch 09] proposed copy gesture (right). Both gestures were user-defined, which means they were the most agreed upon gestures for the copy action in the respective studies. In both cases, users had little or no experience with multitouch interaction and they were allowed to use both hands. The main difference between the two experiments is the context of the node-link diagram in the case of Frisch et al. This illustrates the impact that context has on intuitive interaction.	50
2.1.3	Sulaiman and Oliver [Sulaiman 10] is an example of one study which uses ethnographic observations of traditional tabletops to identify (among other things) how users place their non-dominant and dominant hands when annotating a document (left). Schmidt et al.'s <i>IDLenses</i> (center) provides a solution for user identification on interactive tabletops by identifying a user's hand through biometric measurements and opening a circular lense through which user specific actions may be performed [Schmidt 10]. Widgor et al.'s <i>Rock & Rails</i> (right) uses the shape of the non-dominant hand as a means to trigger different kinds of manipulations such as aligning components to a grid [Wigdor 11].	55
2.2.1	The sensemaking loop, adapted from Pirolli and Card [Pirolli 05].	61
2.2.2	Vogt et al.'s experimental set-up for collaborative sensemaking.	62
2.2.3	<i>WeSearch</i> tabletop application for collaborative search and sense-making [Morris 10].	64

2.2.4	The conversation clock, from Bergstrom and Karahalios [Bergstrom 07].	65
2.2.5	(a) Automatic person-based orientation with neutral zone for group work with Shen et al's <i>UbiTable</i> [Shen 06], (b) automatic person-based orientation during a phase of individual work, and (c) automatic environment-based orientation during a phase of group work on Buisine et al.s <i>Brainpurge</i> application [Buisine 12].	71
2.4.1	Examples of interactive spaces for meeting rooms (left, [Hilliges 07]) and for deskwork environments (right, [Biehl 08]).	80
3.1.1	The single application framework on interactive tabletops, personal file browsing application <i>Focus</i> (left) [Collins 11a, Collins 11b] and a group brainstorming activity <i>Firestorm</i> (right) [Clayphan 11].	91
3.1.2	An illustrated scenario of a preliminary design project (originally presented in [Guerra 12]).	92
3.1.3	The relationships between project and user used when designing the TATIN-PIC information landscape.	93
3.1.4	The <i>AffinityTable</i> proposed by Geyer et al. (left, [Geyer 11a, Geyer 11b]) and the shareability of the devices with regard to entry and access to teamwork (right).	97
3.1.5	The interactive space proposed by Hilliges et al. (left, [Hilliges 07]) and the shareability of the devices with regard to entry and access points.	98
3.1.6	Wigdor et al.'s <i>WeSpace</i> [Wigdor 09b] with regard to their entry and access points to team and personal work.	99
3.1.7	The TATIN-PIC platform with regard to its entry and access points.	100
3.1.8	Personal and activity layer extending to both surfaces in the interactive tabletop.	102
3.1.9	The circular menu serves as personal access point on the tabletop surface. The keyboard widget is opened on left.	103
3.2.1	Three different approaches to grouping: containers, proximity, and lasso gestures.	106
3.2.2	A hypothetical extension of the bubble technique to include labels and multi-level hierarchies. The outcome of the "split" gesture actions is ambiguous, and this figure illustrates two possible outcomes, though more variations are possible.	107
3.2.3	Creation of a group of objects.	108
3.2.4	Adding an object to a group (right) and adding a label to a group (left).	109

3.2.5	Portals on the interactive tabletop (left) and the portals on the interactive board (right). The items removed or added from either surface.	110
3.2.6	Three different multimodal interactive spaces with virtual agents: moderator assistant agent, personal assistant agents, and oracle agent.	116
4.2.1	The TATIN-PIC architecture, modeled within the context of a single device.	124
4.2.2	A fragment of the TATIN-PIC architecture, emphasizing the distributed components of the system. (This will also be used as part of the environmental model for the Gaia methodology.)	126
4.3.1	PROJECTMANAGEMENT role schema.	135
4.3.2	PROJECTSUPPORT role schema.	136
4.3.3	The interaction model.	139
4.3.4	Interaction model associated with the PROJECTMANAGEMENT and PROJECTSUPPORT roles.	141
4.3.5	Interaction model associated with an activity switch, involving the PROJECTMANAGEMENT, PROJECTSUPPORT, APPLICATION-MANAGER and APPLICATIONSUPPORT roles.	142
4.3.6	An illustration of the instantiated agents at runtime and their organizational structure.	146
5.0.1	Interactive space concept (left) and implementation (right). . . .	151
5.1.1	The interactive tabletop, hardware components (left) and in use (right). Note that the prototype uses two pairs of mirrors, eight lasers and four cameras which are not pictured in the schema for simplicity.	153
5.1.2	Interactive board display, hardware components (left) and in use (right).	155
5.2.1	Login screen of TATIN-PIC on the interactive tabletop.	158
5.2.2	BrainTouch: multi-surface brainstorming application. Screenshot from the interactive board (above) and interactive tabletop (below).	160
5.2.3	ProjectTouch: multi-surface project planning application. Screenshot from the interactive board (above) and interactive tabletop (below).	162
5.2.4	CauseTouch: multi-surface causal analysis application. Screenshot from the interactive board (above) and interactive tabletop (below).	165

5.2.5	Personal tools in the personal layer, opened from the user's circular menu.	167
5.2.6	The vocal interface component, which provides users with feedback regarding the result of the voice recognition module, as well as control over the microphone.	167
5.2.7	The dialogue concept for the insertion action as a finite state machine diagram.	170
5.3.1	A series of messages in the multi-agent system to process the vocal request "Create a Post-it note with content 'Develop business plan'".	178
5.3.2	A series of messages in the multi-agent system to process the response to the vocal request "Create a Post-it note with content 'Develop business plan'".	179
6.1.1	BrainTouch, v0.9.	185
6.1.2	Experimental design with counterbalancing of control and experimental conditions and brainstorming topics.	186
6.1.3	A brainstorming session during the control condition (left) and a brainstorming session during the TATIN-PIC condition (right).	188
6.1.4	Idea production for the least-productive participants to the most-productive participants for each of the conditions.	192
6.1.5	Top 5 ranked items from the general usability of the BrainTouch v0.9 application.	194
6.1.6	Bottom 5 ranked items from the general usability of the BrainTouch v0.9 application.	195
6.1.7	General procedure for planning evaluation. Note that activity times in practice may vary as some groups may finish earlier.	200
6.1.8	Control condition (left) and TATIN-PIC condition (right). This image is taken after the task identification sub-activity is complete and the project flowchart has begun. Participants in the control condition are using the vertical board to layout tasks, while the participants of the TATIN-PIC condition must group tasks into phases before importing them into the flowchart.	201
6.1.9	The duration of each sub-activity in the control and TATIN-PIC conditions.	205
6.1.10	Accuracy of resulting project plan in each flowchart for each group in each section.	205
6.1.11	Flowcharts from Group C, control condition, top and Group B, TATIN-PIC condition, bottom.	206

6.1.12	A participant in Group C restructuring the project flowchart, and communicating changes to another participant who updates an electronic version (left) and participants in Group D discussing the planning optimization (right).	208
6.1.13	A group performing causal analysis. The group (right) creates causes and effects as the moderator (left) structures the input from the group.	215
A.1	Ontology describing collective actions that are undertaken by the team.	235
A.2	Ontology describing documents used in TATIN-PIC.	236
B.1	Creation of a group of objects.	238
B.2	Adding an object or a label to a group.	239
B.3	Expansion and contraction of groups.	240
B.4	Adding a subgroup.	240
B.5	Expanding groups and subgroups.	241
B.6	Removing an object from a group. This gesture is illustrated here with an item, but also applies to subgroups.	242
B.7	Removing the last object from a group.	242
C.1	PROJECTMANAGEMENT role schema.	244
C.2	PROJECTSUPPORT role schema.	245
C.3	APPLICATIONMANAGEMENT role schema.	246
C.4	APPLICATIONSUPPORT role schema.	247
C.5	PERSISTENCEMANAGEMENT role schema.	248
C.6	PERSISTENCESUPPORT role schema.	248
C.7	USERIDENTIFICATIONONSHAREDSPACE role schema.	249
C.8	USERPRIMARYSHAREDDEVICESTSUPPORT role schema.	250
C.9	USERPERSONALDEVICESTSUPPORT role schema.	251
C.10	USERPERSONALASSISTANT role schema.	252
C.1	The interaction model legend.	253
C.2	Interaction model associated with PROJECTMANAGER and PROJECTSUPPORT roles.	254
C.3	Interaction model associated with an activity switch, involving the PROJECTMANAGER, PROJECTSUPPORT, APPLICATIONMANAGER and APPLICATIONSUPPORT roles.	255
C.4	Interaction model associated with APPLICATIONMANAGER and APPLICATIONSUPPORT roles regarding document importation. . .	256
C.5	Interaction model associated with APPLICATIONMANAGER and APPLICATIONSUPPORT roles regarding inter-device functionality. .	256

C.6	Interaction model associated with PERSISTENCEMANAGER and PERSISTENCESUPPORT roles.	257
C.7	Interaction model associated with USERIDENTIFICATIONONSHARED-SURFACE and USERPRIMARYSHAREDDEVICESUPPORT role. . . .	257
C.8	Interaction model associated with USERPERSONALDEVICESUPPORT role and USERPERSONALASSISTANT role.	258
D.1	Complete results from the usability criteria questionnaire from the brainstorming evaluations. This was distributed to the participants in the experimental TATIN-PIC condition only.	262

List of Tables

2.1	Taxonomy of surface gestures from Wobbrock et al. [Wobbrock 09].	49
2.2	Existing interactive spaces for co-located collaborative work. This list focuses on meeting room implementations that have been evaluated with potential users engaged in a task.	81
2.3	Design guidelines for interactive spaces by domain.	84
4.1	Shoham’s comparison of object-oriented programming to agent-oriented programming [Shoham 93].	120
4.2	Operators for liveness expression, from [Wooldridge 00].	131
4.3	TATIN-PIC environmental model.	134
4.4	Names and description of roles, and the reference of their role schemas in Appendix C.	137
5.1	Vocal and multimodal commands for creating, editing and deleting Post-it notes.	168
5.2	Keywords used for JSON object messages on the TATIN-PIC platform.	175
5.3	Keywords used for JSON <code>content</code> structure.	176
5.4	Keywords used for JSON <code>args</code> structure.	177
6.1	The different metrics and data gathering tools used in the user evaluations to meet the objectives.	184
6.2	The mean (m) and standard deviation (SD) for speaking time (in seconds), speaking inequity and handling gesture occurrences and handling gesture inequity for all participants in each of the conditions are calculated. The results for each of the criteria are also evaluated using an unpaired Student’s t-test, which is presented in the last four columns: degrees of freedom (DOF), F-values (F), corresponding p-value, and statistical significance (sig).	191

LIST OF TABLES

6.3	Results from the 7-point Likert scale questions regarding the participant's motivation and involvement, and the results of a paired Student's t-test for the five statements.	191
6.4	Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.	193
6.5	Results from the comparative subjective evaluation on criteria for the overall interactive space, with corresponding paired Student's t-test between the two conditions.	193
6.6	Experimental Design over two sessions.	200
6.7	Results from the comparative subjective evaluation on criteria for the group dynamics in the interactive space for the first session, with corresponding paired Student's t-test between the two conditions.	207
6.8	Results from the comparative subjective evaluation on criteria for the group dynamics in the interactive space for the second session, with corresponding paired Student's t-test between the two conditions.	208
6.9	Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.	210
6.10	Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.	211
6.11	Comparison of different approaches to distributed user interfaces.	225
C.1	The services model for the Project and Persistence sub-organizations.	259
C.2	The services model for the user focused sub-organization.	260

Introduction

Computing technology plays a central role in the modern practices of design and engineering projects. This is particularly true within the resource-intensive detailed design phase, which occurs toward the middle of a project lifecycle. This phase has significant computer-support with a vast array of computer-aided design (CAD) applications and project management software, allowing businesses to be quite sophisticated in their development of services or products. The degree to which computing power has been integrated into this process is in stark contrast to the seemingly non-existent availability of computer support for *the preliminary design phase*. With this in mind, this research will contribute a solution for computer-support during this phase of engineering projects.

Context

The preliminary design phase (or conceptual design phase) begins with a high-level description of the problem and ends with a high-level description of a solution. It also includes a project plan that a team will follow to implement this solution [Wang 02, Neill 98].

The importance of the preliminary design phase in design and engineering projects can be explained by the effort curve (Figure 0.0.1). The effort curve illustrates that as a project moves forward in time the ability to make decisions on the final deliverable of the project will decrease while the cost of these decisions will increase. With this constraint in mind, it becomes appropriate for project managers to focus a great deal of time and resources early in the project, in the preliminary design phase, while decisions are cheap and impactful. The problem is that though they may be cheap and impactful, the act of making these decisions is more difficult earlier than later. Concepts, constraints, functional requirements, risks, causes, and effects are still abstract, and decision-making is difficult without concrete information.

This problem has led to the development of new methods and processes for the preliminary design phase which provide designers with a framework to express ill-defined problems, understand end-user requirements, analyze existing knowledge

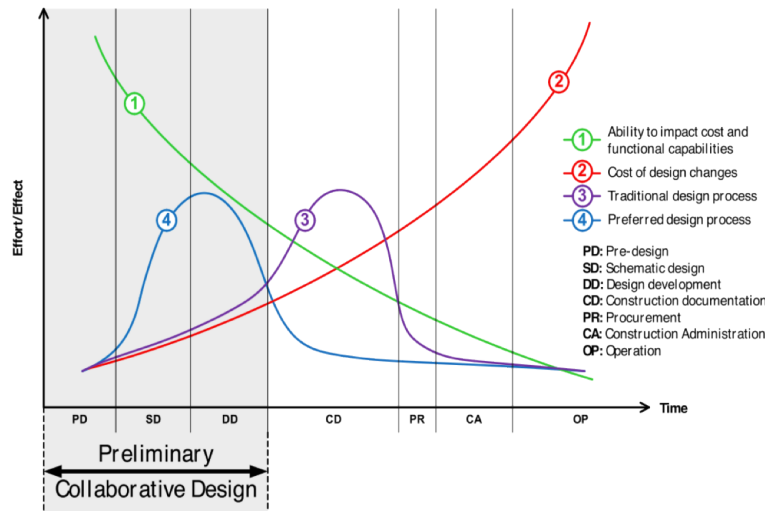


Figure 0.0.1: The preliminary design curve (also known as the MacLeamy curve) was proposed by Patrick MacLeamy to illustrate the importance of shifting effort earlier in the timeline of the project [MacLeamy 04].

and construct solutions. Through continued research and practice, these new evolutions in the preliminary design phase are perhaps most indebted to the approach of Design Thinking. The adoption of this approach within businesses has been recently been championed by IDEO, a critically acclaimed design firm, and Stanford University’s Hasso Plattner Institute of Design, where its theory is taught and put into practice regularly.

At its core, Design Thinking’s methods are structured creativity techniques for problem solving which require all members of the design team to be present and engaged in a collaborative effort. Examples of such techniques are brainstorming (Figure 0.0.2), causal analysis, and visual project planning. Each have their own rules, models, and social structures, but all depend upon proper group communication, collaboration, and creativity to be as complete as possible.

These techniques are conducted inside meeting rooms whose tables, walls, and whiteboards are covered with information in the form of Post-it notes, photographs, sketches, and printouts. These are used to visualize, organize and structure heterogeneous content to provide teams with a deeper, shared insight into a problem. As information is added to this environment, the overhead of managing the information becomes increasingly important. For example, if a modification is made to one part of a flowchart, the changes must be manually propagated throughout the remainder of the flowchart. With this traditional pen-and-paper method of work, it can also be difficult to integrate or reuse information from previous projects and existing knowledge bases. This process is especially time-consuming after the work



Figure 0.0.2: A six-person team engaged in brainstorming at an IDEO make-a-thon event. (source: <http://www.flickr.com/photos/ideopostcards/>)

has been completed, when one member must spend many hours copying down the information and redrawing or photographing sketches so that it may be shared with the rest of the team.

Technology can augment the collaborative work of these meeting spaces. This objective has been the focus of much research on the subject of *co-located collaboration*, providing a fertile foundation on which this thesis is positioned. Co-located collaboration is a subdomain of computer-support collaborative work (itself a subdomain of human-computer interaction) which addresses the design and implementation of computer systems for co-located (i.e. physically present) users engaged in synchronous collaboration (Figure 0.0.3). It is particularly challenging because any tools that are chosen to provide computer-supported collaboration, also have the risk of encumbering *natural collaboration*. This was found to be true by early attempts at first-generation meeting support systems where computer workstations and personal displays were embedded into conference tables for each user in the room. According to Nunamaker et al's decade-long experimentations with such installations, they found that the physical environment ultimately fragmented group focus, obstructed the line of sight of participants, and worked against the natural, traditional collaboration style of the group [Nunamaker 97].

During the past decade, technologies such as interactive tabletops, multitouch board displays, voice recognition software, 3D motion sensing devices, and handheld tablet and smartphone computing are all undergoing significant advances in maturity all the while reaching more affordable prices and off-the-shelf availability. Their form factors resemble the physical configuration of traditional pen-and-paper

	Same time	Different time
Same place	Co-located synchronous collaboration Meeting support systems, brainstorming tools	Shared-space asynchronous collaboration Specialized project management and task displays, systems for shift-work
Different place	Distributed synchronous collaboration Chat systems, video conferencing, real-time collaboration in word processors	Distributed asynchronous collaboration E-mail, forums, shared calendars, edit-tracking features in word processors

Figure 0.0.3: The CSCW matrix, originally proposed by Johansen et al [Johansen 88], provides a broad, but effective, categorization of groupware system.

environments, while their “natural” input devices (based on multi-touch, gestures, voice, tangibles, etc.) allow them to leverage a user’s pre-existing verbal, spatial, social, motor and cognitive skills. Researchers hypothesize that having these devices working in concert inside *interactive spaces* could augment *co-located collaboration* for groups of users.

This is an idea that originates from Marc Weiser’s vision of *ubiquitous computing* where computers would disappear and “weave themselves into the fabric of everyday life until they are indistinguishable from it”[Weiser 91]. These ubiquitous computing rooms have been explored under several different topics in the literature, such as roomware [Streitz 99, Tandler 04], interactive landscapes [Streitz 99], distributed user interfaces [Melchior 09], multi-display environments [Nacenta 09, Bachl 11, Seyed 12], multi-display groupware [Wallace 09], multi-surface environments [Beaudouin-Lafon 11, Gjerlufsen 11, Burns 12] and multi-person-display ecosystems [Terrenghi 09]. Listing these terms together resembles some of their definitions.

These are all appropriate terms to describe the kind of second-generation meeting room that is the subject of this thesis. However, I choose to use the term *interactive spaces* as proposed by Jetter et al [Jetter 12a]. Interactive spaces are “ubiquitous computing environments for computer-supported collaboration that exploit and enhance the existing cognitive, physical and social skills of users or groups of users” [Jetter 12a]. Accommodating these existing skills requires designing interfaces that move beyond mouse-and-keyboard computing and into post-WIMP (Windows, Icons, Menu, Pointer Device) interaction techniques provided by new input devices. This style of computing can then integrate “seamlessly into

established work practices and work environments to achieve a natural, unobtrusive support for collaborative activities”[Jetter 12a].

Research goal

There currently exists a handful of examples of interactive spaces in the scientific literature, as well as a rapidly growing and diverging collection of new input devices. These illustrate a wide range of potential hardware configurations and post-WIMP interaction techniques which are appropriate for many different kinds of activities.

The goal of this thesis is to research and develop an interactive space to facilitate preliminary design activities for design and engineering teams. Therefore, there are two principal research questions that should be answered.

- *How should the interactive space be designed in order to augment co-located collaboration in preliminary design activities?*

Answering this question requires adopting a human-computer interaction perspective and therefore my approach includes several user-centered design methods and principles. In my analysis, I use observations of groups engaged in traditional preliminary design activities and a literature review of existing interactive spaces in related work so that I may compile a list of design guidelines. The literature review also provides a more in-depth definition of co-located collaboration as well as an introduction to the design principle of shareability in the context of interactive spaces.

During the second part of this thesis, I propose an interactive space, which at its core uses a configuration not yet fully explored in previous literature: a large multitouch tabletop and a large multitouch interactive board display. In presenting this interactive space, I use the design principal of shareability to explain how this configuration is appropriate for preliminary design activities. I also give special consideration to the information landscape (i.e. the visual and interactive organization of information) of the interactive space, and propose a user interface which extends across multiple shared displays, and maintains a separation of concerns regarding personal and group work. I also present several interaction techniques whose conception was necessary to allow for certain kinds of multi-surface, multi-modal, and multi-user interaction.

The interactive space was also fully implemented and subject to numerous user evaluations to understand its impact on co-located collaboration of groups engaged in preliminary design activities. Our conclusions show how the interactive space preserves many of the positive qualities of collaboration, while offering a considerable reduction in the overhead and maintenance of preliminary design activities.

- *How can the distributed computing infrastructure be designed to support the requirements of this interactive space?*

Answering this question requires adopting a distributed computing perspective. The proposed interactive space and its interaction techniques calls for a computing infrastructure which can properly support a set of heterogeneous and distributed devices working in concert. Therefore, my approach uses a multi-agent infrastructure which has been designed using the Wooldridge et al's Gaia methodology [Wooldridge 00]. This allows for the prescription of a computing infrastructure which is both platform and language independent while also remaining modular, scalable and reasonably tolerant of network faults.

In order to demonstrate its platform independence, the computing infrastructure is fully implemented using two different multi-agent toolkits, JADE and OMAS. This infrastructure is also used during the extensive user evaluations of the interactive space. A comparison to existing infrastructure solutions in related work is also provided. This serves to highlight some of the advantages of the multi-agent infrastructure, particularly with regard to its run-time modularity, and also suggests some avenues for future work which could make certain aspects of its distributed data sharing strategy more accessible to developers of interactive spaces.

This dissertation represents the culmination of a three year thesis research project, beginning in October 2010 at the Heudiasyc laboratory at the Université de Technologie de Compiègne. This research has been conducted as part of the multi-disciplinary TATIN-PIC (french for Table Tactile INteractive - Plateform Intelligente de Conception), involving collaboration from the Costech laboratory and the Innovation Centre, as well as the MIS laboratory at the Université de Picardie Jules Verne. The research project, including this thesis, was co-funded by the region of Picardy and the European Union through the European Regional Development Fund.

Organization of the thesis

This thesis uses a variety of methods from different disciplines to obtain answers to the two central research questions.

- **Chapter 1: Observation** - In this chapter, I present observations of groups engaged in two different preliminary design activities in traditional meeting room environments. I present and analyze the observations using the theoretical framework of Activity Theory. This provides a *qualitative understanding of the preliminary design activities* that will be supported by the interactive space.

-
- **Chapter 2: Related work** - With a frame of reference established for preliminary design activities, I perform a review of the literature in order to extract *design guidelines* with regard to four domains of interactive space: personal interaction, social interaction, community/workflow, and the physical environment. The literature review also serves to clarify the concept of co-located collaboration, and introduces the design principal of shareability.
 - **Chapter 3: TATIN-PIC information landscape and interaction techniques** - In this chapter, I present the multi-user and multi-surface interactive space, giving special consideration to its '*information landscape*', i.e. the visual and interactive organization of information. I also discuss the *physical environment and input devices*, and show how different configuration can impact the shareability of the interactive space. This chapter also presents several interaction techniques:
 - A multitouch grouping interaction technique that allows users to group virtual components into multi-level hierarchies.
 - A multi-surface interaction technique that allows components to be transferred from one shared surface to another.
 - A lightweight and flexible software solution that allows user identification on the surface of the tabletop in multiuser settings.
 - A set of vocal and multimodal interaction techniques designed to be used with a personal assistant agent.
 - **Chapter 4: Multi-agent infrastructure for interactive spaces** - With the basic components of the interactive space established, I turn my attention to the needs of the distributed computing infrastructure. I begin by discussing several desirable qualities of the infrastructure for interactive spaces at development-time and run-time. I also discuss agent-oriented programming and multi-agent systems and explain why these are appropriate for distributed computing in interactive spaces. The remainder of the chapter is dedicated to an application of the Gaia methodology which results in *an implementation plan for a multi-agent infrastructure for interactive spaces*.
 - **Chapter 5: Implementation** - This chapter presents our multitouch, multi-user, multi-surface, and multimodal interactive space for preliminary design activities. I also present the implementation of our multi-agent infrastructure which uses two different multi-agent system toolkits, JADE and OMAS.
 - **Chapter 6: Evaluation** - This section present three different usability evaluations for the interactive space. These involve groups of five or six

users engaged in preliminary design activities such as brainstorming, project planning, and causal analysis. Despite initial difficulties with the learning of the certain interactive techniques, the interactive space was found to preserve many of the desirable qualities of co-located collaboration while reducing the overhead in performing preliminary design activities. In the second section of this chapter, I also discuss several alternate solutions to the distributed computing infrastructure in interactive spaces by a comparison of related work.

For those who are interested in the human-computer interaction approach to interactive spaces for preliminary design, I recommend chapters 1, 2, 3 and the first sections of chapters 5 and 6. For those who are interested in distributed computing architecture and multi-agent systems, I recommend the first section of chapter 3, as an introduction to the interactive space, and then chapter 4 and the second sections of chapter 5 and 6.

Part 1: Analysis

"Are natural user interfaces natural? No. But they will be useful""

(Don Norman)

Chapter 1

Observation

To better understand collaborative activity in traditional environments, I performed observations of potential users engaged in preliminary design activities. The first activity I observed was a collaborative causal analysis. Causal analysis is a technique very much related to Root Cause Analysis, and uses a diagram inspired by Ishikawa fishbone cause-and-effect diagram. Causal analysis can also be used as a tool by product designers to help them explore the design space of problem. The second activity I observed is a collaborative project planning technique informally known as the brown paper method. Its purpose is to visualize the entire workflow of a project on the wall of a meeting room, so that key players can gather together, understand what work is going on in parallel to their own, and attempt to optimize the project workflow. They are both collaborative tools used in meetings, though one is more product-oriented while the other is project-oriented.

My observations were guided by the Activity Theory (AT) checklist which was designed by Kaptelinin et al. [Kaptelinin 99] to allow evaluators to be more thorough during their observations of people performing activities. Above all, AT provides a “lens” through which interaction designers can understand collaboration by focusing on how social artifacts and social organization mediate social action [Baumer 11].

1.1 Activity Theory

Activity Theory (AT) is a theoretical framework particularly adept in modeling context and collaboration. It is considered to be a “second-wave” theory of human-computer interaction, which was necessary after it became clear that “first-wave” theories, such as the cognitivist, information processing approach, were not sufficient for understanding interfaces and their context.

According to Kaptelinin et al. [Kaptelinin 99], there are two concepts which

form the general approach used in AT: first, the human mind can only exist and be understood within the context of human interaction with the world and, second, this interaction with the world - the activity - is socially and culturally mediated. These are concepts originating from the theoretical explorations of a collective of prominent Russian and Soviet psychologists, notably Lev Vygotsky and Sergei Rubinstein, in the 1920s and 1930s. Half a century later, their work heavily influenced Aleksei Leontiev, who rendered much of the modern framework of AT, and in turn influenced work by Yrjö Engeström on AT in social organizations [Kaptelinin 13]. Finally, Susan Bødker’s work, notably her 1991 book *Through the Interface*, fully integrated AT into the field of HCI and contributed to fundamental sea change in the HCI agenda from “interface” to “interaction” [Christiansen 13].

Before discussing the activity theory checklist used in the observations, I provide a brief primer on AT, which introduces concepts that have proven to be useful and influential to this thesis.

1.1.1 Subjects, objects and tool-mediated action

At the core of AT, subjects perform tool-mediated actions on objects in order to produce an outcome. Here, a “subject” can range from an individual or a collective and must possess agency, or “the ability and the need to act”, which drives the motive of the outcome (Figure 1.1.1) [Kaptelinin 06]. This means that a computer is rarely the object of an activity but rather a tool which mediates activity. From a HCI perspective, Kaptelinin explains subjects are not interacting *with* computers, but rather *through* computers [Kaptelinin 13]. This resonates with Beaudouin-Lafon’s instrumental interaction, where an instrument is a mediator, or “two-way transducer” between the user and domain objects [Beaudouin-Lafon 00]. In AT, the terms “instrument” and “tool” are often used interchangeably, though “instrument” has an added connotation which can imply not only tools but also methods.

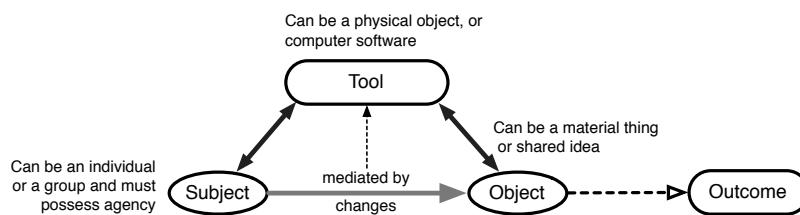


Figure 1.1.1: Basic model for tool-mediated action in AT.

1.1.2 Community, division of labor and rules

By itself, the basic model does provide a way of describing the foundational components of how a social artifact can mediate collaborative action inside a group. However, Engeström's provides a useful extension of such model which captures the social organization inside which an activity takes place [Engeström 99]. For Engeström, the subject-object tool-mediated activity is part of a cultural context. This cultural context is explicitly modeled by introducing a “community” component to the subject and object components of activity. With this community component, activity is not only mediated by tools, but also by “rules” (both implicit and explicit regulations of the interaction between a subject and the relevant community for an activity) and “division of labor” (how a community organizes work with around a relevant object) (Figure 1.1.2) [Engeström 99]. Though there are connections not explicitly represented in this pyramid model, Kutti explains that Engeström views an activity as “actually a systemic whole in the sense that all elements have a relationship to other elements, but all those connections have not been presented in the picture” for the sake of clarity [Kuutti 95].

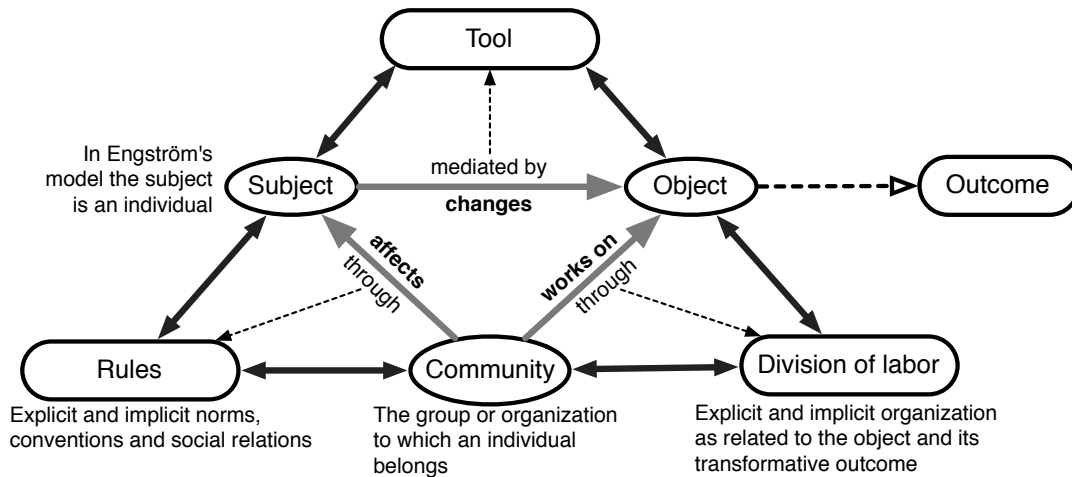


Figure 1.1.2: Socially mediated activity in AT according to Engeström's model. Adapted from [Kharrufa 10] and from [Kuutti 95].

1.1.3 Hierarchy of activity, actions, and operations

Engeström also provides another important contribution to AT: the notion of the *hierarchical structure* of an activity. In AT, an activity is seen a sequence of *actions*, which themselves are composed of a sequence of *operations*. Activities are driven

by the subject's *motives*, actions are driven by lower-level *goals*, while operations are largely reactive and are driven by contextual stimuli.

This hierarchical structure is useful for explaining further concepts, such as *development* and *breakdowns*. The concept of *development* in AT addresses how a subject's cognitive processes changes over time with regards to how they use tools. Development can occur through practice, causing actions (conscious acts) to become a sequence of operations (subconscious acts). When a *breakdown* in the interaction occurs, each operation of an action becomes its own goal-directed action (Figure 1.1.3). Breakdowns are to be mostly avoided because they shift the focus away from higher-level motives, but they can also be helpful for learning and development during an activity by assisting in slowing the user down and shifting focus to finer-grained tasks when necessary [Kharrufa 10]. This notion of development is not limited only to the subject but also applies to all aspects of the activity - community, division of labor, rules, objects and even the tools themselves [Kaptelinin 99].

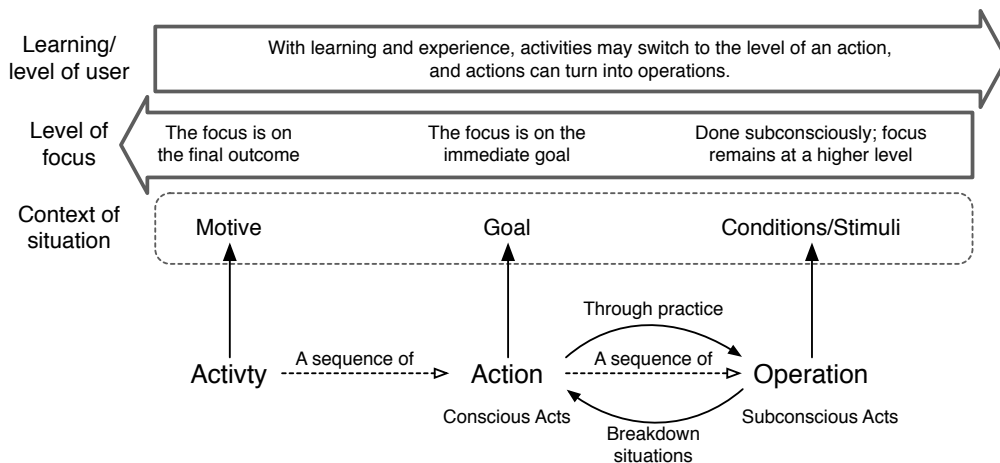


Figure 1.1.3: Hierarchical structure of activity, with actions and operations. Adapted from [Kharrufa 10].

1.1.4 Common objectification

A fundamental idea of AT is that social action is mediated by social artifacts and social organization. Fjeld et al. [Fjeld 02] explore the implications that this idea has for social cognition and groupware. From a cognitivist point of view, an individual's development of thoughts and cognitive activity is brought about through interaction and exchange with a physical environment. Such interaction with artifacts spurs mental activity, referred to *internalization*, allowing an individual to

handle increasingly abstract objects and concepts as part of their own cognitive development. Intertwined with the concept of internalization, is the *externalization* of abstract thoughts and memory into the environment using artifacts. To cite a commonly used example, children use their fingers for counting and adding. Fingers provide an external memory aid and tool for children, which at the same time are internalized to represent the abstract numbers that they use while performing their mathematical operations.

Within in a social and work context, this leads to the idea of *common objectification*. According to Weber, “the process of common objectification is understood as a process by which all (or several) members of a workgroup mutually transfer their individual knowledge, expertise, and experience into a material form. By doing this, they make their materialized knowledge available to other group members” [Weber 00]. The results of common objectification can take the form of sketches, diagrams, written ideas, verbalized ideas, etc. These are externalized versions of abstract concepts, which can be internalized by other colleagues, for the purpose of sharing information and establishing consensus. Of course, no two colleagues will internalize an external diagram the same way; therefore, common objectification is never perfect or complete. It is, however, the means through which groups transfer and build knowledge.

When considering how groups may build knowledge using these mutually objectified artifacts, it is helpful to consider the nature of the actions that may be performed on them. In this regard, Fjeld et al. provide a distinction between *pragmatic* actions and *epistemic* actions (as illustrated in Figure 1.1.4) [Fjeld 02]. Pragmatic actions are goal-directed actions and are especially relevant in coordinated group work. Epistemic actions are exploratory actions and refer to insight that may arrive through simple manipulations of information. If the tools and outcomes used in groupware allow for a clear understanding and reasoning of the problem, goals will be more relevant, effective and easier to establish. If the tools or methods used for interacting with an object are lightweight, fast, natural, and intuitive, the epistemic action cycle is performed more quickly and with less effort, resulting in better exploration. According to Fjeld et al., supporting these two kinds of interactions are important in groupware, as they provide two different ways for groups to reason about the task at hand [Fjeld 02].

1.1.5 Shifting the focus from community to groupware

Going even further, Döweling et al. [Döweling 12] points out that although Engeström model has proven useful for understanding activity inside an object-oriented, socially and culturally-mediated framework, it was *specifically tailored for the analysis of organizations*.

One of the issues with Engeström’s model as suggested by Döweling et al. is that

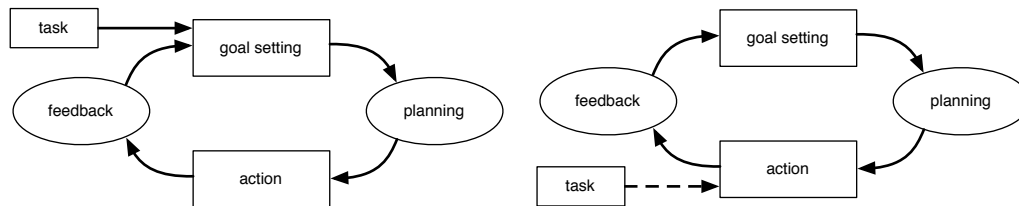


Figure 1.1.4: Pragmatic, goal-directed, action cycle (left) and epistemic, exploratory action cycle (right), adapted from [Fjeld 02].

there is a dichotomy that is difficult to discern between collective subjects that perform an action and the community they participate in. Engeström’s model focuses on one user as the subject, and places the user’s collaborators into the community component. While not untrue, it becomes problematic when analyzing groupware which involves multiple co-located people who share the same goals and perform joint actions.

Döweling et al. suggests a model which brings the focus back on the context of the design of collaborative systems, and does so with a few minor but specific changes to Engeström’s model. First, they rename “division of labor” to “work-flow”, which adds a notion of temporality to the division of labor. Second, they add a concept of “activity context” which accounts for the *environment* and general context of the activity, i.e. the situation that the users are in and acting in. Finally, they remove the explicit community component from the model; because their subject component can now include collective subjects, it can represent the members of the community actively participating in the analyzed subject-object interaction, while the impact of other members of the community can be accounted for in the context, as well as workflows and rules. They also add explicit connections between all components of the activity (which are implicit in Engeström’s model). None of these modifications contradict the theoretical foundation provided by Engeström but rather serves to refocus them onto groupware.

1.1.6 Kaptelinin et al.’s activity theory checklist

Mobilizing a theoretical framework as conceptual and comprehensive as AT for direct use in research, design, and evaluation can be difficult. This is primarily because the models used in Activity Theory are not models designed to predict phenomena that can later be validated by empirical experiments. Though, some research has attempted to use these models in an analytical based approach, such as Mwanza’s conflict analysis [Mwanza 01], they better serve analysis by providing

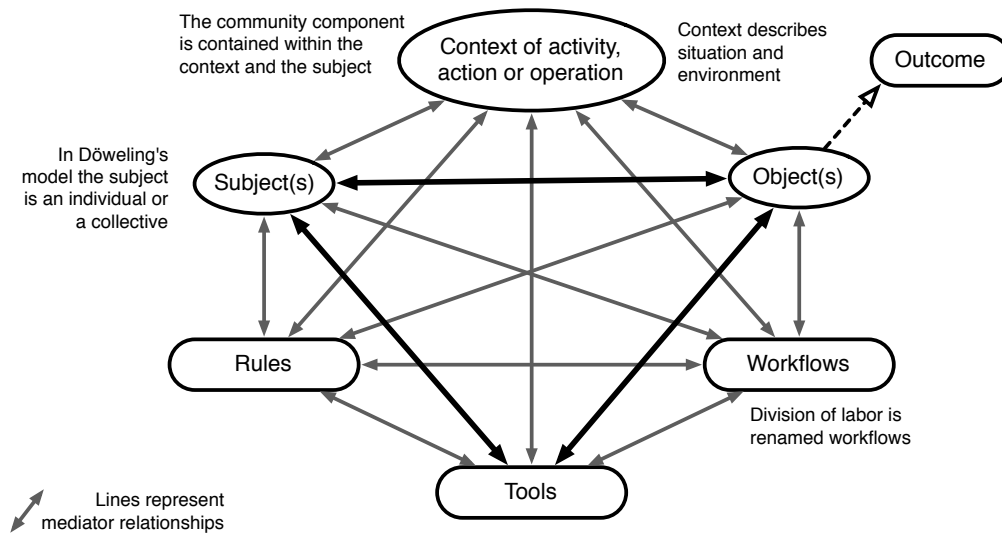


Figure 1.1.5: Döweling et al.'s [Döweling 12] revised model of an activity specifically tailored for the analysis of groupware.

a model for which researchers might understand and describe activity.

The most frequently applied uses of AT, however come in the form of a specialized checklists, which help designers generate observations on an activity so that analysis may be as complete and thorough as possible when designing and evaluating groupware. One of the most straightforward approaches to building a checklist might be as simple as asking researchers to describe each of the components in Engeström's or Döweling's model. However, this would perhaps neglect some of the more useful aspects of AT, such as development or common objectification.

The issue of carefully curating and designing a checklist to guide researchers is resolved by Kaptelinin et al. in [Kaptelinin 99]. Their activity theory checklist is designed to be as comprehensive as possible, covering many of AT's principles such as the hierarchy of activity, actions, and operations, tool mediation, object-orientedness, internalization, externalization, and development. The checklist attempts to generally orient researchers' thought and provoke reflections during design and evaluation activities, providing a list of questions and issues which are centered around four foci:

- Means and ends: What are the motives, goals, and subgoals of the users?
- Environment: In what environment (social and cultural) are these goals taking place? What tools are the participants using to accomplish their goals?
- Learning/cognition/articulation: What mental processes are being articu-

lated with this technology and how do these relate to the objects being used?

- Development: What noticeable changes in the environment are occurring?

The complete list of questions and issues of these four foci are presented in [Kaptelinin 99], and will be directly used in discussing the observations of preliminary design activities.

1.1.7 Conclusion

The first section of the first chapter has been dedicated to AT, because the idea that social artifacts and social organization mediate social action accurately encompasses the frame of reference with which this thesis was created. Co-located collaboration can be extremely complex, and AT has been true to the words written by many researchers ([Baumer 11, Barab 04, Jonassen 99, Kaptelinin 13]) and provided a “lens” through which collaboration can be understood. Among its most helpful insights is the concept of common objectification, which is seemingly ubiquitous throughout every aspect of collaborative work.

In this regard, it’s possible that the theory of distributed cognition (DCog) could have been equally as useful for describing collaboration, because DCog’s theoretical framework is well equipped for explaining the process of common objectification. As explained by [Hollan 00], in DCog, knowledge and cognition are distributed not only inside the minds of actors but also throughout the artifacts in the environment. It contributes to the idea of common objectification by focusing on how the representation of knowledge in one medium can be propagated to another. This propagation occurs across time using tools which transform (or more accurately, transduce) knowledge into various physical or digital media, through verbal communication, etc. For example, an idea can be transferred from one person to another through a pen and paper sketch, an email sent from a computer, or a telephone conversation. This framework can be used to describe how information gets represented and re-represented in a network of cognitive processes, spanning individuals, artifacts and even organizations [Baumer 11].

DCog provides a complementary “lens” to AT for analyzing groupware and collaboration, providing a different focus on the same phenomena. Nardi, in her book *Context and Creativity*, provides a detailed comparison of the two theories [Nardi 96]. Ultimately, the use of one or the other depends on how the analyst chooses to orient their analysis, and there are advantages for using both simultaneously, as demonstrated by Baumer et al. [Baumer 11, Kharrufa 10].

From a pragmatic standpoint, I focus on AT because the activity theory checklist was instrumental in guiding the analysis of the observations, but it is important

to acknowledge the influence of both theories, particularly with regard to the mediation of representational knowledge and common objectification.

1.2 Observation of Causal Analysis Method

The observation of a causal analysis activity was conducted over one 90-minute period. The participants of the observation were five engineering students (all males) from the Université de Technologie de Compiègne who were all enrolled in a course dedicated to teaching value analysis methodologies. All students had attended a two hour lecture on the use of causal analysis, which covered the theoretical groundings of the tool as well as a participatory in-class example. This would be the first time they use the tool on a problem in a small group. Several aspects of the observations were simulated. These were communicated in a design brief, which explained the context and goal of the exercise and asked for certain deliverables.

The primary method of capturing information was a handheld camera, as well as note-taking, with special consideration to the issues raised in the activity checklist.

1.2.1 General Overview

The model used in this method is a *cause-and-effect diagram* (Figure 1.2.1). This is a modified version of the fishbone diagram, originally presented by Ishikawa in [Ishikawa 76] in the domain of management and quality control. The fishbone diagram captures and visualizes the different causes of a problem, the causes of those causes, and so forth, until a suitable level of depth is reached. The cause-and-effect diagram extends the fishbone diagram by adding a second branch dedicated to the consequences of a problem (and the consequences of those consequences, and so forth).

The collaborative process through which the cause-and-effect diagram is constructed is called *causal analysis*. This allows a team to explore the chain of causality of a given problem in a structured and coordinated manner. The outcome of performing the process is a formalized understanding of a problem, as well as validation for a solution's purpose and functionality. It also gives a team alternatives so that the problem may be solved differently, either by removing one of the causes of the problem so that it may never occur, or by removing the need to resolve the problem, by proposing alternative solutions which mitigate its negative effects.

The problem, explained in the design brief in the form of a scenario, is that a potential user has emerged from his home one morning on his way to work only to find that heavy ice has formed on the windshield of his car. The user begins

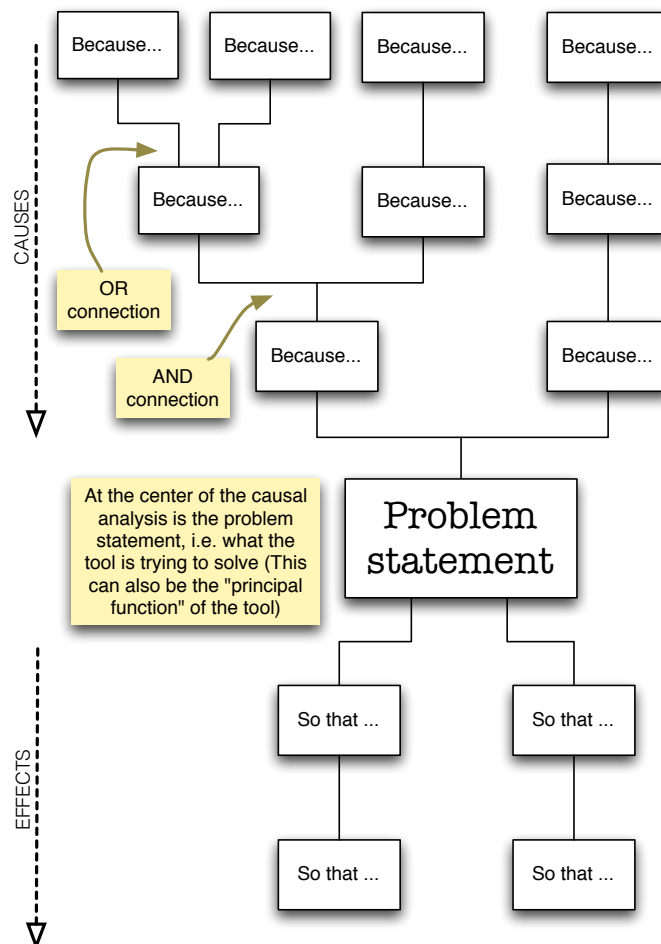


Figure 1.2.1: A cause-and-effect diagram as presented to the participants of the observation.

removing the ice with an ice scrapper, which requires time and physical effort, and wonders if there might be another way to approach the problem. The participants are instructed to build a cause-and-effect diagram around this problem, and frame their analysis around the principal function of the ice scrapper.

The participants of the observation are provided with a table, a whiteboard, several packets of Post-it notes and dry-erase and permanent markers. They bring with them their personal items (pen case, notebooks, and mobile phones) and their notes on the causal analysis process which they have taken from the lecture.

1.2.2 Means & Ends

In order to better understand the process, a hierarchical decomposition with four different activities, has been produced (Figure 1.2.2). The end goal of this exercise, as given to the group from the design brief, is to identify potential alternatives to an ice scraper so that they may consider different ways of solving this problem. The principal criterion for success in this exercise is that the group identifies the greatest number of alternative solutions as possible. Achieving this goal requires a diagram which is both complete and structurally sound. This leads to two sub-goals of completeness and soundness. These were never directly defined by the users. Completeness refers to the generation of the maximum number of relevant cause and effects, while soundness refers to the proper composition of these cause and effects into the chain of causality using the diagram. These are conflicting sub-goals because the greater the number of cause and effects that are generated, the more difficult it is to coherently integrate these cause and effects into the diagram.

The group's ability to evaluate their success in meeting these sub-goals is also difficult. There is no single correct solution for a diagram. They must use their own discretion to determine if they have explored the chain of causality to an appropriate depth. Their principal constraint is their time limit (one hour and half).

The activity took place as follows: the participants read their design briefs, confirmed with each other their understanding of the task at hand, and elected a moderator. The users then prepared the environment for their activity. The moderator wrote down the question on the whiteboard and began the diagram by drawing a box with the word "FP" (for "principal function") on the center of the board.

The group distributed permanent markers and Post-it notes so they would be accessible to all members around the tabletop. Next, the group began an informal brainstorming session where each member speaks an idea for an event on the chain of causality to the group and this event is written down on a Post-it note. The Post-it note is then placed on the tabletop near the whiteboard. The group generated seven potential events during this initial 12-minute brainstorming.

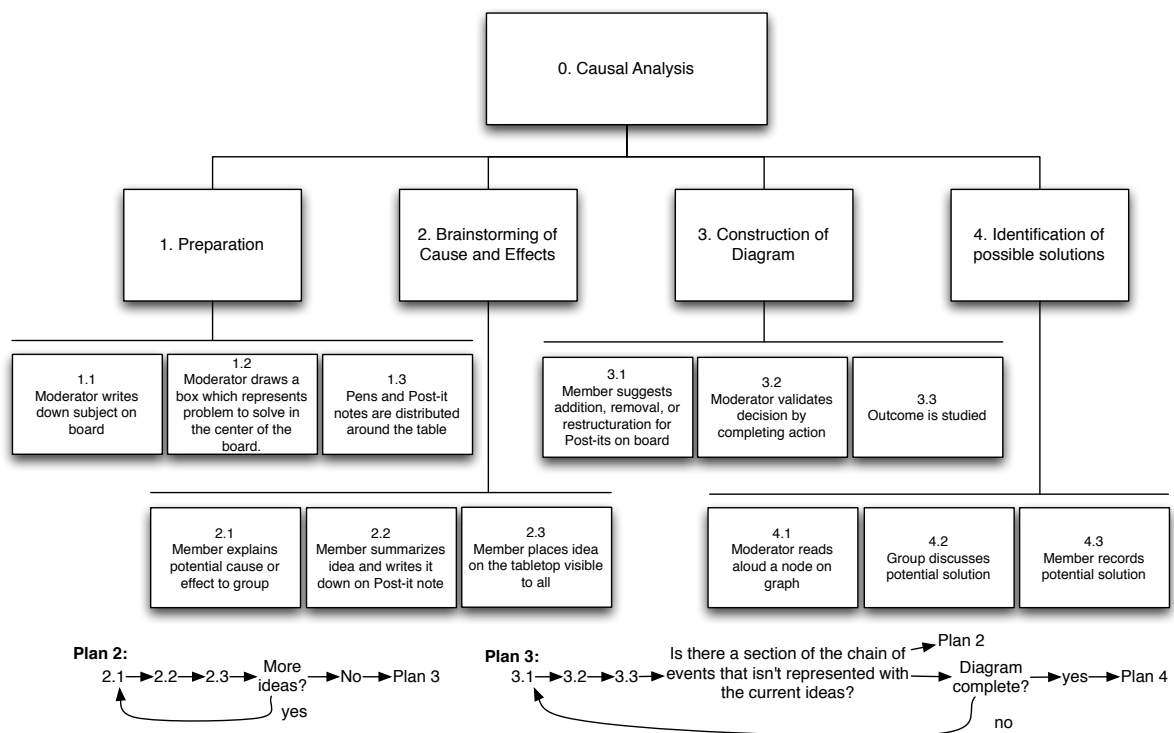


Figure 1.2.2: Hierarchical task decomposition for causal analysis.

The moderator then suggested that they begin the construction of the diagram. They began by selecting one primary cause and one primary consequence from the events on the tabletop and place these above and below the principal function on the board. This is a diagram that meets its goal of soundness, but not of completeness. From this point, they spend the next hour and 10 minutes alternating between either brainstorming and generating additional events, or discussing and modifying the diagram. Finally, with five minutes remaining, they quickly revisit each event in the diagram and identify five alternative solutions. The participants submit their cause-and-effect diagram and their list of alternative solutions.

1.2.3 Environment

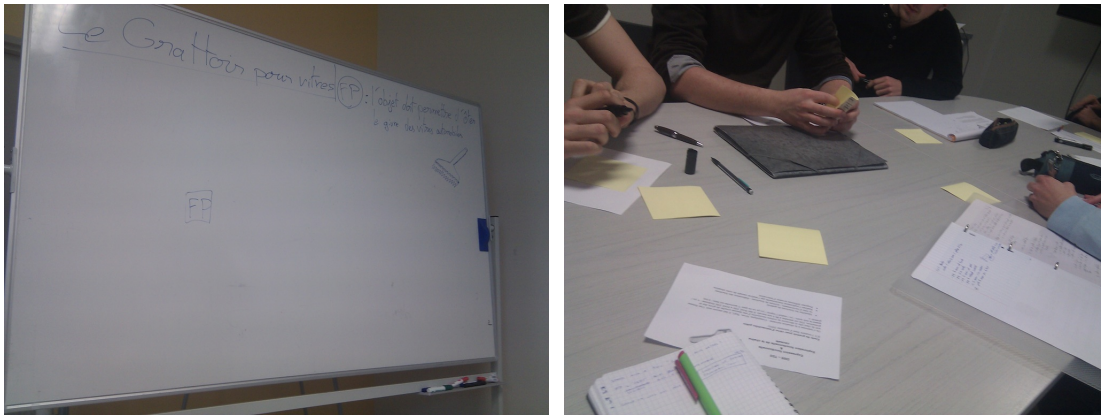
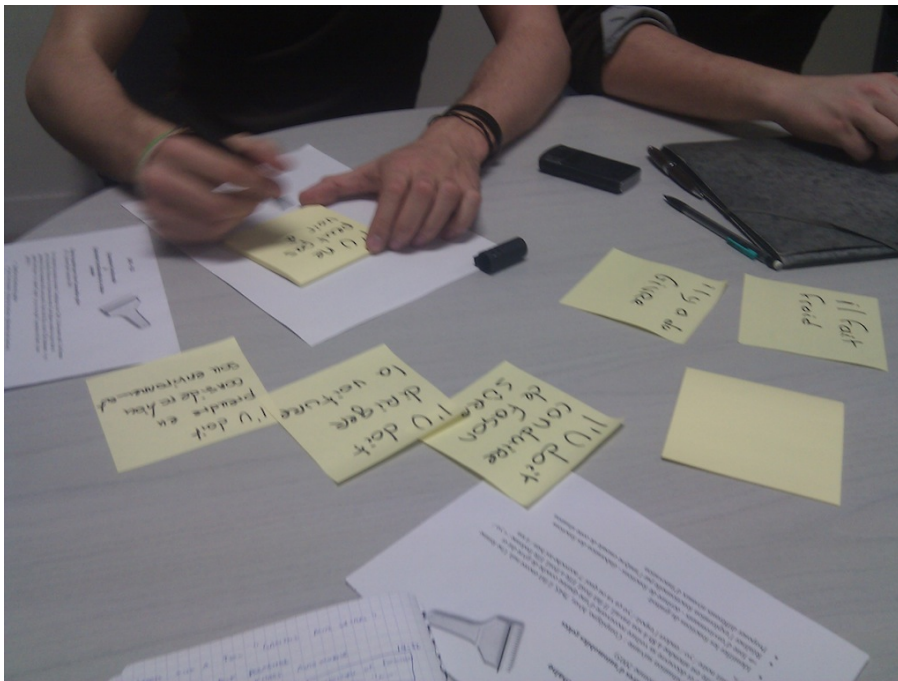


Figure 1.2.3: The table and whiteboard environment after preparation.

The two most important tools in the environment are the tabletop and the whiteboard (Figure 1.2.3). The tabletop (in addition to being a support for all users' personal items) was used for the creation of all Post-it notes, as well as “banking” all of the Post-it notes that have yet to be placed inside the diagram. The whiteboard was the primary tool of the moderator, who used it to maintain the current state of the diagram per the discussion and instructions of the group. Writing each event in the chain of causality on individual Post-it notes provides a great deal of flexibility to the team when constructing the diagram. This allows the diagram to be highly configurable as the Post-its can be repositioned with little effort. This ease of use and operation-level of interaction gives the moderator the ability to manage a diagram which follows the discussion in real-time, and, to a certain extent, supports the cognitive activity of the group through reflection (as discussed in the following subsection).

The moderator rarely used dry-erase markers to draw lines between the events, because, first, the relations between events could generally be understood simply

With this environment and these tools, participants played several roles to help coordinate their activity. During the brainstorming activity, even though Post-it notes were distributed to all users, only one participant ever wrote down the ideas that were spoken by group. This role, which I refer to as the scribe (per [Hunter 11]), emerged naturally, without assignment, but was acknowledged by the time the second idea was spoken (Participant: “Are you writing that down?”, Scribe: “Yes, and I will write everything down as we say it”). The scribe was the participant that was seated closest to the whiteboard and the moderator, and was also responsible for managing the bank of Post-its (Figure 1.2.4).



the diagram. The role of the moderator is to support the discussion of the group members as necessary by reflecting the changes in the diagram in the whiteboard, as well as to shift the focus of the group onto a different task when the discussion begins diverging.

The role of the moderator could sometimes change hands (Figure 1.2.5). For example, at one point the group began discussing a new idea for restructuring the diagram which the moderator was unable to understand, and a group member stood up, approached the board and began explaining the new idea to the moderator. After a moment, the original moderator, seated himself to allow the other participant to temporarily moderate the discussion. Once the discussion shifted to another topic, the temporary moderator sat down, indicating that he no longer wanted to moderate the discussion, leaving no participant at the whiteboard to directly manage the construction of diagram. At this point, original moderator returned and recommenced his moderation activity. Throughout this entire activity, the diagram on the whiteboard was never directly manipulated by more than one user at a time.

Other natural coordination mechanisms emerged during this activity. When the group was (informally) brainstorming for new ideas, each idea was spoken aloud and the generation of new ideas was not parallelized. This sequential idea generation was how brainstorming was originally conceived by Alex Osborn [Osborn 57] so as to allow the group members to respond and build on the ideas of others (though many later revisions of the brainstorming procedure encourage parallel idea generation, by writing ideas, so as not to block the production of new ideas by waiting for speaking turns.) The group naturally coordinated itself in this manner due to the highly interdependent nature of this exercise. For example, in traditional brainstorming, an affinity diagram is used which can better support and categorize widely divergent information. In causal analysis, during the construction of the cause-and-effect diagram, each event is directly related to another event, this requires more coordination between the users, as one idea impacts another. Therefore, they managed this through turn-taking, verbally expressing their idea, and building off of these ideas.

1.2.4 Learning/cognition/articulation

Because the teams used whiteboards, pens, and Post-it notes, no learning of new technology was required.

The majority of the discussion was dedicated to two topics. The first topic of frequent discussion was balancing the two sub-goals, i.e. the completeness and the structural soundness of the diagram. With every modification that was made, the diagram took new meaning and required reevaluation. The group asked themselves question directly addressing this matter. i.e. “Are we missing some events before

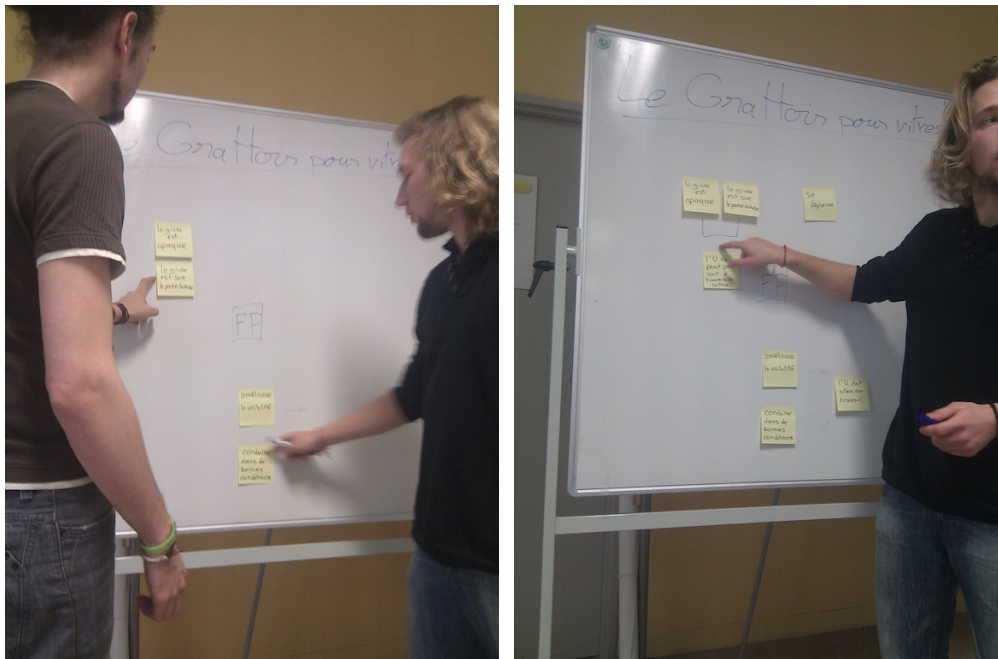


Figure 1.2.5: The role of the moderator can naturally and temporarily rotate among participants.

this event?” (completeness) or “Don’t these two causes have an ‘or’ relationship instead of an ‘and’” (soundness). These can be seen respectively as a diverging and converging activity. The technology was used to support each of these activities of discussion. When divergence was required, the moderator would ask the group for possible causes of a particular event on the board. When convergence was required, the moderator would reconfigure the diagram, by drawing/erasing lines or repositioning Post-it notes according to an active hypothesis proposed by a group member. This allowed for the current discussion to be externalized using the artifacts representing the diagram which could be then be interpreted by the group. The group members maintained internalized notions of the chain of causality, which was shared through constant discussion regarding their interpretations of the diagram. Participants would sometimes “position” themselves at one of the nodes, and discuss the direct causes of this event and the effects of this event.

The second topic of discussion was trying to distinguish cause from effect. Though this is related to the structural soundness of the diagram, this discussion was more abstract, and not framed directly around the causal relation of two specific events. The exercise of temporally organizing cause and effect might seem straightforward, in reality complications of interpretation arise, and the diagram played a role in highlighting these discrepancies.

For example, the group discussed at length the user's need to use his car. Was the user scraping the ice off the windshield "...*because* he wants to use his car" or "...*so that* he may use his car" ? The discussion began when moderator took the "cause" in question: "the user wants to use his car" and placed it in a general area under the principal function so that it was now an "effect". This is an exploratory epistemic action. This new "effect" was then be compared to the other effects in its proximity in order to better understand its new relationships. After some discussion comparing this effect to other effects, the group determines this event should be a cause after all, and the moderator moves it back up to the top of the diagram. Then they create a new event "...so that the user may go to work" and add it the effects of the diagram (a goal-directed pragmatic action). The diagram plays a central role in this deliberation because it allowed the participants to tentatively externalize a potential configuration and make new connections simply through their reflection of the diagram.

1.2.5 Development

The effort that participants put into learning this system is relatively small, when compared to the effort required to learn the brown paper method of project planning (c.f. the following section). Unlike the observation of the project planning which required some explanation to understand the procedure, no additional training or instruction after the two-hour lecture on how to set up the causal analysis was required. The use of these tools in building the diagram was mutually understood by all the participants.

Development in Activity Theory is used to describe the evolution of the activity and tool-mediated interaction over a period of time typically much longer than 90 minutes. Still, within this time frame, some developmental aspects of the activity do appear. As participants advance in their understanding of the chain of causality, larger pieces of the diagram were manipulated albeit with more difficulty. For example, collections of events, instead of individual events, are repositioned on the board and this makes it more difficult for the group to consider epistemic, exploratory actions.

Motivation varied throughout the session, and generally decreased as time progressed. When such discussions began slowing down, some participants would become less active in the discussion and reread their lecture notes and the problem description on the tabletop, until the moderator intervened and moved the discussion along.

The reason for the decrease in motivation and participation is open to interpretation, but I primarily attribute it to two reasons. First, the involvement can be impacted by the demands of the roles played by the participants. For example, the moderator and the scribe remained the most engaged throughout the activity, as

they had to be responsible for maintaining the diagram on the whiteboard and the bank of Post-its on the tabletop, respectively. Second, as time progressed, manipulating the diagram required more effort. Pragmatic actions were slower, simply because it was easier to reason about a diagram with three components than it was to reason about a diagram with fifteen or twenty. Similarly, epistemic actions are slower as it is more difficult to explore different configurations involving more Post-its and then revert back to the original configuration if necessary.

1.2.6 Summary

In summary, below is a list of the more relevant points that are taken away from the observation of the causal analysis session, as they relate to activity theory:

- Requiring the use of a cause-and-effect diagram as a tool, encourages the group to think pragmatically about the problem.
- Requiring that the cause-and-effect diagram be constructed with Post-it notes and dry-erase markers, encourages the group to use epistemic actions during its construction.
- When building the diagram using the whiteboard and Post-it notes as tools, the diagram's formal semantic structure is less relevant than the improvisational use of the Post-it note's position and spatial arrangement to communicate meaning.
- The diagram on the whiteboard is a tool used by the moderator to visually support the discussion at hand, and to promote reflection and convergence through common objectification.
- The group work conducted on the vertical surface (diagram structurization) is convergent in nature, while the group work conducted on the horizontal surface (idea generation) is divergent in nature.
- The role of the scribe emerged naturally and facilitated the creation and management of a bank of Post-its on surface of the tabletop near the whiteboard. This led to much more control and involvement being attributed to the scribe, than to the other participants.
- All participants begin with equal knowledge and ability to physically manipulate all tools in the environment (pens, Post-it notes, whiteboard). In the anticipation of the introduction of new technology, this level of accessibility should be considered.

1.3 Observation of Project Planning using the “Brown Paper” method

The observation of a project planning process was conducted over a two 3-hour periods with a 1-week long break in between the two. The participants of the observation were four engineering students (three males, one female) from the Université de Technologie de Compiègne. The participants knew each other and were all enrolled in a project management course. Several aspects of the observations were simulated. These were communicated in a design brief which explained the context of the exercise and asked for certain deliverables.

The primary method of capturing information was a handheld camera, as well as note-taking, with special consideration to the issues raised in the activity checklist.

1.3.1 General Overview

The objective of this observation was to see how participants coordinated and completed work using the brown paper method of collaborative project planning. The brown paper method provides a group with a way to visualize the entire process of project, and does so in a way that does not bely the heavy coordination and interdependence that is present among the multiple departments of an organization. It serves best as an exercise at the beginning of the project, during a meeting in which all department chiefs and relevant stakeholders are together, so that each department can contribute their own tasks and assert their influence in the planning process from the start. The brown paper method is much better suited when there is an existing project process already in place so that it may be optimized, and is generally not recommended as a tool that a project planner can use to start from scratch.

The brown paper method is one that has been informally developed by project managers and consultants, and does not have formal procedures or specifications attached to it. The name of the brown paper method refers to the brown butcher’s paper that is attached to a wall to serve as a surface on which a diagram is built. The diagram is a specialized version of a flowchart which contains project *tasks* and *milestones*, and the *dependency* relationships between them (Figure 1.3.1). The flowchart is placed inside a matrix, with an organization’s departments on the y-axis, and the phases of the project along the x-axis. The location of a task, or tasks, inside a cell indicates it will be completed during the column’s phase under the responsibility of the row’s department. Milestones denote points in time where management will review existing work and decide if the project can continue into the following phase. A milestone can be considered as a task with a duration of zero that must be completed by the organization’s higher level management,

and therefore almost always appears in the highest row, which by convention is reserved for the department of direction.

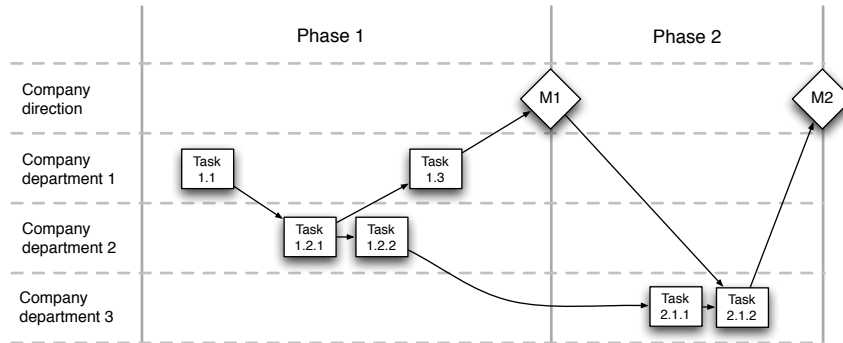


Figure 1.3.1: An example of the brown paper model for two phases of project.

There is no formal and standardized version of the brown paper process. Customization might be appropriate depending on the discretion of the project leader and the group. One such customization is an extension which allows the brown paper method to be used for PERT analysis. PERT, or “Program Evaluation and Review Technique”, is a process which uses the tasks’ durations and dependencies (successors and predecessors) to calculate the “critical path” i.e. the list of consecutive tasks which make up the maximum project duration. Having this information is useful, because a group can focus on either parallelizing these tasks or removing them (if possible), to shorten the duration of the project. PERT analysis requires calculating additional information for each task, such as the earliest start date, earliest end date, latest start date, latest end date, free float, and total float. In the brown paper method, these values are written on a Post-it which is placed next to the Post-it of the associated task (Figure 1.3.2). Here, *free float* is the maximum time a task can be delayed without subsequently delaying any of its successors, and *total float* is the maximum amount of time a task can be delayed with delaying the entire project. These two pieces of information are especially helpful when determining the critical path and reducing project duration. The formulas for calculating this information are based on the values of its predecessors (for the earliest state date, earliest finish date and free float) or its successors (for the latest start date and latest end date), this requires at least two passes through the network where every node is visited (one pass from starting milestone to ending milestone, and another from ending milestone to starting milestone). Figure 1.3.3 explains this calculation process in more detail.

The observations began with a 20-minute presentation of the brown paper method. Then, a design brief was distributed to the participants which explains


	Task ID	Task duration
	Earliest start date (ES)	Earliest finish date (EF)
	Latest start date (LS)	Latest finish date (LF)
	Free float (FF)	Total float (TF)

Figure 1.3.2: An example of a task Post-it in the brown paper method with the calculations from the PERT analysis displayed underneath (left) with the legend used denote the significance of the values (right) .

<p>1) Set the first task's ES to 0. 1st pass: Visit every task from first to last (in a breadth-first manner) and calculate ES and EF.</p> <p>ES = The latest EF of its predecessors EF = ES + task duration</p> <p>2) Set the last task's LF to its EF. 2nd pass: Visit every task from last to first (in a breadth-first manner) and calculate LS and LF.</p> <p>LF = The earliest LS its successors LS = LF - task duration</p> <p>3) 3rd pass: Visit every task and calculate FF and TF</p> <p>FF = The earliest ES of its predecessors - EF TF = LF - EF</p>

Figure 1.3.3: The procedure given to the participants instructing how to perform PERT analysis.

the following scenario: A medium-sized company has, in recent years, established itself as a specialized and important vendor of machinery and engines for heavy construction equipment. The director of the company wishes to branch out into a new product area and develop a new all-terrain vehicle (ATV). The director knows that the existing product development schedule takes two years to complete, and due to commercial and financial reasons, he hopes to reduce this to 18 months. The management committee designates a project leader who is charged with analyzing the existing development cycle (first three-hour session) and then optimizing it (second three-hour session). For this, the project leader has organized a brown paper session and requested the presence of key members from each department of his company. Because the project leader is trying to optimize an existing process, he has compiled a list of the 34 tasks (including their estimated duration) across 7 different phases which were completed by the company in previous product cycles. The design brief also explains some of the more nuanced scheduling conflicts and constraints inherent in the project so that participants can correctly design a coherent project workflow (e.g. “Purchases” can and should begin before “Detailed Design”, etc.).

The participants of the observation are provided with a meeting room with a table, a whiteboard, and an LCD display monitor mounted on the wall which the participants can either use with their own personal laptops as a second screen or with a computer provided for them. They also have brown paper, adhesive tape, scissors, several packets of yellow and pink Post-it notes and dry-erase and permanent markers. The participants bring with them their personal items (pen case, notebooks, project management textbooks and cellphones), as well as any notes that they have taken during the presentation of the brown paper method.

1.3.2 Means & Ends

In order to better analyze the project planning activity, a hierarchical task decomposition with five different activities has been produced (Figure 1.3.4). The final goal of this activity is to produce a more efficient project plan which has a duration of at most 18 months. This is done through meeting two sub-goals: first, the reduction of the existing project’s duration by as many weeks are possible, and second, the production of a project workflow which correctly accounts for all environmental and organizational constraints, as explained by the design brief.

The design brief requests certain deliverables from the participants. For the first session, participants are required to submit two intermediary deliverables by paper: an organigram of the company and a Gantt chart of the existing process that is constructed using the PERT analysis. The basic structure for the Gantt chart is provided, and the group only needs to pencil in the task durations at the appropriate point in the timeline. For the second session (dedicated to the

optimization of the workflow), the participants are required to submit a document summarizing the new project and any impact it has on the organization of company (new organigram) as well as a Microsoft Project file with the revised project workflow.

The principal method of evaluation employed by the participants is to calculate the project duration and the critical path. This allows the participants to understand how close they are to meeting their goal of an 24-month duration and where they can make improvements. Due to the nature of the procedure for performing the PERT analysis, it is difficult to validate the correctness of the entire network calculations until the end of the second pass, where the first task’s earliest start date and its latest start date should equal zero.

To complement the hierarchical task decomposition, the remainder of this subsection provides a textual description of how the activity unfolded in the environment, with photographs in Figure 1.3.1. After the presentation of the brown paper method, the participants sat themselves around the tabletop in the meeting room. They cut a large piece of brown paper and fastened it to the wall. The participants then read their design briefs, discussed the objectives, and asked each other questions to clarify their objectives and their understanding of the context of the exercise. During these discussions, they elected a moderator. One participant begins using the whiteboard to sketch the organigram (task 1.2), while two others (moderator included) are seated at the table generating Post-its with the task names written on them (as given to them by the list in the design brief) (task 2.2). These Post-its were immediately placed in tentative positions on the brown paper. Once the participant who was working on organigram was finished, he was joined by the moderator and together they began assigning which project task would be assigned to what department (task 1.3). With the organigram finished, another participant began preparing the matrix of the brown paper by using a permanent marker to write organization’s departments on the left and the phases on the top (Figure 1.3.5).

With approximately 1 hour and 45 minutes remaining, the group had completed the first step, as well as tasks 2.1, 2.2, 2.3 and 2.4 (according to the task decomposition). Three group members spent 30 minutes placing each of the 34 task Post-its into the flowchart (task 2.5), while one participant begins preparing the matrices with each task’s ID and duration for the PERT Post-its (Figure 1.3.2). After the structure of the flowchart was in place, they began discussing how best to proceed with the calculations in the PERT analysis. They realized that they may be able to bypass some of the tedious PERT calculations by using the model of the Gantt chart that was provided to them. Therefore, two group members begin filling out the Gantt chart by hand at the table, while the other two members begin performing the PERT calculations on the brown paper.

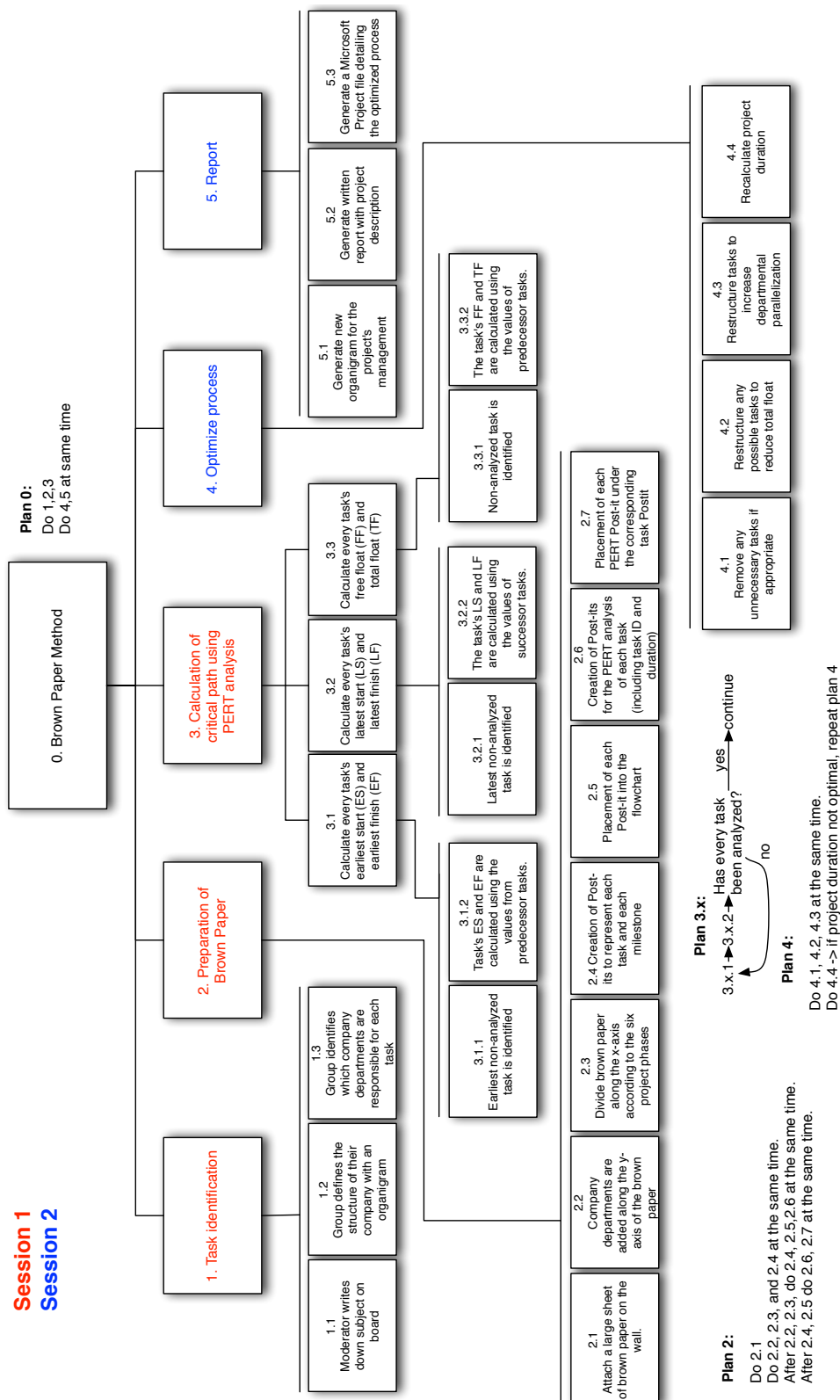


Figure 1.3.4: Hierarchical task decomposition for project planning activity.

With 45 minutes remaining they have completed the first pass of PERT calculations, as well as the Gantt chart. Upon completing the second pass, they discovered that the first task has a negative latest start date value. This indicated that they have made an error somewhere in the calculations of the PERT analysis. They began revisiting all their calculations, cross-referencing them with their Gantt chart, and discovering calculation errors in about half of the PERT calculations. Therefore, they spent the remaining time correcting these errors, which involved the time-consuming recreation of PERT Post-its. This occupied the group for the remaining time. At the end of the session they removed the brown paper from the wall and rolled it up for storage, concluding the first session.

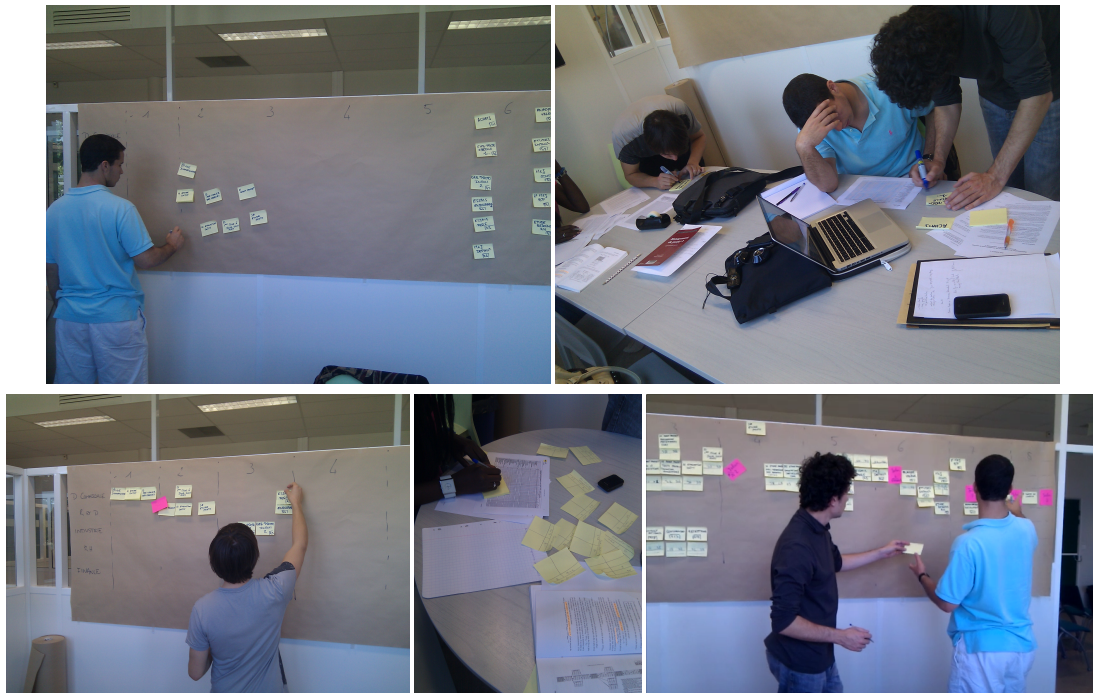


Figure 1.3.5: Preparation of brown paper. From left to right, by row: participant divides brown paper along x-axis to denote project phases (task 2.3); two participants generate task Post-it notes (task 2.4); participant places tasks into the flowchart (task 2.5); participant generates PERT Post-its (task 2.6); participants place PERT Post-its under their corresponding task (task 2.7).

The second session last 2 hours and 45 minutes and occurred one week after the first session. The participants returned to the same room, with the same materials provided to them. The group began this second session by unrolling and reattaching the brown paper. Any Post-its which were no longer properly adhering

to the paper were secured with scotch tape. Then, they read their second design brief which contained more information and instruction relative to the second session.

Instead of restructuring the flowchart (task 4), they decided to begin by drafting a first version of the project report (task 5.1), so as to clarify issues and constraints which might later help during the optimization. Two participants shared the computer terminal to begin writing the document outline and project description in a Microsoft Word document, while the other two use pen and paper to begin sketching the updated version of the organigram. This takes a great deal of time, as they review the project brief and their text books in order to complete their task.

After an hour, one group member opened his laptop to begin preparing a Microsoft Project file. They did this before the optimization of the flowchart began so as to have a electronic version of the flowchart ready. It was decided that the electronic version could be updated conjunction with the brown paper, so as to provide automatic project duration calculations. This involves retyping and structuring all of the tasks in Microsoft Project.

When the Microsoft Project file is nearly ready, they began restructuring the flowchart. First, they removed two of the seven milestones, as instructed by the design brief. Then, one participant at the brown paper, began taking each task individually, reading it out loud, and moving it to its appropriate position in the workflow, which may or may not be contested by other participants. After a first pass is complete, they recalculated the project duration using the Microsoft Project document. This involved having one participant reading out loud each task and its successors, so that the other participant who is building the Microsoft Project document can copy this information into the digital flowchart.

Once the project duration calculation is completed a first time, the team determines the project duration is still too long. They began reinvestigating tasks on the critical path, to understand better where to focus their optimization efforts. Upon reaching a suitable project duration, the participants spent the remaining ten minutes finalizing the report for submission.

1.3.3 Environment

The most important tool for collaboration in the brown paper method is the brown paper itself. Activity around the brown paper was of two different kinds. First, there was a considerable amount of repetitive and procedural construction and calculation involved in actually building the basic components of the diagram, and, second, there was also a significant amount of interpretation and manipulation in order to identify and optimize the dependency relationships between the tasks and milestones.

This led to *two different patterns of coordination*. During procedural tasks, the group was highly parallelized, performing three or four different activities at once, and transferring their output to another. For example, one participant would hand the completed tasks in their chronological order to another, who would begin placing task Post-its in the diagram on the brown paper, while, on the tabletop, a third was drawing matrices for the PERT Post-its, and passing these to a fourth, who was adding task IDs and durations to the matrices.

For tasks which required more interpretation, work was typically coordinated among a sub-group of three participants. For example, in the first session, one participant stood at the brown paper moderating this activity, tracing the visual workflow task by task, searching for discrepancies and requesting information from his colleagues if required. Another participant would also stand near the first, following his activity and maintaining an overview of the brown paper, ready to provide clarification or to hand him a requested Post-it that was out of reach, and a third would be seated at the table, with the design brief in hand, ready to interject with information from the scenario when it would be relevant. This is an example of a highly (and naturally) coordinated task which spans two different supports (brown paper and design brief) and three participants.

One of the most significant breakdowns with the brown paper tool was the lack of explicitly defined successor and predecessor relationships. Normally, these would be drawn with a marker directly on the brown paper, but the group chose not to do so because permanently drawn lines could not follow the optimization process. Instead, most of the dependency relationships were communicated by placing tasks one after the other, even if they were not in the same row. This was mostly effective, except in cases where inserting a task in a workflow would then require shifting the remaining workflow by the width of one Post-it.

Several breakdowns also occurred around the PERT analysis. For example, when there was one mistake in the calculations which propagated throughout the entire network, they could not detect the presence of the error until they reached the end of the second pass. Once they had retraced their calculations, and found the Post-its which were affected, they tried crossing out values on the PERT Post-it, which led to them being less easy to read. They finally discarded these Post-its and began recreating new PERT Post-its to replace them.

Much later, while they were performing the optimization task in the second session, they found that recalculations would be too time-intensive and chose to abandon the manual calculation of the critical path (they even spent time tearing down the PERT matrices that were taped to the task Post-its). The critical path was still of great interest to them, which is why they began maintaining a digital version of the workflow in Microsoft Project during the optimization process. This in and of itself became an activity which required its own coordination. Often, the

participant seated at his laptop could not successfully maintain focus across the activities of the optimization process and his own personal work in Microsoft Project, and therefore would stop the group intermittently so he could catch up. During these periods, the moderator would verbally retrace the workflow on the brown paper, task by task, so the participants could check for inconsistencies present in the Microsoft Project diagram. Once they had successfully manually synchronized the diagrams, the Microsoft Project document was useful for providing information at the request of the moderator or other participants regarding the critical path and project duration.

Another tool in the environment was the tabletop around which most of the tasks that were procedural and easily parallelized were conducted (e.g. creation of all Post-it notes). The whiteboard was used to construct the first version of the organigram, and then assign each of the tasks to their respective departments. This made for an easily glanceable diagram which would assist in the organization of tasks on the brown paper. The workstation with the overhead monitor was used to write the project report. Though there were ergonomic issues with this setup (e.g. the keyboard had to be taken into the participant's lap so that he may face the screen properly and the display was too high for the participants seated at the table), the tool still afforded collaboration for two people during the writing of the document.

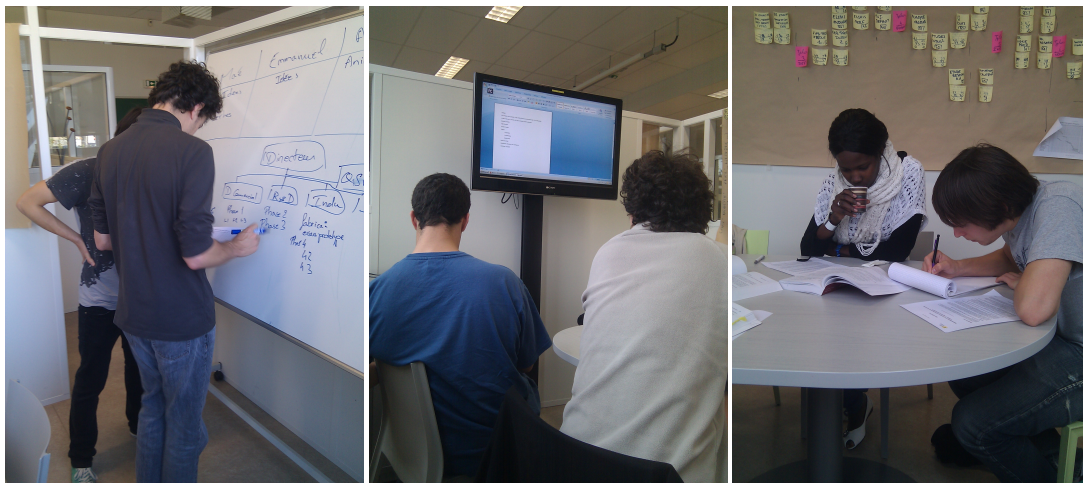


Figure 1.3.6: Use of the whiteboard for the task of building the organigram and performing task identification (1.2 & 1.3), the PC workstation with overhead display (5.2), and the tabletop during the construction of the revised organigram (5.1). These represent two-person collaborative efforts which involved interpretive tasks.

Division of labor was explicit at first. The newly elected moderator wrote down

the names of the participants on the whiteboard, with an assigned task for each participant. After the initial assignment, the whiteboard as a tool for coordinating activity became immediately irrelevant, and the remainder of the coordination was communicated verbally. Participants simply volunteered to do a task, which would be validated by the moderator, or the moderator would request that they perform a task. This was especially useful for coordinating procedural tasks. But even this method of coordination was increasingly loosely defined over time, especially once the moderator role began naturally rotating among participants. As time progressed, and with the entire workflow displayed on the brown paper, group awareness had increased to a significant point where even explicit verbal communication was no longer necessary for coordination. For example, the roles described in the interpretation and organization activity described in the first paragraph were all naturally developed. The role of the moderator was available to all members of the group, provided they had specific requests for action and information that would assist them in the cognitive task they were trying to complete.

1.3.4 Learning/cognition/articulation

Perhaps the most important aspect of the brown paper method that is externalized is the dependency network of the tasks. The ability to visualize the flow of the entire project on the wall articulated the current state of affairs and assisted in the more straightforward, procedural activities as much as the interpretative, reorganizational activities. For example, during the PERT analysis, each of the formulas were easily internalized and then were used to operate upon a task to produce the calculations. A participant would examine a task, examine its successors and then write down (externalize) the result of the mental calculations (internalization). Other participants could then validate the calculations, and follow the progress of the users. The more interpretative tasks, notably the optimization and reorganizational activity, were perhaps the only time that the participants were actively and verbally considering their two sub-goals (reduction of project duration and proper consideration for environmental and organization constraints). The optimization was directly performed by one participant who did his best to reduce the project as much as possible. During this process, two other participants, one maintaining an overview of the entire project, and the other relying on information in the design brief, assisted the first participant by interjecting with information regarding relevant constraints which might prohibited the action that is being performed. The brown paper was of great assistance in this process because it allowed the first participant to externalize a possible optimization of the dependency network, which could be internalized and validated by the two other participants. These concepts echo many other items on the activity checklist for cognition and problem articulation. Notably, it allows the group to simulate possible actions before they

are actually implemented by trying new, easily undone, spatial configurations with the Post-it notes. It also allows the group to monitor and reflect upon what is being externalized, and naturally provide assistance and coordination to assist in the activity.

Some aspects of the brown paper project planning make it difficult to follow. For example, participants used flexible and informal notion of dependency, which were communicated not through lines drawn on the board, but rather through the spatial configuration of the Post-its. Understanding the explicit nature of successor and predecessors relationships required time and attention. When participants would mental trace their way through their diagram, they did so with careful focus so as not to miss a dependency relationship.

1.3.5 Development

Because the activity took place over two different sessions, participants were required to clean up the meeting room at the end of the first session and setup their environment again in the beginning of the second session. This involved detaching the brown paper from the wall, rolling it up with the Post-it notes still attached and storing it in a safe place. Then, when the brown paper was unrolled and re-attached to the wall, the Post-it notes were bent and lost their ability to adhere to the brown paper, causing several Post-it to fall. This was quickly rectified by reinforcing the Post-its with tape, which meant they were more difficult to move later, during optimization.

The session began with heavy involvement from the moderator for coordinating roles and parallel work. This gradually decreased over time because once the diagram was partially completed, the group had an overview of the work that was done, and could decide for themselves what they could do next to help their team progress (discussed in Section 1.3.3).

1.3.6 Summary

Observing collaborative project planning provided an example of a much more complex and varied activity than causal analysis. In summary, below is a list of the more relevant points that are taken away from the observation of the project planning session, as they relate to activity theory:

- Coordination can be explicitly enforced by the moderator, or simply deduced by observing an overview of the state of the work and deciding what needs to be done.
- During highly collaborative activities, people assumed direct responsibility for certain information resources. This *social mapping of resources* can

be seen, for example, during optimization, when one person manages the flowchart, another the design brief, and another the Microsoft Project file.

- Similar to causal analysis, participants relied upon the use of position and spatial arrangements of Post-its more than explicitly defined lines representing dependencies. This allowed the components of the flowchart to remain easily manipulable and support epistemic actions.
- Similar to causal analysis, the group work conducted on the vertical surface (diagram structurization) is convergent in nature, while the group work conducted on the horizontal surface (task generation, component building) is highly parallelized by the participants.
- Because there were no explicit lines marking dependencies, this led to errors in the calculations of PERT analysis.
- PERT calculations revealed the critical path of a project, and allowed the team to be more pragmatic in its optimization. Manual calculations required too much effort and therefore, participants maintained a paper project flowchart and a digital Microsoft Project file simultaneously.

1.4 Conclusion

How collaborative work can span multiple surfaces, and assist in coordination, problem articulation, and communication is of great interest to this thesis. The whiteboard in the causal analysis and the wall in the brown paper method, as well as the role of the moderator, were instrumental in promoting awareness, reflection, and an overview of the status of the group's progress. The tabletop facilitated parallel and procedural work which was then fed into the diagram on the vertical surface. With these tools participants acted naturally through informal roles which collectively advanced the group toward meeting their goals. These are all aspects which properly designed groupware must preserve.

At a first glance, building diagrams with Post-it notes and dry-erase markers appears to have many advantages. They are expressive, accessible, colorful, and versatile but upon closer examination, they require an excessive amount of preparation, overhead and upkeep, especially as the diagrams grow more complex. Moreover, particularly in the case of project planning, transferring these diagrams into a digital form so that they may be preserved and used post-activity requires a significant amount of time and effort. Also, the time-consuming and error prone PERT calculations caused them to be quickly abandoned. These are all aspects that may be facilitated by computer support.

Chapter 2

Related work in interactive spaces

After observing traditional work environments and activities, I now turn to the related work in interactive spaces for co-located collaboration. Interactive spaces draws from wide range of different fields - CSCW, HCI, ambient intelligence, and social sciences to name a few - which would make an exhaustive review of the literature infeasible. I use the lessons learnt from conducting the previous observations of traditional work to help curate the selection and to help bridge of research which has occurred across multiple domains.

At its foundation, research on the theme of interactive spaces is perhaps most indebted to Mark Weiser's vision of *ubiquitous computing*. Weiser was interested in the new possibilities of computing systems afforded by wireless networking, increasingly portable computers, and touchscreen interfaces, and believed these devices could bring a new paradigm on interaction beyond personal computing. In his 1991 seminal paper *Computer for the 21st century* [Weiser 91], envisioned a ubiquitous computing environment where processors and radio receivers were hidden from users and embedded into a multitude of everyday devices which could provide seamless and concerted interaction. He proposed a coarse classification for the main input and output devices in this environment: tabs, small handheld badge-sized personal devices; pads, medium paper-sized devices; and boards, large shared displays such as interactive whiteboards, all of which were context-aware and could respond to interactions such as direct touch, pen, and even voice. It was remarkably forward-thinking at the time, and also, with the rise of commercial smartphones, tablets and interactive whiteboards, quite accurate. As a result, Weiser's proposition of a ubiquitous computing paradigm opened a wealth of new research questions, and over the two decades since these initial proposals, software engineers, interface designers and social scientists continue to make advancements in their understanding of these environments and the challenges they present.

At a recent international workshop on the design of interactive spaces for collab-

orative work, creativity and learning, the central role of ubiquitous computing was acknowledged by the organizers. In their closing remarks, they also captured why designing ubiquitous computing environments for co-located work is so difficult [Jetter 12a]. Such environments require a holistic understanding of

1. a user's individual interaction,
2. the users' social interaction and communication,
3. the workflows inside the organization and
4. the physical environment.

Proper consideration for these four domains provides a comprehensive view of interactive spaces. Though they are difficult to consider atomically, as a design choice in one category will impact the other three, I use this to help frame the review of the state-of-the-art in interactive spaces.

The following subsection will address the state-of-the-art of a user's individual interaction through multitouch and multimodal gestures. The second subsection addresses social interaction and collaboration with multi-user groupware. The third subsection discusses workflows and organizational factors that impact a collaborative environment. The final subsection will examine how the design factors in the first three leverage the physical environment to augment co-located collaboration.

2.1 State-of-the-art of multitouch and multimodal gestures

Multitouch input devices have a long history in research and development (even longer than the mouse) but they have only entered the public consciousness recently. One of the most compelling and highly publicized demonstrations of multitouch interaction was done by Jeff Han at a TED conference in 2006. Han presented a new interactive tabletop capable of fluid, 10-finger interaction on a large surface at an affordable price [Han 06]. Following the presentation, Han formed his own company, Perspective Pixel, and continued to make such tabletops, notably for media outlets such as CNN, allowing them to gain even more exposure and eventually leading to Han appearing on *Time* magazine's list of the *100 Most Influential People in the World* [King 08].

Though his tabletops were specialized and not sold commercially, the average user would have the opportunity to test multitouch interaction for themselves with the release of Apple's iPhone in 2007. Though not the first multitouch device on

the market, it sold remarkably well and packaged a handheld multitouch interface inside a mainstream product. Much was written about the emerging paradigm of interaction: users could tap buttons, pinch photographs and swipe through lists, providing an intuitive way to interact with components of the user interface. The potential for interaction designers is enormous as the number of unique gestures that a user can perform is essentially without limit.

These claims of intuitive interaction were not completely unfounded. Before multitouch interfaces made their way into commercial applications, a theoretical foundation for describing the intuitiveness of such interfaces was already well established in research communities. A notable example was with Bill Buxton's idea of chunking and phrasing [Buxton 86b]. A *chunk* represents a unit of interaction, and an interface designed to enhance the *phrasing* of these chunks could allow the user to string together small chunks to form larger chunks. Gesture based interfaces, which engage the user's motor system and enhance learning, help the user associate action to outcome. Kurtenbach and Buxton proposed *marking menus*, resembling hierarchical pie menus, to show how users can navigate through several levels of items and, after practice, use seemingly atomic gestures to trigger actions in a multi-level menu as if it were one chunk of interaction [Kurtenbach 93]. (Khurruafa points out that the idea behind Buxton's chunking and phrasing would later parallel developments in Activity Theory for HCI, particular with regards to actions and operations and a user's development [Kharrufa 10].)

Yet even for the level of intuition it leverages, multitouch remains a generally tricky, highly researched, sometimes controversial form of interaction, especially when the technology is translated to large board-sized surfaces. As the present research joins this effort, it will be critical to examine the advantages and challenges that are inherent in designing for this input device.

2.1.1 Designing multitouch gesture sets

When designing an application for a board-size multitouch device, one of the first objectives that an interaction designer or software developer must accomplish is to select a coherent set of multitouch gestures and their associated user interface components. At the start of my research, my intuition was to design a vocabulary of multitouch gestures by analyzing and classifying experimental gestures from an extensive literature review. I felt that if I could visualize all the gesture *actions* (e.g. one-finger tap, one-finger hold, two-finger pinch, and so forth) and their *outcomes* (e.g. selection, mode switch, zoom), patterns would emerge, and a multitouch grammar could be defined.

I quickly found this approach, and several others, had already been used to define multitouch gesture sets, with limited success. These approaches to the problem are important because they are successful in producing a taxonomy of gesture actions.

Connecting these actions to a coherent set of *outcomes*, however, is a goal that is wrought with numerous conflicts and challenges.

2.1.1.1 Norms and conflicts in gesture sets

An application designer who must choose a set of multi-touch gestures and the associated user interface will encounter a number of problems. First and foremost is a problem of norms in existing gesture sets. Ingram et al. [Ingram 12] examines this problem by surveying the literature and documenting all mentions of gesture *actions* and their *outcome* throughout commercial operating systems and research propositions. They found that commercial solutions for mobile operating systems (Apple's iOS, Microsoft's Windows Phone 7 and Google's Android) generally provide consistent action-outcome implementations of gestures such as tap, double tap, dwell, drag, flick and pinch.

Comparison with the gesture sets of their desktop counterparts, however, produces *conflicts across different interfaces*. For example, a five-finger scrunch on the tablet version of Apple's iOS will close the current application, while the same gesture on the multi-touch trackpad in OSX 10.8 will open an application launcher (Launchpad). This gesture also provides an example of another problem: *conflicts within the same interface*. An application designer for Apple's iPad cannot include five-finger multi-touch gestures in their own application, as it conflicts with the five-finger scrunch gesture of the operating system.

Outside of commercial applications, Ingram et al. [Ingram 12] indicate that research applications are decidedly worse in maintaining consistency in multi-touch interaction. They illustrate this claim by examining the task of rotating the user viewpoint. Such a rotation is implemented with as much as six different gestures in research applications: one finger drag, two finger rotational drag, three finger drag, multiple finger rotational drag, one finger hold with one finger drag, flat hand rotation. This is an example of six different actions mapping to the same outcome.

For another example, depending on the research application, the gesture of a one-finger drag can produce outcomes of selection, camera panning, camera rotation, and object displacement. This is an example of the one action mapping to many different outcomes.

Research teams are understandably working independently of one another, with an objective of proposing and researching specialized and experimental interaction techniques, and this makes gathering these interaction techniques into the context of a single application difficult. Researchers are beginning to take notice of these problems. Their sentiment is shared by Wigdor et al. [Wigdor 09a] in an introductory paper for the *CHI Special Interest Group for Designing User Interfaces for Multi-touch and Gesture Devices* where they explain that “with this increase

of devices and interfaces, comes an unfortunate increase of non-standard design patterns and methods of user interaction.”

2.1.1.2 Recognition vs. recall in Natural User Interfaces

Compounding these conflicts in gesture sets, are inherent problems of recognition vs. recall. This problem is better framed and more pronounced when discussing *natural user interfaces* (NUI), which Donald Norman [Norman 10] describes as a nebulously defined umbrella term used equally by designers and marketers. I will defer to the NUI Group Community’s definition:

Natural User Interface (NUI) is an emerging concept in Human-Computer Interaction that refers to a interface that is effectively invisible, or becomes invisible to its user with successive learned interactions. A NUI relies on a user being able to carry out relatively natural motions, movements or gestures that they quickly discover [and] control the computer application or manipulate the digital content.[nui 09]

This is a good definition because it hints at some potential modalities of interaction, and it expresses the importance of allowing users to leverage their pre-existing skills that they have acquired through a lifetime of interaction with their physical environment. This is one of the most important advantages of NUIs.

But this definition also captures some of the design trade-offs that occur with NUIs. Norman, in a journal article bluntly titled *Natural User Interfaces Are Not Natural*, explains that there are inherent problems with NUIs particularly with regards to the user’s visibility of available actions [Norman 10]. NUIs which incorrectly emphasize the invisibility of the user interface are in violation of the design principal of “*recognition rather than recall*” that was popularized by Jakob Nielsen’s heuristic evaluations [Nielsen 94].

Lucas Mathis in his book *Designed for Use* [Mathis 11] puts this problem in context with previous paradigms of interaction when he compares command-line (CLI) interfaces with graphical user interfaces (GUIs) and NUIs. He explains how CLIs oblige users to recall their commands while GUIs, in particular the ubiquitous WIMP (windows, icons, menus, pointing device) style of GUIs, provides users with visual menus of actions allowing users to recognize the commands (Figure 2.1.1). As recognition is more efficient than recall, these interfaces provided users with the possibility to explore and discover new functionality, through better visibility [Norman 10, Nielsen 94], without necessarily having to execute the command to discover its function. With NUIs, Mathis argues users return to something similar to command-line interfaces, where users must depend on recall with the hope that the designers have accounted for their intuition, when attempting to manipulate interface components for the first time.

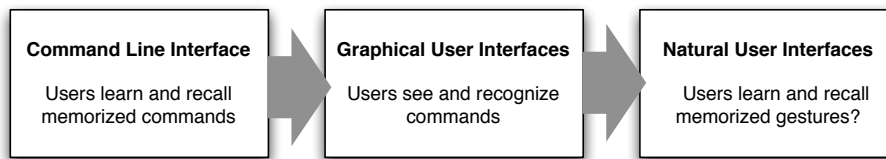


Figure 2.1.1: Progression of user interfaces with regards to their strategies for learning. Adapted from [Mathis 11].

2.1.1.3 User-defined gesture sets

These complications have pushed researchers to develop a coherent strategy for gesture sets for surface computing, and as a result there have been several approaches taken. Wobbrock et al. [Wobbrock 09] developed a taxonomy (Table 2.1) based on notions of *form*, for describing the physical aspects of the gesture in terms of pose, path and touch; *nature*, if the gesture is symbolic, metaphoric, physical, or abstract; *binding*, if the target object of the gesture represents an artifact or environment; and *flow*, which dictates if the response occurs while the gesture is active or after the gesture is completed. Note this taxonomy focuses exclusively on horizontal surface gestures on tabletop surfaces. This taxonomy is valuable for expressing the broad range of gestures and interaction that an input device can achieve. It also provides a theoretical foundation on which user-centered experiments may be framed in order to understand the implications behind each class of gestures.

With this taxonomy, Wobbrock et al. performed an experiment where they presented participants with an animation of a particular “outcome” (moving an object, performing a pan, rotating an object), and then asked the user to perform a gesture which would produce the associated outcome (a strategy originally proposed by Good et al. [Good 84]). They find that 72% of the gestures invented by participants were one-point touch, and one-point path gestures. They also find that nearly half (44%) of the gestures were physical in nature. They then rank the agreeance of the gestures among the participants and select the most commonly agreed upon gesture in an effort to build a curated user-defined gesture set. Though they produce a gesture set that is conflict-free (i.e. no two outcomes share an action), the average level of agreeance is only 32% for one-handed gestures, and 28% for two-handed gestures. This result is significant because it shows that even if a designer chooses the most “intuitive” gesture set, it will likely not be intuitive for most users.

Regardless of this result, the theoretical grounding of this work and its outcomes have influenced other researchers to produce similar experiments. Frisch et al. [Frisch 09] reuse this show-the-outcome-and-ask-for-the-gesture experimental

Table 2.1: Taxonomy of surface gestures from Wobbrock et al. [Wobbrock 09].

TAXONOMY OF SURFACE GESTURES		
Form	<i>static pose</i>	Hand pose is held in one location
	<i>dynamic pose</i>	Hand pose changes in one location
	<i>static pose and path</i>	Hand pose is held as hand moves
	<i>dynamic pose and path</i>	Hand pose changes as hand moves
	<i>one-point touch</i>	Static pose with one finger
	<i>one-point path</i>	Static pose & path with one finger
Nature	<i>symbolic</i>	Gesture visually depicts a symbol
	<i>physical</i>	Gesture acts physically on objects
	<i>metaphorical</i>	Gesture indicates a metaphor
	<i>abstract</i>	Gesture-referent mapping is arbitrary
Binding	<i>object-centric</i>	Location defined w.r.t object features
	<i>world-dependent</i>	Location defined w.r.t world features
	<i>world-independent</i>	Location can ignore world features
	<i>mixed dependencies</i>	World-independent plus another
Flow	<i>discrete</i>	Response occurs after the user acts
	<i>continuous</i>	Response occurs while the user acts

protocol with the difference that they ask the participant to perform the gesture with one hand, with two hands, and then with a stylus in one hand. Ingram et al. [Ingram 12] compares Frisch et al.’s results [Frisch 09] with those of Wobbrock et al. [Wobbrock 09] and uncovers conflicts even between these user-defined gesture sets. For example, there were two solutions proposed for the copy action in each study (Figure 2.1.2). This provides further evidence that designing an “intuitive” gesture set is entirely dependent upon the *context* of the interaction, and this could potentially mean that this methodology for designing intuitive gestures sets is flawed because the results do not generalize across all applications.

Another approach is to design systems which help users learn complicated gestures. Norman cites gesture guides and proper feedback as two attributes that should be considered in to increase the acceptance of gesture-based interfaces by users [Norman 10]. Researchers such as Freeman et al. [Freeman 09] and Bragdon et al. [Bragdon 10] use a simplified version of Wobbrock et al.’s taxonomy [Wobbrock 09] in order to identify different classes of gestures which can be learned by users. They then construct a specialized learning environment with gesture guides which allow users to learn a large cross-section of different sets of gestures. These solutions have a similar flaw as the gestures they choose are performed only as arbitrary actions and without any link to any particular outcome. Though this was designed to have learned gestures generalized across different applications af-

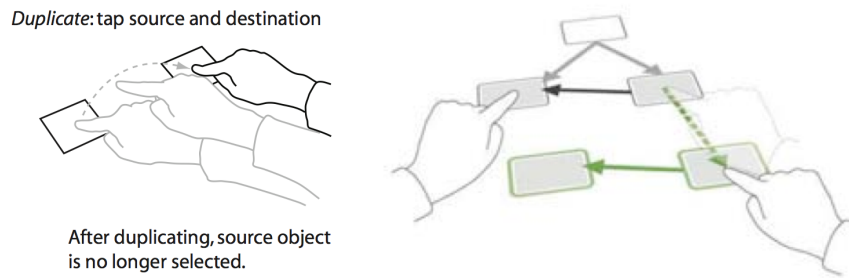


Figure 2.1.2: An example of a conflict between gesture sets: Wobbrock et al. [Wobbrock 09] proposed copy gesture (left) and Frisch et al. [Frisch 09] proposed copy gesture (right). Both gestures were user-defined, which means they were the most agreed upon gestures for the copy action in the respective studies. In both cases, users had little or no experience with multitouch interaction and they were allowed to use both hands. The main difference between the two experiments is the context of the node-link diagram in the case of Frisch et al. This illustrates the impact that context has on intuitive interaction.

ter they are committed to memory, they have removed all notions of the context in which the gestures are performed. More recently, Nacenta et al. [Nacenta 13] advocate using user-customizable gesture sets, where each user can create their own action-outcome bindings, and show next day recall rates of 97% in their experiments. Despite these encouraging results, Nacenta et al. also explain that there are many situations where user-customizable gesture sets are not desirable, for example, when consistency across new and different applications is important or in collaborative settings when awareness and transferability of other users' gestures should be encouraged [Nacenta 13].

With conflicting gesture sets and person-specific notions of intuitiveness, the idea of having an “invisible” interface through which a user can naturally interact is unreasonable, particularly when considering the complexity of the tasks witnessed in the user observations. This does not mean that NUIs are to be avoided, but that NUIs do have some inherent and unobvious limitations. Norman explains that gestural systems are one of the most important future paths for more holistic human interaction with technology, and that conventions and standards, rather than optimization, will be more important for progress. He concludes his article by writing “Are natural user interfaces natural? No. But they will be useful.” This advocates a more reflective and principled approach in designing gesture interfaces so that the progress achieved with standard human-computer interaction principles are considered.

2.1.2 Design principles for multitouch gestures

If an all-encompassing, intuitive gesture set is not feasible without ambiguities and conflicts, there has been much research dedicated to uncovering several design principals to help understand how users interact with tabletop surfaces *within* the context of the task they are try to perform.

2.1.2.1 Direct manipulation:

Ben Shneiderman introduced the notion of direct manipulation interfaces. These feature a continuous representation of the object of interest, physical actions or labeled button presses instead of complex syntax, and rapid incremental, reversible operation with immediate visual feedback [Shneiderman 83]. For tactile surfaces, according to Wobbrock et al's taxonomy [Wobbrock 09], this would imply interactions which are physical in nature, object-centric in binding, and continuous in flow. This is confirmed by their own experiments, where they find a plurality of gestures are physical and object-centric and nearly half were continuous.

Moreover, Ingram et al. [Ingram 12] found the most agreed upon gestures, whether proposed by users or designers, to be a one-finger drag for moving objects, and the two-finger spread for expanding (or zooming) objects. These gestures are also highly visible and easily understood, which would mean they would be suitable for transferability from expert users to novice users.

2.1.2.2 Physics-based gestures:

Building on the notion of physical actions from Shneiderman [Shneiderman 83], are such the concepts of pseudo-physicality and meta-physicality as discussed by Hilliges et al. [Hilliges 07]. *Pseudo-physicality* describes visual elements which resemble their real-world counterparts and afford a certain class of pseudo-physical interaction, (i.e. a virtual Post-it note can be dragged and moved on the surface of the tabletop). *Meta-physicality* refers to the behavior of virtual components which may not have any meaning in the physical world, but maintain a high-fidelity to the effects of the virtual environment. Hilliges et al. find that "as long as objects have a clearly distinct and explainable behavior, users seem to be willing to accept and use a technique even if it is unrealistic in the strict sense" [Hilliges 07]. This means that even though a physical Post-it cannot be slid across a tabletop and bounce at the edge of the surface, the behavior of the virtual Post-it can adhere to the effects of a physics-simulation engine without a detrimental impact on the intuitiveness of the interaction.

This is again supported by the findings of Wobbrock et al. [Wobbrock 09], who state that users found physical gestures to be the most intuitive category of gesture nature (more so than symbolic, metaphorical, and abstract) and were employed

nearly half of the time (43%). The instances where users did not use physical gesture were only for actions which had an abstract outcome (e.g. duplicating a Post-it cannot be done in reality). The intuitiveness of physics gestures can be accentuate when the outcomes coherently accounts for gesture size and speed.

It is worth noting that the nomenclature of such interactions are not entirely clear: Kruger et al. [Kruger 04] use “pseudo-physics” to describe “meta-physics”, and Butz [Butz 10] distinguish some aspects of meta-physicality with hyper-physicality.

2.1.2.3 “Big” and “small” gestures:

Wobbrock et al. [Wobbrock 09] found that users intuitively seemed to distinguish only two broad categories of gesture: touches with one, two or three fingers and touches with five fingers or whole handed gestures. This corroborates with Hinrichs et al.’s ethnographic studies which found that users “in the wild” do not distinguish between a one-finger drag and a three-finger drag of an object [Hinrichs 11]. This also seems to align itself with the concept of pseudo-physicality as a physical piece of paper can be slid across a tabletop with more than just one finger.

There also seems to be a notion that larger gestures imply a higher level of interaction. Ingram et al. [Ingram 12] explains how this notion is integrated into several commercial systems such as iOS and OSX, where four and five finger gestures are reserved for window and desktop management functionality.

2.1.2.4 Bi-manual manipulation:

Some studies, such as Hinrichs et al. [Hinrichs 11], have also found that users primarily interact with only one hand. There are some limitations to this conclusion because it was derived from experiments which used novice users in a casual social setting who were unfamiliar with interactive tabletops rather than expert users in a work-related context. This indicates that such a system might want to support one-handed interaction for novice users, and consider bi-manual interaction as the users progress in the learning of the interface.

Bi-manual interaction can either be *synchronous*, where both hands perform the same or similar tasks (such as a two-handed zoom gesture or displacing multiple widgets with two hands) or *asynchronous*, where each hand is performing two different and independent components of a single gesture (discussed in the following subsection). The advantage of using bi-manual interaction, as explained by Buxton et al. [Buxton 86a], is that, by engaging both hands, users can perform two complementary and simultaneous tasks in a manner which reflects how users interact with objects in everyday tasks. Users naturally understand how to channel different tasks to each of their hands and supporting such natural actions can lead

to more effective interaction.

2.1.2.5 A design caveat: *prior experience*

It's important to note that previous user experience will play an important part throughout these four principles. Mauney et al. [Mauney 10] noted that 70% of users who owned a multi-touch device swiped up to scroll down, while 50% of users without a multi-touch device did the opposite. Wobbrock et al. [Wobbrock 09] observed participants tapping index and middle fingers as if clicking, and using imaginary widgets at the top-right and bottom left of the screen for task switching and closing. Mauney et al. [Mauney 10] also found there may be some cultural differences for what is perceived to be intuitive. For example, in their study participants with Chinese as their native language used significantly more gestures that were symbolic in nature.

2.1.3 Extending direct manipulation through instrumental interaction

The advantages of using an approach with direct manipulation techniques are clear, but such an approach is also limited, particularly for when interactions imply outcomes outside the physical realm, or when one direct manipulation technique conflicts with another. Consequently, as indicated by Freeman et al. [Freeman 09], few commercial devices have included multi-touch gestures beyond the basic spatial manipulations described by Shneiderman [Shneiderman 83]. This frequently leads to direct-touch interaction techniques to be multiplexed, through mode switching. Mode switching is a popular research question for gestural interfaces, as each mode can change the outcomes of actions and unlock new functionality. Therefore, there has been much research in augmenting direct manipulation with other modalities and techniques.

Beaudouin-Lafon's instrumental interaction [Beaudouin-Lafon 00] conceptualizes tool-use in WIMP interfaces in a manner which extends them to post-WIMP interfaces. Here, "tool use" mediates activity between the user and domain objects. WIMP interfaces do use some concepts of direct manipulation, but extend direct manipulation by offloading a great deal of functionality into tools such as menus, scrollbars, and toolbars.

If direct manipulation in WIMP interfaces is augmented through instrumental interaction, how can direct manipulation in post-WIMP interfaces be augmented as well? This subsection examines this question by covering some of the more common solutions from the literature: asynchronous bimanual interaction, stylus-based interaction, tangible and mixed reality user interfaces, and multimodal vocal and touch interaction.

2.1.3.1 Asynchronous bimanual interaction

One effective solution could be in leveraging the inherent differences in how a user performs tasks with their dominant hand and non-dominant hand when working on interactive tabletops. This style of interaction is based on studies of every day occurrences of bi-manual manipulation, such as those of O'Hara and Sellen [O'Hara 97] and Guiard [Guiard 87], which analyze how people use their hands when doing traditional work (Figure 2.1.3). One notable development is that of Guiard's *Kinetic Chain Model*, where the non-dominant hand serves as a frame of reference which coarsely represents the context of the task while the non-dominant hand performs the detailed work within that context [Guiard 87]. This notion is echoed in O'Hara and Sellen's explanation of the non-dominant hand being an "anchor" for the activity performed by the dominant hand [O'Hara 97]. For example, imagine how people position their non-dominant hand on a sheet of paper while they are writing with their dominant hand. Another inspiration, as cited by Wigdor et al. [Wigdor 11], is existing GUI implementations where different mouse functionality can be trigger by simultaneous keyboard shortcuts.

These observations and models for bimanual manipulations are the leading inspiration for several interaction techniques. Schmidt et al.'s *IDLenses* [Schmidt 10] uses asynchronous gestures to provide a way for user-specific identification, based on biometric measurements of a user's non-dominant hand. Placing the non-dominant hand on the tabletop surface, opens a specialized "window" through which the dominant hand may perform specific interactions (Figure 2.1.3). *Rock & Rails*, as proposed by Wigdor et al. [Wigdor 11], is another solution which demonstrates mode switching with dominant and non-dominant hand gestures (Figure 2.1.3).

The disadvantage of such an approach is that it requires the tracking of not only the users in collaborative setting, but also the tracking of (or identifying) every user's left and right hands, a solution that often requires sophisticated tracking technologies and robust gesture recognizers to achieve. Another occasional disadvantage, particularly with *Rock & Rails*, is their difficulty to adapt to multi-user settings with open orientations; *Rock & Rails* concept helps a user align objects to a grid which would be orientation-dependent.

2.1.3.2 Stylus-based interaction

Many touch surfaces use a stylus for input, including several commercial interactive whiteboard and handheld mobile devices. One of the first implementations to do so on large surfaces was the *Liveboard* developed at Xerox Parc [Elrod 92], where they place particular importance on the user experience of the pen-based interaction, citing the ease with which users can sketch as it reuses existing skills.

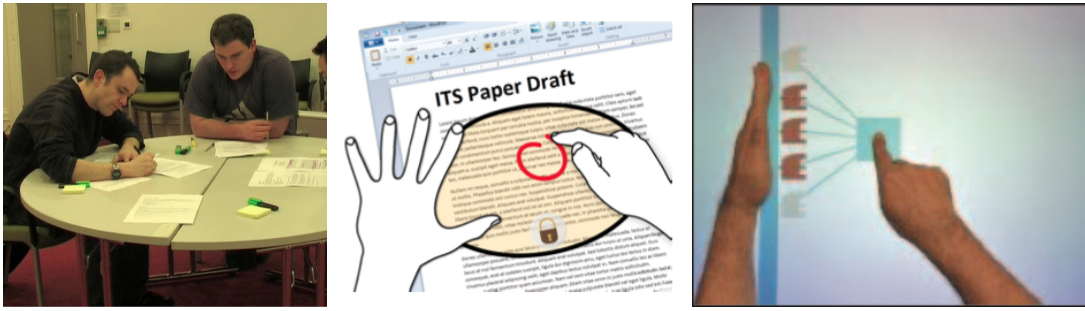


Figure 2.1.3: Sulaiman and Oliver [Sulaiman 10] is an example of one study which uses ethnographic observations of traditional tabletops to identify (among other things) how users place their non-dominant and dominant hands when annotating a document (left). Schmidt et al.’s *IDLenses* (center) provides a solution for user identification on interactive tabletops by identifying a user’s hand through biometric measurements and opening a circular lense through which user specific actions may be performed [Schmidt 10]. Widgor et al.’s *Rock & Rails* (right) uses the shape of the non-dominant hand as a means to trigger different kinds of manipulations such as aligning components to a grid [Wigdor 11].

They also add buttons directly on the pen that may be pressed while holding it. These buttons, which simulate the right and left click of a mouse, were found to be difficult to use.

Buxton, who was also working at Xerox Parc at that time, took the lesson learned from the Liveboard and applied them to a horizontal drafting board called the Active Desk [Buxton 97], which eventually led to one of the first commercial tabletops for graphic designers. Buxton attributes the main advantage of a stylus to its familiarity, which allows it to be used with extreme precision with almost no training. Leitner et al. [Leitner 13], with their *Kolibri* pen, provide an excellent example of the precision that users can achieve with a stylus by implementing sub-pixel micro-gestures on an interactive whiteboard sketching application. By using these small gestures, users can trigger different actions directly with the movement of the pen, without having to use specialized buttons.

Brandl et al. [Brandl 08] hypothesize that pen-based interaction on interactive tabletops is perhaps most effective when used in conjunction with principals of bi-manual input. Such an approach would allow users to perform context switches with touch interaction on the non-dominant hand and precise interaction with the dominant hand’s stylus. Frisch et al. [Frisch 09] draw inspiration from Brandl et al. [Brandl 08], as well as Wobbrock et al.’s previously mentioned research on user-defined gesture sets [Wobbrock 09], to perform experiments where they observe

inexperienced users performing bimanual pen and touch gestures, resulting in a compiled gesture set. This gesture set, though not entirely conflict-free or universally intuitive (as analyzed by [Ingram 12]), shows the expressive potential of such interaction strategies.

2.1.3.3 Tangible User Interfaces

Tangible User Interfaces (TUIs) use a set of specialized physical interface widgets which are graspable and instantly manipulable on a physical level. Much of the initial research work of TUIs was initially outlined by Fitzmaurice et al. in 1995 [Fitzmaurice 95] and Fitzmaurice and Buxton in 1997 [Fitzmaurice 97] (at the time, they were referred to “graspable user interfaces”). According to them, these physical artifacts are “essentially a collection of input devices that can be tightly coupled or ‘attached’ to virtual objects for manipulation or for expressing action” [Fitzmaurice 97]. Their purpose is to blend virtual and physical artifacts which would augment the virtual object with physical handles in order to facilitate bimanual interactions, spatial configurations, and position and orientation control [Fitzmaurice 95].

TUIs have continued to be an active area of research, with a dedicated international conference now in its 7th year [tei 13], and are often used in conjunction with interactive tabletop surfaces. On the surface of the tabletop, objects can be moved, rotated, flipped, stacked and removed to provide different functionality. Two of the most important qualities of these tangible interfaces as identified by user evaluations conducted by Fernaeus and Tholander are persistence and readability [Fernaeus 06]. This makes TUIs particularly attractive when considering novice users or collaborative settings, thanks to the engaging and easily understood physical interface.

There are several examples of well executed TUIs on interactive tabletops. The ReacTable [Jordà 07] is a music-generating collaborative tabletop application which uses abstract shapes to produce different musical effects in real time. Their open-source toolkit ReacTIVision [Kaltenbrunner 07] for fiduciary marker tracking has been widely downloaded and cited over 240 times in research papers since 2007.

TangiSense, presented by Kubicki et al. [Kubicki 12, Kubicki 11], uses an RFID approach to object tracking on tabletop surfaces and demonstrates several applications for collaborative work such as automobile traffic control simulators. Object tracking by RFID allows for certain interactions such as stacking objects, which is difficult to implement otherwise, as well as on-artifact storage space, so that objects may remember their previous state even if their contexts change. Another example of work applications for TUIs and interactive surfaces is Jetter et al. [Jetter 11] who propose Facet-Streams which uses abstract objects as logical units

that can be strung together to form a complex search query.

TUIs and interactive surfaces do not need to only be limited to the surface of a device. Gjerlufsen et al. [Gjerlufsen 11] propose an application for neurobiologists on an interactive wall display which can display up to 64 high-resolution 3D brain scans. Inspired by Hinckley et al.'s prop-based interaction [Hinckley 94], they use a small handheld plastic brain as a prop in one hand and specialized pointing wand in the other, allowing a user to change the orientation of the on-screen brain scans by pointing directly to the prop.

The design space for TUIs and interactive surfaces is extremely large, but require special considerations. For example, TUIs on interactive tabletops cannot rearrange themselves automatically as virtual widgets would. However, even this problem is being addressed in current research. Nowacka et al. [Nowacka 13] propose a solution, *Touchbug*, which uses vibrating motors that can slide and reposition themselves automatically on the surface an interactive tabletop. While this is still extremely experimental, progress has been made with traditional TUIs as well: Lebrun et al. [Lebrun 13] emphasize a principled approach, which clearly identifies a separation of responsibilities regarding virtual objects and tangible objects working together.

2.1.3.4 Mixed-reality interfaces

Mixed-reality interfaces (or augmented reality interfaces) are a style of TUI which are with considering separately. As explained by Wendy McKay's conclusion in [Mackay 03], instead designing new physical tools to serve as handles on virtual artifacts, mixed-reality reuses the existing physical tools in their original form, allowing the traditional methods of work to be augmented with virtual functionality.

This approach is rooted in one of the first interactive tabletops; Pierre Wellner's *DigitalDesk* [Wellner 93] used on overhead projector and camera to allow users to tap on numbers on a sheet of physical paper to have them entered into a virtual calculator, which was projected onto the tabletop. The Affinity Table, designed by Geyer et al. [Geyer 11a], also provides an example of such an approach: users can use a pen and Post-it note to write down an idea, and then they may create a virtual copy of this Post-it note by placing it on the surface of the tabletop. This allows users to perform handwriting tasks with the instruments they are most familiar with, such as a pen, and then use the computer interface for tasks such as duplicating or changing the color of a Post-it note. Anoto pens, which can store and transmit handwriting data in real time thanks to an embedded processor and bluetooth connection, also allowing for interesting interaction techniques in mixed-reality settings, such as Haller et al.'s *Shared Design Space* [Haller 06]. Hunter et al. [Hunter 11] make use of these on their tabletop computing interface, the MemTable, to create virtual copies of handwritten notes for digital manipulation

and storage.

2.1.3.5 Voice and touch-based multimodal gestures

In 1980, Richard Bolt presented a multimodal interaction technique referred to as “Put-that-there” which demonstrated gesture and voice commands with object-aware and location-aware pronouns for distal manipulation on large screens. Since then, Sharon Oviatt’s corpus of work on multimodality (as summarized in *10 Myths of Multimodal Interaction* [Oviatt 99]) has significantly matured the field’s understanding of how users interact with multimodal systems, beyond Bolt’s simple “verb-object-location” constructions. Oviatt’s research supports a complementary usage of multimodal input, and explains how users often employ different modalities sequentially rather than simultaneously and redundantly. Moreover, a user’s preference for unimodal or multimodal interaction is directly related to the nature of the task; for example, through experimentation with users [Oviatt 99, Oviatt 97], Oviatt found spatial location commands are often composed using multimodal input, but selection commands and general commands (such as printing) were often performed unimodally.

Schnelle-Walker and Döweling’s recent work on design patterns for multimodal interfaces also explains the importance of using inputs in a complementary fashion and reflects upon many of the advancements that were made since Bolt’s “Put-that-there” [Schnelle-Walka 12]. Among their proposed design patterns are *auditory mode switching* and *select-by-touch, operate-by-voice*, where touch interactions are augmented with voice commands to provide additional functionality. Another design pattern, *voice as a private output channel*, can be especially useful when interacting with shared displays (as demonstrated by tabletop research by Morris et al. [Morris 04b]).

In 1999, Cassell et al. [Cassell 99] presented an advanced multimodal interface which uses a sophisticated conversational virtual agent to embody the human-computer interaction. User-testing with similar systems, such as with Nguyen and Wobcke [Nguyen 05], show how users can employ natural conversational skills such as negotiation and turn taking to perform complex interaction with such systems. Finally, Tse et al. [Tse 08] are perhaps one of the only examples of multimodal fusion of voice commands and touch gestures on an interactive tabletop. They present a sophisticated solution which is even capable of understanding *multi-user* multimodal commands, in a “put that there” and “this one too” style, though no user evaluation with this system was conducted.

2.1.3.6 Reflections on extending direct manipulation

With regards to the above interaction techniques, several of these examples are motivated by their desire to extend direct manipulation in the case of *collaborative settings*. For example, this is explicitly the case with Schmidt et al.'s bimanual *IDLenses* [Schmidt 10], Jetter et al.'s *Facet-streams* [Jetter 11], and Tse et al.'s multi-user multimodal commands [Tse 08].

Conversely, some of these input devices still require significant maturation to be properly executed in a multi-user settings. For example, some pen and touch input setups, such as with Brandl et al. [Brandl 08], are explicitly designed for a single user setting; the tabletop screen is divided into two sections, one for the dominant hand's touch input and the other for the non-dominant hand's input. Dividing the tabletop into regions for every user's left and right hand might not be appropriate for many kinds of collaborative setting. Other, more flexible, pen-only tabletop solutions such as with Hilliges et al. [Hilliges 07] are used for two-person work, but require users to hold the pen in an uncomfortable manner so as not touch the tabletop with their hands.

Ultimately, the modalities of personal interaction with such systems will also be affected by the collaborative context in which they are produced. Multi-user interactive tabletops or board displays carry with them entirely new design considerations, which will be addressed in the following section.

2.2 State-of-the-art of multi-user surfaces

This section will examine surface computing within the context of group work, i.e. co-located synchronous computer-supported cooperative work. The first part will examine related work on collaboration and identify the aspects of collaboration that should be either preserved or augmented. The second part will discuss specific design principles regarding social interaction and communication that will have a positive impact on collaboration.

2.2.1 Support for collaboration

Researchers have been investigating the use of CSCW systems to support and augment collaboration for a variety of activities. The purpose of this thesis is to effectively engage a group in preliminary design activities in the context of a multi-disciplinary engineering project. The term "preliminary design" has not been directly discussed in the CSCW literature, but there has been a wide range of topics which share many of its qualities. These topics include meeting support system, information sharing systems, collaborative sensemaking, collaborative decision-

making, collaborative knowledge building, collaborative learning and collaborative problem-solving.

Of these terms, collaborative problem-solving is perhaps the closest term to preliminary design. There exist a few different definitions. Ullman discuss problem solving in the context of meeting support system and defines it as “generating and refining information punctuated by decision-making” [Ullman 04]. Hilliges et al. [Hilliges 07] discuss the notion *creative collaborative problem solving* as a group activity which requires interpretation, coordination, and communication. They derive this definition from social science research conducted by Teresa Amabile [Amabile 83].

The importance of interpretation, coordination and communication particularly resonated with the observations conducted in the previous chapter. It is also similar to other models of collaboration such as the “3C” model (communication, coordination, cooperation) of Fuks et al. [Fuks 05]. More importantly, it provides three different subjects which can bridge the wide range of research topics concerning co-located collaborative activities. The following three subsections will address examines the importance of interpretation, coordination and communication in collaborative interactive spaces.

2.2.1.1 Interpretation

The *interpretation* of data and information is necessary for understanding a complex problem, providing insight, and making decisions. This is relevant for several different domains, including meeting support systems. Ullman explains that the task of interpretation and informed decision-making in meeting support systems must be done with inconsistent, incomplete, uncertain, and evolving information [Ullman 04]. Therefore CSCW systems must be capable of expressing such information, without hiding inconsistencies and conflicts in data.

This is exactly what is done with such traditional problem solving methods such as causal analysis and project planning from the observations. Throughout the development of the cause and effect diagram and project flowchart, they are used to represent data which is inconsistent, incomplete, uncertain and evolving. This goes beyond the actual semantic structure of the cause-and-effect diagram and the project flowchart and extends to the tools used to externalize them. For example, in the causal analysis observation, an “uncertain” connection between two components was communicated by placing the Post-it notes close together, while an actual drawn line between two components communicated a “certain” connection. These same characteristics apply to brainstorming and the process of constructing an affinity diagram.

Outside of the field of meeting support systems, there is much relevant work within the information visualization community centered around interpretation,

and especially around the topic of collaborative sensemaking. Sensemaking is the process of creating understanding and generating insight from data, and conceptually, focuses on the user as an analyst who works with data to extract patterns and test hypotheses. This is immediately evident when considering Pirolli and Card's model of sensemaking activities (Figure 2.2.1) [Pirolli 05].

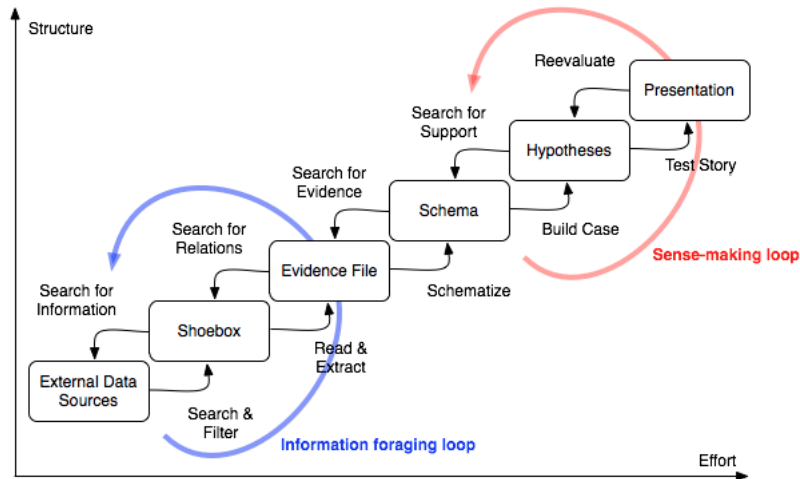


Figure 2.2.1: The sensemaking loop, adapted from Pirolli and Card [Pirolli 05].

There has been some recent work dedicated to understanding sensemaking in co-located CSCW settings. Vogt et al. [Vogt 11] investigate the role that groupware can play in sensemaking (Figure 2.2.2). They consider sensemaking among pairs in front of a *large shared display*. The first stage of their analysis is focused on understanding how the large shared display can assist in interpretative tasks. They find that that the display is useful tool which allows users to *spatially arrange* information while also providing them with an *overview*. These results are corroborated with other, similar user evaluations, e.g., Andrews et al. [Andrews 10] confirm the positive impact of a large vertical surface on which a user might attribute spatial meaning to objects on a large display.

The second stage of Vogt et al.'s analysis [Vogt 11] is to understand how collaboration in sensemaking impacts the classic model of sensemaking from Pirolli and Card. They identify five activities performed by groups using the large vertical shared display: Extract, Cluster (which both relate to the information foraging loop), Record (Schematizing), Connect (Hypothesis) and Review. According to their conclusions, the use of the shared display impacted the process of sensemaking of data in two ways: First, it allows the groups to spatially and visually arrange all of the information at once, which allows for the process of sensemaking to be streamlined (for example, the Evidence File and Schema step from Pirolli's

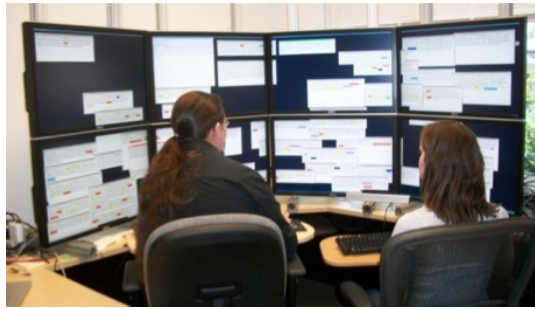


Figure 2.2.2: Vogt et al.'s experimental set-up for collaborative sensemaking.

model are combined here), and second, it introduces a new notion of Review, where groups begin rereading and reinterpreting information they have already seen, while discussing current and alternative hypotheses (this review step is not directly addressed in Pirolli and Card's model, although it can be considered as a set of rapid ascensions and descensions of the entire loop). Their work concludes that the process of sensemaking is streamlined and extended in the collaborative, interactive setting.

Wallace et al. [Wallace 13] confirm the presence of the additional review stage in group sensemaking tasks, and show how a large horizontal display (an interactive tabletop) can support interpretative tasks for teams of four. They similarly emphasize the use of spatial arrangements of information and the use of an overview [Wallace 13]. In their experiment, the authors distributed to each participant a set of presentation slides which contained information in the form of charts, tables, and statements. The participants collaborated by arranging the information in a "tableaux", or grid of documents, whose order and position convey abstract meaning. The arrangement of the tableaux might be difficult for an outsider to observe and understand, but for the people actively participating in its construction, the tableaux had a clear and shared meaning. This is also an excellent example of common objectification at play.

This research indicates that large shared displays are useful tools for sensemaking in collaborative settings. These are effective especially when they allow for the spatial arrangement of information and they allow participants to establish their own overview of the information. Moreover, the spatial arrangement of information as well as the organization of the overview conveys meaning that is shared by all participants.

Beyond using a large shared display, there exists another general design strategy for supporting sensemaking and interpretation tasks in co-located collaboration: *low-viscosity interaction*. This is proposed by Jetter et al. [Jetter 11] and corroborated by Wallace et al. [Wallace 13] for sensemaking on collaborative interactive tabletops. Here, low-viscosity means "low resistance to change". This implies that

interaction is designed using rapid and reversible operations which can be quickly done and, if necessary, undone. Additionally, if these operations can be performed with a low cognitive load, much of the focus of the user can remain on a higher level goals. Therefore, the advantage of low-viscosity interaction, as explained by activity theory, is that it seems to increase the efficiency of the group's pragmatic and epistemic action cycles.

2.2.1.2 Coordination

Coordination refers to how a group manages the division of labor during an activity. Vogt et al.'s [Vogt 11] previously mentioned study on collaborative sensemaking also reveals the positive impact that coordination has on interpretation. In their experiments, 5 out of 8 of the pairs they observed naturally formed two distinct roles: sensemakers and foragers. The sensemaker was the dominant participant who took notes on a whiteboard and directed his partner's effort with requests for information. The forager would support and challenge the sensemaker's current hypothesis with evidence and would organize and maintain awareness of the data. These roles appeared naturally during the activity and correspond to the lower-level foraging loop and the higher-level sensemaking loop in Pirolli and Card's model (Figure 2.2.1). They also were attributable to the groups with the highest performance scores.

This illustrates the importance and efficiency of *natural roles of collaboration* in interpretative tasks. Users have aligned themselves into roles which address the most crucial aspects in sensemaking without instruction. These low-level and high-level roles appear in some of the observations. For example, the dynamic between the scribe and the moderator resembles the dynamic between the forager and the sensemaker. It is important for CSCW systems to consider how the environment can bring about or impact these roles. This is especially important when designing an environment for a group of more than two people. For example, a tightly partnered sensemaker-forager team, might exclude other members of the group.

There are many other studies in the literature that address how interactive tabletops can support coordination and division of labor. There is a general consensus that this can be achieved by designing tabletop interfaces that allow for both *shared awareness* and *parallel work*.

Morris et al. [Morris 10] cite support for division of labor as one of seven design criteria in the requirements of their interactive tabletop collaborative search and sensemaking environment. Some of their proposed interaction techniques for their tabletop environment *WeSearch* were explicitly designed to support coordination between participants (Figure 2.2.3). For example, they allowed users to "clip" webpages into separate pieces to allow for the division of labor when analyzing

the content of a website. The motivation for this idea was to provide a way that allowed participants to parallelize the work of analyzing the website. Their evaluations show that this failed to support coordination as the designers had intended: users simply did not use this feature. They did observe that the shared display of the interactive tabletop itself “seemed to adequately support division of labor by providing each group member with space to interact and facilitating conversation and awareness through co-presence and co-visibility” [Morris 10].



Figure 2.2.3: *WeSearch* tabletop application for collaborative search and sense-making [Morris 10].

Isenberg et al. [Isenberg 10], with their interactive tabletop *Cambiera*, design their workspace environment with considerable flexibility to allow for parallel work. In their design, the individual’s activity space was augmented with subtle awareness cues from the activity of other participants. Each user was able to perform their own searches and individually analyze documents, but see color-coded awareness cues, allowing them to be conscious of their partner’s search queries when they are contextually relevant to their own queries. Following their evaluation, Isenberg et al. explicitly maintain their recommendation “that collaborative systems continue to allow teams to smoothly choose which collaboration style best fits their task requirements and work styles.” They also found that some pairs spent too much time working individually, in parallel, without reconvening to merge their work, leading to lower performance scores. Although some groups responded favorably to the indicators, and they were perhaps *too* subtle. Therefore, Isenberg et al. [Isenberg 10] recommend that stronger indicators be used to help coordinate parallel work.

Finally, Hilliges et al. [Hilliges 07] highlight the necessity of interactive tabletops supporting parallel work in idea generation and brainstorming tasks. Parallelizing the individual work of idea generation is hard-coded into the protocols of many brainstorming techniques, and designing groupware that can support this is essential. Jetter et al. [Jetter 11] also cite parallel work as a design objective in their tabletop search environment. They do so by using a tangible user interface for con-

structuring search queries which allows for parallel work to be conducted individually first and merged later.

One of the most important aspects of coordination is the *equity of participation*. A concept originating from collaborative learning studies, the idea is that groups perform better at the task at hand if the participation from every member is as equal as possible. A group’s performance may be negatively impacted if one member refuses to participate or if a few members dominate all conversation. Woolley et al. [Woolley 10] find a positive correlation between coordinated turn-taking in conversation and the “collective intelligence” and performance of the group.

One of the advantages of measuring equity in participation is that it can be processed in real time, and the information can be shared on supporting displays during group work as coordination mechanism. Bergstrom and Karahalios [Bergstrom 07] seek to encourage equity of participation by allowing users to self-monitor. They use a real-time circular information visualization to display talk patterns on an interactive tabletop, as shown in Figure 2.2.4. From their user evaluations they found that users are able to adjust their behavior, and coordinate turn taking better, even though some users found such a large visualization distracting. Bacher et al.’s *Reflect* [Bachour 08] and Ogawa’s *Table talk enhancer* [Ogawa 12] rely on similar setups and report positive effects on turn taking as well.

Wallace et al. [Wallace 13], when comparing the use of interactive tabletops and to traditional tabletops, find that interactive tabletops support the equity of participation better than traditional ones, and this equity of participation correlates to higher performance scores for sensemaking tasks.

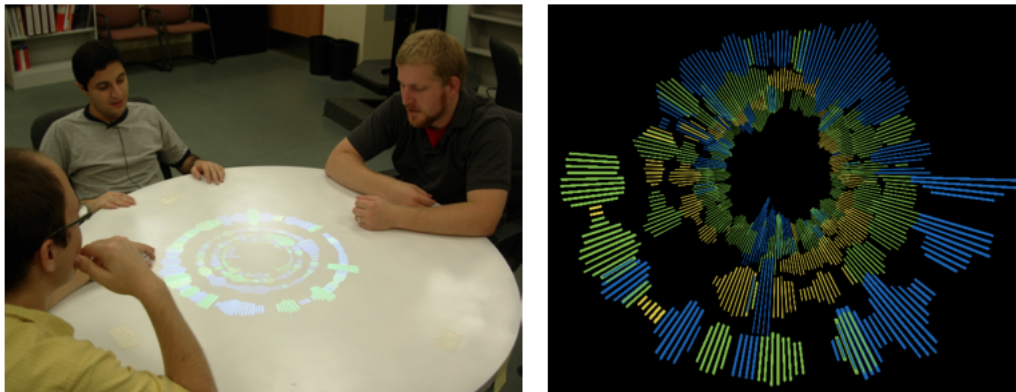


Figure 2.2.4: The conversation clock, from Bergstrom and Karahalios [Bergstrom 07].

Some results on the equity of participation can be nuanced. For example, Mar-

shall et al. [Marshall 08], when comparing multitouch tabletops to single-touch tabletops in collaborative learning applications, found that multitouch led to more equitable participation in terms of interaction events with the interface but not necessarily in terms of conversation and turn-taking. Understanding how equity of participation can be measured requires an understanding of how groupware impacts communication.

2.2.1.3 Communication

Communication in group work is a means through which interpretation is shared and coordination is managed. I have already discussed some aspects of communication, as it is difficult to discuss interpretation and coordination without discussing communication. The ramifications for not properly considering communication in groupware can be tremendous. This is certainly the case with Nunamaker et al.'s decade long experience with meeting support systems in the 1990s [Nunamaker 91]. The meeting support system's used in their experimentation repurpose traditional PCs by embedding them into C-shaped tables which are positioned around an interactive whiteboard. When every user has their own workstation and personal display, the line of sight between participants is disrupted making it difficult to support natural conversation; participants exert more effort to peer over monitors and engage in communication. Nunamaker et al.'s emphasis on preserving line of sight shows the importance of considering both *verbal and non-verbal communication* in collaborative activity [Nunamaker 97].

The variety of gestures in nonverbal communication is extremely rich. It's important to note that nonverbal communication refers to any gesture, pose, or gaze made by the body. A person will never stop "gesturing", and will always be communicating something nonverbally. While this is an important aspect of communication, it is also difficult to analyze non-verbal communication. Bressolle et al. [Bressolle 98] explain "a fundamental aspect of human communication is its non-deterministic character, in part due to the inferential processes which enters into play in mutual understanding." They investigate participants verbal and non-verbal communication during the use of their groupware system designed for air traffic controllers. Their evaluation uncovers many instances of gesturing which contribute to "the establishing and updating of a mutual cognitive environment" (similar to the concept of common objectification) and an increase in the team's efficiency.

Because the nonverbal communication will be used in the evaluations presented later in this dissertation, I will quickly illustrate several different kinds of nonverbal gestures which are pertinent to collaboration in groupware environments. Many of these come directly from Paul Ekman's seminal work *The Repertoire of Nonverbal Behavior*, or from works which build upon his.

- Handling gestures: Gestures where one performs an operation on an object that changes the state of that object. This includes all aspects related to the physical interface of the system. This could be manipulating a virtual object to move and reoriented it, or writing with a pen or virtual keyboard [Ekman 69].
- Deictic gestures: Also referred to as pointing gesture, these are frequently done with hands, fingers and gaze to identify either real or imaginary people, objects, and directions [Ekman 69].
- Adaptor gestures: Often unintentional, adaptor gestures can be identified self-touching, such as scratching or crossing arms, or object-oriented, such as playing with a pen. Researchers generally believe these associated to with negative emotional states like distraction, boredom, fear, embarrassment, and irritation, but such conclusions are controversial without thoroughly examining the context [Ekman 69]. Other researchers claim that adaptors can also be more positive and are used during reflection and cognitive thought, such as the classic image of a philosopher stroking his beard [Masse 00].
- Metaphoric gestures: Such gestures are used to illustrate speech by forming an analogy between a concept and a particular action or pose done with the body. An example would be of a speaker describing their layout of a house and drawing the shape of the rooms with their finger in the air. They can not only serve to illustrate speech [Goldin-Meadow 99], but can also instantiate abstract thinking in the speaker [McNeil 96].
- Punctuator gestures: Punctuators are the only one of the five types of gestures that must accompany some form of verbal communication and can be performed by either the speaker or the listener. Punctuators will help carry the rhythm of the conversation. Examples of these gestures are nodding the head after hearing each item of a list of groceries, or open palms and raised eyebrows at the end of an assertion to indicate turn-taking (also known as beats and batons, respectively). Such gestures serve to complement, modulate and emphasize the message of the speaker [Goldin-Meadow 99], and occur frequently in argumentative discourse [Masse 00].

Supporting both verbal and nonverbal communication is one of the leading motivations for using interactive tabletops which can accommodate natural *face-to-face* communication. Positioning the users in front of each other can have a positive impact on collaboration. For example, Rogers and Lindley [Rogers 04] compares interactive tabletops to interactive whiteboards and generally finds that the former has a positive impact on group collaboration for shared awareness and parallelized

work. They also note several advantages that interactive whiteboards have for certain kinds of collaborative work: they find interactive whiteboards are better at providing “a shared surface for communal and audience-based viewing and annotating of information, that is to be talked about and referred to, e.g., the showing of visualizations, slideshows and the sharing of video and other media”. Ahmed Kharrufa [Kharrufa 10] describes this as *shoulder-to-shoulder* collaborative activities, which represents a different but equally important style of communication and coordination.

Communication is common measure for evaluating groupware systems. Communication between participants allows evaluators a medium through which they may understand how groups are progressing in their interpretation of a problem and coordinating their actions. Generally, the more communication in a group related to the task at hand, the more interpretation and coordination is occurring. Jamil et al. [Jamil 11] stress the importance of not only measuring the amount of talk by each participant but also the content of the talk. By coding participants’s time spent discussing the task, group organization and roles, and requests for information, they find that interactive tabletops are as good as traditional tabletops in supporting a group’s effort in coordination and reflection. Buisine et al. [Buisine 12], Marshall et al. [Marshall 08], and Martínez et al. [Martínez 11] stress the importance of measuring nonverbal and verbal communication in collaborative setting, particularly when measuring the equity of participation, as one modality might uncover information that is not present in the other.

2.2.2 Design considerations for collaboration on multi-user surfaces

Gathering multiple users around an interactive surfaces introduces new constraints and requirements that are not present in single-user gestural interfaces. These issues were directly addressed when Stewart et al. [Stewart 99] formalized the idea of *single-display groupware* (SDG), where multiple users have simultaneous private input channels to a system which provides a shared output on a large display. Such a configuration, as argued by Stewart et al. [Stewart 99], would provide new parallel collaboration previously inhibited by social and technical barriers, but at the same time would produce new conflicts and time-consuming frustrations between users sharing the same interface and screen space.

Interactive tabletops and whiteboards are some of the more common examples of SDG, and the previous subsection has discussed their impact on the interpretation, coordination, and communication in group settings. However, there exists much related work on SDGs that has been conducted without these high-level goals in mind, but rather with more focus on addressing specific conflicts and frustrations

that Stewart et al. predicted would arise from these interfaces [Stewart 99].

This section will focus more on design trade-offs and considerations when implementing interfaces on SDG. The topics I address - orientation, space, user identification, clutter, and feedthrough - are taken from common themes found in related work on the subject, but are also largely inspired by much of the work of Meredith Ringel Morris and her lists of design considerations for SDG (these have evolved over time and through context, and can be found in [Morris 04a, Morris 06b, Morris 10]). I use this as the basis of a curated list of design considerations for interactive tabletops in the following subsections.

2.2.2.1 Orientation

There is no privileged viewing angle on an interactive tabletop. A meeting participant could be seated anywhere along the borders of the tabletop. This constraint poses an enormous problem when trying to adapt legacy software and existing operating systems to tabletop computers, and it presents challenges that were never necessary to consider on single-user desktop applications. Consequently, there have been many studies which directly examine how and why participants rotate objects on a tabletop surface. Kruger et al. [Kruger 04] perform an observational study of pairs of users collaborating on a traditional tabletop to uncover many of the nuances involved in orientation.

According to their observations, orientation serves three major roles: *comprehension* (a user will reorient an object to comfortably accommodate reading, writing or drawing, or view the contents of an object from another different angle), *coordination* (a user will rely on orientation to express which objects he or she is currently using or part of their personal workspace, and which objects are available to the group or part of the group workspace) and *communication* (objects can be reoriented to convey support nonverbally).

Kruger et al. explain that “rotation is a relatively ‘lightweight’ communication that people do naturally, quickly, intuitively, and without explicit consideration” [Kruger 04]. In their experiments, “people rarely asked for explanation or commented on items that were already positioned on the table. That is, the meaning of the orientation was self-explanatory” [Kruger 04]. These findings show how orientation plays a crucial role in supporting collaboration, should be implemented with explicit consideration to low viscosity interaction.

Orientation on collaborative tabletops has been supported in several different ways:

- Fixed orientation: Such systems assume one fixed orientation. This is less desirable, but can be appropriate for some scenarios when users are working side-by-side on simple tasks.

- Manual orientation: Conceptually, supporting the direct, manual rotation of objects would allow for the greatest support of natural styles of comprehension, coordination, and communication. This approach, when compared to how easily users manipulate traditional media, is limited by the efficiency and effectiveness of the input devices that are used. The limit in terms of the degrees of freedom of some stylus and single touch interfaces, force many tabletop designers to use a dedicated rotation “handle” on each object, such as is done in the *DiamondSpin* toolkit [Shen 04]. (Analysis of interaction techniques by degrees of freedom is proposed by Beaudouin-Lafon in [Beaudouin-Lafon 00] as part of the framework of instrumental interaction.) Solutions which have capitalized upon multitouch gestures, with two finger rotations and translations, are more natural, but are still slower when compared to traditional media.
- Automatic orientation: Automatic orientation can either be person-based or environment-based. With person-based orientation, objects are automatically oriented based on the position of a participants around the tabletop. This can either be assumed, based on an understanding of seating arrangements of users, or detected, typically with user-tracking solutions. Items will be automatically repositioned either according to the position of closest user, or the user which has most recently accessed this information. More sophisticated extensions of this kind of solution might even offer user-neutral zones on the surface for group work, where objects are manual oriented, as suggested by Shen et al. [Shen 06] (Figure 2.2.5). Environment-based solutions assume that the appropriate orientation of an object will always be toward the outside edge of the tabletop. These solutions are interesting because they are lightweight, as they use basic translation actions for managing orientation, and they automatically enforce some aspects of Kruger et al.’s principal of coordination by respecting personal and group orientations [Kruger 04]. When designing a tabletop application for brainstorming, Buisine et al. [Buisine 12] find that both solutions can be appropriate for different situations; they use person-based automatic orientation for the individual idea generation task and environment-based orientation for the group classification task (Figure 2.2.5).
- Hybrid methods of orientation: Following their formative study of rotation on interactive tabletop, Kruger et al. [Kruger 04] followed up with a hybrid interaction technique, *Rotate ’N Translate* (RNT), which allows a user to swing an object with one touch point using meta-physical interaction and some notions of environment-augmented orientation. Lui et al. [Liu 06] also introduce several orientation interaction techniques collectively named TNT

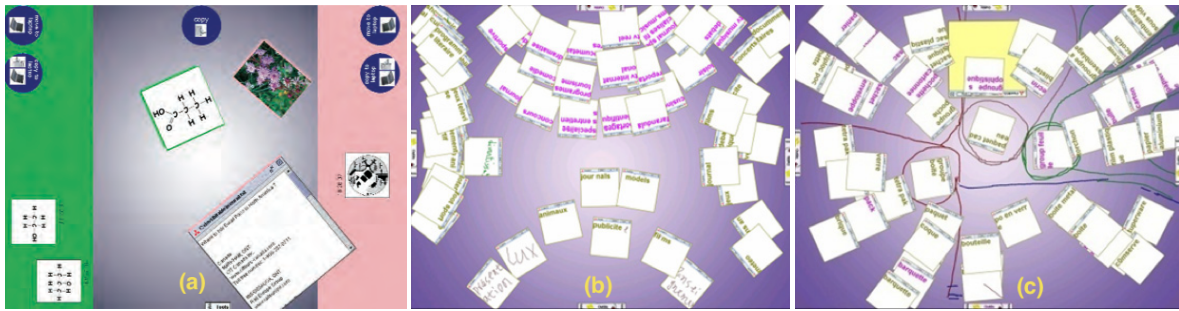


Figure 2.2.5: (a) Automatic person-based orientation with neutral zone for group work with Shen et al's *UbiTable* [Shen 06], (b) automatic person-based orientation during a phase of individual work, and (c) automatic environment-based orientation during a phase of group work on Buisine et al.s *Brainpurge* application [Buisine 12].

which were based on tangibles or open palm orientations. Both of these propositions use a center zone on each of the objects where they can be manipulated only by translation, which provides some finer degree of control. Also, providing a more natural reorientation (and translation) technique was a primary motivation for initiating tangible interface research on tabletops as indicated by Fitzmaurice et al. [Fitzmaurice 95, Fitzmaurice 97].

2.2.2.2 Space

The large screen size of collaborative interactive tabletops gives users an often unfamiliar amount of screen space to arrange and layout components. While Kruger et al. focuses on orientation, their analysis actually considers orientation in conjunction with *translation* [Kruger 04]. They cite Wang et al.'s work which shows that the two concepts are inseparable interactions for a user [Wang 98]. Therefore, many of design implications of Kruger et al.'s study, which stresses the need low-viscosity interaction techniques supporting interpretation, coordination and communication, also apply to translation.

These finding are often cited as the motivation for a wealth of different experimental translation techniques found in the literature. Beyond traditional direct manipulation techniques which can be used with meta-physics to allow objects to be thrown, there exists cursor-extensions, long distance pointing, and radar techniques.

Cursor extensions, such as the pentagraph technique [Nacenta 07], moves the user's cursor position by a factor proportional to the user's contact point and allowing them to reach the whole surface of a tabletop with little movement. Long distance pointing uses input devices to assist users in reaching objects. For ex-

ample, Parker et al.'s *TractorBeam* [Parker 05] is a handheld pointing device and Rekimoto and Saitoh developed *HyperDragging* [Rekimoto 99], which allows a user to employ their laptop's trackpad to move objects on tabletops and whiteboard. Radar techniques give users a special virtual widget on the tabletop display which reproduces the world in miniature, and allows users to interact through it. This was originally designed by Swaminathan and Sato as a "dollhouse" metaphor for large vertical display [Swaminathan 97].

Reetz et al. [Reetz 06] survey these interaction techniques before presenting a hybrid method of translation "Superflick" which combines the natural flick gesture to send an item across the tabletop, with a tap-to-stop gesture, which stops the item once it reaches its target. Their evaluation finds a significant increase in precision with technique, as opposed to the simple flick gesture. They acknowledge, however, that the simple flick gesture for passing components in collaborative settings is subjectively found to be easier to use, and "accurate enough" as other participants can catch or reach the items that are passed to them.

In the observations discussed in the previous chapter and the related work for interpretation in collaborative setting, the use of spatial arrangements of components was important for communicating relationships between components. This observation is of particular relevance when considering *grouping tasks* on an interactive tabletop: Watanabe et al. [Watanabe 07] posit that grouping techniques based on spatial proximity would be closer to user's natural method of categorization and develop a bubble-based grouping technique where items can be grouped and regrouped depending on their proximity to other items. This allows for an informal, spatial method of grouping, based low-viscosity interaction, and Höchtl et al. [Höchtl 12] find this method outperforms a more structured bin-based categorization technique. Their interaction technique, however, does not support multiple levels of hierarchy, as is sometimes necessary with affinity diagrams.

The space on a large surface with multiple users also gives rise to issues of territoriality, a popular topic for researchers and interaction designers. Some researchers explicitly divide the interface of the tabletop into separate zones for each user, as is done in Shen et al. [Shen 06] and Morris et al. [Morris 06a]. Following Kruger et al. research in how the orientation and the position of components can naturally communicate notions of ownership [Kruger 04], Martínez et al. [Martínez 11] believes that the interactive tabletop surface is best left open and undivided to provide more space for collaboration.

2.2.2.3 User identification

User identification can open new functionality in multi-user setting, such as personalized clipboards or storage spaces, personal and multi-device interaction, and multimodal fusion. There exists a wide range of solutions in the literature ded-

icated for the tracking of users. The DiamondTouch table, developed by MERL [Dietz 01], proposes an effective solution by using an array of antennas on the surface of the tabletop, with each antenna emitting a unique signal. Once contact is made with surface, the signal travels through the user and into a receiver located in the user's chair, allowing the system to determine which user is touching where. Through a university loan program, the DiamondTouch table was (and still is) a popular device among researchers and it is the surface on which much of the initial work on tabletop collaboration was conducted. This had a positive impact on the research community by providing a standardized tabletop and an elegant solution to user identification, but some of the early interaction techniques developed for this hardware can be difficult to reproduce on systems with different kinds of hardware and user identification set-ups. DiamondTouch tabletops also requires users to stay seated in chairs and connected to receivers which might not be appropriate for larger interactive spaces.

As the hardware design of interactive tabletops grew more diverse, newer more flexible user identification systems began to emerge. Meyer et al.'s *IdWristbands* [Schmidt 10] and Roth et al.'s [Roth 10] *IR Ring* provide effective solutions with wearable devices with LED and IR transmitters, respectively. Others use motion tracking such as Microsoft's Kinect to track user's hands and bodies, as is done on with Martínez et al.'s *Collaid* interactive tabletop [Martínez 11] and Turnwall et al.'s interactive wall-sized whiteboard [Turnwald 12]. Finally, more avant-garde solutions such as *Bootstrapper* use cameras underneath the tabletop to recognize users by their shoes which achieves successful recognition 92.3% of the time during five-person collaboration. These solutions all require specialized physical hardware, place different limitations on the number of users as well as their movement in rooms, and require some form of user authentication. They are particularly well adapted for an interactive tabletop or whiteboard, but have not been effectively adapted for tracking in larger interactive spaces. Some lightweight software solutions exist such as dividing up the surface of the screen into explicit user territories but this places restriction on space and collaboration.

On the other hand, many multi-user interactive tabletops and whiteboards are simply designed to be user-agnostic. These limit some of the extra functionality that could be made available to users, but they still allow multi-user interaction and parallel work to be present in the interface in a way that does not discriminate between users.

2.2.2.4 Clutter

Clutter is mentioned as problem for interactive tabletops in the several different research projects: [Morris 10, Pinelle 08, Leithinger 07]. Managing clutter is perhaps the most surprising design consideration for interactive tabletops, but is one

that manifests itself because of several compounding factors. First, as Stewart et al. mention, multiple users share the surface and therefore screen real estate for on screen components can be limited [Stewart 99]. Second, this is especially troublesome considering that most tabletops use projected displays whose limited resolution require that text and other artifacts to be rendered slightly larger than on a personal display. Third, interactive components on the display, such as buttons must be displayed slightly larger than on mouse-based interfaces to help offset “fat-finger” errors [Vaida 09]. Fourth, many of the traditional methods of managing clutter and occlusion on a computer display in WIMP interfaces (e.g. minimizing applications into a task bar at the bottom of the screen, rotating through windows with keyboard shortcuts) must be rethought for SDG. Finally, as is common with direct manipulation techniques implemented on gesture-based interfaces, interactions are frequently object-oriented, without many environmental-level gestures, leaving users with no easy way of moving and manipulating multiple items without directly selecting or manipulating them all, which can make managing a great number of widgets tedious.

2.2.2.5 Feedthrough and group awareness

From the very first shared display such as Xerox Parc’s Liveboard [Elrod 92], group awareness is explicitly taken into account in the interfaces. Maintaining the focus of the group is especially difficult on a interactive tabletops with multiple viewpoints. On the other hand, providing overwhelming support for group focus can detract from individual focus when it is required during parallel work. Therefore, managing group awareness is a balancing act between group focus and individual focus. Careful consideration for the visibility of gestures is instrumental for maintaining group awareness.

Actions that are beneficial for an individual may also be detrimental for the group. The term *feedthrough* refers to the feedback produced by interaction techniques when artifacts are manipulated by the actions of other users [Nacenta 07]. Jamil et al. [Jamil 11] compare two different interaction techniques on tabletops for a collaborative learning activity for children; one version of their tabletop application used only direct manipulation techniques, while the other uses the pantograph interaction technique (see Section 2.2.2.2). This grants users with a low-visibility far-reaching gesture which allows them to perform large interactions over the entire screen. This resulted in confusion, laughter and play when users would steal “objects” away from each other, ultimately negatively impacting group performance. The direct manipulation techniques did not have a detrimental impact of collaboration, as users were forced to reach across the tabletop surface, and therefore perform interactions with high-visibility, providing natural feedthrough. Too much emphasis on feedthrough can perhaps also have negative effects on group

awareness. For example, Morris et al. gave examples of multi-user gestures where participants must hold hands in order to perform collective interactions for high-level tasks which required group consensus (e.g. changing applications), but found that asking participants to touch was uncomfortable [Morris 06c].

Interactive tabletop systems by Geyer et al. [Geyer 11b] and Hilliges et al. [Hilliges 07], strike a nice balance for group awareness by providing a vertical display next to their tabletop system which provides an “overview” and “shared reflection space” respectively. These supporting displays are updated in real-time according to the actions of the users. Feedthrough, and highly visible gestures, also positively impacts gesture transferability from expert user to novice users.

From exploring the design considerations which impact orientation, space, user-identification, clutter, and groupaware, its interesting to note the different concerns that arise when comparing interactive tabletops and interactive board displays. Board displays can almost be considered as a specialized version of an interactive tabletops, as the orientation constraint is relaxed, simplifying space management and providing some control over clutter. Because of this, and because they are often displayed prominently and in the sight-line of meeting participants, visibility and group awareness is easier to consider.

2.3 Design consideration for established workflows

Collaborative problem solving in the context of a design and engineering project is knowledge intensive work which involves departmental coordination across several different workflows. When different project stakeholders gather for a meeting, the workflows of the organization are converging for a brief but important moment in the timeline of the project. The design of interactive space should be conscious of the larger work processes of the group members and should address this issue through several different approaches.

This concept has been introduced in the previous chapter when discussing Activity Theory. For Engeström, a community mediates an activity through implicit and explicit social rules and division of labor [Engeström 99]. Döweling et al. rephrase Engeström’s vision of an activity, by expressing community as only part of the context which mediates an activity, and by renaming “division of labor” to “workflows” [Döweling 12]. These activity models both emphasize the importance of considering rules and workflows when designing for collaborative activity based systems.

Scott et al. in their 2003 article *System Guidelines for Co-located, Collaborative Work on a Tabletop Display* [Scott 03] discuss several guidelines which are related to workflows, including “support for fluid transitions between activities”, “support

for transitions between personal and group work,” and “support for transitions between tabletop collaboration and external work.” These are discussed below.

Jetter et al. [Jetter 11] elevate the notion of established workflows into a high-level domain for the design of interactive spaces, alongside personal interaction, social interaction, and physical space. For an illustration, they give the example of a sketching session undertaken by designers, which is a collaborative process with time limits, roles, iterations, breaks. The sketchers can even be overseen by a supervisor or a facilitator who should have the option of visualizing the entire process, without disrupting their activity, to be able to intervene and gives the sketchers direction only if necessary.

Using the aforementioned related work, and the observations of preliminary design in the previous chapter, I present a list of design considerations for workflows in interactive spaces.

2.3.1 Organizational roles

Complementing the roles that can naturally emerge in collaboration are the roles inherent to the organization and its social structure. Each participant will have different motives and goals which should not be encumbered by the interface. The moderator role (also referred to as a facilitator [Jetter 12a]) is responsible for proposing activities and topics of debate, synthesizing discussion, extracting decisions when necessary. The role of the scribe [Hunter 11, Pinelle 03], when not naturally occurring, can also be enforced by organizational rules such with team assistants or project assistants. Their responsibilities could be take notes and diffuse information as required (c.f. Section 2.3.3).

Groupware can assist in these organizational roles by supporting them when they are useful (e.g. moderator can have access to more functionality than other participants, allowing him or her to manage current activities, change the meeting agenda, and to view an overview of the activity) or the groupware may simply assume the roles, if it can effectively do so (e.g. automatic logging and note-taking can replace some of the work of a scribe). People may play multiple roles during a meeting. For example, a project leader might participate as a moderator of a meeting, as well as its scribe. Roles can also rotate among participants. For example, if another member has a better understanding of a certain topic of discussion, he may temporarily take on the role of the moderator.

2.3.2 Access to personal documents and information

In a multidisciplinary engineering team, each participant will have personal documents which could be relevant to the activity taking place. This could be information pertaining to previous projects, e.g. schemas, photographs, gantt charts,

evaluation results, or information pertaining to their own domain, e.g. standards, norms, and design guidelines. This can also refer to any work that a participant has done before the start of a meeting, such as a presentation or meeting agenda. Such information can exist on network servers or on mobile devices. This notion, however, is not limited to digital information but also includes physical, traditional media. In the observation of project planning, participants constantly referred to information given to them in the design briefs, as well as their own personal documents on PERT analysis. These documents complemented and informed the collaborative activity taking place.

Personal access to information was also mentioned as a design guideline by Scott et al. [Scott 03]. File management on interactive tabletops is very much an open question. Such systems must balance digital concepts such as file access permissions and file navigation, with social concepts, such as private information and document sharing. These issues were recently brought to light through the work of Judy Kay and Anthony Collins [Collins 07, Collins 09, Collins 08, Collins 11a, Collins 11b].

2.3.3 Output and distribution

One of the primary usability issues with traditional whiteboards and Post-it based creativity techniques, is the amount of post-work they require. After the activity is complete, one member of the team, often the project leader who called the meeting, must stay in the room or take a digital photograph of the content in order to duplicate and distribute all the information that was produced. When information is distributed, it is often rendered into a different, more usable format than the result of the activity. This was witnessed in the observations when one user spent a great deal of time reproducing and updating a digital copy of the flowchart on the brown paper in Microsoft Project. Interactive spaces, through logging and persistence can reduce this amount of time significantly, by facilitating the automatic export of documents, and in different formats. There are some examples of this taken into consideration in interactive spaces. For example, in Geyer et al. [Geyer 11b], consider the exportation and post-work aspects of their interactive spaces by letting users open a screen copy of the result of a brainstorming. This approach is similar to taking a digital photograph and still requires significant person-hours to process. It would be more beneficial to export the resulting into different formats which would be more appropriate for their post-activity use.

2.3.4 Pre-existing workflows during an activity

Groupware can eliminate certain tasks by automating them when possible and by rendering the existing workflow more efficient. This requires an in-depth under-

standing of users' motives and the users' activities (such as through task analysis and hierarchical task decomposition) in order to identify which task can be automated. Examples of this include *WeSpace*, by Wigdor et al. [Wigdor 09b], which an interactive space used to encourage information sharing in astrophysicists to discuss and share data in a more efficient manner. They optimize existing workflows by interface directly with their data gathering applications and accommodating file formats that facilitate the sharing of data.

2.3.5 Providing transitions between activities

Ullman [Ullman 04], using the analysis of topics of discussion in meetings from Olson et al. [Olson 92], emphasizes the importance of supporting fast transition of activities during the same meeting. This is also mentioned by Scott et al. in their tabletop design guidelines [Scott 03].

Taking this idea even further, if the results of activities are in an easily exploitable digital format, groupware can not only support rapid transitions between activities but also reuse existing information in new and different ways. This includes reopening and exploiting activities of the same type (e.g. the tasks from a previous planning project are used as input into a new project planning) and of a different type (e.g. the results of a brainstorming can be used to begin a causal analysis). By promoting the reuse of existing work in future work, efficiency is increased. By facilitating activity transitions between different kinds of tools, more freedom is given to the moderator to use these tools in the context of collaborative problem solving.

2.4 State-of-the-art of augmented physical environments

After giving consideration to the domains of interactive spaces in a personal, collaborative, and organizational sense, the fourth domain, the physical environment, gives consideration to “everything else”. According to Jetter et al. [Jetter 12a], this refers to tables, chairs, lighting, walls, windows, and all aspects of the room, as well as physical constraints placed by the displays' sizes and shapes and users' movement and perception in these environments.

This section first begins by discussing related work in interactive space which has bridged these four domains. Then, this section presents a new recent design principle, *shareablity*, which can assist in understanding the composition of these interactive spaces.

2.4.1 Related work in interactive spaces

The tables, chairs, lighting, walls, windows, etc. of interactive spaces were the focus of Streitz et al.'s foundational work on *roomware*. In 2001, their concept space *i-Land* [Streitz 99] began exploring interactive tabletops, board-sized displays, object tracking, and even specialized office chairs with embedded tablets to develop interaction techniques for transferring information from one surface to another inside the physical space. Around the same time, Johanson et al. [Johanson 02a] launched the Stanford Interactive Workspace project, or *iRoom*, which also explored similar concepts with high-resolution whiteboards, interactive tabletops and information transfer techniques.

Other projects have since continued their efforts in exploring the design space of roomware. Haller et al. [Haller 06] developed the *Shared Design Space*, a sophisticated prototype of a interactive tabletop linked to a board display, and developed several interaction techniques which merged TUIs, stylus and multitouch to provide a sketching and information sharing environment for designers. More recently, Beaudouin-Lafon et al. [Beaudouin-Lafon 11, Gjerlufsen 11] presented the WILD room which features an ultra-high resolution wall with 32 monitors for purpose of information visualization activities. Because their wall is not directly manipulable with touch or stylus based interaction, they have developed a number of information-sharing and space-reaching techniques using laptops, tablets, smartphones and objects. Though these projects are responsible for significantly advancing the state-of-the-art in roomware, they do not provide in-depth formal user evaluations involving groups engaged in collaborative work (*i-Land*, *SDS*, *WILD*), or they do so only with certain components of their room activated at a time (*iRoom*).

In order to better understand roomware and its varieties, Terrenghi and Dix propose a taxonomy to classify multi-person-display ecosystems based on their physical-size (on the scale of an inch, foot, yard, “perch”, or “chain”) and the style of collaboration they were designed for (one-to-one, one-to-few, few-to-few, one-to-many, many-to-many) [Terrenghi 09]. Because of the wide range of eco-systems covered (social environments, work environments, and learning environments), the taxonomy is not fine-grained enough to distinguish between different kinds of interactive spaces for collaborative work. All interactive spaces for co-located collaborative work are essentially *perch-sized* (1 “perch” = 5 meters), *few-to-few* person multi-display eco-systems.

Table 2.2 provides a general overview of the past decade of perch-size, few-to-few interactive spaces, with a special emphasis on systems which have undergone collaborative user evaluations. There are several trends which are apparent (especially when considering research which occurred in parallel in single-display groupware). Interactive whiteboards were incorporated into interactive spaces

earlier than tabletops because they are easier to design and implement (e.g. no constraints of orientation, users have more reach on the surface). Multitouch and stylus interaction are still active areas of research for both vertical and horizontal surfaces. Both interactive tabletops and whiteboards have been increasing in size and resolution, especially as the technology which allows for multitouch interaction at a larger scale is maturing. DiamondTouch tabletops, with their user identification system [Dietz 01], have become less common, while digital Anoto pens/styluses have become popular for the flexible cross-surface interaction they can provide. Despite these two common components, interactive spaces use a range of heterogeneous and customized hardware systems to achieve their goals. Because multitouch technology is maturing and becoming increasingly used, I believe there will be some standardization around this technology in the future, and interactive space will include both horizontal and vertical displays. For comparison, Table 2.2 also includes our own working prototype, the TATIN-PIC interactive space, which will be introduced in later chapters.

There exists many different styles of interactive spaces for collaborative work. The design choices depend on what kind of activities researchers wish to support in the interactive space. The difference in the goals of each interactive spaces makes comparison difficult. It drives the overall strategy for the layout of the room and the physical devices inside, making an interactive space more appropriate for one type of activity but not another. For example, returning to some examples of the related work, Hilliges et al. [Hilliges 07] and Geyer et al.’s *AffinityTable* [Geyer 11a] are interactive spaces for meeting rooms, while Biehl et al.’s *Impromptu* multi-display environment [Biehl 08] aims at supporting collaborative activity outside the context of these meetings and in everyday work (Figure 2.4.1). Table 2.2 focuses on the concept of interactive meeting rooms as much as possible, though some of the interactive spaces are actually presented as shared design studios [Haller 10] or information sharing spaces [Wigdor 09b].



Figure 2.4.1: Examples of interactive spaces for meeting rooms (left, [Hilliges 07]) and for deskwork environments (right, [Biehl 08]).

Of the evaluations that have been conducting in interactive meeting spaces, the

Table 2.2: Existing interactive spaces for co-located collaborative work. This list focuses on meeting room implementations that have been evaluated with potential users engaged in a task.

Name	Year	Citation	Tabletop (Size) Input kind, multitouch	Vertical display	Other input devices	Evaluation task	Group size
Interactive Mural (running PostBrainstorm)	2002	[Guimbretière 01a, Guimbretière 01b]	Traditional	Interactive (1.83m by 1.07m), Stylus, single-user	Some tangible tools, laptops	Brainstorming % Sketching (product ideas)	4 to 7
Dynamo	2003	[Izadi 03]	Traditional	Interactive (2.5m by 2m), Remote mouse and keyboard, User-specific	Laptops	Creating an informational poster	4
MultiSpace Project	2006	[Everitt 06]	Interactive (0.85m by 0.64m), Multitouch, DiamondTouch	Interactive (1.6m by 1.1m) Single touch or stylus, Single user	Laptop, tablet, wireless keyboard and mouse	Drafting exam questions	3
Shared Design Space	2006	[Haller 06, Brandl 07]	Interactive (1.68m by 1.19m), Stylus, User-specific stylus	Interactive (1.19m by 0.84) Multitouch or stylus, User specific	Laptop, tablet, tangible palettes for menus	No evaluation	-
<i>No name</i>	2007	[Hilliges 07]	Interactive (1m by 0.6m), Stylus, Divided surface	Interactive, (< 5m by 2.5m), 1/3 multitouch 2/3 hand tracking, User agnostic	-	Brainstorming (equipment list for an arctic expedition)	2
WeSpace	2009	[Wigdor 09b]	Interactive (1.04m by 0.86m), Multitouch, DiamondTouch	Display (3.04m 1.52m), input from tabletop and laptops, user-specific	Laptop	Researchers discussing ideas for a scientific proposal	3
MemTable	2009	[Hunter 11]	Interactive (1.06m by 1.63m), Multitouch & Stylus, User agnostic	none	Laptops, tangibles, keyboard	Planning and design for a restaurant	4
NiCE discussion room	2010	[Haller 10]	Traditional (N/A), Anoto pens and paper, User agnostic	Interactive (4.5m by 1.1m), Stylus & handheld TUIs, user-specific	Laptop	Planning and design for a restaurant	3
AffinityTable	2011	[Geyer 11b, Geyer 11a]	Interactive (1.37m by 1.08m), Multitouch & TUI	Display (1.62m by 0.75m), Input from tabletop, user agnostic	-	N/A	3 to 4
WILD room	2011	[Beaudouin-Lafon 11, Gjerlufsen 11]	Interactive (1m by 0.56m), Multitouch, User agnostic	Interactive, Input from tabletop, laptops, smartphones, user agnostic	Laptops, tablets, smartphone, tangibles, space-wide object tracking	N/A	-
TATIN-PIC	2013		Interactive (1.6m by 1.4m), Multitouch, User agnostic	Interactive (2.5m by 1.15m), Multitouch, User agnostic	Tablets, smartphones, vocal interaction	Engineering and Design meeting scenarios	5 to 6

results have emphasized the importance of both the interactive tabletops and interactive board displays for collaborative work. Hilliges et al. show how the form factor of the tabletop supports information generation tasks and divergent thinking, which complements the role of the whiteboard for convergent thinking and structural tasks [Hilliges 07]. The *AffinityTable* [Geyer 11a] also exploits both surfaces to provide groups with both a “shared action space” and a “shared reflection space”, though in this system only the tabletop is interactive. These are important strategies which demonstrate how a horizontal display and a vertical display can be used in tandem to augment the collaboration of a group. These resemble how vertical and horizontal surfaces were used in the observations or preliminary design activities. None of these interactive spaces actually use both a *multitouch* tabletop and *multitouch* whiteboard and therefore it is difficult to extrapolate conclusions about using both in tandem without first testing our own.

There exists another notable deficiency in this list, and that is of the limited amount of users that were used in group evaluations. Each user adds another level of complexity in the system and to the dynamics of the group. The observations from the previous chapter included 4 or 5 users, and based on Olson’s research, small meeting groups in reality can include anywhere from 3 to 7 users [Olson 92]. The related work uses 2, 3 and sometimes 4 users. Moreover, of all the evaluations, only two of them use a comparative evaluation to traditional setups ([Hilliges 07, Hunter 11]). Comparing traditional spaces to interactive ones is sometimes controversial: Nunamaker et al. explain that the difference in media makes comparisons problematic [Nunamaker 91]. However, these comparisons are necessary to better understand the impact that the new media has on group dynamics. Rogers et al. explains that these kinds of comparisons are possible by using a more global approach to measure group dynamics, e.g. through communication and equity of participation [Rogers 09]. Moreover, much can be learnt by measuring the quality of the group’s output at the conclusion of the activity.

One of the goals of this thesis is to explore the advantages of an interactive space which uses both a *multitouch* interactive tabletop and a *multitouch* interactive vertical display to understand how this configuration aids collaborative problem solving over traditional methods.

2.4.2 Shareability in interactive spaces

Shareability, as defined by Hornecker et al. [Hornecker 07], is the extent to which an interface can engage a group of co-located people in shared interaction around the same artifacts. This is a design principle that is especially relevant when considering the physical layout and interface of a multi-user multi-surface interactive space, as the size, orientation, and input capabilities of the system will have an effect on the emergent interaction of group [Hornecker 07].

Two core components of shareability are access points and entry points. *Entry points* are environmental or virtual structures and cues which draw a user in and invite interaction [Hornecker 07]. Examples of entry points include a reminder written on a Post-it note attached to a display, an unfinished document left open on a computer, or the headlines of a newspaper. For an example which revisits coordination in co-located collaborative work, providing an overview of the work would be considered an entry point which allows a new group member understand what work remains to be done. Entry points, as suggested by Rogers et al. [Rogers 09], are very much linked to a user's perception of the state of digital and physical resources. Their presence or their absence can either encourage or inhibit a user from interacting with artifacts, thus dictating their shareability in the digital space.

If entry points provide users with perception and understanding of the state of artifacts in a digital realm, than *access points* provide users with tools and functionality which might mediate interaction with these artifacts. In multi-user systems, these are often designed to control which participants can interact with what, through notions of permissions and territoriality. Often times access points are directly related to the type input device that is used. For example, traditional mouse and keyboard workstations provide access points for only one user, which makes them inappropriate for equitable collaborative work. Large multitouch surfaces, on the other hand, support egalitarian access in terms of an input device, but consideration must be given to virtual access points with regard to the users' reach and movement. There are also perceptual aspects of access points that are important for collaboration, such as visibility and feedthrough [Hornecker 07]. For example, early versions of Wigdor et al.'s WeSpace [Wigdor 09b] (Table 2.2) used long distance pointing and space-reaching techniques from user's laptops to manipulate content on a large vertical display. They found that even though these access points were egalitarian, they disrupted the perceptual access of other user's actions. They resolved this issue by adding an multiuser interactive tabletop to manage the vertical display. The virtual controls on the interactive tabletop proved to support more shareable and perceptible interaction techniques than the controls on the individual's laptops.

There does not exist a sharp categorical distinction between entry point and access points. As Hornecker et al. explain, "access refers largely to making it possible to do something while entry refers to inviting people in, showing them how they can move into the space and help detect access points." For Hamden et al. [Hamdan 13], access points can be considered a subset of entry points in tabletop workspace design.

These notions might seem to be already intuitively understood and executed by designers, but in reality, the interfaces of single display groupware can quickly

become muddled with the notions of ownership and control. For example, Wang et al. [Wang 06] experiment with an interactive tabletop which give users the ability to activate personal ownership markers on shared virtual components. Activating an ownership marker removed access points for other participants for these components and, in turn, decreased the efficiency of the group's work. This allowed a participant to dominate access to virtual components through "aggressive and conquering behavior" which increased frustration and discomfort in groups. Therefore, designers must consider the difference between personal and groups access points which is still very much an open question in interactive spaces.

2.5 Conclusions

I have reviewed the state-of-the-art for four domains of interactive spaces: user's personal interaction (with a special consideration given to multitouch interaction), group's social interaction and collaboration on shared surfaces, interaction within the context of a larger community, and the physical aspects of the environment. The results of this literature review have been developed into design considerations proposed in Table 2.3.

Table 2.3: Design guidelines for interactive spaces by domain.

Multitouch	Shared surfaces	Workflow	Environmental
Favor direct manipulation techniques	Use lightweight orientation techniques	Optimize existing workflow processes	Use tabletops and whiteboards in conjunction
Use physics-based gestures	Use lightweight translation techniques	Consider the roles played in the community	Consider space, movement, and interaction for up to 6 users
Design for "big" and "small" gestures	Limit clutter	Allow access to and sharing of personal documents and information	Evaluate with users within the context of a design activity
Engage user in bimanual manipulation	Consider user-identification systems	Automate output and distribution	Use "shareable" input devices and interaction techniques
Consider extending direct manipulation through instrumental or multimodal interaction.	Consider feedthrough and overview to help group awareness	Facilitate transitions between activities	Design with consideration for entry and access points

Research which pertains to surface computing from the perspective of an individual's interaction is reasonably advanced. Multitouch gestures should be designed

with consideration for direct manipulation techniques, physical effect, a distinction between the large and small categories of gestures, and bimanual manipulation. There also exist many styles of instrumental interaction which can complement direct-manipulation-style multitouch.

Research on shared surfaces is much more complex, for two main reasons. First, researchers from different fields (e.g. CSCW, CSCL, etc) will have domain-specific objectives and different styles of collaboration that they wish to support. Second, the hardware on which research is conducted is extremely diverse and is still rapidly evolving today. Though there have been some standard interactive tabletops distributed within the research community (e.g. Microsoft Surface, Merl DiamondTouch), the vast majority differ in terms of the technology they use for an individual's interaction. Nonetheless, when examining the literature on groupware and the multi-disciplinary approach to collaboration, there are many design considerations that arise. These include lightweight orientation and translation techniques, clutter reduction, user identification and managing group awareness.

Activity theory is especially useful for understanding the mediating role that a community plays in an activity. Proper consideration should be given to the existing social rules and workflows. By analyzing and understanding these process, as I have done in the previous chapter, interactive spaces have an opportunity to optimize workflows, facilitate different roles played by group members, assist users in sharing documents, aid in the output and distribution of the result of the activity, and facilitate transition between activities.

Interactive spaces are an opportunity to unify these three domains into a physical and interactive environment. Multitouch tabletops and whiteboard can be linked together through interaction techniques which facilitate the transfer of information. These physical environments can engage small to medium-sized groups in computer-supported co-located collaboration.

For creative collaborative problem solving, I believe interactive surfaces with these design consideration can augment the interpretation, coordination, and communication of a group. Interpretation in a group is traditionally supported through externalization of complex information and data through well-established creativity and project planning techniques. Groupware can facilitate the use of such representations of information by providing an interface capable of supporting their construction in a more fluid and less time consuming manner. The computing system can be more flexible than traditional media by providing a lower viscosity in interaction and supporting an epistemic exploration of the problem space. Such system can also provide an overview of the current relevant information allowing the team to be more pragmatic in its direction. Coordination in collaboration is just as relevant. Groupware must allow for parallel work, group awareness, and natural roles of collaboration. Finally, unencumbered interpersonal communication during

collaboration is critical for supporting collective interpretation and coordination, and verbal talk is equally as important as nonverbal gestures. Finally, one of the best measures of collaboration is the equity of participation of the group members, which has been shown to directly correlate to group performance.

Part 2: Proposition

"Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods"

(Marc Weiser)

Chapter 3

TATIN-PIC Information Landscape & Interaction Techniques

The TATIN-PIC Project (french for **T**able **T**actile **I**nteractive - **P**lateform **I**ntelligente de **C**onception) at the Université de Technologie de Compiègne is an investigation into multi-user multi-surface computing environments for performing preliminary design activities. Before addressing its implications for distributed infrastructure, I must address the conceptual organization of the user interface as well as the design of new interaction techniques for the environment.

3.1 Information landscape for project-oriented and user-oriented activities

The existing literature only provides glimpses of how interactive spaces manage the high-level, environmental tasks involved in a multi-user multi-surface environment. In order to have a clearer picture of what these tasks might entail, I consider these tasks from the perspective of an operating system shell. The operating system shell in a single-user desktop environment allows a user to address such aspects as user login, file access, and application management. For example, the user interface mechanisms in common WIMP-based operating systems allow users type their username and password to access personal documents which are stored in their own private hierarchy of files. Through a file browser interface or by using desktop icon shortcuts, they also allow users to launch different applications inside of frames or windows. This allows users to view and manipulate file data inside windows, and suspend and resume multiple computing processes by using these windows.

This approach uses a widespread and well-established WIMP metaphor for man-

aging the high-level tasks of a single-user desktop system. It provides an operating system shell with a clear visual and interactive hierarchy of information. I refer to this as an environment's *information landscape*. This is similar to the notion of information hierarchy or visual hierarchy in interaction design, but "landscape" is more general and better applied to interactive spaces because a strict notion of hierarchy should not be assumed.

Our first task will be to discuss how the information landscape of an interactive space should be presented and managed. Typically, interactive spaces in research are built within a single application framework, where only one application at a time runs in full-screen. This allows researchers to (justifiably) side-step the problem while they focus on specific lower-level activities or interaction techniques. For example, much of the work reviewed in the previous chapter successfully presented and validated their experimental propositions within the context of a single application framework. But considering these interaction techniques in the context of a high-level information landscape could potentially be problematic, *especially in the context of gesture-based systems* because of conflicts of interest that might occur between the lower-level and higher-level aspects of the system. These conflicts are very much present in today's existing gesture-based systems. For example, applications developed for Apple's iOS and the iPad can only use three simultaneous points of contact, because gestures using four or five points of contact will automatically be understood as a high-level interaction for closing and changing applications. This inherently limits the possibilities of interaction provided to application designers.

There have been a handful of researchers that have directly addressed this in the literature, mostly in the context of interactive tabletops. The problem of activity launching and activity switching has been identified and discussed by Ackhad et al. on interactive tabletops [Ackad 10]. They present a solution named *Switch* for activity switching on interactive tabletops which uses a menu component that can be activated by any participant. Hamdan et al. [Hamdan 13] presented a conceptual framework for identifying the qualities that a "surface manager" should provide for a multi-user interactive tabletop. In their work-in-progress presentation, they address the need for accessing information resources, partitioning the surface of the tabletop for different activities, and transitioning between activities.

The problem of providing user login for access to personal documents or user-specific functionality has also been addressed at a high-level within the context of interactive tabletops. For example, Anthony Collins [Collins 11a] discusses file navigation on the interactive tabletop and presents *Focus* which allows users to browse and view files they have previously made available by using the *Focus Exporter* on their personal machines. Again, this occurs within a single application framework, and it is unclear how their collaborative file browser application can

be used in conjunction with other solutions such as the same researcher’s group brainstorming application, *Firestorm*, especially if there are conflicts with gesture primitives. Figure 3.1.1 shows an example of the two different applications.

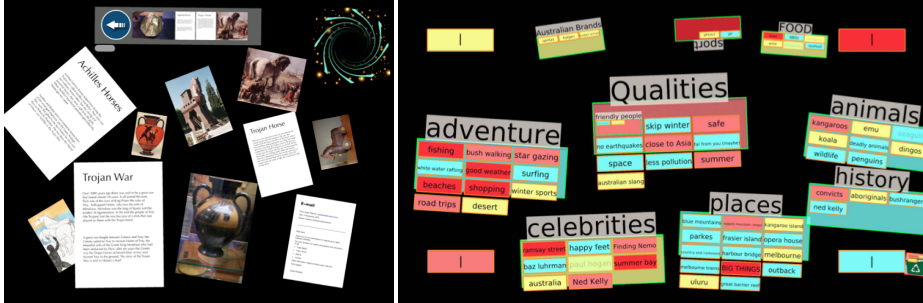


Figure 3.1.1: The single application framework on interactive tabletops, personal file browsing application *Focus* (left) [Collins 11a, Collins 11b] and a group brainstorming activity *Firestorm* (right) [Clayphan 11].

In this section, I present the approach used in the TATIN-PIC project which clarifies and extends these issues in the context of an interactive space. The following subsection uses a scenario to identify constraints which originate from the perspective of preliminary design team’s workflow and community. Then, before discussing the information landscape of the interactive space, it is necessary to discuss the physical environment of the interactive landscape. Therefore, I consider the physical environment of the interactive space and the impact that the physical configuration has on the *shareability* of the interactive space for preliminary design. With the workflow and the physical environment introduced, the information landscape of the interactive space is presented.

3.1.1 Workflow and community in interactive spaces for preliminary design

Informed by the observations conducted in Chapter 1 and motivated by the design guidelines for workflow in Chapter 2, in this subsection I present a general scenario for preliminary design activities and define several terms regarding community and workflow. This work was also informed by research conducted with members of my team, such as Joiron et al.’s preliminary design project ontology [Joiron 13b] and Guerra et al.’s [Guerra 12] study of preliminary design methodology.

Figure 3.1.2 presents an illustrated example scenario of the preliminary design phase of a mechanical engineering project. This is framed in terms of a problem-solving exercise, with each colleague exploring different diverging and converging pieces of the problem space. Therefore, the width of the colored line in each user’s

timeline represents how divergent (large) or convergent (small) their work activity is. The end result of the preliminary design phase is a detailed project plan for the company which presents a solution and project timetable. After the preliminary design phase, the project will likely continue through a phase of detailed design, industrialization, and ultimately commercialization.

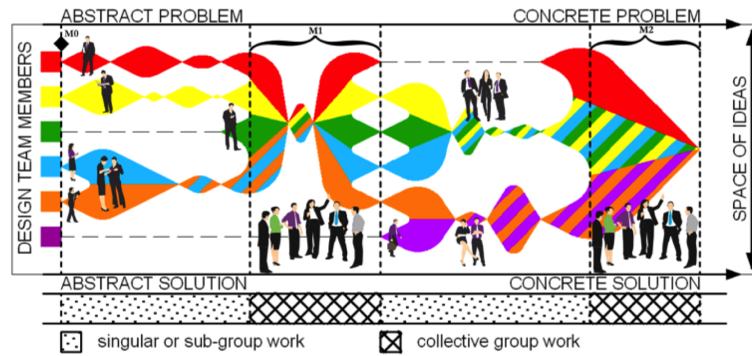


Figure 3.1.2: An illustrated scenario of a preliminary design project (originally presented in [Guerra 12]).

The colleagues in this scenario are perhaps from different departments of a company (e.g. Engineering, R&D, Commercial, Human Resources, etc.) but collaborate closely with one another for the purpose of the project. After a phase of personal, parallel work (which can sometimes be collaborative in nature), these colleagues unite for a co-located meetings. These meetings are initiated by a project manager who invites personnel, schedules an appropriate date and time, and reserves the interactive space. During these meetings, the participants will want to share any relevant personal work or advancements that they have previously conducted. The project leader, or moderator, may see fit to engage the group in any number of collaborative problem-solving activities (e.g. causal analysis, brainstorming, project planning, etc.), whether they be premeditated or spontaneous, depending on the nature of the discussion and the scope of the problem at hand. If one of these team activities is conducted, the project manager will digitally distributed the results of the activity to the team at the end of a meeting. Additional objectives might be set for the team, and a new meeting might be scheduled in the future, launching a new round of personal, parallel work for the colleagues.

A group develops a preliminary design project through one or more sessions, to which users are connected. A *session* represents a single meeting of a group of users. This may last an entire workday or half an hour. Different sessions can involve different users as some users may not be present in every session. Because TATIN-PIC is a meeting room, it may be used for any number of projects, by any employee or project leader who reserves it. This requires two different kinds

of login, a project login to access project-related information resources (collaborative activities) and a personal login, to access personal information resources (for information sharing, presenting, etc).

Therefore, the interactive space must manage not only project-centered activities, but also participant-centered activities.

- A *project-centered activity* involves the active construction and manipulation of project-related information resources, such as the cause-and-effect diagram in causal analysis, the flowchart for project planning, and the affinity diagram for brainstorming. This could also include other collaborative exercises (which I have studied but not directly observed) include risk assessment, using a risk matrix, and functional analysis, using a FAST diagram.
- A *participant-centered activity* involves any presentation, annotation and sharing of documents on a personal and interpersonal level, e.g. accessing a website to find information on norms and standards, or photographs of prototypes from previous work.

There is by no means a dichotomy between the two types of activities in terms of their motives. They are meant to be concurrent and mutually inform one another. They only pertain to different kinds of information resources. Figure 3.1.3 represents the relationship between these concepts.

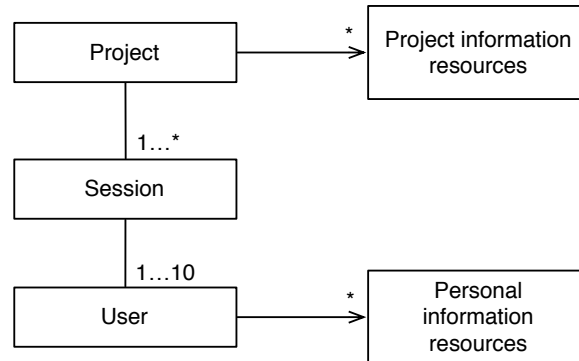


Figure 3.1.3: The relationships between project and user used when designing the TATIN-PIC information landscape.

Finally, the team's community and organization also introduces two different kinds of members of the session.

- A *participant* is a primary user of the interactive environment. He or she arrives at the meeting inside the interactive environment with or without

mobile devices such as a tablet, smartphone or laptop. In terms of the scenario, a participant can be a specialist of a certain domain, such a mechanical engineer, an industrial designer, or even perhaps a client who happens to be on-site for the meeting. Therefore, their familiarity with the environment is not assumed. They may also be able suspend and resume any participant-centered activities throughout the meeting.

- A *moderator* is a specialized participant, with project-level responsibility. This will often times be a project manager. The moderator has the authority to suspend and resume the project-centered activities for the group.

These two roles, and the two kinds of activities they involve, are useful for organizing the information landscape of the interactive space. With this perspective, it is clear that a participant will have to perform a user authentication step in order to have access to their personal documents, while a moderator will have to perform an additional project authentication step, to access project related documents.

3.1.2 Environmental concepts of meeting rooms and interactive spaces

As discussed above, there exist two broad categories of activities that will be conducted in the interactive space. This subsection investigates how the devices from the physical environment can support these activities. Based on our review of related interactive space (Table 2.2), I propose four coarse categories of devices used in these interactive space. Then, using the principal of shareability, I examine how these devices can be used to support the two different kinds of activities in the interactive space.

3.1.2.1 Categories of devices in meeting rooms and interactive spaces

Interactive space have four kinds of devices: primary shared surface, ancillary shared surface, mobile personal device, and specialized I/O devices. These were derived by examining related work and adopting a “user view” of the devices mentioned in these interactive spaces. These are introduced and explained below:

- The *primary shared surface* is the main platform of collaboration in an interactive space, providing a surface for shared action for multiple users inside the environment. It is a board or wall-sized surface with enough room to accommodate concurrent actions from multiple users. There is only one such surface in an interactive environment. This will be typically an interactive wall/board display or an interactive tabletop. The device can either allow for *direct* interaction (e.g. tangibles, stylus, multitouch gestures) or *distal*

interaction through specialized I/O device (which will be discussed below). In both cases, a primary shared surface can be either *multi-user, identity-specific*, where the display identifies the owner of an interaction event by using specialized tracking hardware or by requiring that each user interact with an assigned input device (e.g. stylus), or *multi-user, identity-agnostic*, where the surface has no hardware device to identify users but has an interface which is designed with certain territorial conditions so as not to require it.

- *Ancillary shared surface* are surfaces which support complementary aspects of collaborative activity. These again may be interactive tabletops, or board-sized displays. They are either multi-user/identity-specific, multi-user/identity-agnostic, or simply single-user interfaces (e.g. certain interactive board displays are design to be used by one member of the team at a time in a presenter-audience style of interaction). The environment can include any number of ancillary supporting device, provided the interactive space can physically accommodate them.
- The environment also includes any number of users' *mobile personal devices*. These are the devices which belong to the participants and are brought to a meeting so that information resources on these devices may be used for reference or shared with other users. Examples of such devices are laptop, tablets, or smartphones.
- Finally, *specialized I/O devices* are devices that provide specialized input or output devices. They can be *space-wide input devices* such as cameras, 3D tracking systems, omnidirectional microphones or *user-specific input devices* such as headset microphones, styluses, and digital pen and paper. The same is true for output devices; for example, sound can come from either speakers in the room (*space-wide*), or headphones worn by the users (*user-specific*). The input devices from mobile personal devices (mouse, trackpad, touchscreen) can also be extended to the interactive space, using space reaching techniques, and used as user-specific input devices. Therefore, a user's smartphone could be considered to be both a mobile personal device and a user-specific I/O device.

Outside of the interactive and digital aspects of the system, it may also be helpful to provide groups with complementary traditional surfaces as well, such as traditional whiteboards and paperboards. These can be completely independent from the system, or coupled with space-wide input devices such as cameras which can store and distribute their information digitally.

3.1.2.2 Device configurations and the principle of shareability

The actual configuration and implementation of an interactive meeting space from the above components should be done with respect to the activities that the interactive space will be supporting. Such choices will have a particularly strong impact on the *shareability* of the interactive space, i.e. the extent to which an interactive space can engage a group of co-located people in shared interaction. (This design principal is discussed in Section 2.4.2). Using different physical device configurations will provide different entry points and access points in the interactive space. These entry and access points will be more appropriate for certain kinds of activities than others.

There is no formal methodology for analysis-by-shareability. Similar to Activity Theory discussed in Chapter 1, it is a lens through which researchers can observe interaction. I am applying it to the physical form factor of the components in the environment, to the input devices used to interact with the environment, and to the general high-level functionality of an environment, but it could very well be used on a much smaller scale to analyze the entry points and access points of individual virtual components on the interactive tabletop. However, this analysis will remain high-level.

I propose in this subsection a conceptualization of interactive spaces based on the entry and access points afforded by the devices in the environment and the general class of functionality they provide (either project-oriented or participant-oriented). By analyzing interactive spaces in terms of their entry and access points, one can understand how different device configurations might impact the shareability of the environment. This will be important for deriving a new configuration that is appropriate for supporting the two different kinds of activities (project-centered and participant-centered) that I identified in the preliminary design scenario. This requires understanding how a device supports *entry and access points for teamwork* and *entry and access points for personal work*.

Interactive meeting spaces should manage teamwork through project-oriented information resources and personal work through participant-oriented information resources (Figure 3.1.3). The devices in interactive spaces can provide entry and access points to teamwork, as well as entry and access points to personal work.

- Devices and functionality that emphasize teamwork entry points display the state of collaborative artifacts and motivate activity and actions on these shared artifacts. Teamwork access points provide users with the means to manipulate shared artifacts and advance the state of collaborative work.
- Devices which emphasize personal entry points and personal access points allow users to perform a personal task, such as creating a reminder to perform

a certain task or referencing information from a personal document during discussion.

Personal access points can, and often do, play an explicit role in collaborative activity. For example, if the device on which the user has accessed personal information also provides some form of a teamwork access point, then the user would be able to share the personal document with the team by transferring it to a shared surface and thereby entering the document into the collaborative activity.

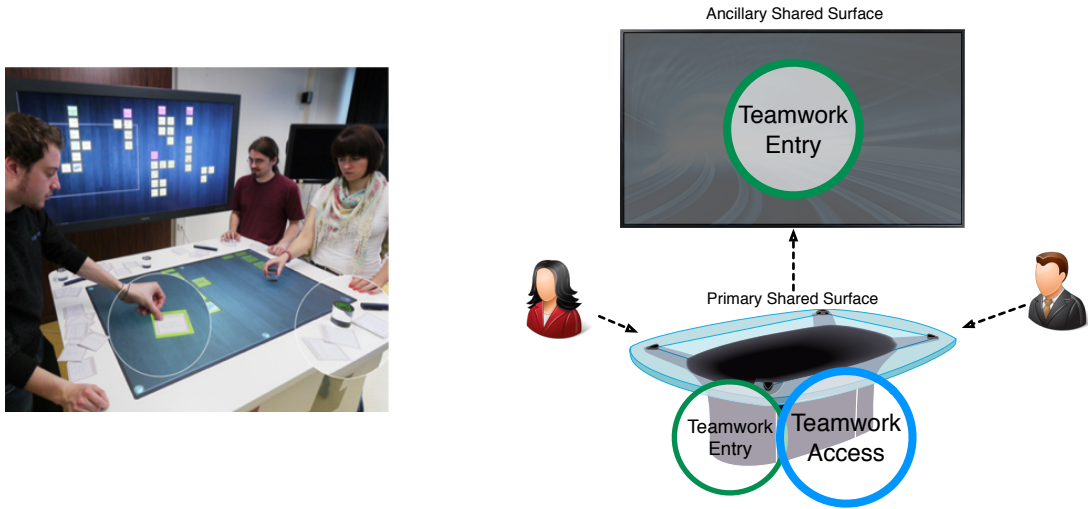


Figure 3.1.4: The *AffinityTable* proposed by Geyer et al. (left, [Geyer 11a, Geyer 11b]) and the shareability of the devices with regard to entry and access to teamwork (right).

I begin by considering the *AffinityTable* proposed by Geyer et al. [Geyer 11b, Geyer 11a]. The purpose of this interactive space is to provide an environment where participants can build affinity diagrams through a brainstorming session, i.e. project-oriented work. This interactive space was not designed to provide entry and access points for personal work. It uses an interactive tabletop as a primary shared surface and a vertical display as an ancillary shared surface. These are both entry points into the group work. The tabletop is directly manipulable through a rich set of tangible and mixed-reality interactions, and therefore it provides a rich set of access points to the teamwork. The vertical display is not directly manipulable but rather provides an overview of the teamwork. Therefore, it provides an excellent entry point in the group work, but it does not have any direct access points.

Next, the interactive space proposed by Hilliges et al. [Hilliges 07] shares many common attributes with the *AffinityTable*. They both share a common goal of

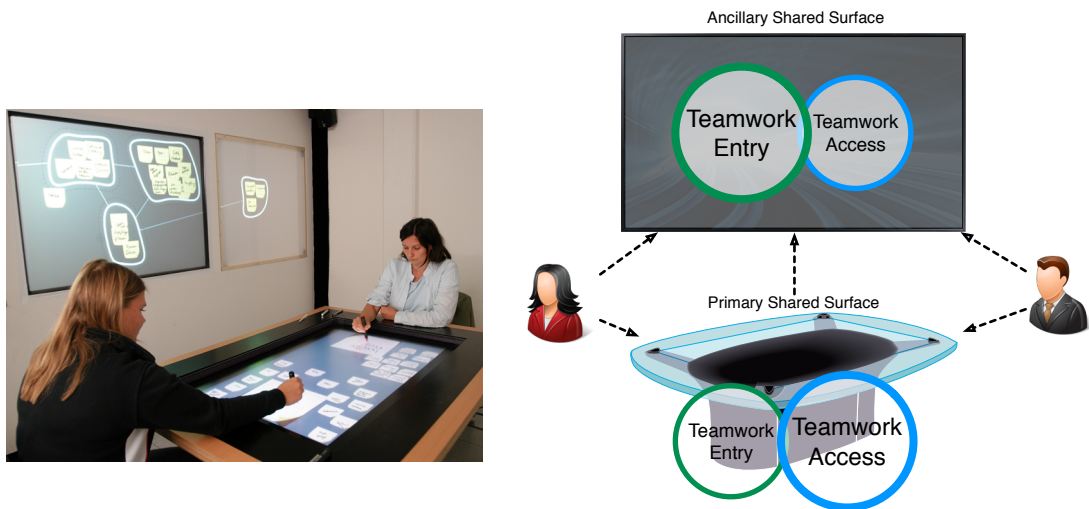


Figure 3.1.5: The interactive space proposed by Hilliges et al. (left, [Hilliges 07]) and the shareability of the devices with regard to entry and access points.

supporting the construction of affinity diagrams through brainstorming sessions. However, in the case of Hilliges et al., both surfaces are directly manipulable: the tabletop surface uses stylus input while the vertical board uses stylus and multitouch. Therefore, the vertical surface also provides an access point into teamwork. This is illustrated in Figure 3.1.5. Note in the figure I am emphasizing the interactive tabletop as a larger access point than entry point, and the opposite with interactive board. This is because, as the authors indicate, the interactive board provides an overview, which serves as a greater entry point into teamwork, while the tabletop is used to parallelize work, which places more emphasis on egalitarian access points in teamwork.

The difference between the AffinityTable and Hilliges et al. 's approach is a subtle, but can impact collaboration in a number of ways. Collaboration with the AffinityTable remains balanced among all participants gathered around the tabletop, because access points are equally distributed among all participants. Access points are only limited where physical reach and territoriality come into play. This system has been carefully crafted to maintain the egalitarian notion of contribution that comes with idea generation and brainstorming.

Hilliges et al., on the other hand, clearly provide two different way to engage with the affinity diagram, by providing access points to the collaborative work on the vertical display. The authors indicate that this approach was helpful because the interactive tabletop is better at supporting diverging, parallel work, while the

vertical surface is better at supporting converging work. This allows a team to shift between face-to-face collaboration and shoulder-to-shoulder collaboration.

Another kind of interactive meeting space is Wigdor et al.'s WeSpace, which use a large wall-sized display as their primary shared surface of collaboration. The wall-sized display captures the current state of collaborative artifacts of the group, providing a strong overview and teamwork entry point to the users. Because the wall display is not directly manipulable, an ancillary surface (an interactive tabletop) reproduces the content displayed on wall and allows users the ability to manipulate the content, providing teamwork access points, and to a lesser extent, teamwork entry points. Finally, the interactive space also supports personal devices from which personal windows and artifacts may be replicated and transferred onto the primary displays. This occurs on personal laptops, which provide both personal entry and access points. Because of the software on their computers, they can add and remove artifacts to the vertical display, which means that these personal devices provide access points to teamwork. This style of interactive space is designed for meetings which focus on information sharing, information visualization, discussion, sensemaking, and reflection (Figure 3.1.6, left).

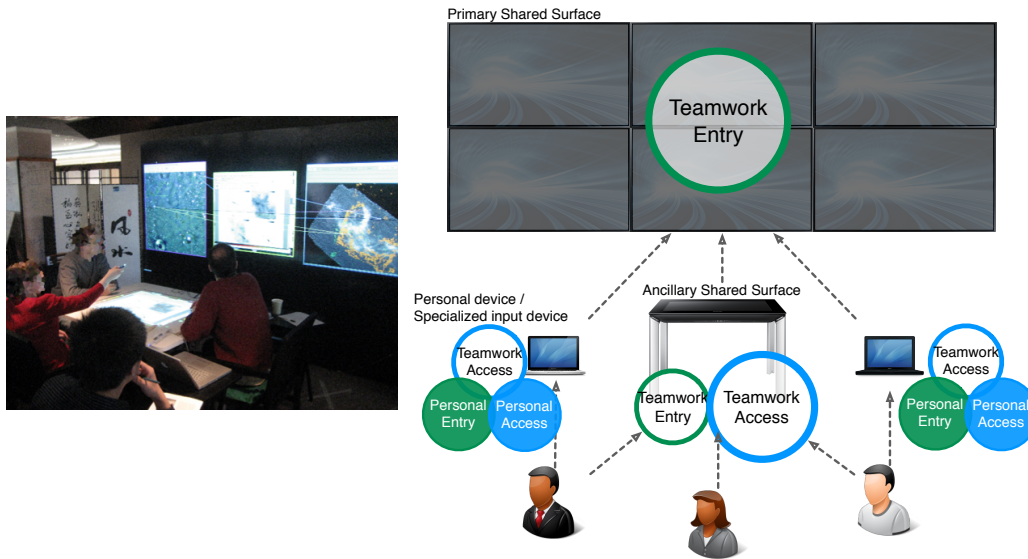


Figure 3.1.6: Wigdor et al.'s *WeSpace* [Wigdor 09b] with regard to their entry and access points to team and personal work.

In order to better support the activities I observed in Chapter 1, teams require an interactive space which can support access to and sharing of personal information resources, while still allowing the group to perform collaborative exercises such as project planning and causal analysis, therefore I propose a combination of the two

kinds of interactive spaces (Figure 3.1.7). The TATIN-PIC interactive space uses an interactive tabletop and an interactive board display which are both directly manipulable through multitouch sensors, similar to Hilliges et al. [Hilliges 07]. Wigdor et al.'s support for personal devices is a sound strategy for allow entry and access points to personal, shareable work [Wigdor 09b], but it is also important to give users who have not transported their personal devices the ability to remotely access relevant information resources from the surfaces of the interactive tabletop. Therefore, it would be beneficial to provide access points to personal work, directly on the tabletop as well.

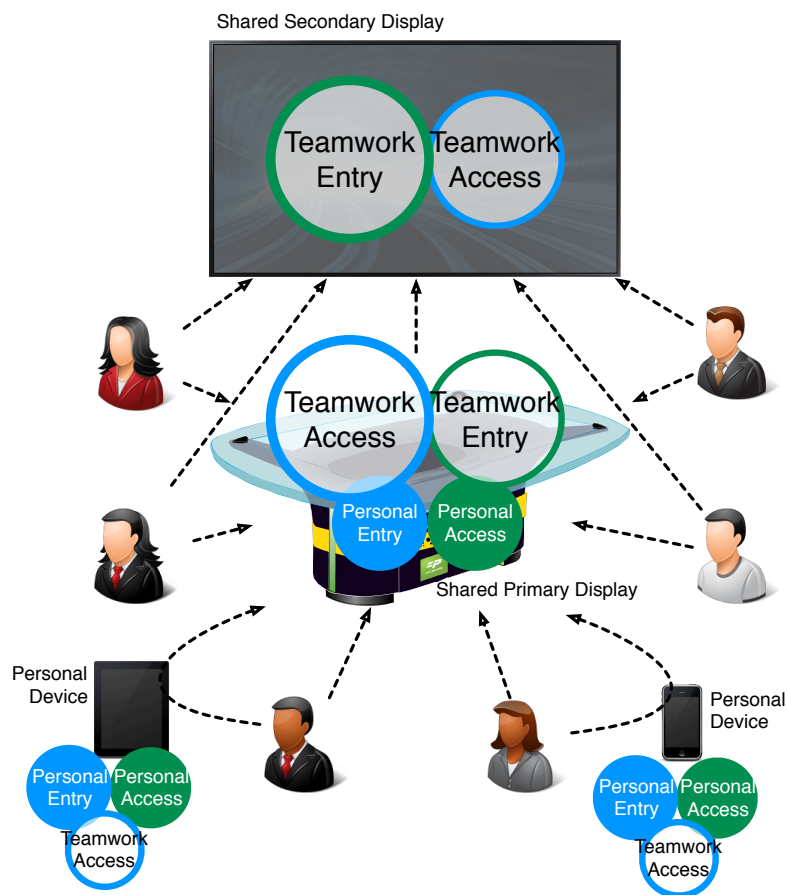


Figure 3.1.7: The TATIN-PIC platform with regard to its entry and access points.

3.1.3 Multi-surface layered information landscape

With the physical form of the interactive space established, my goal here is to provide clear visual and interactive landscape of information which maintains a separation of concerns for the two activities (project-centered and participant-centered) identified in the preliminary design scenario. This should also address such functionality such as launching and changing applications, accessing files, and user login.

The interactive space approaches this problem by separating the user interface of our shared displays into two layers. These layers extend to all shared surfaces in the interactive space as illustrated in Figure 3.1.8.

3.1.3.1 Participant-oriented layer

The first layer is dedicated to participant-oriented activity and information resources, allowing a user to open and display personal PDFs, pictures, websites, and other documents. These are widget-based documents that do not run in full screen mode. Though they are launched by a single user, who has access only to the documents in her own personal file system, they can still be used collectively. For example, participants may use the PDF widget to clone a PDF to share with another user, or they may also activate a menu in the PDF widget that allows them to directly transfer the file to another user.

The interface uses a *movable circular menu* as the personal access point to this functionality. Users may access their personal circular menu after they have completed a login identifying themselves. User login has been done any number of ways on interactive tabletops in related work, such as using tangible badges or phones, or the biometrics of the user hand. (In our current prototype, users must type their names into a virtual keyboard which is activated by the moderator of the session. We do not consider the security aspects related to personal identification on groupware surfaces.)

Once the personal circular menu is activated, users may open a file navigator to browse and view their personal files that they have placed on some shared network drive, or an Internet navigator for accessing websites. All widgets and menus that belong to a user are coded with a designated color. An example screenshot in Figure 3.1.9 from the final implementation shows this personal access points, with a user's virtual keyboard.

3.1.3.2 Project-oriented layer

The second layer runs behind the first and is dedicated to project-centered activities and information resources. These activities are contained within *applications*

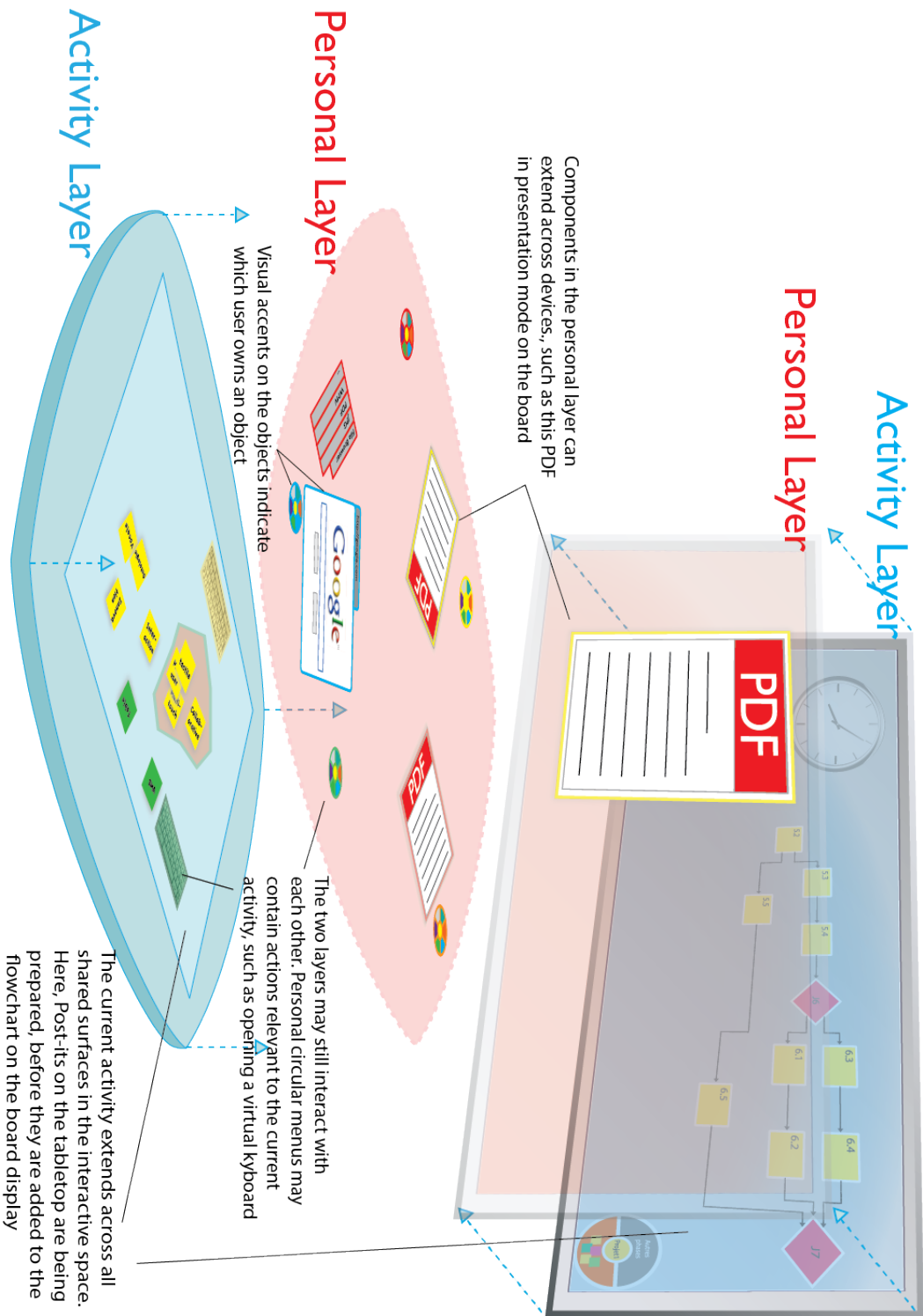


Figure 3.1.8: Personal and activity layer extending to both surfaces in the interactive tabletop.

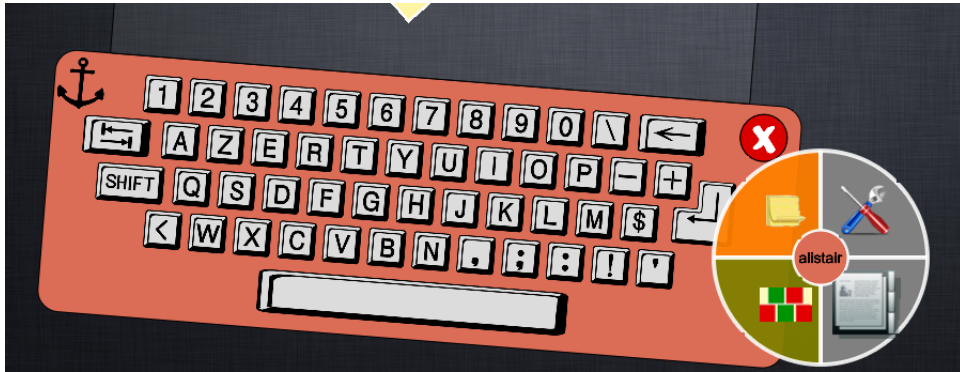


Figure 3.1.9: The circular menu serves as personal access point on the tabletop surface. The keyboard widget is opened on left.

which are each dedicated to their own collaborative exercise such as brainstorming, project planning or causal analysis. Because the purpose of such activities is to engage the entire group in focused problem-solving and decision-making, an application is constrained to always run in full-screen mode and across all shared surfaces in the meeting space. In doing so, there is a large surface area for the collaborative activity to support spatial reasoning, and give users freedom to frequently pivot from a horizontal surface to a vertical surface as is naturally done during the collaborative activity I observed.

Note in Figure 3.1.8 that there can still be notions of user ownership of components in the project-oriented layer. For example, the virtual keyboards that are part of the project application still belong to certain participants, but these components contribute to the project-oriented information resources, and not to the user's own personal information resources. Therefore, they are still included in the project layer.

Resuming and suspending applications is part of the responsibility of the moderator, who may access such functionality from the personal circular menu. When the moderator changes from one application to another, the circular menus of all participants may change to include items that are relevant to the new context. These new menu items may launch widgets which are both personal, and also part of the current activity. For example, in the causal analysis activity, users can open personal virtual keyboards to create Post-it notes. Other functionality that the moderator may have in his personal menu includes automatic export and distributing (via a shared network drive) the contents of the current group work.

3.1.3.3 Emphasizing the difference between layers

To further emphasize the difference between the two layers, visual and interactive cues are used. The project-oriented layer adheres to the effects of a physics engine, with components sliding and rotating even after they have been touched, while the personal layer does not. The components of the collaborative activity layer are therefore, slightly more engaging and easier to flick to other group members, while the personal layer provides finer control and organization. By using a physics engine, one can assign objects their own mass, friction, and center of gravity, allowing them to be rotated with only one finger. These effects, also allow for intuitive translation effects, where objects are displaced with respect to the speed of a user's drag gesture. Designers can also specify how objects collide and/or stick together, allowing them, for example, to be aligned to the borders of the tabletop relatively easily.

3.1.3.4 Using project-oriented layers for complementary tasks in the activities

I have discussed how the project-level layer extends to the every shared surface in the room, but this is not to imply that the same the functionality should be exactly the same on both surfaces. In the project activities observed in Chapter 1, the vertical surface is used for a different purpose than the horizontal surface. For the TATIN-PIC interactive space, the vertical surface can have a different context than the horizontal surface. For example, when the causal analysis activity is initiated by the moderator, the context of the tabletop will switch to one that is appropriate for creating Post-it notes, while the vertical display will switch to one that tailored to the creation of the cause-and-effect diagram.

There are a few differences between the activity layer on the tabletop and the activity layer on the board display. The aforementioned meta-physical interactions, felt more appropriate on the horizontal surface, rather than on the vertical board display. This decision was made for a pseudo-physical reason: a virtual component of the interactive tabletop sliding on the surface of the tabletop seems appropriate, while the same effect on the vertical display seems to deny the effects of gravity. Also, as the horizontal display is inherently multi-user, the canvas on the tabletop cannot be panned and zoomed without disrupting the activity for all users. The vertical display however, is design to be used more often by a single user at a time, and therefore the canvas can be moved using “large” gestures, i.e. four-finger drag action to pan, and a five-finger pinch action to zoom.

3.1.4 Conclusion

This section presented a layered approach to the information landscape of interactive space for meeting rooms. This approach maintains a clear separation of concerns which arise from requirements originating from the workflow of a multi-disciplinary preliminary design project. This information landscape is applied to a physical and interactive environment, which has optimized for preliminary design activities.

In the context of interactive spaces, the problem of a clear multi-surface information landscape has also been recently addressed by Jetter et al. [Jetter 12b], from whom I borrow the term “information landscape”. They consider a zoomable, object-oriented information landscape (ZOIL), which provides many advantages, some which were not considered in this approach. However, they do not maintain a separation of concerns between participant-centered and project-centered activities.

3.2 User interaction techniques

Designing the applications required creating a set interaction techniques which work across applications, some of which are entirely new, and some of which are derived and modified from existing work. These interaction techniques address grouping and hierarchization of components, transferring components from one surface to another, user-specific interaction, and multimodal interaction techniques.

3.2.1 Interaction techniques for grouping

Existing grouping interaction techniques can be generalized into three categories. These are illustrated in Figure 3.2.1.

- Containment techniques involve a container component from which objects maybe be added and removed. An examples of this is the “bin” technique presented in [Geyer 12], which allows a user to interact directly with the container making it larger or smaller to accommodate more objects .
- Proximity techniques allow the grouping of objects based on their proximity to other objects. Watanbe et al design a “bubble” technique which gives each object a surrounding bubble, which can be merged with other bubbles or split into different bubbles [Watanabe 07].
- Finally, there are gesture-based techniques which contrast the object-orientedness of the first two techniques. Gesture-based grouping techniques are

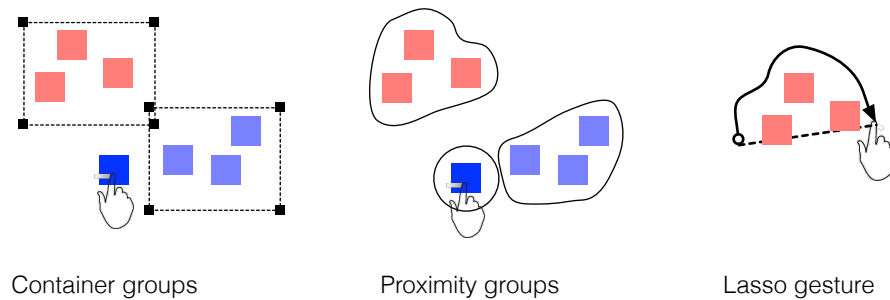


Figure 3.2.1: Three different approaches to grouping: containers, proximity, and lasso gestures.

triggered by placing two vertical hands on the surface of the tabletop such as done in [Tse 08], or by drawing a “lasso” around objects.

Gesture-based grouping techniques can also be mixed with the first two techniques, such as is done in *Firestorm*, a tabletop brainstorming application (Figure 3.1.1, right). In this application, users perform a lasso gesture around objects and then release the lasso inside a container [Clayphan 11].

Strong grouping techniques should allow low-viscosity interaction and functionality which facilitates the creation of groups, the removal of items from group, the addition of items to groups, the modification of items in groups, merging groups, splitting groups, adding labels, removing labels, and modifying labels in groups. The gesture-based techniques usually only consider the initial grouping action, and special consideration must be given to the remainder of possible actions. It would be unrealistic to assume that a solution could be designed which allows a user to perform each of these solutions effortlessly: there are inherent design trade-offs in each approach that will under-prioritize certain actions and emphasize others.

For example, each of these three interaction techniques for grouping do not directly address how groups may be labelled, nor do they address how groups may be used to form multi-level hierarchies i.e. groups and subgroups. They can be modified and extended to include this labeling and multi-level functionality, but this does cause certain aspects of their behavior become unclear. For example, with the bubble technique, users can draw a line across a bubble to split a group into two groups. But, with labelled bubbles, it is uncertain what would happen to the label of the old group and how this should affect the labels of the two new groups. Moreover, if the split action crosses a subgroup, it is unclear if this also divide the subgroup into two subgroups. These ambiguities are illustrated in Figure 3.2.2.

The metaphor used for a grouping techniques should not have ambiguities when considering multi-level hierarchies in groups. This introduces new functionality

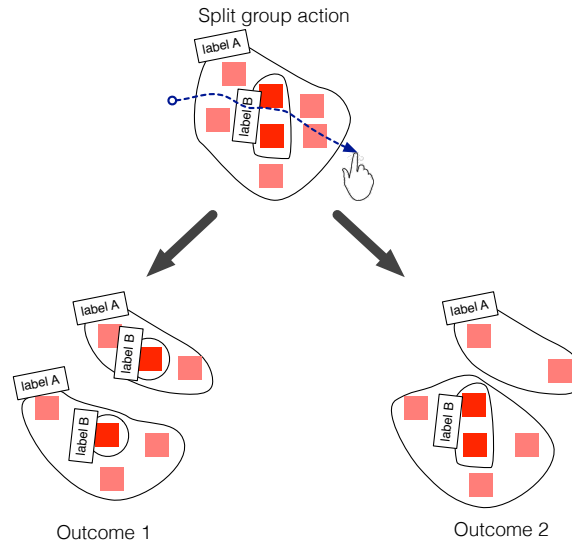


Figure 3.2.2: A hypothetical extension of the bubble technique to include labels and multi-level hierarchies. The outcome of the “split” gesture actions is ambiguous, and this figure illustrates two possible outcomes, though more variations are possible.

such as adding a group as a member to an existing group, removing a subgroup from a group, adding labels to subgroups, and removing labels to subgroups. This can conflict with existing functionality, such as merging groups. For example, consider the surface of a tabletop which contains a group of two items and a group four items. In the bin technique, if the user drags the two-member group onto four-member group it is unclear if this should add the two-member group as a subgroup, or if this should merge the groups to create a new six-member group.

Given the range of functionality required to support multi-level, labelled groups, and the ambiguities that arise with these metaphors, a new interaction technique that can effectively handle as much of this functionality as possible is required. I present a new grouping interaction, which combines proximity-based grouping interaction and a directed graph to capture multi-level hierarchies. This allows us to leverage the effectiveness of multitouch proximity-based grouping, as found by [Geyer 12], into a metaphor which can accommodate multi-level hierarchies.

Figure 3.2.3 and figure 3.2.4 illustrate three basic functions: the creation of the group, adding an item to a group, and adding a label to a group. These are done using direct manipulation multitouch gestures. Note that the distinction between adding an item to a group and adding an item as a label of a group relies on a distance threshold.

The interaction technique also addresses expansion and contraction of groups (tapping on the center circle of a group), adding subgroups (similar to adding an item), removing items (done by “tearing away” an item from the group), and replacing the label of group. The remainder of the illustrations are provided in the appendices. Our team has also registered a patent for this interaction technique.

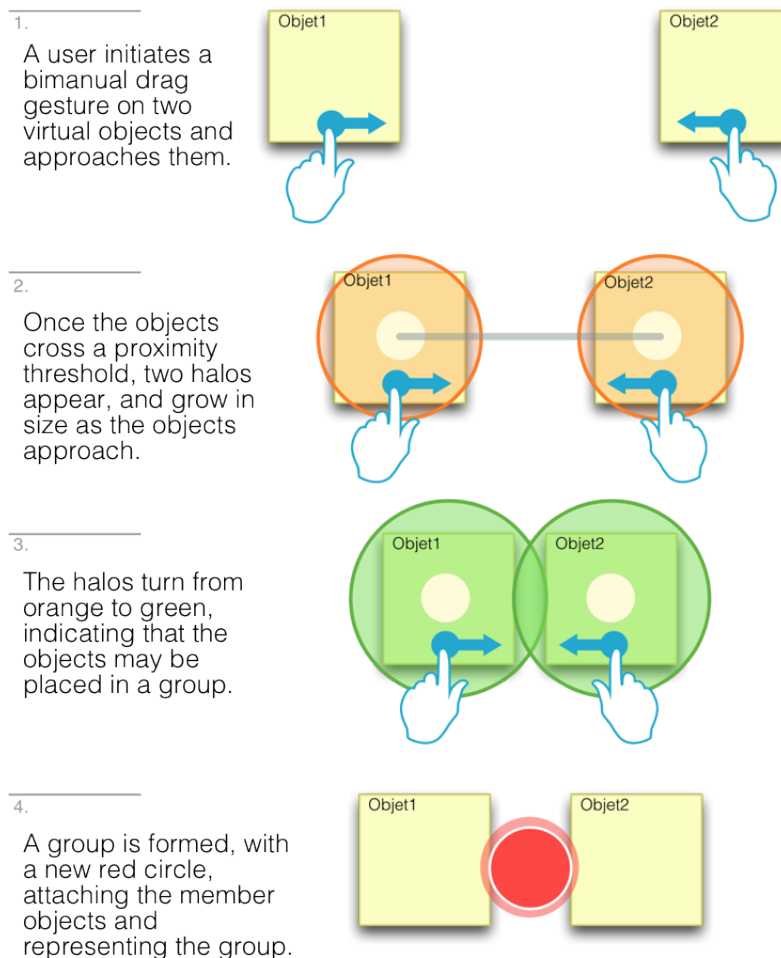


Figure 3.2.3: Creation of a group of objects.

3.2.2 Interaction techniques for transferring information across surfaces

Proper consideration for interaction techniques which allow for the transfer of information from different surfaces is critical when building interactive spaces. I

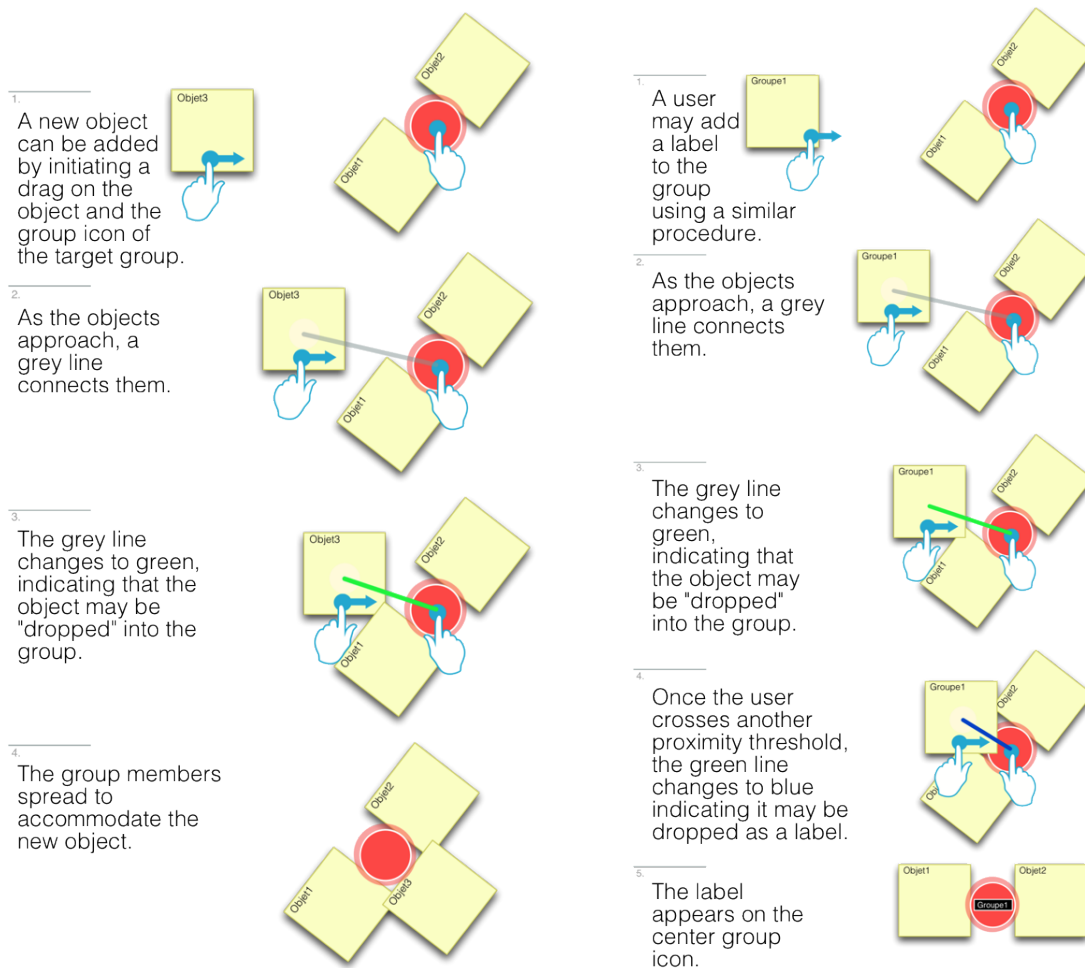


Figure 3.2.4: Adding an object to a group (right) and adding a label to a group (left).

consider two different kinds of multi-surface information transfer, one which is object-oriented and the other which is activity-oriented.

3.2.2.1 Portals

In Wu and Balakrishnan’s work [Wu 03] on an interactive tabletop application for two-person groups performing a room planning and furniture layout task, they briefly mention a discarded idea for an interaction technique based on “wormholes,” which connects one user’s workspace at one end of a tabletop, to another user’s workspace at the other end of the tabletop. They use a UI component which allows objects to be dragged on it and transported to other side of the table [Wu 03]. Nacenta et al. [Nacenta 09] reference Wu and Balakrishnan’s idea and note that this interaction technique can also be used as multi-surface strategy. They build upon the original idea by explaining that users should be able to move wormhole components into any convenient position on the surface, and they named the interaction technique “spatially arranged wormholes,” though, again, no implementation is tested [Nacenta 09]. In this section, I build upon this interaction technique, renamed *portals*, and provide details for its design in the TATIN-PIC interactive space.

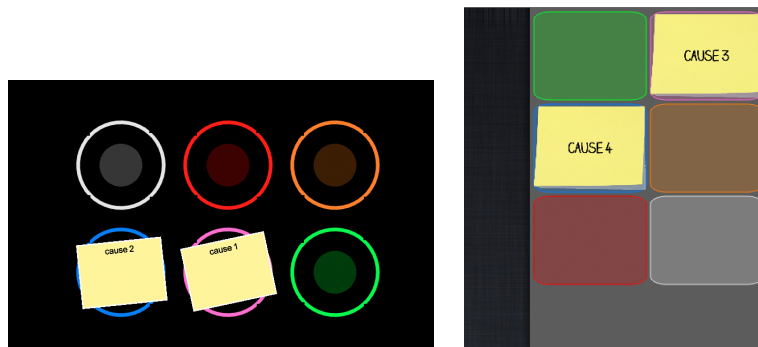


Figure 3.2.5: Portals on the interactive tabletop (left) and the portals on the interactive board (right). The items removed or added from either surface.

Portals are artifacts which do not necessarily “transport” an object, but rather cause it to appear on two surfaces at once, and allow it to be transported through basic drag gestures (Figure 3.2.5). On the surface of the tabletop, a user may drag an object onto a portal (or simply throw it into a portal from a distance.) Once the object has entered a portal, it remains visible and manipulable on the original side of the portal, as well on the portal’s artifact on the board display, allowing the transfer to be completed from another drag gesture by the user located at the board display, at which point it disappears from the tabletop’s portal. The user

at the board display can also drag objects onto the board's portals, allowing them to be accessible from the tabletop as well. The portals on the board display are located in a virtual drawer, which can be slid open and closed from the right side of the display. The portals on the tabletop can also be dragged into any position they want, and are designed with physics-based interaction which allows users to throw the objects from afar and have them be pulled into the center of the portal when they come in contact with it.

Nacenta et al. provide an excellent literature review and taxonomy for multi-display information transfer [Nacenta 09]. Many of the interaction techniques that are reviewed emphasize the needs of a single-user to either control or "get" objects on distal surfaces. In this case, one of the criteria for a strong multi-surface interaction technique is to maintain a physical and spatial mapping to the different surfaces in the room. This is usually achieved by cursor extensions or canvas stitching, which allows objects to be passed from one surface to the other, as if the surfaces were located next to each other on a planar surface.

While the design on the portal technique eschews the spatial mapping that is typically emphasized in multi-display environments, I believe it provides a certain level of flexibility to the users, allowing them to "map" the portals to their coordinated roles. (This social mapping of users to information resources was noted in the observations of the environment.) That is, the techniques favors a role-oriented multi-user mapping rather than a multi-surface spatially-oriented mapping. This metaphor was conceived when observing how meeting participants hand off Post-it notes to the moderator at the whiteboard, and was designed to allow multiple participants a flexible and organized way to transfer Post-it notes to the user located near the board. This lifts certain restrictions on multi-user coordination. For example, a user generating objects on the surface of the tabletop does not need to be seated at the edge of the tabletop closest to the board with his or her back to the display. They may sit facing the board at the other end of the tabletop and communicate and coordinate freely with the moderator.

The technique does have some restrictions, notably a portal may only be used to hold one object at a time, and requires that the object be removed at the other end, thus opening the portal again. This restriction slows the throughput of object transfer between surfaces, but this also has certain advantages. It affords the moderator, and the group, the focus and the time to properly consider how the often diverging output of the entire team might be structurally converged on the board. This constraint also serves as a coordination mechanism.

3.2.2.2 Exporting and importing between applications

Applications, as defined in Section 3.1.3, occupy multiple shared surfaces at once. They are software dedicated to the co-construction of specialized artifacts, such

as affinity diagrams, project flowcharts, and cause-and-effects diagrams, which I refer to as their *application data*. These applications can be used in conjunction with social creativity and collaborative exercises such as brainstorming, project planning, and causal analysis. I believe that there is opportunity to allow new group workflows to be constructed across applications and to emphasize the reuse of previously constructed application data in different contexts, through flexible and context-dependent application data transfer.

This idea is better explained with an example scenario of a group starting a project planning from scratch. The moderator will begin a new brainstorming session by asking each participant (who is representing his or her respective departments of an organization) to generate a list of tasks that they must complete over the course of a project. Here, a member of the sales department might generate tasks related to drafting a business plan and launching a marketing campaign, while a member of the R&D department will generate tasks for constructing prototypes and conducting evaluations. Once these tasks are generated, the moderator asks them to organize the tasks on the tabletop by potential phases of a project, such as project initiation, product design, manufacturing, etc. Once the tasks are organized into cross-disciplinary project phases (using the grouping interaction technique mentioned above, for example), the moderator then opens the project planning application and then imports this “task brainstorming” into a project flowchart. The tasks are grouped by project phase on the board display, are all that remains is establishing their sequential, successor and predecessor relationships for the PERT analysis. Because of the way the applications are designed, the majority of the collaborative brainstorming takes place on the tabletop, as the nature of the technique emphasizes individual idea generation, parallel work, and egalitarian collaboration for grouping. Once the project planning application is opened, the information is automatically structured and presented on the board display, allowing members of the team to engage in moderated collaboration, as the information has converged into a more concrete representation of the project. The components of the diagram can still be entirely restructured and modified, by adding task through the use of portals, for example.

The project lead can save this project planning application data for future use, and perhaps a few months later, when the same or similar team has reconvened for a different project, he may export their project planning back into brainstorming activity, allowing all participants to reuse and review the tasks that they have once generated, and engage with the data in this context, before moving it to a new project planning.

This scenario involves importing a brainstorming into a project planning activity (and vice versa). The project planning application was specifically designed to accept application data from a brainstorming (*context-dependent transfer*), but

the information can also be transferred in a more general way. For example, when the application data from a brainstorming is imported into a causal analysis, this is done so by adding the brainstorming data in a list of Post-it notes in a virtual drawer at the right side of the board display (*general transfer*).

I see the ability to import and export application data as advanced functionality accessible to the moderator of the session from his personal menu. From this menu, the moderator also has functionality to export and automatically distribute the application data through a shared network drive which is instantly accessible to all team members. Depending on the nature of the application data, the moderator may have a choice of export formats. For example, a brainstorming may be exported as a text file or as a mind-map. Users with personal devices can instantly open this file on their personal devices. This is a third kind of activity-based multi-surface information transfer.

3.2.3 User-specific interaction on the tabletop

User-specific interaction is often a desired component of an interactive work environment, and gives designers access to data regarding the owner of actions. This can lead to intuitive multi-user functionality, such as personal clipboards, user-specific undo and redo actions, or even a shared paint canvas application where each user can control their own color and brush strokes without interfering with others. In this case, user identification is required for multimodal interaction, which involves fusing a user's vocal commands with their gestures.

User identification on interactive tabletops can be achieved using a range of hardware devices (see Section 2.2.2.3) to variable but mostly great effect. While considering these options, I designed several lightweight software techniques which could be used during the development of the system. This proved to be more than adequate for our purposes, and therefore a brief review of the approach is included here, while a more detailed examination is provided in work published with my colleagues in [Barthès 12].

The first software technique I considered was dividing the tabletop into separate pre-defined territories, as is done in systems such as [Shen 06]. This idea was discarded to maximize the surface space for shared work, rather than partition it. Next, I considered special gestures from which orientation could be extracted and associated to a user. For example, when selecting items, users might use a special flick gesture (that would be fast enough to be above a certain threshold) or place three fingers on an object in the form of a triangle. From the shape of the three points of contact, a direction could be followed to the edge of the tabletop which could be associated with a user. This would allow a user to select any item within their reach, but problems might occur if the gesture is difficult to execute or the direction is ambiguous. Similarly, I considered extracting direction from

the shape of the fingers pressed against the surface of the tabletop, but believed this to be unreliable given the variety of shapes that might occur from the all of the users' different finger and thumb touches. This approach is also dependent on the specific technical characteristics of multitouch sensing device (e.g. capacitive-based, optical-based, etc.).

Finally, I chose to rely on the user's personal access point (their circular menu), and use a simple distance metric so that every gesture that occurs would be attributed to the user with the closest circular menu. It is similar to the idea of territories, but does not visually demarcate the shared workspace, and is more flexible as users can move their menus with drag gestures. It could still be ambiguous, if an object lies at the midpoint of two menus, but with proper feedback, these ambiguities can be quickly understood and corrected. I considered some modifications of this technique such as introducing a bias toward the user which created the object, or even a bias which fades over time, but felt this might make resolving ambiguities more difficult to understand.

3.2.4 Vocal and multimodal interaction techniques

The core of the shared project-oriented layer of the interface is designed to be used with direct manipulation, one-finger touch and one-finger drag techniques. This approach provides an extremely low-threshold gesture set designed to engage novice users and quickly give them access points to the same functionality used by expert users. The simplicity of these direct manipulation techniques, also allows for gesture transferability between novice and expert users.

Nevertheless, there are limits to the power of direct manipulation techniques, and therefore I considered other modalities of interaction which might also be as approachable and accessible as multitouch. Tse et al. [Tse 08] demonstrate a proof-of-concept of multi-user multimodal (voice and touch) interactions for interactive surface computing. In their prototype brainstorming application, they allow users to interact with design patterns such as *select by touch*, *operate by voice* (e.g. "Group selected items" or "Link this item <tap> to this item <tap>") and *auditory mode switching* [Schnelle-Walka 12] (e.g. "Annotation mode" changes the effects of direct manipulation techniques, allowing participants to draw lines on a Post-it, instead of moving, using a one-finger drag). The system also grants the user the ability to select a Post-it by referring to it by its text. Tse et al. also considers joint multimodal commands, where users can perform multi-user, multimodal interaction. For example, if a user says, "Group selected items" five seconds *after* another user selects two items, the two items are also included in the selection of the user's grouping.

I believe these multimodal techniques are an interesting approach, and they have been added to the design of the TATIN-PIC system. Because of the vocal nature

of these commands, they also may be transferred just as easily from expert users to novice users, and they add power and new functionality to direct manipulation multitouch gestures. Still, because there does not exist a 100% reliable speech system, and given the variety of ways that a user may formulate vocal commands, I have chosen to embody the multimodal interface inside an intelligent virtual agent. This agent has a library of tasks which it can perform when requested by the user, and it can understand a multitude of differently formulated phrases related to these tasks. The virtual agent would remain mostly silent, relying on the visual feedback from the result of the request, but when a request is not clear, it can speak through speech synthesis to ask the user for clarification.

In this section, I identify three different ways in which voice-activated intelligent agents could be introduced into the interactive space. These are illustrated in Figure 3.2.6. In the first configuration, the virtual assistant would be reserved for only a single user: the moderator. The agent could assist the moderator, who wears a headset with an earpiece and a microphone, in performing specialized functionality dedicated to the high-level organization of the meeting and the activities of the participants. The second approach would be to provide every user with headsets and their own personal assistant agent, which could provide functionality for performing group work (with such interactions as proposed by [Tse 08]), as well as access to personal information resources, such as calendars and email. The third configuration uses an *oracle agent*: users interact with one intelligent agent for the entire group, using only one microphone and shared audio with room-wide speakers. Though this configuration is not the best suited for performing user's individual tasks, it still can perform global information requests and present the results to the group on the display. Considering these three styles of interaction is important, because each will impact the group's natural conversation and turn-taking in different ways.

Moving forward with the implementation of this system, I chose to implement the second solution. This solution provides the user with a range of actions so that they may manipulate information resources at a personal level and at an activity level. Also because voice input and output is handled by an earpiece, user-specific voice recognition and voice synthesis is more easily achieved than with the oracle agent. By implementing this option first, it also reduces the effort necessary to later adapt the other two.

3.3 Conclusion

In this chapter, I have presented the organization of physical environment of the TATIN-PIC system, which emphasizes distributed access and entry points on each shared surface to allow teams to engage in group work, while still supporting nat-

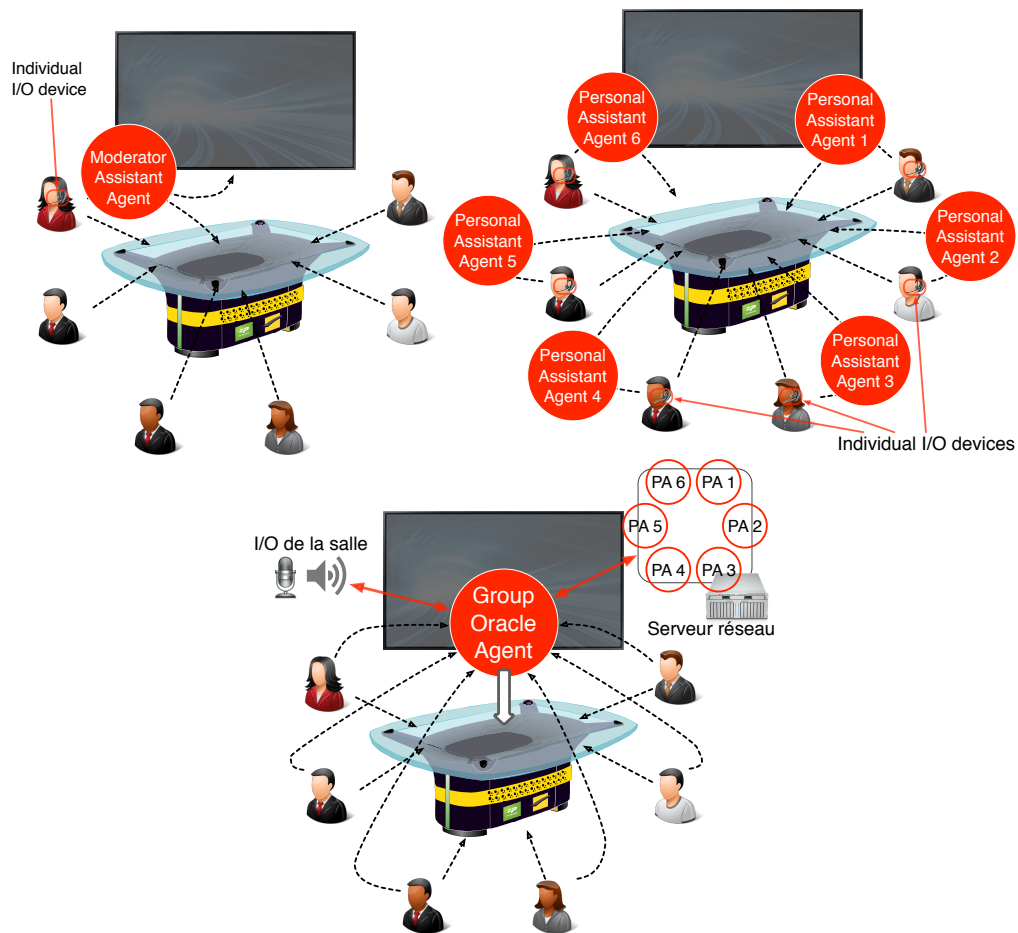


Figure 3.2.6: Three different multimodal interactive spaces with virtual agents: moderator assistant agent, personal assistant agents, and oracle agent.

ural collaboration styles. I have also presented the organization of the information landscape, which is designed to support both activity-level information resources and individual-level information resources, through two separate layers.

I also presented several general interaction techniques for the interactive space. These include a multitouch grouping technique, which gives users better control over hierarchies and labels, two different methods of multi-surface information transfer, one object-oriented and another activity-oriented, a lightweight software technique for user identification (provided the users have persistent, individual access points), and finally, multimodal interaction techniques within the context of teamwork in interactive spaces.

Each of these concepts has been conceived with respect to the design goals presented at the end of Chapter 2. The core functionality of the system is accessible through direct manipulation techniques, with “small” gestures such as the one-finger tap and the one-finger drag. Bigger gestures, for higher-level interaction, are added to the board display to allow a user to pan and zoom. On the tabletop, the activity layer of the information landscape allows for applications to run inside a physics-based environment, with effects allowing for lightweight orientation and translation techniques. These physical effects are also present in the grouping and portal techniques, which are again achieved through small, direct manipulation interaction. The grouping technique also engages the user in synchronous, bimanual manipulation and has specific functionality for managing clutter via reducing or expanding the size of groups. Software-based user identification techniques are used to allow for multimodal interaction on the surface of the tabletop. Finally, because all techniques are based on direction manipulation, and I do not present any space-reaching or cursor extension techniques. This choice was made by design, because direct manipulation techniques allow for inherent, nonverbal communication in the gestures, which provides a form of feedthrough to help maintain awareness of other user’s actions.

On the personal layer of the information landscape, participants can use their personal access points to view and share personal documents. The moderator’s access point, also includes functionality for exporting and distributing group work, as well as transferring information between activities. These are important actions for supporting the role of the moderator and they have to potential to impact the role of the moderator.

I examined several different kinds of interactive spaces, and explained how they differed in terms of the group and individual entry and access points. From this I derived the configuration of an interactive space, using a tabletop and a whiteboard which are both multitouch enabled. The tabletop, through the personal and activity layers of the information landscape, provides a platform which is as much for the group’s artifacts as it is an individual’s.

These concepts will be fully realized in the implementation of our system in Chapter 5. Chapter 6 will also address the evaluation of certain aspects of the system in the context of preliminary design activities.

Chapter 4

Multi-agent infrastructure for interactive spaces

The previous chapter presented the TATIN-PIC distributed and multi-user interactive space, along with several multi-device interaction techniques. This chapter studies the implications of such an interactive space with regard to its distributed computing infrastructure. This is neither a new research question nor one with a conclusive answer: providing coordination mechanisms for heterogeneous devices working in concerted action requires an infrastructure which holistically integrates networking and software I/O principles. There exist a variety of approaches, but due to the modular and distributed nature of agent-oriented programming, multi-agent systems are particularly well suited for this task. The process of designing a multi-agent system for the purpose of an interactive space can be difficult, therefore an implementation plan can be helpful guiding developers on interactive spaces.

The first section on this chapter will provide more information on agents, agent-oriented programming and the Gaia methodology for the design of multi-agent systems. The second section will revisit the TATIN-PIC interactive space from a distributing computing perspective and discuss the problems presented by its distributed environment. The second section also presents several design objectives for the infrastructure, and discusses how multi-agent systems can meet these objectives. The final section will apply the Gaia methodology for the design of the TATIN-PIC infrastructure, resulting in an implementation plan consisting of the agents, their roles and their protocols. This application will be parameterized according to the number and type physical devices used in the environment, and can be applied to different configurations of interactive spaces beyond TATIN-PIC.

4.1 Software agents and societies

In this section, I provide a brief introduction to software agents, multi-agents systems, and Wooldridge et al.'s [Wooldridge 00] Gaia methodology for the design of multi-agent systems.

4.1.1 Agent-oriented programming and multi-agent systems

At their conception, software agents were created from a need to parallelize processing and distribute control inside a computing system. Nwana [Nwana 96] attributes their first appearance to Hewitt and Baker's 1977 proposition of concurrent actors [Hewitt 77]. The actor model is derived by placing constraints on objects in object-oriented programming (OOP), restricting them to only having the ability to send and reply to messages from other concurrently executing actors, with only a limited notion of internal state. Around the same time, other researchers also began investigating how agents, with an actor's message-passing capabilities coupled with a formal, symbolic internal state, could be used for distributed problem-solving and artificial intelligence applications. These focused on interaction and communication issues of societies of agents and addressed issues such as task decomposition, task distribution, cooperation, uncertainty and conflict resolution. Examples of such formative contributions are Smith's *Contract Net Protocol* [Smith 80] and Erman et al.'s *Hearsay* Speech-understanding system [Erman 80].

Moldt et al. [Moldt 97] attribute much of the theoretical formalization of the concept of agents as a general software engineering paradigm to Yoav Shoham, who in 1993 proposed the concept of agent-oriented programming (AOP) [Shoham 93]. Shoham illustrates how AOP is derived from OOP in Table 4.1.

Table 4.1: Shoham's comparison of object-oriented programming to agent-oriented programming [Shoham 93].

	OOP	AOP
Basic unit	object	agent
Parameters defining state of basic unit	unconstrained	beliefs, commitments, capabilities, choice, ...
Process of computation	message passing and response methods	message passing and response methods
Types of message	unconstrained	inform, request, offer, promise, decline, ...
Constraints on methods	none	honesty, consistency

In order to describe an agent's internal state, Shoham introduces a modal logic language which uses discrete time points to index fact statements. These fact states can refer to environmental conditions or even actions, which are, according to Shoham, facts that become true once the action is performed. The language also uses several modal operators for defining *beliefs* (opinions of the truth value of facts), *commitments* (which can express an agent's obligation or intention to perform an action, perhaps with other agents), *capabilities* (the tasks that an agent is able to perform), and *choice* (decisions which the agent can make). Like OOP, the process of computation in AOP takes place through message passing and response methods. In AOP, however, Shoham does not consider messages as simple information passing mechanisms but rather emphasizes them as an agent's speech acts which are taken from social science literature on verbal communication (notably [Searle 69]). Their type is defined by their purpose; they can either be to inform, request, offer, promise or decline information or action. The modal logic language is used to also define *constraints* that are placed on the methods, allowing them to preserve some basic properties of an agent's mental state, such as maintaining local consistency between beliefs and commitments and assuring that they will do whatever is within their capabilities to uphold their commitments.

Shoham's agent-oriented programming provides a clear view of exactly what an agent is, but it does not determine what roles these agents should play and how they should coherently interact with each other or the environment. Nwana [Nwana 96] surveys the kinds of software agents that have been used in research applications and identifies seven different kinds of roles that they play: collaborative agents, interface agents, mobile agents, information agents, reactive agents, hybrid agents, and smart agents. The identification of these seven categories depends on the extent to which they cooperate, learn, and perform autonomously [Nwana 96]. The potential applications for societies of these kinds of software agents inside *multi-agent systems* are far-reaching. From their conception, they have been designed to be an integral part of distributed artificial intelligence research, but at the same time they also have been proven to be adept for simulation, distributed algorithms, and robotics. For example, the Lhommet et al. [Lhommet 11] use smart agents to simulate crowd activity in moments of crisis and catastrophes. Another example of a different kind of application is Kubicki et al.'s [Kubicki 12, Kubicki 13] interactive tabletop which uses a society of interface agents to manage and reason about virtual and physical items on the surface of the tabletop.

4.1.2 The Gaia methodology for the design of multi-agent systems

Multi-agent systems have also become relevant as an architecture pattern which can apply to more traditional software engineering problems. Wooldridge et al. [Wooldridge 00] explain that agent-oriented programming represents a powerful and high-level abstraction which can be used to model and develop complex and distributed systems. Due to the versatility and power that comes with software agents, the process with which developers may layout their agent-oriented software architecture is not necessarily clear. Wooldridge et al. therefore present the Gaia methodology for the purpose of building multi-agent systems through a process of organizational design [Wooldridge 00].

The Gaia methodology is founded upon a certain set of domain characteristics: agents are coarse-grained computational system, each executing in their own concurrent process. The agents must be cooperating, not competing, with each other, and their organizational structure, relationships, and capabilities are static and do not change at run time. Finally, it is designed to be used with a comparatively small number of agent types, e.g. no more than 100. The Gaia methodology begins after a detailed requirements summary of a system and goes through two phases: an abstract, analysis phase, in which roles and interactions are established, and a design phase in which agents, services, and acquaintance models are derived. These last three models results in a high-level implementation plan, but do not prescribe any tools or methods for the actual implementation, leaving decisions regarding the actual framework of multi-agent systems up to the needs of the developers.

The Gaia methodology was revised by Zambonelli et al. in 2003 [Zambonelli 03], providing several additions which have been helpful during my application of this methodology primarily because of their organizational abstractions. Among the most relevant changes are the addition of an environmental model, organizational rules, and an organizational structure.

- The environmental model lists information resources that an agent may have access to and how these information resources are physically distributed throughout the system. This is especially relevant in the context of interactive spaces.
- The organizational rules describe the high-level responsibilities of the organization, and they are important for any multi-agent system which may support openness and self-interested behavior.
- The organizational structure describes the relationships between the roles

that are played in terms of their topology, effectively replacing the acquaintance model with a more comprehensive approach.

In reality, I applied the first version of Gaia in its entirety, and during this process, I found myself developing much additional information such as the environmental model, organizational rules, and the organizational structure out of necessity. Upon discovering the revised version of Gaia, these organizational abstractions were very much welcomed and fully integrated into my application of this methodology. Indeed, it is difficult to design a physically distributed multi-agent society without conceptualizing these subjects in some form. Jamont and Occello's DI-AMOND method, an alternative to Gaia for the design of multi-agent systems, similarly places a great deal of importance on the environmental model, before defining interactions and roles [Jamont 07].

In this chapter, I will use the Gaia methodology to introduce and describe the experimental infrastructure that was developed during the software engineering process. I do so with parameterized roles and interactions so that my method may be easily applicable to other kinds of interactive spaces with different configurations. Also, because I follow the recommendations of the Zambonelli et al. [Zambonelli 03], the multi-agent system can be considered to be open for new additional agents introduced at run-time, provided they know how to communicate with the existing agents in the system. This will depend on the common agent communication language that is chosen, but this is not an issue addressed by the Gaia methodology. Therefore, I reserve the discussion of the agent communication language for the following chapter on the implementation of the interactive space.

Before I engage in the Gaia-certified agent-making, I must directly address the precondition to the methodology: the system requirements of interactive spaces.

4.2 Requirements summary of interactive spaces

This subsection has several objectives. First, I will introduce some basic concepts related to the distributed environmental structure of the TATIN-PIC interactive space. Then, I will also present explicit development-time and run-time objectives for the infrastructure, and discuss how a multi-agent infrastructure can work towards meeting these goals.

4.2.1 Single-device architecture for groupware

Before considering the nature of distributed computing system for groupware, it may be helpful to introduce some concepts in the context of a single-device system for groupware. If the “interactive space” were to consist only of an interactive

tabletop (i.e. one primary shared surface), modeling the TATIN-PIC architecture would be straightforward. For example, the Model-View-Controller pattern, a time-tested standard of software I/O architecture, might be helpful in implementing such a user interface. It provides a clean dichotomy between data and the interface to this data, which is useful for the maintenance and reuse of code. Given the project and application requirements presented in Chapter 3, a single device architecture for TATIN-PIC would resemble the architecture presented in Figure 4.2.1. Here, a project contains a list of application documents, which each have their own specific data models and associated views, or *scenes*. A project is also worked upon through a number of sessions, which consist of a certain subset of participants from a organization's roster.

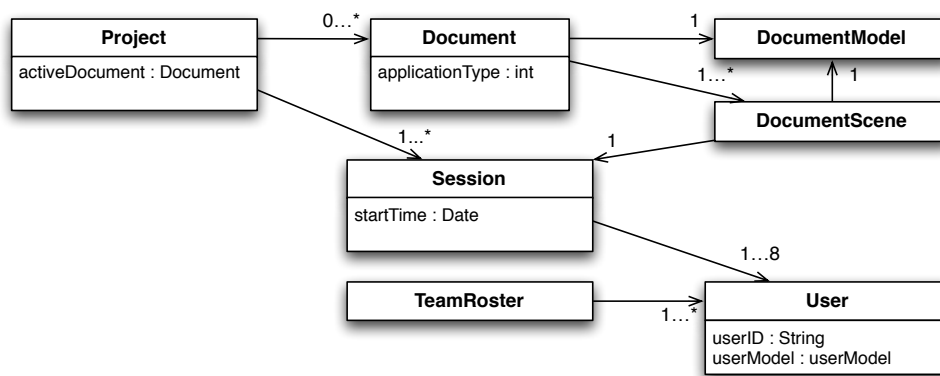


Figure 4.2.1: The TATIN-PIC architecture, modeled within the context of a single device.

This may or may not be implemented through a multi-agent system. Traditional object-oriented patterns are powerful enough to handle this architecture with ease. The concurrency provided by agent-oriented programming could be helpful in providing a layer of abstraction for the concurrency of user's actions, but it is by no means necessary.

4.2.2 Distributed architecture for groupware

Now in the perspective of a distributed system, this model must be replicated and distributed to a variety of different devices (Figure 4.2.2). First, I make a few assumptions in order to make the presentation of the infrastructure clearer: There exist some *master project database* on the primary shared device. This database contains *master projects*, which in turn contain documents whose models are relevant to all shared devices in the interactive space. For example, this means

that a *master document model* of a document of type “causal analysis” contains all data necessary for the tabletop device to generate its scene (e.g. a collection of virtual Post-it notes which are displayed on the tabletop), and for the whiteboard to generate its scene (i.e. the actual cause and effect diagram). This assumption, i.e. one device contains a master version of all projects, is made in order to facilitate the use of a database. Therefore, it will be the responsibility of the infrastructure to distribute this master document model to other shared devices, and provide some level of synchronization for this document model. I describe the *master project database* as being on the primary shared device, not only for the sake of clarity in my presentation, but also because (for the purposes of the TATIN-PIC project) we wished to build a stand-alone mode for the interactive tabletop, which could run some considerably reduced version of the software without the rest of the interactive space (for logistical and transportation reasons). Without this constraint, it is easy to imagine that the project database could also be stored on a network server, and the multi-agent infrastructure that I will present will be able to support this kind of adjustment in an obvious way.

I now provide some examples of how the infrastructure should behave for the sharing and distributing of changes to project information at run-time. When the moderator creates a new document on the primary surface, the ancillary shared surface (i.e. the interactive board) must be notified that an additional project document has been created. Then, when the new document is opened, the ancillary shared surface must then be able to receive a copy of the document model so that it may generate its corresponding scene view. If the moderator chooses to switch back to a document that has already been opened, the shared display needs to be informed of this activity switch but it does not require another transfer of the document model; it will already have a local copy of the document model ready to display.

The primary shared device also contains a *userModel* for each user who has logged in on the table. The data that is stored in the *userModel* might be, for example, a list of items that the user has created, or the items that the user has selected in the interactive space. A personal device, such as a smartphone or a tablet, will also have some notion of a *userModel*, with similar information, and the same goes for the personal assistant module. In the interactive space, accessing the personal assistant is done through a user-specific I/O device: a bluetooth headset. Because there are multiple headsets active at any given time, and because they require resource-intensive voice recognition and voice synthesizing software, each headset is connected to their own mini-PC which are housed inside the chassis of the interactive tabletop. It is on this device that a personal assistant will run, and maintain their own version of a *userModel*. Not only should the infrastructure support some level of synchronization between all *userModels*, it should also allow

a user's actions from one device to directly modify information on another. For example, the personal assistant module should be able to stay informed of the selection events on the primary shared device, and then when the user instructs a personal assistant to create a group from the selected items, the user assistant should be to formulate that command and add a group to the document model that is displayed on the tabletop.

Figure 4.2.2 illustrates the aspects of the interface and the application model that are to be distributed throughout the system.

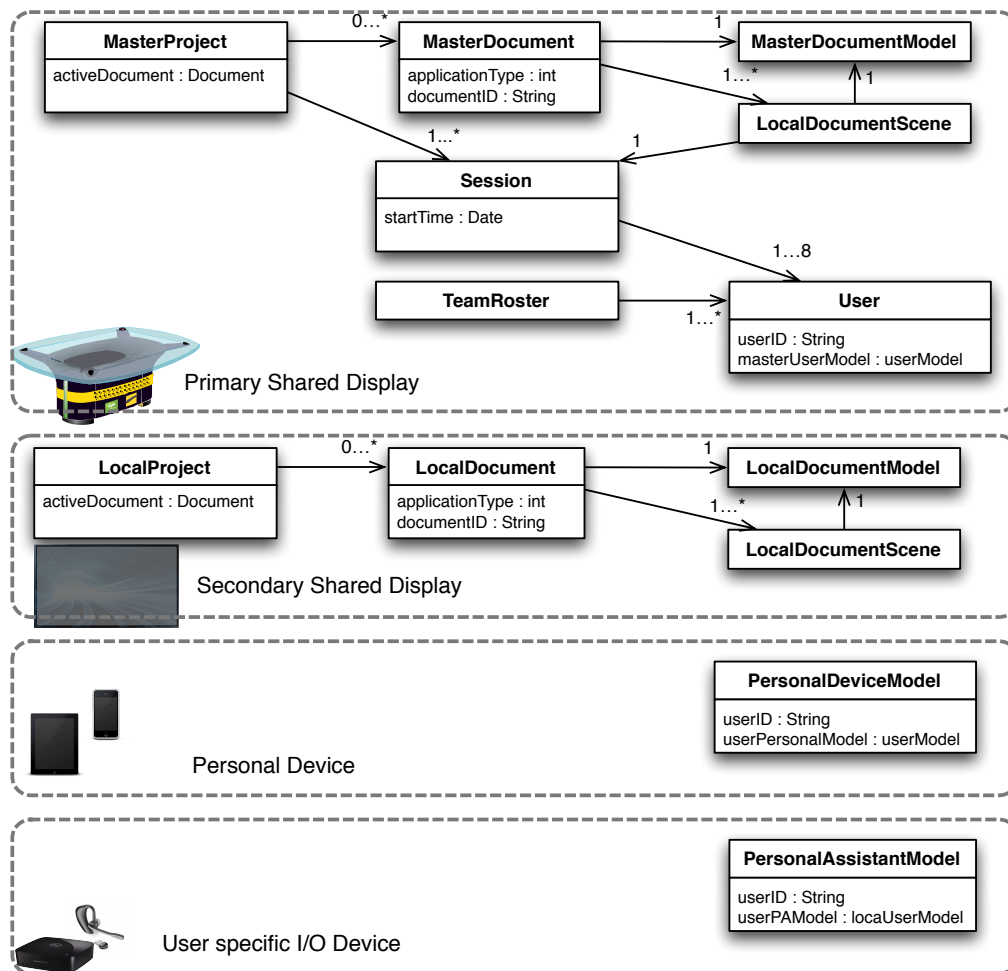


Figure 4.2.2: A fragment of the TATIN-PIC architecture, emphasizing the distributed components of the system. (This will also be used as part of the environmental model for the Gaia methodology.)

4.2.3 Design objectives for interactive space infrastructure

This distributed architecture is what motivates my investigation into distributed computing systems. There exists a variety of networking and distributed system toolkits and software libraries which could be used to accomplish such distributed application models. This is why it is important to introduce several infrastructure design goals and discuss how multi-agent systems play their part in addressing these need. Some of the infrastructure design goals are inspired by or corroborate those which are presented in related work in scientific literature, such as Banavar's distributed application model requirements for pervasive computing [Banavar 00], *iRos* middleware in the Stanford iRoom [Johanson 02b], the *Shared Substance* middleware of the WILD room [Gjerlufsen 11].¹

General objectives

- G1: Modular and distributed project model: The infrastructure must support a distributed model for data structures and user interfaces which are located on different devices and must perform together.
- G2: Distributed and parallel multi-user interaction techniques: The infrastructure should facilitate the development of interaction techniques that are adapted to interactive spaces and cross-device user-specific interaction, such as the multimodal fusion of input or multi-display interaction techniques.

Development-time objectives

- D1: Platform and toolkit independence for user interface: Due to the heterogeneous set of personal devices and I/O device that are used in interactive spaces, the infrastructure should not impose constraints on developers regarding the choice of platforms, programming languages and toolkits that are used to build the user interface.
- D2: Platform and toolkit independence for distributed infrastructure: Because of the various platforms and user interface toolkits that developers may use, the implementation of the infrastructure, which will integrate and coordinate data from a heterogeneous set of input devices, should not prescribe the use of certain toolkits or programming languages. The infrastructure should provide a standard and accessible way for transferring data which can be implemented in variety of different languages. This provides more freedom for developers to integrate user interface elements directly into the infrastructure at their discretion.

¹As mentioned in Chapter 2, these examples of interactive spaces in related work did not focus heavily on user experimentation and therefore had limited applicability for influencing the design of our interactive space. They are however highly influential when considering the infrastructure requirements of our system.

Run-time objectives

- R1: Rapid prototyping: Once the system is running, developers might want to continue modifying programs on certain devices, necessitating their disconnection, re-compilation and reconnection at run-time to facilitate debugging. The infrastructure should respect the modular nature of the distributed project model, and therefore should allow developers to reboot devices with a limited impact to the overall infrastructure of the system. Developers should also be able to design scripts, which can be run locally and trigger actions on remote devices, in order to assist in debugging. This is as much a run-time requirement as it is a development-time requirement, but it is included here because it is related to R2.
- R2: Flexible network connections: It is important that the infrastructure be able to accommodate unexpected situations and have a tolerance for network errors. The distributed infrastructure should be able to handle new connections, unexpected disconnections, and eventual reconnections. This gives flexibility to the users, whose devices might turn off or disconnect from the network if they leave the interactive space.
- R3: Persistence and recovery: The infrastructure must render the distributed project and document models persistent so that they be stored and accessed in a database. Persistence is also important to allow for recovery in the case of a system-wide failure.

4.2.4 Using multi-agent systems and the Gaia methodology to meet the infrastructure requirements

I believe that a multi-agent system would be particularly effective in meeting these requirements. Agents by nature are autonomous, coarse-grained computation systems which are designed to be modular, distributed, and concurrently executing. Therefore, an infrastructure implemented using agent-oriented programming has the potential to support the distributed data structures (G1) and parallel interaction (G2). This will be addressed in detail in the following section, throughout the application of the Gaia methodology.

The Gaia methodology is also appropriate because it does not prescribe a low-level implementation plan, leaving how the agents are actually implemented to the discretion of the developers. To facilitate the implementation of agent societies, designers may choose from several sophisticated and robust multi-agent libraries, such as JADE (Java Agent Development Environment²), SPADE (Smart Python

²<http://jade.tilab.com>

multi-Agent Development Environment³), and OMAS (Open Multi-Agent System, written in Lisp⁴). Each of these might provide different advantages. For example, JADE provides robust agent container mechanisms for managing agents and their relationships running in different environments across the network, while the OMAS provides several intelligent conversational agent models for rapid artificial intelligence prototyping. More importantly, this is not a choice of one or the other. These three multi-agent systems are FIPA compliant, allowing them to be used in conjunction with each other. The Agent Communication Language specification is a standard established by the Foundation for Intelligent Physical Agents (FIPA) which details how agents can transfer information. This allows agents from different multi-agent systems to communicate with each other. Therefore, developers are not limited in their choices of infrastructure toolkits (D2).

Because the agents exist in multiple programming languages, these can be directly integrated into the software of many different kinds of platforms. For example, JADE agents can be directly integrated in an application built using the Android SDK⁵, while a SPADE agent could be integrated into an application built using Kivy⁶, a multitouch Python toolkit. Therefore, developers are not limited in their choices of user interface toolkits.

Using FIPA compliant and open multi-agent systems also have additional advantages. Users may write scripts which send messages simulating agent activity on remote devices, giving flexible ways to assist in the debugging of the system (R1). Also, given their granular design, one agent may crash, be rebooted, and request a new copy of the model without much impact on the entirety of the system. Moreover, messages can be directed to an agent running on an individual device, or may be broadcast to all agents indiscriminately, allowing them to react to the messages only if they know how to appropriately respond. If a message does not receive a response, after a certain time interval, an agent may take the necessary steps to resolve or ignore this issue. Of course, these are strategies which depend on the actual implementation, but they demonstrate great potential for flexible patterns of network communication (R2). The remaining run-time goal is one that is resolved by designing dedicated agents whose role it is to ensure the persistence of artifacts (R3). R3 will be directly addressed in the application of the Gaia methodology in the next section.

³<https://pypi.python.org/pypi/SPADE>

⁴<http://www.utc.fr/~barthes/OMAS/>

⁵<http://developer.android.com/sdk/index.html>

⁶<http://kivy.org/>

4.3 TATIN-PIC multi-agent infrastructure: Application of Gaia methodology

The multi-agent system will be presented using the Gaia methodology. The first phase, the analysis phase, requires that the *roles* in the society and the *interactions* between roles be identified and iterated upon. In the first subsection, the roles are presented alongside the *environment* model, which is especially important for conveying the physically distributed nature of the society of agents. In the second subsection, the interaction model is presented alongside the *organization rules*, which describes the qualities or rules that should be upheld in the society of agents. These are four components of Gaia are used to define the *organizational structure* of the society, which is presented in the third subsection. The second phase, dedicated to design, requires the construction of the *agents* model and the *services* model, these are discussed in the fourth and fifth subsection, respectively.

Within the following subsections, I will introduce Gaia's concepts and notation systems as they become relevant.

4.3.1 Roles and Environment

I will first begin by describing roles played in the organization. These roles are PROJECTMANAGEMENT, PROJECTSUPPORT, APPLICATIONMANAGEMENT, APPLICATIONSUPPORT, PERSISTENCEMANAGEMENT, PERSISTENCESUPPORT, USERIDENTIFICATIONONSHARED SURFACE, USERPRIMARYSHAREDDEVICEDEVICESUPPORT, USERPERSONALDEVICESTUPPORT and PERSONALASSISTANT.

Before discussing these roles further, it is necessary to present Gaia's role schema, as well as discuss the environmental model of TATIN-PIC.

4.3.1.1 Gaia's role schema

These roles are defined through Gaia's *role schema*, which consists of four separate parts. The first part is a plain text *description* of what the role entails. The second part addresses *protocols and activities*. Protocols define how agents perform actions with other agents. They are listed in the role schema in a sans-serif font. Activities are computations that the agent carries out in "private," without interacting with other agents. They are listed in a sans-serif and underlined font. The third part of the role schema is its *permissions*. Permissions address the role's access rights to information resources that are required for the role's responsibilities. A role may either read, change, or generate information resources.

The final and most important part of the role schema is dedicated to *responsibilities*, which explain how a role functions. These are divided into two categories

liveness properties and safety properties. Liveness properties are the results that an agent must bring about, in response to certain environmental conditions, essentially describing the normal *modus operandi* of an agent. Safety properties are invariants, or conditions that the role must respect or preserve. For example, the liveness properties of a chef might be to receive customer orders, check ingredients, execute the recipe for a certain dish, and transfer the dish to a server. The safety properties associated with the role might be ensure that the stock of a certain ingredients do not decrease below a certain threshold. Under the condition that they do, the chef can perform other activities or protocols to replenish them. Safety requirements in the role schema are described by a list predicates. In our chef example, this would be “*pastaDoughStock* > 2 units”. Liveness properties are described with liveness expressions, which use the protocols and activities as atomic elements, e.g. “CHEF= (AwaitOrder. CheckIngredients. [ReplenishIngredients]. ExecuteRecipe. TransferPlateToServer)^ω”. The operators for liveness expressions are taken from the fusion notation for operation schemata and are provided in Table 4.2.

Table 4.2: Operators for liveness expression, from [Wooldridge 00].

Operator	Interpretation
$x.y$	x followed by y
$x y$	x or y occurs
x^*	x occurs 0 or more times
x^+	x occurs 1 or more times
x^ω	x occurs infinitely often
$[x]$	x is optional
$x y$	x and y interleaved

4.3.1.2 TATIN-PIC environmental model

The information resources that are used for the permissions for Gaia’s agent model are derived from the illustration of the environment in Figure 4.2.2. There is no official method for presenting the environmental model in Gaia, though they recommend listing the environment variables and their permissions in a table. This table for the TATIN-PIC environment model is provided in Table 4.3, and is meant to complement the illustration of the environment in Figure 4.2.2.

This environmental model makes use of certain global parameters. These parameters will allow the environmental model, and the entire multi-agent infrastructure, to be extended and applied to an interactive space with different configurations than the one presented in the previous chapter. These global parameters, derived

from the specifications of the interactive space in Chapter 3, are:

- *total_participants* : the number of participants which are participating in a session in the interactive space
- *total_applications* : the number of different space-wide, project-oriented applications that the platform has available for execution
- *total_ancillary_shared_device* : number of supporting ancillary shared devices
- *total_personal_device* : number of supporting personal devices
- *total_personal_assistant* : number of personal assistant devices

Note that many of these parameters can and will be modified at run-time. For example, a user may wish to activate a personal device or personal assistant after they have been working in the interactive space for some time.

Table 4.3 provides an overview of the variables used in the environment, but they are also discussed in more detail below:

- For the purposes of the Gaia methodology, there exists a primary shared surface which contains a master version of the project, i.e. a *masterProjectDocumentList*. On this same device there is also a link to the *masterActiveDocument*. In the yet to be presented role schemas, these are marked as “supplied”, because they are variables created by the environment. These will be accessed and managed by the PROJECTMANAGEMENT role.
- There also exists a *localProjectDocumentList* on shared devices, along with a local *activeDocument* link for these devices. These are information resources which must be managed and synchronized by the multi-agent system, via the PROJECTSUPPORT role.
- The *masterProjectDocumentList* and the *localProjectDocumentList* must also be made persistent so that group work can be saved and reopened by teams. This is done by creating a *projectDocumentDatabaseEntry*. The system must synchronize the distributed documents and create the database entry from the distributed project model. This is the responsibility of the PERSISTENCEMANAGER and PERSISTENCESUPPORT roles.
- The environment will also generate an *interactionEvent* any time a user completes an action that will have some signification for other devices in the environment. These *interactionEvents* describe the nature of the action that was completed in the environment. The multi-agent system typically does

not exchange *interactionEvents* in its messages, but rather *userEvents* and *applicationEvents*.

- *userEvents* are similar to *interactionEvents* except they also contain the identity of the owner of the action. If the *interactionEvent* occurs on a personal device or a user-specific I/O device, adding a *userID* to an *interactionEvent* to create a *userEvent* is trivial. However, if an *interactionEvent* occurs on a shared surface which is user-agnostic, identifying the *userID* of an event is more difficult. This is handled by the role USERIDENTIFICATIONON-SHAREDSURFACE.
- Sharing and acting upon *userEvents* and maintaining local copies of *userModels* are handled by agents of several roles. The role USERPERSONALDEVICESUPPORT handles the *userModel* located on a user's personal device. USERPERSONALASSISTANT handles a local *userModel*, as well as vocal recognition and synthesis. USERPRIMARYSHAREDDEVICESTUPPORT handles the *userModels* located on the primary shared surface. Note that this role also has access to the local project model. This role serves as a bridge between user's actions and the mutually shared project-related information resources.
- If an *interactionEvent* is also relevant to the application, then it may be packaged as an *applicationEvent*. This allows an application, which run on multiple shared surfaces in the environment, the ability to share information across multiple devices. For example, an *applicationEvent* is used in the implementation of the portal interaction technique, to allow an item to be transported across multiple surfaces. These *applicationEvents* will serve to add, remove or modify components of the distributed local *activeDocuments*, are managed by and APPLICATIONMANAGEMENT and APPLICATIONSUPPORT roles.

4.3.1.3 TATIN-PIC role schemas

For space reasons, the complete collection of Gaia role schemas for the TATIN-PIC interface are provided in the Appendix C. However, in this section I present two role schemas for the PROJECTMANAGEMENT (Figure 4.3.1) and PROJECTSUPPORT roles (Figure 4.3.2).

Table 4.3: TATIN-PIC environmental model.

	Variables	Comments
reads	<i>masterProjectDocumentList</i>	// List of all documents of a project (for simplicity, this is located on the primary shared device)
	<i>userID[p]</i> , $p = 0, \text{total_participants}$	// UserID for each participant
changes	<i>masterActiveDocument</i>	// Link to the active document which is currently open on primary device
	<i>localProjectDocumentList[i]</i> , $i = 0, \text{total_ancillary_shared_device}$	// Local list of all project documents which exist across ancillary devices
	<i>activeDocument[i]</i> , $i = 0, \text{total_ancillary_shared_device}$	// Link to the active document which is currently open on all shared devices
	<i>userModel[p]</i> , $p = 0, \text{total_participants}$	// A user model (e.g. recent actions, selected actions) for each user which logs into a shared devices.
	<i>userPersonalDeviceModel[q]</i> , $q = 0, \text{total_personal_device}$	// The user-specific model which is used on each personal devices.
	<i>userPAModel[r]</i> , $r = 0, \text{total_personal_assistant}$	// The user-specific model which is used by the personal assistant.
	<i>projectDatabase</i>	// Database which contains every project
generates	<i>interactionEvent</i>	// A device will generate a generic interface event for each action performed. The multi-agent system will interpret these as either <i>userEvents</i> or <i>interdeviceData</i> .
	<i>userEvent</i>	// Created whenever a user completes an action on a surface that should be shared with other devices serving the same user.
	<i>applicationEvent</i>	// Created whenever a user completes an action on a surface that should be shared with other devices running the same application.

Role Schema: PROJECTMANAGEMENT	
Description:	
This role involves supporting a project which is created or opened on a primary shared surface and enabling the other project support roles in the environment by granting access them to project data, and informing them of application switches, and new documents that are created.	
Protocols and Activities:	
AwaitActivitySwitch, InformOthersOfActivitySwitch, AwaitTransferRequest, TransferDocument, TransferListOfDocumentNames, <u>AwaitDocumentCreation</u> , InformOthersOfDocumentCreation	
Permissions:	
reads	supplied <i>masterProjectDocumentList</i> // List of all documents of a project
changes	supplied <i>masterActiveDocument</i> // Link to the local active document (scene and model)
Responsibilities	
Liveness:	
PROJECTMANAGEMENT = (ManageCurrentActivity ManageResourceAccess ManageDocumentCreation)	
MANAGECURRENTACTIVITY = (AwaitActivitySwitch. InformOthersOfActivitySwitch) ^ω	
MANAGERESOURCEACCESS = (AwaitTransferRequest. (TransferDocument TransferListOfDocumentNames)) ^ω	
MANAGEDOCUMENTCREATION = (<u>AwaitDocumentCreation</u> . InformOthersOfDocumentCreation) ^ω	
Safety:	
true	

Figure 4.3.1: PROJECTMANAGEMENT role schema.

As described when discussing the environmental model, the PROJECTMANAGEMENT has access to the *masterProjectDocumentList* and the *masterActiveDocument*. The PROJECTMANAGEMENT role has three main responsibilities that are carried out simultaneously and repeatedly. The first involves waiting for activity switches from the environment (e.g. when the moderator opens a new application document using his circular menu). Upon receiving the activity switch request, the PROJECTMANAGEMENT role will load the new application document and will inform all agents playing the PROJECTSUPPORTER and APPLICATIONMANAGER roles that a new application document has been selected. The second responsibility of the PROJECTMANAGEMENT is to await any information transfer requests from other agents. This could be an agent which is requesting either the list of documents of a project, or the contents of a specific document. The third responsibility of the PROJECTMANAGEMENT is to wait for new document requests originating from the environment (e.g. when the moderator creates a new project planning document for the team). Upon receiving such a request, the PROJECTMANAGEMENT will add a new document to the *masterProjectDocumentList*, and then inform all PROJECTSUPPORT roles of the new addition to the list.

The ProjectSupport role runs on an ancillary share device and therefore has ac-

cess to the local list of project documents and a link to the local active document. First, upon initialization, it will send a request for a copy of the list of all application documents that exist for the current project. Then, after receiving this list of documents, it will spend the remainder of the time awaiting and following any activity switches and document creations that happen in the environment. When informed of an activity switch, it will check to see if it has a local copy of the document, and if it does not have a copy of this document it will additionally request a copy (c.f. its safety responsibility). With this local copy of the document, it performs a local application switch while informing the local APPLICATIONSUPPORT role. When informed of a document creation, it will add the document to the *localProjectList*.

A brief description for the remainder of the roles is provided in Table 4.4, along with the location of their Gaia role schemas in Appendix C.

Role Schema: PROJECTSUPPORT[<i>i</i>], <i>i</i> = <i>total_ancillary_shared_devices</i>	
Description:	
This roles involves supporting an ancillary shared device in the environment by managing local access to project data, and keeping the device in sync with the activity of the primary shared device.	
Protocols and Activities:	
RequestListOfDocumentsNames, AwaitListOfDocumentNames, RequestDocument, AwaitActivitySwitch, SwitchActivity, AwaitNewDocumentInfo, AddDocumentToList	
Permissions:	
changes	supplied <i>localProjectDocumentList</i> [<i>i</i>] // List of all documents of a project
	supplied <i>activeDocument</i> [<i>i</i>] // Link to the local active document (scene and model)
Responsibilities	
Liveness:	
PROJECTSUPPORT[<i>i</i>] = (RequestListOfDocumentNames. AwaitListOfDocumentNames. (MaintainActivity ManageDocuments) ^ω	
MAINTAINACTIVITY = (AwaitActivitySwitch. [RequestDocument]. SwitchActivity)	
MANAGEDOCUMENTS = (AwaitNewDocumentInfo. AddDocumentToList)	
Safety:	
<i>activeDocument</i> = <i>nil</i> => RequestDocument	

Figure 4.3.2: PROJECTSUPPORT role schema.

4.3.1.4 Discussion

When conceptualizing the environment, its important to note, as explained by [Zambonelli 03] that the dichotomy between the environment and the multi-agent system might not always be clear. The environment itself is made up of several fully

Table 4.4: Names and description of roles, and the reference of their role schemas in Appendix C.

Role Name	Description	Schema
APPLICATIONMANAGEMENT[k], $k =$ <i>total_applications</i>	Because applications run across multiple devices, coordination is required between devices to manage user interface components. The application manager role can trigger and respond to application-level events on other devices. The application manager role is also responsible for <i>importing</i> another document into the current document. (e.g. importing brainstorming document into project planning)	C.3
APPLICATIONSUPPORT[i][k], $i =$ <i>total_ancillary_shared_devices</i> , $k =$ <i>total_applications</i>	The application support role can trigger and respond to application-level events on other devices. (e.g. inter-device portal interaction technique)	C.4
PERSISTENCEMANAGEMENT	This role is responsible for storing the current project information, including all shared document data on other devices, into a persistent database or file structure.	C.5
PERSISTENCESUPPORT	This role is responsible for coordinating with the Persistent Manager Role and sending local document information to be synchronized and rendered persistent.	C.6
USERIDENTIFICATIONONSHAREDSPACE	Takes <i>interactionEvents</i> that occur on a user-agnostic shared surface, identifies their owner, and transfers them to the corresponding user's local agent (e.g. on the primary shared surface, this would be the agent with the USERPRIMARYSHAREDDEVICESUPPORT role.)	C.7
USERPRIMARYSHAREDDEVICESUPPORT[p], $p =$ <i>total_participants</i>	This role serves as link between the active document on a shared device and the user's personal access point on the that shared device. This role is responsible for executing actions on the information resources on this device which come from other roles. Requests to modify the local document model or scene, may come from other agents serving the same user.	C.8
USERPERSONALDEVICESUPPORT[p], $p =$ <i>total_participants</i>	If a user has a personal device, then this role is responsible for the coordination (with the user's other agents on other devices) of the user's information resources across all devices. It may follow the user's actions in the environment or it request specific modifications of the project document on the primary shared device.	C.9
USERPERSONALASSISTANT[r], $p =$ <i>total_participants</i>	This role is responsible for all verbal exchanges with the user, including information requests, and multimodal fusion.	C.10

functioning user interfaces, which exist across several devices. Although the multi-agent system is only a true necessity when the environment engages in inter-device communication, it may still be helpful in managing local user interface components as well. However, for the purposes of the infrastructure design, I focus *almost* entirely on the needs of inter-device communication, but other practitioners may add additional agents and agent organizations to this infrastructure to suit their own local user interface needs. In doing so, I respect the goals of a distributed and modular environment (G1), as well as user-interface toolkit independence (D1).

Another issue that has yet to be discussed is exactly how the environment interfaces with the multi-agent system. Depending on the toolkits and programming languages which are used, this can be as simple as local method calls, or remote method invocation, but this is more an implementation problem than a conceptual one, and will be addressed in a later chapter.

4.3.2 Interaction model and organizational rules

In the previous subsection, I presented the role schemas while focusing on the responsibilities of the roles and their relationship with the environmental resources. This emphasizes the physically distributed nature of these resources, which will be the most influential factor on the overall structure of the multi-agent system. What is difficult to intuitively grasp with the role schemas alone is the structured patterns of interactions that occur across the entire organization and the general relationship between different roles. This will be explored in this section through the elaboration of organizational rules, and the interaction model.

4.3.2.1 Gaia's interaction model and organizational rules

The interaction model is responsible for capturing the dependencies and relationships between the various roles in a multi-agent system. This information complements the role model, by focusing on the essential nature and purpose of interaction between roles, rather than on the precise ordering of particular message exchanges. The legend of the interaction model is described in Figure 4.3.3.

The organizational rules in the Gaia methodology are similar to the responsibilities of a role, but at an organizational level. They can be captured using liveness properties and safety properties. Here, safety properties describe time-independent global invariants that the organization as a whole must respect. Liveness properties describe the patterns of execution in an organization. These properties can be described using a textual description or they can be provided using some formal notation. Gaia does not prescribe a formal notation, but they do provide some suggestions, which I use here. For example, $P(R1) \rightarrow Q(R2)$ indicates that protocol P must necessarily be executed by role R1 before R2 can execute protocol

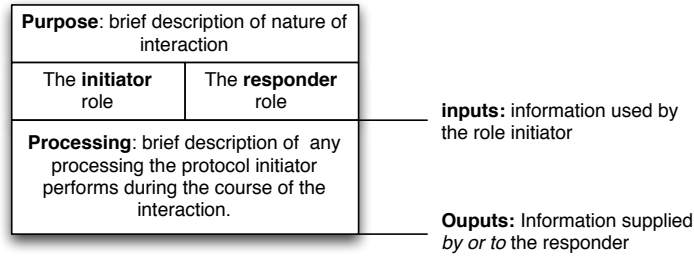


Figure 4.3.3: The interaction model.

Q. Another notation I use is $R(property)$, $R = Role1, Role2, \dots, RoleN$, which means that *property* is spread throughout all roles listed in R . Additionally, I also use $Role1 \leftrightarrow Role2$ to indicate a necessary concurrency between two roles.

4.3.2.2 TATIN-PIC's organizational rules

There are two main objectives that the organization is trying to achieve. The first is related to goal G1, and concerns the fact that the infrastructure must manage distributed project documents across environment. There is the same active document of a project that must be opened on all shared devices in the interactive space. This means that the *documentID* of the *activeDocument* and the *masterActiveDocument* which exists across all shared device must be the same. Also, because there can be one and only one application running at a time, it will be appropriate if the APPLICATIONMANAGEMENT and APPLICATIONSUPPORT roles could do so as well, by suspending and resuming their activity when required..

$$R_1(activeDocument.documentID(x)), R_1 = (PROJECTSUPPORT[i], APPLICATIONSUPPORT[k][i], PERSISTENCESUPPORT[i])$$

$$PROJECTMANAGEMENT(masterActiveDocument.documentID(x)) \leftrightarrow R_1(activeDocument.documentID(x))$$

$$\begin{aligned} & \text{SuspendActivity}(APPLICATIONMANAGEMENT[k_i]) \rightarrow \\ & \text{ResumeActivity}(APPLICATIONMANAGEMENT[k_j]), \text{ such that } i \neq j \end{aligned}$$

The second internal objective of interactive space's infrastructure (related to G2) is that *userEvents* are shared and distributed to all other roles which are serving the same user. For every *userID*, there exists a group of agents which serve the user.

$$R_y(userID(y)), R_y = (USERPRIMARYSHAREDDEVICESUPPORT[p_y], \\ USERPERSONALDEVICESUPPORT[q_y], USERPERSONALASSISTANT[r_y])$$

$$R_y(userEvent(z)), \forall z, y \text{ such that } userEvent(z).userID = userID(y)$$

4.3.2.3 TATIN-PIC interaction model

These organizational safety properties are conditions which must be upheld throughout the design of the protocols used by the agents. These protocols are expressed using Gaia's interaction model. I present three interaction models, which are related to the PROJECTMANAGEMENT role and the PROJECTSUPPORT role as described in role schemas.

- Figure 4.3.4 illustrates the exchange between the PROJECTMANAGEMENT role and the PROJECTSUPPORT role in order to synchronize project lists. The first exchange occurs when the PROJECTSUPPORT role is initialized, and the second occurs when the moderator creates a new document in the environment.
- Figure 4.3.5 illustrates the exchange between the ProjectManagement role, ProjectSupport role, ApplicationManagement role and the ApplicationSupport role during activity switches.

The remainder of the interaction models can be found in Appendix C. I provide a brief description of these below:

- Figure C.4 illustrates the exchange between the APPLICATIONMANAGEMENT role and the APPLICATIONSUPPORT role when one document is imported into an existing document.
- Figure C.5 illustrates the exchange, at an application-level, of modifications that occurs to the active document between the APPLICATIONMANAGEMENT and APPLICATIONSUPPORT roles. Note that their interaction patterns, and roles, are similar. Its possible to capture this using only one role, but I use two different roles to emphasize the different information resources that the roles have access to (*masterActiveDocument* and *activeDocument*).
- Figure C.6 illustrates the exchange between the PERSISTENCEMANAGER and PERSISTENCESUPPORT, when the latter sends the former a copy of its local document model for storage.

- Figure C.7 illustrates how *interactionEvents* are passed from the USERIDENTIFICATIONONSHAREDDEVICE role to the appropriate USERPRIMARYSHAREDDEVICESUPPORT role, which then generates a *userEvent* and transfers this to other roles serving the same user.
- Figure C.8 illustrates the exchanges which are initiated by USERPERSONALDEVICE role and the USERPERSONALASSISTANT role.

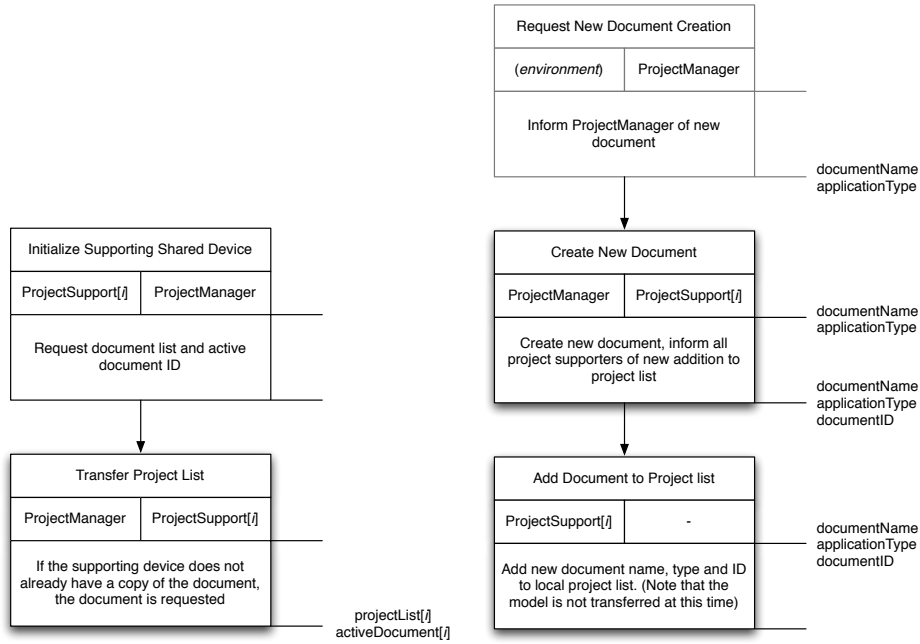


Figure 4.3.4: Interaction model associated with the PROJECTMANAGEMENT and PROJECTSUPPORT roles.

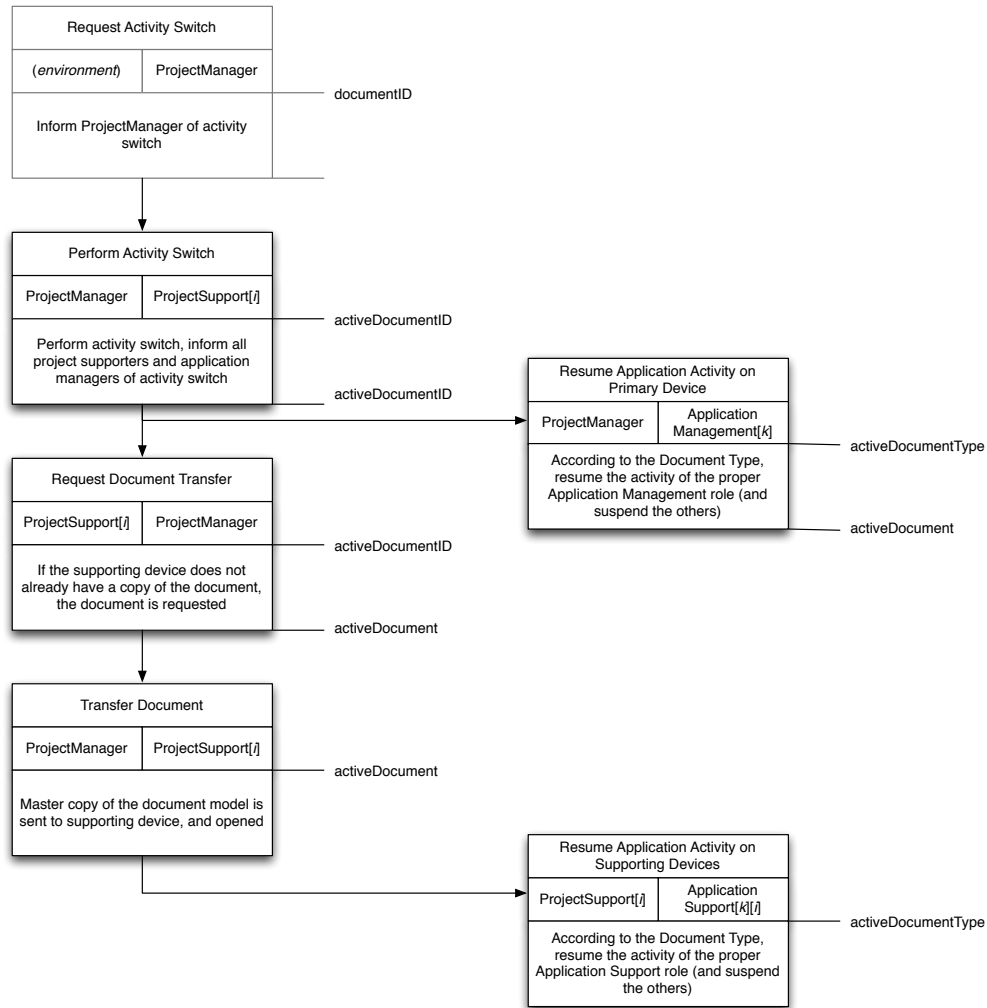


Figure 4.3.5: Interaction model associated with an activity switch, involving the PROJECTMANAGEMENT, PROJECTSUPPORT, APPLICATIONMANAGER and APPLICATIONSUPPORT roles.

These role and interaction models, and the environmental model and sub-organizational rules conclude the analysis stage of the Gaia methodology for TATIN-PIC

4.3.3 Organizational Structure

With the analysis stage of the Gaia methodology finished, I may now define some of the organizational structure for the roles. This will make the relationships that are implied in the interaction model more explicit. Gaia again does not prescribe any particularly notation for expressing the organizational structure, but suggests

the use of relationship roles through three different kinds of relations: a *control* relationship, when one role has authority over another role allow it to partially control the actions of this subordinate role; a *peer* relationship, where two roles have equal status; and a *dependency* relationship, where one role relies on some resources or knowledge from another role in order to complete its tasks.

With these three kinds of relationships, *the multi-agent system can be divided into three sub-organizations*. The first address project, application, and document aspects of the system:

- $\forall i \text{ PROJECTMANAGEMENT} \xrightarrow{\text{controls}} \text{PROJECTSUPPORT}[i]$
- $\forall i, j \text{ such that } i \neq j, \text{PROJECTSUPPORT}[i] \xrightarrow{\text{peer}} \text{PROJECTSUPPORT}[j]$
- $\forall k \text{ PROJECTMANAGEMENT} \xrightarrow{\text{controls}} \text{APPLICATIONMANAGEMENT}[k]$
- $\forall i, k \text{ APPLICATIONSUPPORT}[k][i] \xrightarrow{\text{depends_on}} \text{APPLICATIONMANAGEMENT}[k]$
- $\forall i, j, k \text{ such that } i \neq j, \text{APPLICATIONSUPPORT}[k][i] \xrightarrow{\text{peer}} \text{APPLICATIONSUPPORT}[k][j]$
- $\forall i, j, \text{ such that } i \neq j \text{PROJECTSUPPORT}[i] \xrightarrow{\text{peer}} \text{PROJECTSUPPORT}[j]$

The second sub-organization involves only two roles, the PERSISTENCEMANAGER and the PERSISTENCESUPPORT.

- $\forall i \text{ PERSISTENCEMANAGEMENT} \xrightarrow{\text{depends_on}} \text{PERSISTENCESUPPORT}[i]$

The third sub-organization deals with the roles which support users. They are linked to the first organization because they depend on the active document specified by the PROJECTMANAGEMENT role.

- $\forall i \text{ USERPRIMARYSHAREDDEVICESUPPORT}[i] \xrightarrow{\text{depends_on}} \text{PROJECTMANAGEMENT}$
- $\forall i \text{ USERIDENTIFICATIONONSHARED SURFACE} \xrightarrow{\text{controls}} \text{USERPRIMARYSHAREDDEVICESUPPORT}[i]$
- $\forall i \text{ USERPRIMARYSHAREDDEVICESUPPORT}[i] \xrightarrow{\text{peer}} \text{USERPERSONALDEVICESUPPORT}[i]$
- $\forall i \text{ USERPERSONALDEVICESUPPORT}[i] \xrightarrow{\text{peer}} \text{USERPERSONALASSISTANT}[i]$
- $\forall i \text{ USERPERSONALASSISTANT}[i] \xrightarrow{\text{peer}} \text{USERPRIMARYSHAREDDEVICESUPPORT}[i]$

In this third sub-organization, I also found it useful specify three additional organizational relationships, which indicate that an entity playing a role for one user is also a peer to all other entities playing this role for the other users. In the application of the Gaia methodology so far, I have not addressed examples of collaboration between agents of different users, which might, for example, allow one personal assistant agent to consult another personal assistant agent, when completing a task. Though these kinds of relationships will be under-exploited in the design and implementation of TATIN-PIC system, they do exist, and therefore they are represented by these three relationships.

- $\forall i, j$ such that $i \neq j$, $\text{USERPRIMARYSHAREDDEVICESUPPORT}[i] \xrightarrow{\text{peer}} \text{USERPRIMARYSHAREDDEVICESUPPORT}[j]$
- $\forall i, j$ such that $i \neq j$, $\text{USERPERSONALDEVICESUPPORT}[i] \xrightarrow{\text{peer}} \text{USERPERSONALDEVICESUPPORT}[j]$
- $\forall i, j$ such that $i \neq j$, $\text{USERPERSONALASSISTANT}[i] \xrightarrow{\text{peer}} \text{USERPERSONALASSISTANT}[j]$

4.3.4 Agent model

The remaining models, the agent model and the service model, belong to the detailed design phase of Gaia. The agent model serves to provide a mapping from the previously established roles to agents, along with the number of such agents that will be instantiated in the multi-agent system.

The agent model for the interactive space is presented as a list below. Here, I parameterize the agents that are instantiated (shown in superscript after the agent name) so that it may apply to multiple configurations of interactive spaces, depending on the number of shared surfaces, personal devices and I/O devices that are used, and also the number of different kinds of applications that the interactive space contains. I use a one-to-one mapping between roles and agents, which is not uncommon in the Gaia methodology [Wooldridge 00].

- $\text{PROJECTMANAGER}^1 \xrightarrow{\text{play}} \text{PROJECTMANAGEMENT}$
- $\text{PROJECTSUPPORTER}^{\text{total_ancillary_shared_device}} \xrightarrow{\text{play}} \text{PROJECTSUPPORT}$
- $\text{APPLICATIONMANAGER}^{\text{total_applications}} \xrightarrow{\text{play}} \text{APPLICATIONMANAGEMENT}$
- $\text{APPLICATIONSUPPORTER}^{\text{total_applications} * \text{total_ancillary_shared_device}} \xrightarrow{\text{play}} \text{APPLICATIONSUPPORT}$
- $\text{PERSISTENCEMANAGER}^1 \xrightarrow{\text{play}} \text{PERSISTENCEMANAGEMENT}$

- $\text{PERSISTENCESUPPORTER}^{\text{total_ancillary_shared_device}} \xrightarrow{\text{play}} \text{PERSISTENCESUPPORT}$
- $\text{USERIDER}^1 \xrightarrow{\text{play}} \text{USERIDENTIFICATIONONSHARED SURFACE}$
- $\text{WORKBENCH}^{\text{total_participants}} \xrightarrow{\text{play}} \text{USERPRIMARYSHAREDDEVICESUPPORT}$
- $\text{PERSONALDEVICE}^{\text{total_personal_device}} \xrightarrow{\text{play}} \text{USERPERSONALDEVICE}$
- $\text{PERSONALASSISTANT}^{\text{total_personal_assistant}} \xrightarrow{\text{play}} \text{USERPERSONALASSISTANT}$

Now with the agents defined, I take this opportunity to reemphasize the organizational structure of the multi-agent system, defined in the previous section, by providing an example. Figure 4.3.6 provides an illustration of an example interactive space infrastructure where the interactive space with one ancillary (or secondary) shared device (an interactive board), four participants, two personal assistant agents, two personal devices. The interactive space also runs two different applications across all shared surfaces.

4.3.5 Service model

The purpose of the service model is to identify the Gaia services that exist in the organization as provided by the roles which the agents are playing. A service in Gaia represents a single coherent block of functionality in which an agent will be engaged, often comprising multiple protocols and activities. A service also does not have to necessarily be triggered by an external request; though this often may be the case, it may also come from the initiative of an agent [Zambonelli 03, Wooldridge 00].

Identifying a service involves identifying the *inputs*, *outputs*, *pre-conditions*, and *post-conditions* of each service. The inputs and outputs can be derived from the interaction model in a straight-forward way. The pre-conditions and post-conditions are derived from a number of sources, such as the organizational rules, the safety conditions of the roles, the protocol description and processing flow of the interaction model. They represent requirements for execution and constraints on outcomes, respectively.

Because of the one-to-one mapping of roles to agents, the TATIN-PIC service model does not capture much additional information that can be gathered from the interaction model alone. However, together with the agent model and the organizational structure of the agents, it is an essential part of the implementation plan of the infrastructure. Therefore, I include it in only in Appendix C:

- Table C.1 presents the service related to the project, application, and persistence functionalities of the infrastructure.

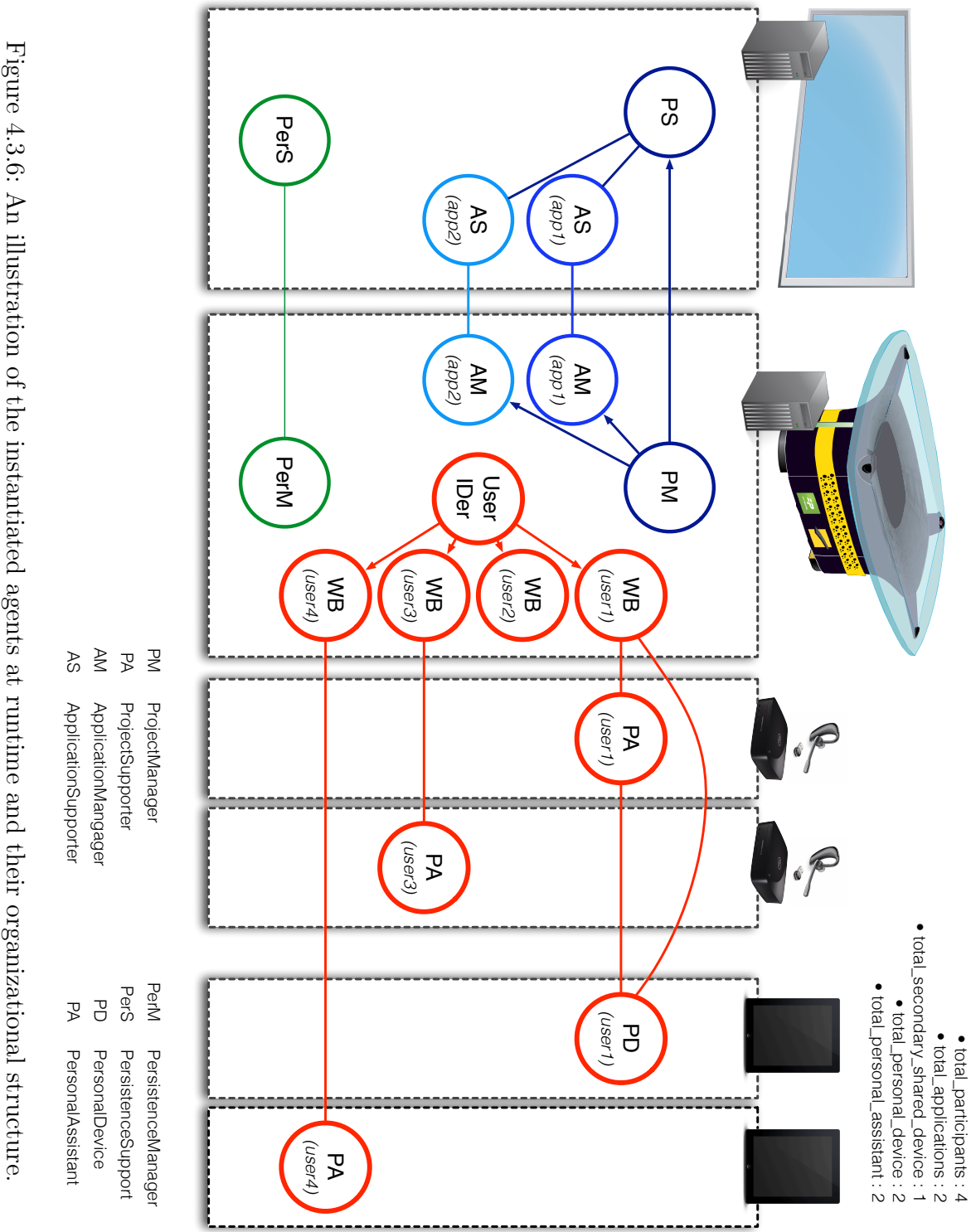


Figure 4.3.6: An illustration of the instantiated agents at runtime and their organizational structure.

- Table C.2 presents the services related to the sub-organization of a user's agents.

4.4 Conclusion

The output of the Gaia methodology is a list of agents and services that can be implemented by software engineers. There could be any number informal methods that could have been used to derive such a list, but the use of Gaia is advantageous for several reasons. First, agents and services are derived from the environmental model, the organizational rules, the organizational structure, the role schemas, and the protocols. This assures a level of quality and structural soundness in the design of the multi-agent system that is offered by the comprehensiveness of the Gaia methodology. Second, the agent model is parameterized so that this infrastructure may apply to any number of interactive spaces, depending on their configuration, allowing it to be applicable to a wide range of scenarios. Finally, and most importantly, through the use of the organizational abstractions that are provided in the revised version of Gaia [Zambonelli 03], the multi-agent system is, to a certain degree, *open*. In an open society, additional agents can still be designed and added at development-time or at run-time, provided they adhere to the organization's rules and structure. The degree to which this multi-agent system is open (particularly at run-time), will also depend on the implementation. For example, if messages are broadcast to all agents in a standard and open communication language, this will greatly facilitate the process of designing additional agents which may listen in on *userEvents*, and perform computations in parallel to the existing services.

If there are issues with the applicability of this multi-agent organization to other interactive space, then it is with the assumptions made in the environment model. Notably, I over-emphasize a reliance on a *masterProjectDocumentList* and its location on the primary shared device, when it reality in may be beneficial to have these kinds of resources running on a network server. However, because I have been detailed in the application of Gaia, these kinds of environmental configurations can still be accommodated for in easily-perceived ways. For example, the PROJECT-MANAGER, APPLICATIONMANAGER and PERSISTENCEMANAGER agents can be moved to this remote server, along with the project database, and the primary shared device can be treated as another ancillary device, complete with its own PROJECTSUPPORTER, APPLICATIONSUPPORTER, and PERSISTENCESUPPORTER agents.

This is an advantage afforded by the “development-time mobility” of agents, and contributes a great deal to the scalability of this approach. For example, interactive spaces for CSCW generally belong to two categories. The first category, which have been discussed at length, are co-located collaborative environments,

where all team members are located inside the same physical environment. The second are distributed collaborative environments, with some team members geographically separated from others involving a hybrid of co-located and distributed interaction, as explored by [Lepreux 11, Gumienny 11]. In this chapter, I propose a software infrastructure which addresses the needs of user interfaces for co-located collaborative environments. Though I consider only co-located collaboration, the design and analysis of this software infrastructure, the devices introduced in the environmental model could be geographically separated, which would allow this infrastructure to apply equally to *distributed* distributed user interfaces as well.

These kinds of strategies are dependent upon the choice of the multi-agent toolkits and platforms that are used in the implementation of the infrastructure. Certain multi-agent toolkits may afford message-passing through HTTP protocols more easily than others. In accordance with the development-time goal of platform independence for infrastructure (D2), these decisions are ultimately left to the designers of distributed computing architecture. However, in order to properly demonstrate the flexibility afforded by this design approach, the implementation of the infrastructure, presented in the following chapter, will be done with two different multi-agent toolkits, JADE and OMAS.

Part 3: Implementation & Evaluation

"Do you mean *this* Post-it?""

(TATIN-PIC Personal
Assistant)

Chapter 5

Implementation

The TATIN-PIC interactive space (Figure 5.0.1) was designed for teams in the preliminary design phase of engineering projects, performing activities such as brainstorming, project planning, and causal analysis in co-located interaction. It includes several custom-built hardware devices (an interactive tabletop, and an interactive board display), personal and mobile device such as smartphones and tablet computers, and headsets linked to small dedicated computers supporting vocal interaction. These work in concert to provide a distributed user interface run by a multi-agent infrastructure.

The process of constructing this interactive space would not have been possible if not for the assistance and expertise of a multi-disciplinary team at the Université de Technologie de Compiègne, comprised of members from the Costech laboratory, the Innovation Center and the Heudiasyc laboratory. Throughout the implementation of this interactive space, numerous design and engineering decisions were made which might be of interest to others developing similar systems. Therefore, they are detailed in this chapter.

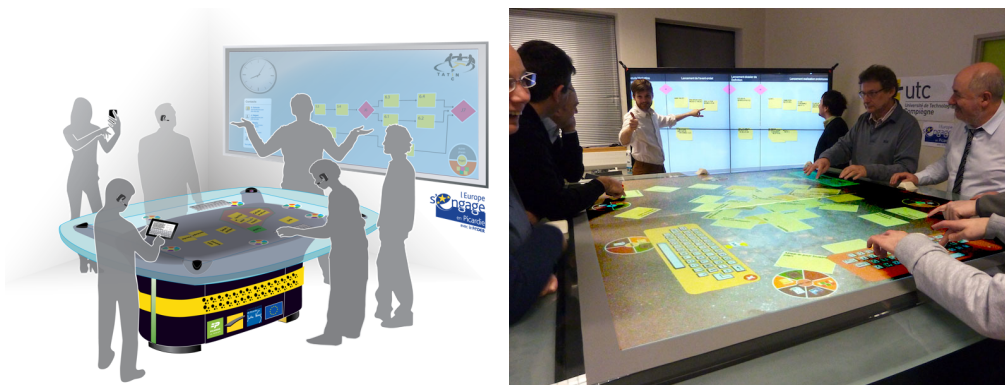


Figure 5.0.1: Interactive space concept (left) and implementation (right).

5.1 TATIN-PIC hardware

In this section, I present the interactive tabletop, the interactive board and the vocal devices that were used in the interactive space. I provide details regarding the hardware of our prototypes, but my primary goal will be to highlight some the design decisions that were involved when constructing these devices for an interactive space for collaborative work.

5.1.1 Interactive tabletop

For the purpose of evaluating our interactive space, we required an interactive tabletop that could comfortably accommodate parallel interaction from six to eight participants. Of the few interactive tabletops that are available commercially, none offer an appropriate surface area for such a group size. Therefore, we were required to construct our own prototype.

First, the physical structure of the table was designed to house the internal devices (computer tower, projectors, mirrors, cameras, and connection cables). The tabletop surface consists of an opaque projection screen and large glass pane. The height of the tabletop surface (1.2 meters) is designed for use while standing or seated on high chairs or stools. This is a relatively high surface, and its design has the intention of encouraging mobility in the interactive space, rather than having the participants stay seated.

The horizontal display of the interactive tabletop is made possible by two full-HD short-throw video projectors, and two pairs of mirrors, which cast an image on the opaque projection screen from underneath the tabletop. The screen size is 1.6m by 1.4m and its screen resolution is 1920 pixels by 2160 pixels. Presently, this is a relatively high number of pixels compared to other interactive tabletop surfaces in related work, but only produces a resolution of approximately 33 PPI. This makes it difficult to display small text, but in practice, and given the distance between the user and the surface of the tabletop, this is only occasionally a constraint (particularly when reading PDFs). For example, in our observations in Chapter 1, the participants wrote on Post-it notes in handwriting that was large enough to be seen from across the room. Displaying text in this size can be easily done on the surface of the tabletop. The tabletop surface also includes a non-interactive border on which participants can rest their hands or place personal items such as notes, smartphones, tablets, or small laptops while they are using with the table.

The surface area of the tabletop is the most important factor when choosing which multitouch input technology to use. Due to its large size, optical sensing solutions, which make use of infrared light, are easily the most feasible in terms of price and effectiveness, and also work well with projectors. There exists several

different kinds of optical set-ups, but they all share a common approach: the interactive surface is illuminated with infrared light from some source, and physical contact with the surface scatters this light, allowing it to be captured by cameras with infrared filters. Perhaps the most widely implemented multi-touch technology is Han's Frustrated Total Internal Reflection (FTIR) technique [Han 05]. This solution uses infrared LEDs mounted around the edges of the glass pane which flood the glass full of infrared light. Light is kept inside the glass pane as it reflects against the exterior of the glass (Total Internal Reflection), until some opaque object comes in contact with the glass pane, "frustrating" and releasing the light. This can then be captured by cameras with infrared filters that are placed beside the projectors underneath the tabletop. One of the problems with this approach is that it does not scale well to a surface as large as ours. The light is not evenly distributed throughout the inside of the glass, making the image processing more difficult because some areas of the surface will have less contrast than others. Diffused Surface Illumination is a similar approach, proposed by Tim Roth [Schöning 08], which uses a special kind of plexiglass that distributes the light much more evenly. This works well, but is expensive and does have some size limitation because the surface material is flexible.

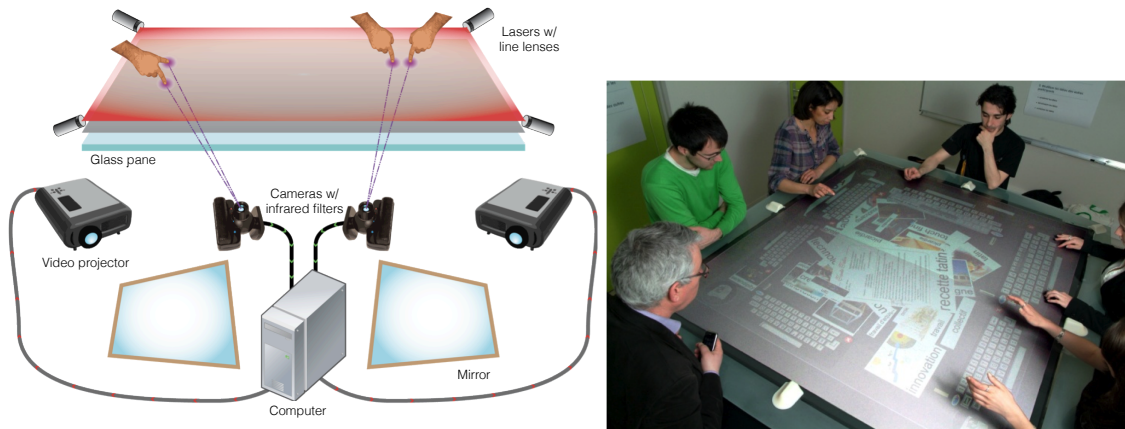


Figure 5.1.1: The interactive tabletop, hardware components (left) and in use (right). Note that the prototype uses two pairs of mirrors, eight lasers and four cameras which are not pictured in the schema for simplicity.

We use a different approach, by the means of a Laser Light Plane (LLP) illumination technique (Figure 5.1.1, left). Instead of LEDs, this approach uses infrared lasers equipped with line generating lenses that stretch the beam into a uniformly illuminated straight line which travels flush with the surface of the tabletop. With proper adjustment and laser positioning, this creates a plane of laser light that is a

millimeter above the surface. Therefore, any contact with the surface also breaks this laser light plane, causing infrared light to scatter and to be detected by the cameras. This solution works much better for large surfaces, and produces higher image contrast and more uniform distribution of light across the entire tabletop. Its important to note certain design trade-offs exist with this option. The light does not come up through the glass but rather is emitted horizontally thus making occlusion a potential issue. However, because we use eight different lasers around the edge of the tabletop, such problems with occlusion for multi-point touch interaction are extremely rare in practice, even when large objects are placed on the tabletop. Still, occlusion is an issue when detecting the contours of a user's hand, for example, because the concave shapes produced by the fingers are not uniformly illuminated as they would be if the light was emitted from underneath the tabletop. Similarly, LLP disallows the use of reflective fiducial markers which could be place underneath objects to have them detected, as the laser light never reaches them. These trade-offs are worth the robust multi-point detection that LLP provides for large surfaces.

In order to capture the infrared touch input we use four webcams 640 x 480 pixels, with a frame rate of 60 Hz and infrared filters which allow them to only detect the infrared light scattered from the laser light plan. These are calibrated and stitched together using Community Core Vision (CCV)¹, an open-source software application developed for optical multitouch sensing. CCV also handles the image processing to transform the camera input into software touch events (background subtraction, noise removal, amplification and connected components analysis) at approximately 60 frames per second. The touch events are expressed using the TUIO protocol [Kaltenbrunner 05] which packages a unique ID, an x,y coordinate, and an orientation for each touch event. These touch events are then made available to multitouch applications through the UDP network protocol on port 3333. This set-up can easily process 80 simultaneous points of interaction, which assures that we can support up to eight users at once. Figure 5.1.1 (right) shows the hardware components and the final interactive tabletop prototype.

5.1.2 Interactive board

The interactive space also includes an interactive board display. Based on the observations from Chapter 1, we see the whiteboard as a tool that is used primarily by either only one participant, or multiple participants taking turns and interacting sequentially. This observation was the most impactful factor when considering the design decisions that were made during the construction of the interactive board prototype.

¹<http://ccv.nuigroup.com>

The interactive board measures 2.05 x 1.16 meters, and has a resolution of 2732 pixels by 1536 pixels with a pixel density of 34 PPI. This again is more than adequate for displaying text in a size that approximates that of the handwriting observed in Chapter 1. The display is comprised of four LCD monitors with extra thin bezels. They are powered by one computer with a high-end graphics card, which stitches the displays together as if they were one display, thereby adjusting for the space lost by the bezels. A large single pane of glass is placed over the four monitors, so that users may have a surface on which they may perform their multitouch gestures. The input technology is a proprietary infrared overlay frame which uses embedded infrared LEDs and sensors to detect touch points. Though the specifications claim that the infrared overlay frame is built to handle 32 points of simultaneous interaction, in reality, it can handle even more. This makes it appropriate for use during the few instances where it is actually used by two participants simultaneously. Figure 5.1.2 shows the hardware components and the final interactive board prototype.

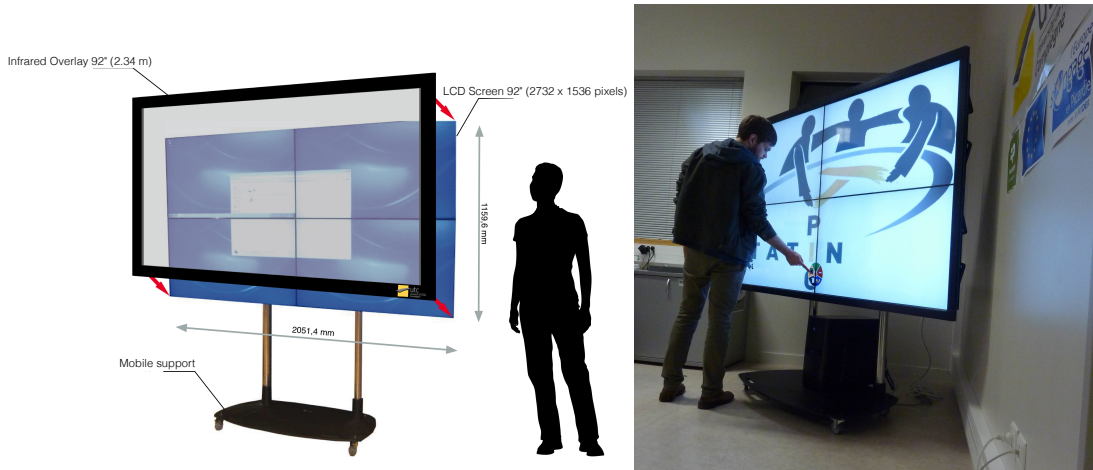


Figure 5.1.2: Interactive board display, hardware components (left) and in use (right).

5.1.3 Voice Boxes

We have designed and implemented vocal interaction for up to four different participants. The most impactful factor when designing the hardware to support vocal interaction is deciding how to allocate enough CPU and memory to perform the resource intensive processes of speech recognition and voice synthesis for multiple participants simultaneously. In order to do so, we use four computers with miniature form factors (one Dell Inspiron Zino HD, and three Mac Minis),

which we refer to as *voice boxes*. Though the space only has four at the moment, this can easily be extended to six in our implementation, so that we can support at least six multimodal participants with little difficulty. Another advantage for using separate machines is that the sound card (and operating system) can only handle one system microphone channel at a time by default. With one computer per participant, this problem is circumvented.

Vocal interaction is an optional features in the environment, because not every participant might be comfortable with this modality of interaction, especially during initial or more causal sessions in the interactive space. If they choose to take advantage of the multimodal interaction, the participants must wear one of the wireless headsets that are provided in interactive space (four Plantronics Voyager Pro which use dual microphones for noise reduction). These headsets are placed just underneath the table at various positions around the tabletop for the purpose of synchronizing a user's login with their headset. The headsets are each associated to one computer, and each voice box is associated to a general position on the interactive tabletop. If the user performs a login in at this position, the tabletop will send a message to the associated voice box communicating the username of the login. The voice box then broadcasts a message on the network asking the user's personal assistant agent to identify itself and its location. If the personal assistant agent exists, it will respond with the IP address and port on which it will listen for speech-to-text user input, and also will remember the location of the voice box so that it may communicate text that must be synthesized into voice for output to the user.

The user can open and close the microphone channel by using the microphone button on the interactive tabletop. The voice box's recognizer continuously listens on the microphone channel and if sounds above a certain threshold are detected by the microphone, the recognizer will perform speech recognition on the sounds. If the match is appropriately strong, it will transfer the text to the personal assistant agent. The software on the voice box also continuously listens on a port for text from the personal assistant so that it may synthesize it. The software uses Microsoft Speech API for both the recognition and synthesis. Speech recognition is fairly consistent in the Microsoft Speech API, but it remains problematic, because it is particularly difficult to programmatically implement and dynamically assign user-specific acoustic model and perform the training thereof. Additionally, despite the noise-canceling headsets, interference from other users can still be an issue. These factors would ultimately limit their use during our formal user evaluation activities.

5.2 TATIN-PIC user interface and applications

This section presents the general user interface of the TATIN-PIC operating system shell, applications, and tools. We discuss the login screen, the brainstorming application, the project planning application and the causal analysis application, and personal tools for the users. The multitouch user interface on the interactive tabletop and the interactive board were designed with the MT4j (MultiTouch for Java) [Laufs 10] toolkit.

5.2.1 Login screen

We have designed the operating system shell to run as a sub-system inside the current operating system on the tabletop and the whiteboard (Windows 7). Upon turning on the tabletop, if our TATIN-PIC subsystem is automatically launched, the first scene the users will encounter is the login screen. This scene enables two things. First, it provides the moderator with a circular menu through which he may create a new project or open an existing project. Then, it provides all participants with a virtual keyboard through which they can type their username, so they can access their personal circular menus. At this time, each user is assigned a color at random, which will highlight any future user-specific components throughout this session. The virtual keyboards also use an autocomplete feature which facilitates the login process. At this time, if the interactive board has booted, it simply displays a background image with no interface features.

After all participants have performed their login the moderator is free to create a new document (or open an existing one) in one of three applications: brainstorming, project planning, and causal analysis.

5.2.2 TouchSuite: applications for the TATIN-PIC interactive space

In this section, we present the multi-surface applications for brainstorming, project planning, and causal analysis inside our interactive space. These applications share similar designs and behave in similar ways. Their most unifying characteristic is that they all rely heavily on a Post-it note metaphor to suggest pseudo-physical possibilities of interaction, such as dragging and rotating. The metaphor is a natural fit for the applications because each of these creativity techniques when executed in a traditional space similarly relies on the physical use of Post-it notes.

In order to create Post-it notes inside these applications, the users must use their personal virtual keyboards (accessible from the personal circular menus). As users type, characters appear in a small white text box in a bar just above the

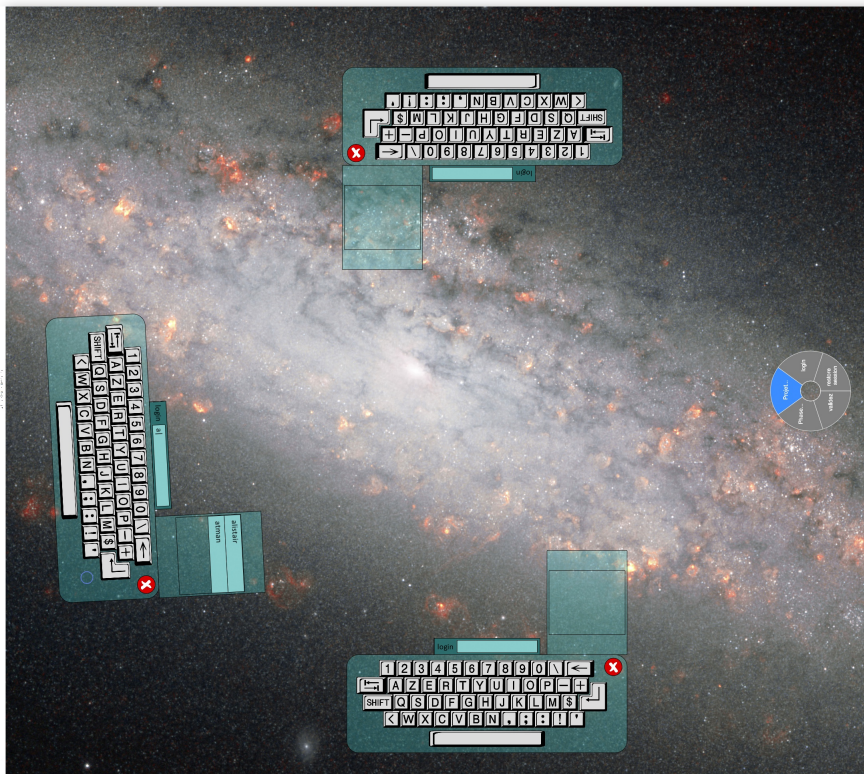


Figure 5.2.1: Login screen of TATIN-PIC on the interactive tabletop.

keyboard. This text box includes a cursor, which can be moved by tapping on a new position in the text, and pins, which can be dragged to highlight text so that it may be replaced. When the user is finished writing the text, he or she may drag the text off the keyboard's edit bar, and onto the surface of the tabletop, creating a virtual Post-it note.

All keyboards and virtual Post-its on the interactive tabletop use direct manipulation techniques and adhere to the effects of a physics engine (implemented with the jBox 2D library). This allows users to use drag gestures to push and pull components with intuitive, meta-physical effects. This facilitates manipulation, allowing users with enough practice to accurately move, throw or spin components with one, two or three fingers ("small"-class gestures). Users can also select Post-its by tapping on them which causes them to be highlighted in their own user-specific color (additionally, they are added to the user model so that they can be made available to personal assistant agents and personal device agents.)

5.2.2.1 BrainTouch : brainstorming application

The BrainTouch application allows users to use the creation of Post-it notes and groups for the purpose of brainstorming. As previously described, user create Post-its from virtual keyboards, which are opened from the circular menus. The grouping techniques presented in Chapter 3 are used to construct labelled groups. The brainstorming application is stateless, and does not impose separate phases such as idea generation, semantic cleaning, and categorization. These kinds of steps are enforced socially, giving more freedom to the moderator and the group to organize, or improvise, their approach according to their own personal preferences for collaborative creativity.

When Post-it notes are created (via dragging a text box out of a keyboard and onto the canvas), they are sent to the board display. The text of the idea appears anonymously, and fades away after a short time. This effect provides feedthrough, helps promote group awareness, and encourages associative thinking with other users' ideas during the idea generation phase of a brainstorming session (Figure 5.2.2).

After the completion of the brainstorming session, the moderator may choose to output the file as text, or as a mind map, and distribute these files to all participants via Dropbox.

5.2.2.2 ProjectTouch: project planning application

The design of ProjectTouch (Figure 5.2.3, the project planning and PERT analysis application, relies heavily on the observations and hierarchical task analysis from the Chapter 1. It also can be used in conjunction with the brainstorming

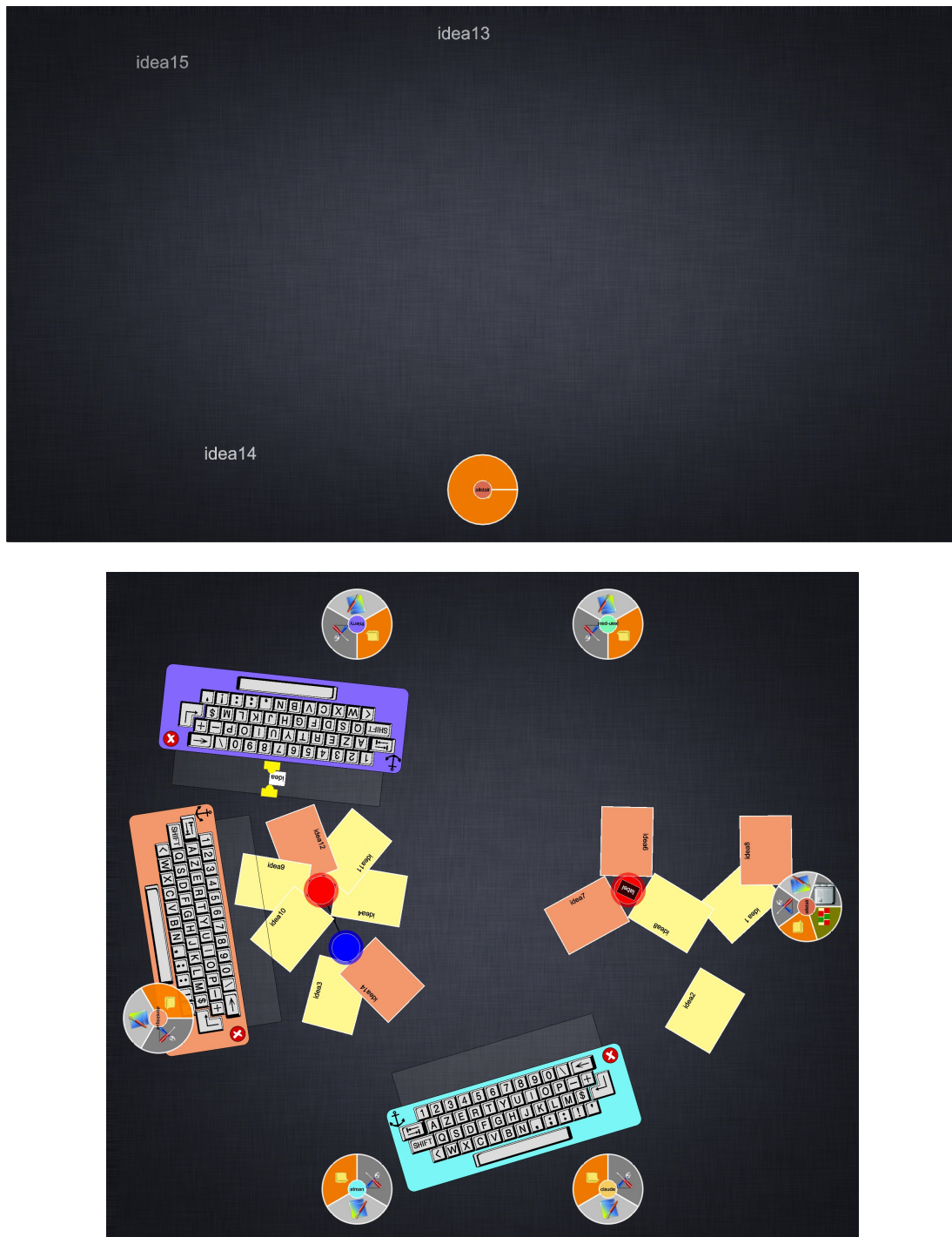


Figure 5.2.2: BrainTouch: multi-surface brainstorming application. Screenshot from the interactive board (above) and interactive tabletop (below).

application, by importing the data from a completed brainstorming document.

Most of the interaction, as is the case in the traditional environment, takes place on the vertical surface. There are three main virtual components to the project planning application on the virtual board: tasks, phases, and milestones. First, *tasks* have the appearance of a yellow Post-it note. They can be created by a symbolic gesture where the user gestures in the shape a square. Similarly, they can be deleted by drawing an “X” over the Post-it note. They are moved with a one-finger drag gesture. A double tap on the Post-it note opens up a keyboard allowing it to be edited. The tasks are also automatically numbered by their task ID in the top left corner of the Post-it note. This similarly can be modified by double tapping on the ID, which opens a number pad to change the ID.²

The second class of components includes the *phases*. They appear as large grey rectangular blocks. The name of the phase is located in a dark-grey header, which users can double-tap to edit with a keyboard. After the last phase, there is a button with a “+” sign which allows users to add additional phases. Phases can be deleted by dragging the header, and throwing the phase up above the display. Phases can be reordered by dragging the header of a phase to its new position.

Phases also include the third component of ProjectTouch: the *milestone*. By convention, because the transition from one phase to another requires that the team pass through a milestone, the milestones are automatically attached at the end of the phase. Therefore, they are created and deleted automatically with the phase component. The milestone also acts as a handle to resize the phase; dragging the component left or right will decrease or increase the area of the phase, providing more or less space to manage Post-its.

A general project plan can be built using these components. This project plan can be exported by the moderator into a Microsoft Project Document and distributed via Dropbox.

The application also includes additional functionality so that users can add durations to task, and dependency relationships to perform a PERT analysis. First, *successor and predecessor relationships* can be added by drawing arrows between the tasks and milestones, using a two-finger drag gesture. These arrows can be deleted in a manner similar to the tasks, by performing a symbolic “X” gesture on the lines. Then, after the moderator activates the PERT analysis view, through a button on the menu on the board display, an additional component appears underneath each task and milestone. The PERT analysis component is a

²This is only necessary when users want fine-grained control over the order of the tasks when they are exported, or if they want to add one task as a subtask of another task. For example, renumbering the ID of Post-it “1.4” to “1.2.1” inserts this task as a sub-task of task “1.2”. Once an ID is changed, all other task are updated to reflect the insertion. This can also be used to group multiple tasks together, although we favor the clustering proximity technique presented in Chapter 3.



Figure 5.2.3: ProjectTouch: multi-surface project planning application. Screenshot from the interactive board (above) and interactive tabletop (below).

small rectangular component which displays *the duration of a task*, as well as the *free float* and the *total float*. The PERT component can be expanded by pulling down on the component, which reveals additional information such as the earliest start date, earliest end date, latest start date, and latest end date. It also provides a dial by which the user can adjust the duration of the task. By modifying the duration of a task, all subsequent tasks are automatically updated to reflect the changes of the PERT calculation in real time. The *critical path* can be identified by following the task with zero total float.

Additionally, the entire view can be panned left and right with a four-finger drag gesture, and zoomed with a five-finger scrunch gesture. These are “big” gestures which allow for control not on a specific component, but rather on the high-level environment. Our only departure in maintaining the “small” and “big” classes of gesture is the two-finger drag for drawing arrows, and the one-finger drag for moving components. This exception did not pose problems for the users of the system during initial, causal experimentation with users. However, maintaining the two gesture classes in the future can be done by adding an small arrow component to the tasks, allowing the user to draw lines by beginning a one-finger drag gesture on this component.

While much of the activity takes place on the whiteboard, the interactive tabletop is reserved primarily for complementary personal tasks. We also developed a voting mechanism which allows the moderator at the board to settle any discrepancies about the duration of tasks by asking the group to vote on a particular task. The participants can then open a number pad from their personal menus and enter a number for the duration of the task. After the voting is done, the moderator can view the results on the board, and use this information to decide what the duration of the task should be.

It is often the case the task should be identified by the group before the start of project planning, or similarly, the tasks from previous project planning can be used as a foundation for the current planning. This functionality is made available through the “import” functionality. Before beginning the planning, a moderator may ask the group to perform a brainstorming to identify the tasks that will be used in the project plan. This allows each member of the team to generate tasks that are relevant to their own department. Then, using the grouping technique, the users can group the tasks by their general phase (i.e. pre-project, prototyping, industrialization, etc.). Upon completion of the brainstorming of task, the results can be imported into the project planning application as phases and tasks. This method directly involves all participants of the team, allowing for the initial project plan to be constructed by a more democratic means than if they were just created by the moderator at the board.

5.2.2.3 CauseTouch: causal analysis application

CauseTouch (Figure 5.2.4) allows groups to perform group causal analysis, much in the style observed in Chapter 1. CauseTouch also makes use of the portal interaction technique, to transfer items between the tabletop and the board. Similar to BrainTouch, the tabletop application allows for the creation of cause-and-effect Post-it notes by the entire group, (or one participant playing the role of the scribe). These Post-it notes can be passed to the board display, through one of six portals. The portals on the tabletop can be dragged in any spatial arrangement which might be convenient to the group.

On the interactive board, the moderator can use the portals on this board surface to transfer Post-it notes from and to the interactive tabletop. The portals are accessible to the moderator through a vertical drawer located at the right side of the screen. This vertical drawer can also serve as a bank for any temporarily unused Post-it notes in the diagram.

The cause-and-effect diagram on the vertical surface of the application uses interaction techniques that are similar to the gesture used in project planning diagram. Post-it notes can be created using a symbolic square gesture, deleted using a symbolic “X” gesture, linked together using a two-finger drag gesture and moved using a one-finger drag gesture. This is similar to how Post-it notes were handled on the interactive board in ProjectTouch, but one main difference between the two applications is that CauseTouch uses *an invisible grid* to help position and align the Post-it notes. For example, when dragging Post-it notes on the CauseTouch board, they move incrementally, and approximately 20 pixels at a time. The purpose of the grid is to allow participants the possibility to easily align Post-it notes in the diagram.

The environment also enforces certain logic and causal analysis constraints. For example, the cause-and-effect diagram uses one tree which represents the causes, and one tree which represents the effects. Connections between the two trees are not possible unless they pass through the immovable center Post-it, which represents the problem or event that is the subject of the analysis. Additionally, cyclical relationships in the tree are not logically sound, and therefore are not possible in the diagram.

CauseTouch also makes use of importation and exportation features. Importing documents from other phases adds the data to Post-it notes which appear in the bank on the right. Exportation is also available to the team in formats such as mind-maps, which are distributed to the team via Dropbox.

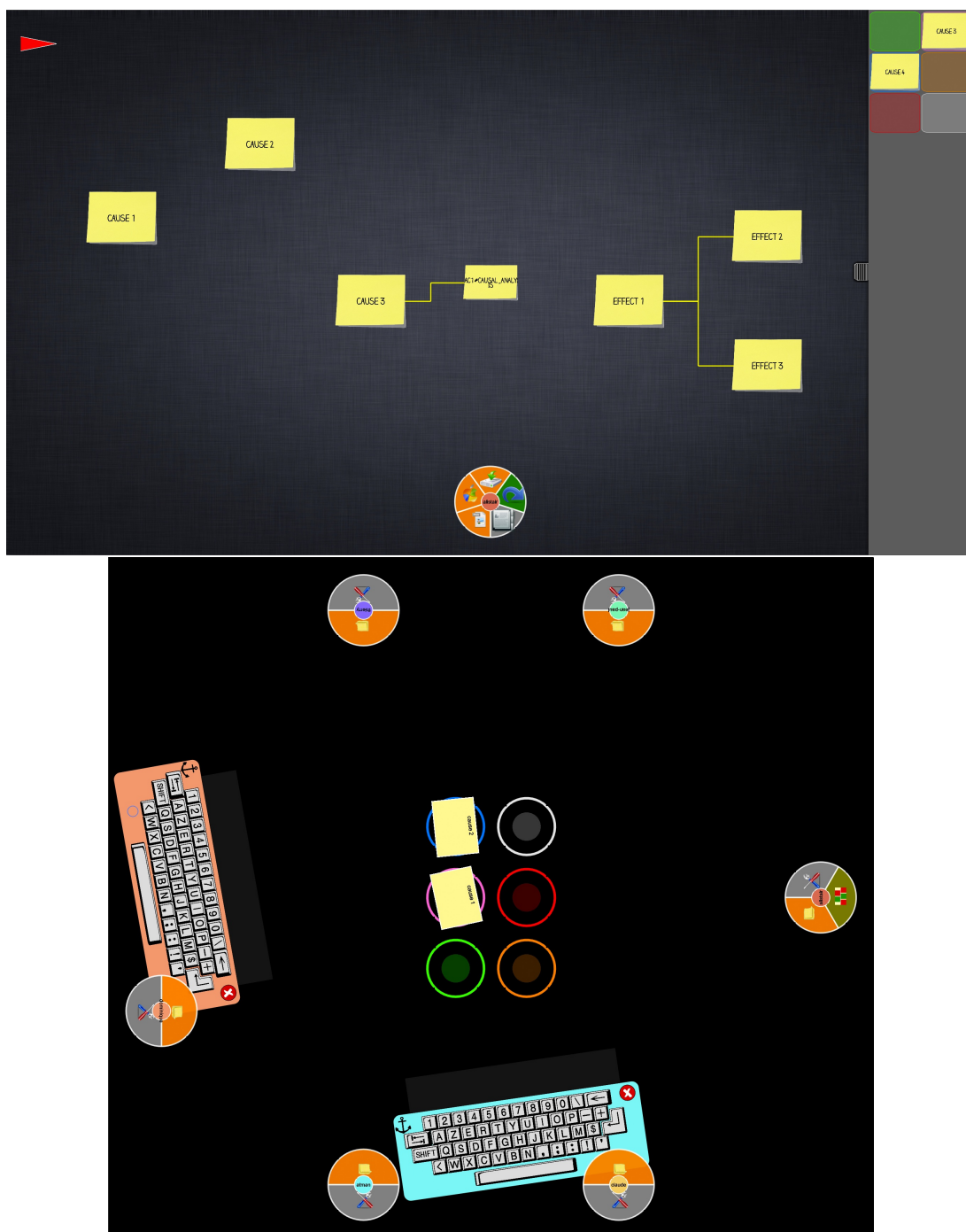


Figure 5.2.4: CauseTouch: multi-surface causal analysis application. Screenshot from the interactive board (above) and interactive tabletop (below).

5.2.3 Personal tools

As described in Chapter 3, the operating system shell of the TATIN-PIC system includes user-centered applications, or *tools*, that exist in the personal layer of the interface. These tools are accessible through the user's circular menu, where they have access to a file browser and a web browser. Each user has access to a preconfigured local or network directory which serves as the root directory for their file browser. Users can share files with other users from the file browser's share menu, which adds the selected files to other participant's directories. Additionally, tapping on a file in the file browser allows users to open files.

Currently, the file browser can launch a PDF viewer for PDF files, a movie viewer, for various file types and video encoders, and an image viewer for PNG and JPG files. These viewers have visual accents in their frames of the color of the user that has opened them. A long press, or dwell, on viewer can open a circular menu with additional functionality, such as sharing this file (an alternative method to sharing through the file browser) or cloning a view. A cloned view can still respond to events from the master view; for example, a user might want to clone a PDF file so that a participant opposite to him may also view the file from the appropriate orientation. The user may do so and pass the clone view to other participants. Then, while the user is discussing the document and flipping through the pages of the original PDF view, the clone view will also follow along. The start, stop and seeking functionality of the movie viewer also behaves similarly with its cloned views. Any file that cannot be opened by a viewer, is opened in an arbitrary file type viewer, which displays the name, and certain meta-data such as the size, last modified date, and file extension. The web browser allows users to access Internet web pages (or if a user taps on an HTML file from the file browser, it will open here). The web browser also includes personalized functionality such as persistent home pages and bookmarks.

These components are shown in Figure 5.2.5.

5.2.4 Personal assistant agent and multimodal interaction

With the headset and connected voice box, users may use multimodal interaction techniques with the applications. Multimodal interaction makes use of an embodied personal assistant agent which can understand and respond to users' vocal requests. Users can access this functionality by activating the vocal interface component from their personal menu (Figure 5.2.6). This component provides users with visual feedback regarding the result of the voice recognition module, as well as control over the microphone. Upon opening this component the personal assistant agent will introduce itself, indicating it is ready for input.

Because the Post-it note metaphor is so useful and prevalent in the TouchSuite,

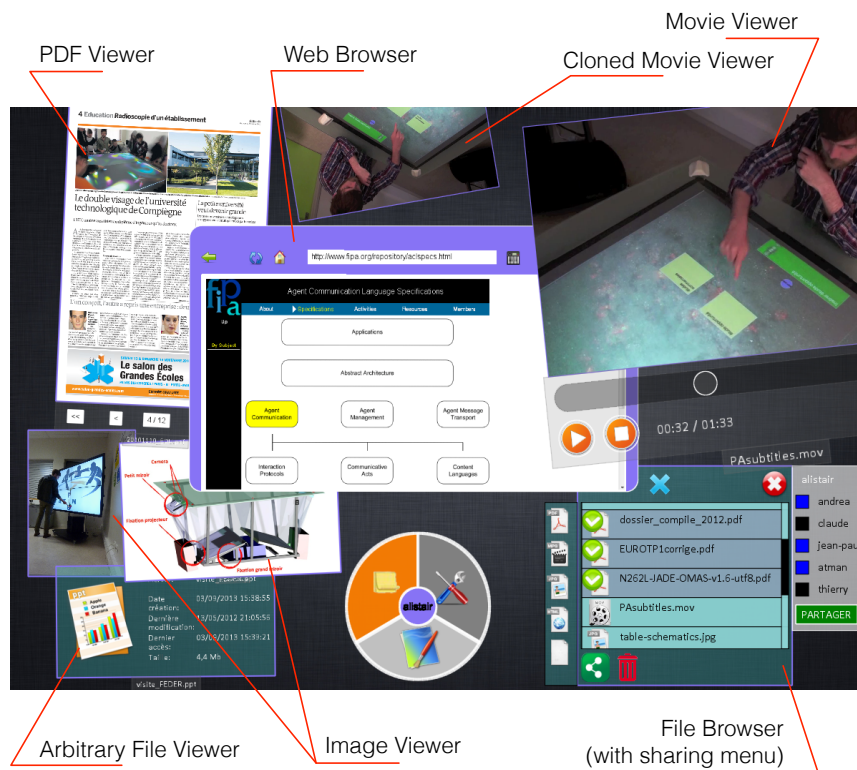


Figure 5.2.5: Personal tools in the personal layer, opened from the user's circular menu.

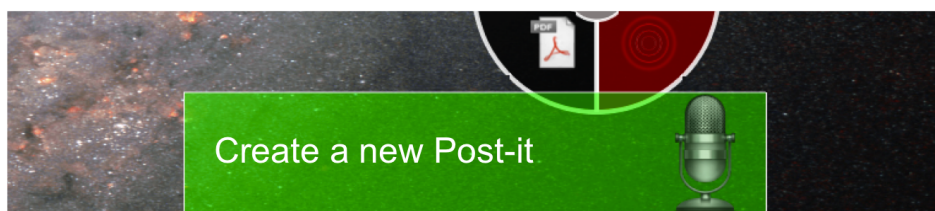


Figure 5.2.6: The vocal interface component, which provides users with feedback regarding the result of the voice recognition module, as well as control over the microphone.

the multimodal interaction techniques we designed are largely focused around the use of Post-it notes. An abridged list of available commands can be found in Table 5.1.

Table 5.1: Vocal and multimodal commands for creating, editing and deleting Post-it notes.

Command	Example vocal command	Processing comments
Creating a new Post-it note	“Create a new Post-it note”; “New note”	-
Adding text to a Post-it note	“Insert <text>”	If the user’s last action references some Post-it note, insert the text into this Post-it note, or if not, than ask the user which note.
Creating a new Post-it note with text	“Create a new Post-it with text <text>”; “New Post-it titled <text>”	-
Finding a Post-it note on the table	“Show me where the <some text> is”; “Find Post-it <some text>”	-
Deleting a Post-it note	“Delete this note”; “Delete these notes”	Personal assistant will ask the user for confirmation.

We use the OMAS (Open Multi-Agent System) toolkit, written in Lisp, to implement the personal assistant agent because it is appropriate for the rapid prototyping of intelligent agents. Each agent has a *task library*, which represents the set of possible actions for an agent. The definition of a task concept requires specifying its name, its linguistic cues, and a link to a dialog concept. The *linguistic cues* of a task determine which task will be selected upon receiving a user’s natural language request originating from the voice recognition process running on the voice box. When defining a task, the `:indexes` parameter allows a programmer to specify the list of linguistic cues. Each word or phrase (cue) has a weight between -1 and 1. If the input phrase contains the linguistic cue, which has a weight of -1, the task cannot be selected. Similarly, if the input phrase contains a linguistic cue with a weight of 1, the task will be calculated to have maximum relevance.

For example, the task named “Insert text into Post-it” might have linguistic cues such as `:indexes ("insert" .6 "append" .6 "assert" .5 "add" .3 "Post-it" .3 "create" -1 "delete" -1)`. When a user speaks an input command, and the voice box transfers the results of the voice recognition to the personal assistant, the personal assistant checks the input string for linguistic cues specified by each task in the library, and computes a score by using a MYCIN-like formula. If two cues a and b are present, the combined score is computed by the formula $a + b - ab$. Using the above example of the insert task, if an input string is “Insert ‘business plan’ into this Post-it”, the weight of the two linguistic “insert”

and “Post-it” will give a score of $(0.3 + 0.6 - 0.3 \times 0.6) = 0.72$ for the task. This will likely be the highest rated task for this input string, and the personal assistant agent will assume responsibility for this task, and automatically pass the “insert” dialog concept to the conversational engine.

This method of task selection is designed for receiving requests vocally and does not include any grammatical analysis in task selection. Spoken language is frequently grammatically incorrect, and voice recognition software may transcribe text incorrectly, therefore sophisticated grammar-based analysis will be error prone. Additionally, if certain words are frequently misinterpreted by the voice recognition machine, a quick fix might be add this word to the linguistic cue. For example, if “insert” is sometimes understood as “assert,” and if “assert” is not used in other tasks, it might be helpful to add it to the list of linguistic cues.

The *dialog concept* is a finite state machine that is handled by an agent’s conversational engine. It is through this mechanism that a personal assistant can request clarification if it is missing certain parameters. Depending on the task, the agent might make use of some rudimentary pattern matching to isolate the parameters in the command. For example, with the insertion task, a personal assistant will attempt to isolate any text after the word “insert” or “append” and the remove any prepositional phrases where “Post-it” or “note” is the object of the preposition. This would correctly isolate the text “Business plan” from the command “Insert ‘business plan’ into this Post-it”.

We also extended the OMAS language to include some functionality for managing implied references to components. The personal assistant, as described in the previous chapter, is informed of any action the users has performed in the environment (i.e. actions that are perform by another user-specific agent in the multi-agent system). The personal assistant can then construct a stack of *salient features*, or previously referenced components. These salient features are stored in *micro-context* concepts, that explain the general action that was performed when referencing these Post-it notes. For example, if the user selects two Post-its with a tap gesture, and deletes them using a vocal command, the IDs of the Post-its will be stored as salient features inside a micro-context of type “delete”. This information is then used during the conversational finite state machine to try to resolve which components are being referenced in vocal commands. Returning to our running insertion example, if the user has just created or selected a Post-it note, the personal assistant agent will find the IDs of these Post-it notes in the most recent micro-context in its stack, and then perform the insertion on these components. If the personal assistant finds only a recent delete action, i.e. a micro-context not relevant to the current action (one cannot insert text into a deleted Post-it), the personal assistant will ask for more information. Figure 5.2.7 illustrates the finite state machine diagram for the insertion dialogue.

It is worth mentioning that for the implementation of the management of the micro-context and salient features stack, we saw it helpful to encapsulate this functionality into another, separate agent. This *storage agent* is a *staff agent* to the personal assistant agent, per Negreponte's vision of a personal assistant as a digital butler [Negroponte 97], who is responsible for the interaction with its master, and delegates much of the details of execution to members of its staff. This allows a programmer to keep any one agent from becoming too large and unwieldy, by distributing functionality among other agents in the system. Similarly, we implement *post-it agents* as staff agents to handle the functionality regarding the Post-it note actions. The inclusion of new agent classes does not necessarily render incorrect the multi-agent society that we proposed with our Gaia methodology; staff agents could have very well been included in our analysis, but their use here is one of convenience in implementation more than a building block of structural design. Their relationships are isolated, and their existence does not impact the other members of their society, because they do not break any organizational rules of the multi-agent society.

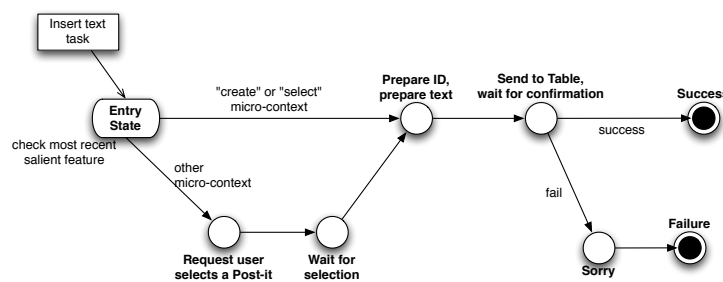


Figure 5.2.7: The dialogue concept for the insertion action as a finite state machine diagram.

5.2.5 Tablets and smartphones

Tablets and smartphones can be incorporated into an interactive space through any number of approaches. Some examples of the more interesting uses include the pointing and cursor extension techniques as implemented WILD room, where the smartphone's position and orientation is tracked by cameras allowing the user to point at distal components and interact with them through the smartphone's interface [Beaudouin-Lafon 11]. Rather than integrating tablets and smartphones into the interactive space as a new source of interaction, we approach them as personal and mobile computers which provide a new source of information.

The primary use case we see with smartphones and tablets involves a user showing and sharing personal media (e.g. photographs, sketches, videos) and referenc-

ing documents (e.g. specifications, standards, reports, charts). We implemented this by including personal Dropbox functionality on the interactive tabletop, which can be used by participants if they have installed the Dropbox application on their personal devices. This approach could be extended to network drives, or other kinds of cloud-based storage services such as Google Drive. This allows the interactive space to be used in flexible ways. For example, toward the end of a brainstorming session, a designer might use a sketching application on a tablet to sketch a few alternate product designs. She can save these to her Dropbox folder. Another participant might also want to share a few pictures of different prototypes that are on her smartphone from a similar previous project, and therefore she adds them to her own Dropbox folder. The sketches and photographs can then be immediately opened and displayed on the tabletop. They can be passed around, compared, organized, and shared with other participants, providing useful, easily shareable visual aids for the conversation.

We also developed a native application for the Android operating system that is designed for a more mobile user. It also includes a Personal Device Agent so that the application can communicate directly with the agent on the tabletop, without having to pass through the user's Dropbox. The application can be used to create and organize Post-it notes before a meeting. For an example use case, the moderator might schedule a meeting for a team to get together and use the ProjectTouch application to organize a new project plan for an upcoming project. A participant can use this Android application to create and organize the tasks that will be carried out by their department before the meeting begins. When the participant arrives in the interactive space, she can transfer the Post-it notes to the interactive tabletop by selecting them on the tablet and then clicking the create button. They appear on the interactive tabletop, and can be easily integrated into the project plan by the group.

5.3 TATIN-PIC infrastructure

In the previous chapter, we presented our organization of the multi-agent infrastructure with respect to certain design goals. Two of these design goals, regarding the independence of toolkits for the user interface (D1), as well as for the infrastructure (D2), were discussed briefly. Indeed, these are freedoms afforded by the Gaia methodology because it does not prescribe a specific implementation plan. In the previous section we have chosen a range of interface toolkits, such as MT4j, the Android SDK, and the Microsoft Speech API³. Using these toolkits has allowed us to focus the modular design of their applications, which were built explicitly

³<http://www.bing.com/dev/en-us/speech>

for the device they are running on. This section will present the implementation of the multi-agent society designed to have these devices working in concert.

5.3.1 Multi-agent systems

The multi-agent infrastructure is built using two different toolkits, JADE and OMAS. JADE, written in Java, was used to integrate agents directly into the toolkits which use Java. OMAS, written in Lisp was used particularly for its capacity to implement conversational intelligent agents. The use of these two systems together requires that we define a coherent communication and content language, so that agents of different type may freely exchange information. A previous version of this architecture was presented in [Moulin 11].

5.3.1.1 JADE Agents

The JADE agents which integrate directly into the tabletop program are one ProjectManager, one PersistenceManager, three ApplicationManagers and as many Workbench agents as there are participants. ApplicationManagers were defined by using separate class types for each application (BrainTouchTableManager, ProjectTouchTableManager, CauseTouchTableManager) at development time, while the Workbench agents are defined at runtime, using user-specific information such as their login name and ID. Contrary to our Gaia agent society in the previous chapter, we did not implement a UserIDer. As discussed in Zambonelli et al. [Zambonelli 03], the dichotomy between functionality that should be handled by the environment and the functionality that should be handled by the agents is not always clear. During our analysis I erred on the side of caution and encapsulated this functionality inside an agent role, but because there exists only a one-way information exchange from the UserIDer to the Workbench agents, the UserIDer can be implemented directly into the system without having to create a separate agent class. Having the UserIDer agent was helpful during the conception of this agent society because it allowed us to consider this functionality throughout the analysis and design of our multi-agent system, but ultimately, its actual implementation may be left to the discretion of the developers.

The JADE agents which integrate directly into the board program are one ProjectSupporter, one PersistenceSupporter, and three ApplicationSupporters, which were defined by using separate class types for each application (BrainTouchBoardSupport, ProjectTouchBoardSupport, and the CauseTouchBoardSupporter).

Each local system of agents runs inside of a JADE agent container. The multiple instances of these containers make up the JADE platform. Because there must be only one parent container inside a platform, we launch the main container on the tabletop and allow the board container to connect at startup. The JADE platform

provides a *directory facilitator*, which indicates the location and availability of all agents, and an *agent management system* which manages the creation and removal of agents and containers. These mechanisms allow the agents on different devices to exchange messages as if they were on the same platform. When passing messages that occur directly between one JADE agent to another JADE agent, JADE uses Java Remote Method invocation to pass objects directly to other agents. If a JADE agent communicates with other non-JADE agent, it must use Postman agents, and an open agent content language to do so. This will be discussed in Section 5.3.2.

5.3.1.2 OMAS agents

The OMAS multi-agent system manages the vocal interface with its user, and interfaces directly with the speech recognition and synthesis program using the Microsoft Speech API. This has been discussed in part when presenting the multimodal conversational user interface. With respect to our application of the Gaia methodology from the previous chapter, the OMAS system implements a Personal Assistant agent for each user. In our current implementation, these are defined in specific hardcoded classes which represents a static personal assistant for members of our laboratory.

As discussed previously, the personal assistant agent also has staff agents: the *post-it agent*, which contains behaviors encapsulating the actions that user can perform with virtual Post-it on the tabletop, as well as *storage agents*, which manages the storage and retrieval of past actions for use during the construction of queries which might include previously referenced components. There exists one set of these agents for each user.

OMAS uses its own message protocol for communicating among agents, which requires that several parameters and their values be defined, such as the nature of the message, the contents of the message, and any sought after skills of other agents (e.g. if an agent needs two numbers a and b to be multiplied, the nature of the message will be a “request”, with skill “multiply”, and contents a and b). OMAS agents then broadcast messages openly, allowing any agent with the necessary skill to respond freely to their request. Additionally, specific receivers may be identified in the parameters of a message for more directed requests.

5.3.2 Inter-platform communication

The infrastructure described in the previous sections has some advantages. Each multi-agent system has its own role. The JADE system helps to manage the activities across different instances of Java programs, and the OMAS system manages the activities with the vocal interface of participants. Internally, they use their own structure and messages dispatch functionality. The main disadvantage of this

infrastructure requires that we give special consideration to how messages may be exchanged between both systems. In answering this question, however, we can open our multi-agent system to a variety of different agents types.

5.3.2.1 Agent Communication Language

The exchange of information in multi-agent systems occurs through messages. The *communication language* defines the structure of these messages. For the purpose of sharing information among agents of different types, the Foundation for Intelligent Physical Agents (FIPA) defined an agent communication language standard. The structure of the FIPA communication language includes a list of keyword and value pairs which define important information relevant to the nature of the message, such as the name of the sender, the name of the receiver, the date, the content, and the performative. The performative, based on Searle's speech acts [Searle 69], refers to the type of message, e.g. whether it is a request for information, an answer, or an error message. The FIPA specifications do not address the implementation of message passing systems or protocols, leaving such decisions to the developers of multi-agent systems.

For example, the JADE multi-agent system uses its own FIPA-compliant message passing system, and encapsulates the message inside a Java object. If the sender agent is located in the same container as a receiver agent, the message is passed as an object event. If the sender agent is located on a different container than the receiver agent, JADE serializes the java object and uses Java's Remote Method Invocation (RMI) libraries to transfer the object to the agent. Similarly, OMAS uses its own, Lisp-friendly internal communication language for formulating messages.

In our case, we wish to transfer messages between a JADE container and the OMAS platform. Because we cannot use Java's object passing functionality, we design our own serialized message constructs and passing mechanisms for agent communication. We design this language to be as open and accessible as possible, first by basing our constructs off the FIPA language specifications and also by formulating messages as JSON⁴ objects. The content of a message is considered as a JSON object whose properties are in accordance with the FIPA standard. Libraries providing JSON writers and parsers are available for numerous programming languages, which facilitate the use of JSON objects during development. JSON objects are essentially a list of keyword/value pairs, where the values can be strings, numbers, objects, or arrays. Table 5.2 lists the keywords and their values used in our inter-platform communication language. We use a subset of performatives from the FIPA language. Their integer IDs are taken from the JADE

⁴<http://www.json.org>

API.

Table 5.2: Keywords used for JSON object messages on the TATIN-PIC platform.

Parameter	Value type	Description
sender	string	Name of the sender
receiver	array	Name(s) of the receiver(s)
profile	string	Profile of the receivers
date	long	The date the message is sent
performative	int	The ID of the “type” of message
language	string	language of message
ip	string	IP of the sender
port	string	The port from which the message leaves
id	string	ID of message
task-id	string	ID of the task that corresponds to the message
content	JSON object	An object describing the actual content of the message

The need to repackage and transfer inter-platform messages introduces a need for new requirements in the infrastructure. To this end, we implement two new agents types, a *JADE-OMAS transfer agent* (written in JADE), and an *OMAS-JADE transfer agent* (written in OMAS). The J-O transfer agent will repackage JADE message objects into a JSON string and transfer these to the O-J transfer agent which will repackage them into OMAS messages, and vice versa. Both transfer agents behave slightly differently. The O-J agent is more proactive, and intercepts the broadcasted messages, while the J-O agent must be directly addressed. For example, if a Workbench agent wishes to send a message to its master’s personal assistant agent, it must direct its message to the J-O agent who will handle the rest of the processing. The O-J agent, on the other hand, listens to any message that is broadcast on the OMAS platform, and if the agent ID belongs to a JADE agent, the O-J agent will package and transfer the message to the JADE platform.

5.3.2.2 Agent Content Language

The content is also a JSON object which is packaged inside the JSON message. The content of a message contains information and data pertaining to the actual action that agents will perform. Table 5.3 details the four parameters that are used when forming messages.

With regard to the applications we have designed that rely on the Post-it note metaphor, we have implemented ten different actions that a meeting participant can perform:

Table 5.3: Keywords used for JSON `content` structure.

Parameter	Value type	Description
<code>action</code>	string	A value corresponding to a skill or a behavior
<code>args</code>	JSON object	List of parameters pertaining to the action that the agent will perform
<code>answer</code>	JSON object	Information or data structure in response to a request
<code>error-contents</code>	string	Explanation if an error occurs

1. `create` : for creating a Post-it or a group.
2. `delete` : for deleting a Post-it or a group.
3. `edit` : for replacing the textual content of a Post-it. NB: We do not explicitly consider deleting content, as the edit action can be performed with an empty string.
4. `insert` : for appending textual content at the end of a Post-it note string.
5. `add` : for adding a Post-it or a group to a group.
6. `remove` : for removing a Post-it or a group from a group.
7. `select` : for selecting a Post-it or a group.
8. `unselect` : for unselecting a Post-it or a group.
9. `unselect-all` : for unselecting all the previously selected Post-its or groups.
10. `highlight` : for triggering a flashing behavior for a Post-it.

These actions represent requests (or *userEvents* that were used in the analysis in the previous chapter) that can be made *from* the personal assistant *to* the workbench agent. Each of these actions will formulate their arguments differently. The parameters that are used when describing the `args` structure are in Table 5.4. Note that not all actions require the use of all `args` parameters. For example, `create` requires `type` (depending on if a Post-it or group is begin created), `value` (the content of the Post-it or the label of the group), and `ref` (list of IDs of Post-its to add into the group, if a group is being created). `add`, on the other hand, requires only `ref` (list of IDs to add) and `to` (group to add them to).

There are many different kinds of messages that an agent can send. For example, for its own purposes, a personal assistant agent may request the entire model of the document that is open on the tabletop (`get-model`) or request all its master's currently selected item (`get-selected-items`). The personal assistant agent is also informed of *userEvents* that occur on the tabletop. Therefore, the personal

Table 5.4: Keywords used for JSON `args` structure.

Parameter	Value type	Description
type	string	Either “postit” or “group”
value	string	Content relate to the
ref	string array	List of Post-it or group IDs
to	string	Explanation if an error occurs

assistant agent can receive messages of `selected-item` and `unselected-item`. These actions are then stored by the *store agent* for future use during the processing of vocal commands.

Additionally, as illustrated in Figure 5.2.6, the personal assistant agent must also inform the tabletop of the results of the voice recognition module so that the user can receive the appropriate feedback on the tabletop. This is done through the `echo` action, while the controls for the microphone are done through the `mic-on` and `mic-off` actions.

5.3.2.3 Message passing example

For an example of these messages in action, imagine the participant speaks a vocal command such as “Create a Post-it note with content ‘Develop business plan.’” The series of ensuing message exchanges is illustrated a sequence diagram in Figure 5.3.1 and Figure 5.3.2.

After receiving the query from the voice box, the personal assistant agent will parse the message, detect the relevance of the “create Post-it” task, and pass the information to the Post-it staff agent. Note in Figure 5.3.1 that the personal assistant agent can specify a “pattern,” indicating to its staff agent exactly what format it would like the information to be returned in.

The Post-it staff agent will then formulate a message using OMAS’s internal communication language, with its master’s JADE workbench agent added as the intended recipient. The personal assistant agent and staff agents know the name of the workbench agent because it has been parameterized as “<login>-workbench” in the JADE system.

Once the message is broadcast on the port the OMAS platform is using, the O-J transfer agent will recognize that the intended recipient is a JADE agent. The transfer agent will repackage the string in JSON, and the content of the message will be:

```
{"action" : "create", "args" :
{"type" : "postit", "value" : "Develop business plan"}}
```

The O-J transfer agent will send the message to the J-O transfer agent which will repackage the message for the JADE platform and send it to the appropriate

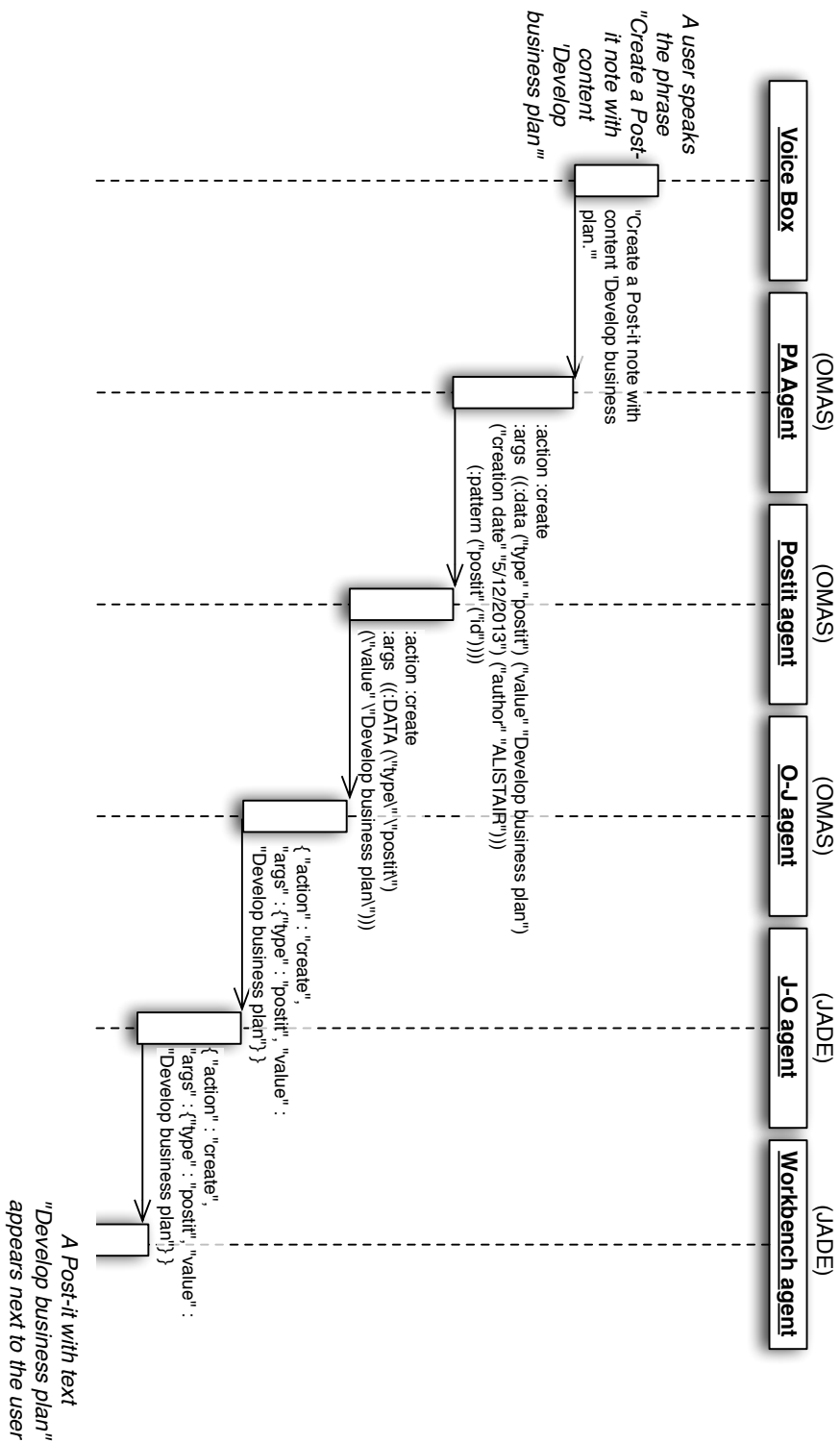


Figure 5.3.1: A series of messages in the multi-agent system to process the vocal request "Create a Post-it note with content 'Develop business plan'".

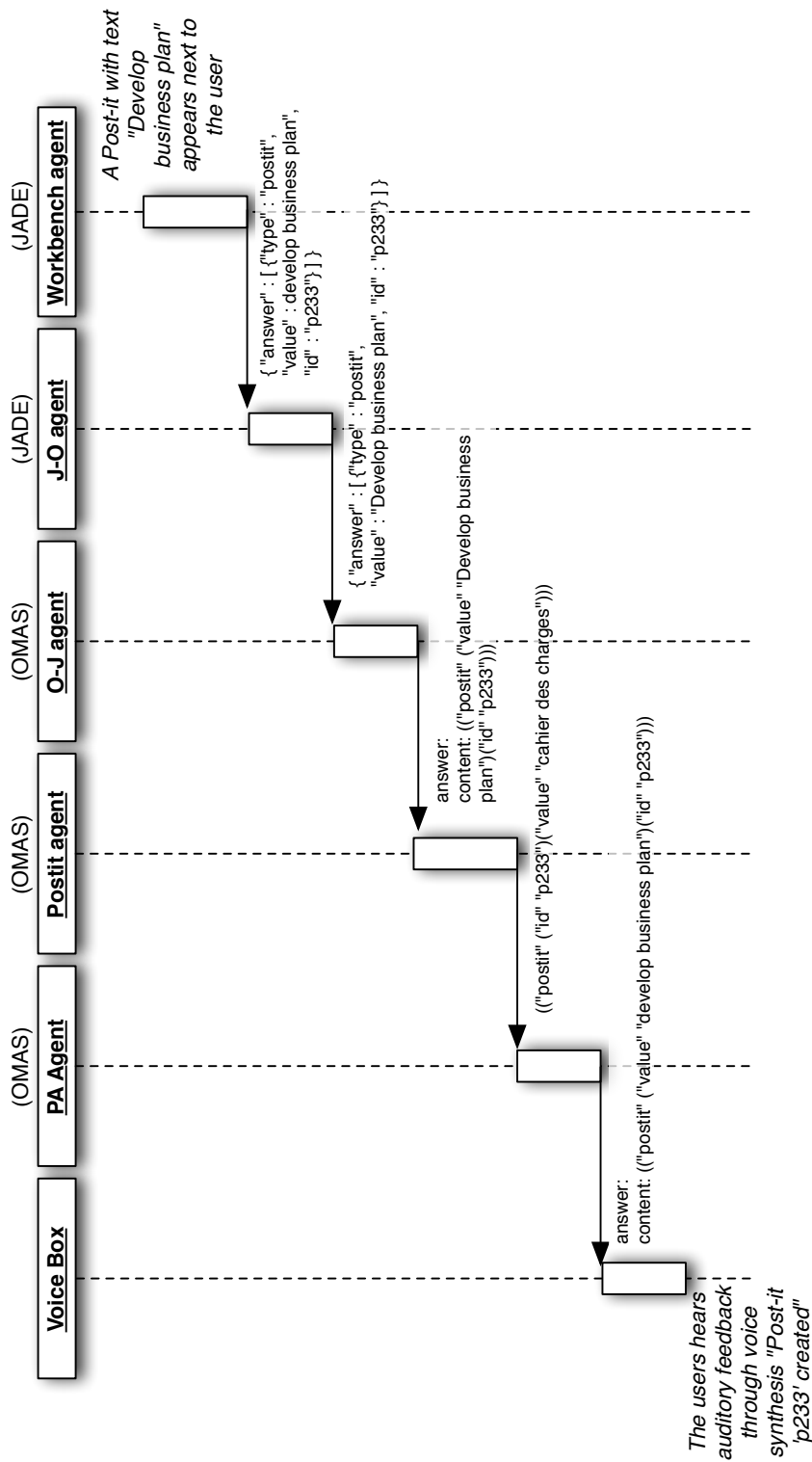


Figure 5.3.2: A series of messages in the multi-agent system to process the response to the vocal request "Create a Post-it note with content 'Develop business plan'".

workbench agent. The workbench agent will create the Post-it note, by adding it to the model of the *activeDocument*, and in doing so will generate a unique ID for it. Now the workbench agent must return a message confirming the action was completed and also include the ID of the new Post-it. It will construct a message and send it directly to the J-O transfer agent, who will package it in JSON and then transfers it to the O-J agent. At this point the content of the message will be:

```
{"answer" : [ {"type" : "postit",  
  "value" : "Develop business plan", "id" : "p233"} ] }
```

The O-J agent will transfer this to the Post-it staff agent, who will prepare the response for the personal assistant agent according to the pattern used in the original request. The personal assistant agent will respond to the user using voice synthesis on the voice box, providing auditory feedback that the Post-it has been correctly created. It will also ask its storage staff agent to store the micro-context (“create”) and the salient feature (the value of “answer”) (not pictured).

5.4 Conclusion

This chapter has presented the hardware devices, the user interface and the multi-agent infrastructure of the interactive space. This implementation was necessary to support the co-located collaboration of groups of up to eight participants (with currently only four vocal headsets available).

The resulting implementation underwent many iterations and changes throughout its development as our conceptualization of the interactive space matured through practice and informal experimentations. At the same time, there were also several formal user experimentations and evaluations that were made possible by our implementation. These will be presented in the next chapter.

Chapter 6

Evaluation

The implementation of the TATIN-PIC interactive space was carried out over a period of three years and, during this time, numerous design sessions involving over 90 different participants were conducted. The majority of this chapter will present three of the experiments which focus on brainstorming, project planning, and causal analysis. The final section is dedicated to the multi-agent system and a comparison to related work on multi-device environments.

6.1 User Evaluation

The user evaluations of the TATIN-PIC interactive space were designed to accomplish three main objectives:

1. To evaluate the general usability of the interactive space and interaction techniques.
2. To understand how the interactive space impacts the completion of specific preliminary design activities (i.e., brainstorming, project planning, causal analysis) when compared to traditional environments.
3. To understand how the interactive space impacts group dynamics and communication when compared to traditional environments.

These objectives are addressed through three different evaluations. The first is a within-subjects comparative evaluation of a 30-minute brainstorming task in an interactive and traditional space. The second is a between-group comparative evaluation of an approximately 4-hour project planning session in an interactive and traditional space. The third evaluation involves a causal analysis case study that is part of a larger less-structured ethnographic observation with participants

using the interactive spaces for extensive periods of time for their own personal class project. These evaluations were conducted over a three-year period at various points during the implementation of the interactive space, so that the results could still have an influence on the design of the system.

These objectives are not meant to be mutually exclusive, but rather serve to orient the analysis of the evaluations. The third objective will play a part in understanding the second, and the second will play a part in understanding the first. This will be discussed in the following subsection.

6.1.1 Discussion of evaluation objectives

Our data gathering techniques focus on a wide range of criteria, and before going further, it is important to provide an introduction to these criteria. These experiments are approached as usability evaluations, and therefore, they concern usability goals. Per Rogers et al. in [Rogers 11], there are six general usability goals: effectiveness, efficiency, safety, utility, learnability, and memorability. Because it is often infeasible to design an experimental protocol and a corresponding interactive prototype which can measure these six usability goals together, experimental protocols often focus on a subset. In our case, the primary concern is the *effectiveness* of the system, and, to a lesser extent, its *efficiency*, as well as the *utility* of some specific functionality and interaction techniques in the system.

Orthogonal to these usability goals are user experience goals, which are less concerned with performance than the general experience a user might have when interacting with the system. With regard to preliminary design, the user experience is important because it can impact the dynamics of the group, which may influence the group's performance. Therefore, three primary user experience goals were identified. The interactive space should be *enjoyable*, *motivational* and *playful*. Additionally, *creativity* is an user experience goal that is considered in the context of the brainstorming activity. These usability and user experience goals are referenced throughout multiple questionnaires in each of the evaluations for the purpose of gathering subjective data from the participants regarding these criteria. Therefore, the first objective will be met by measuring the subjective evaluation of the effectiveness, efficiency and utility of the system, as well as the degree to which it is enjoyable, motivational, playful, and conducive to creativity.

These criteria will also assist in meeting of objective 2. Moreover, the effectiveness of a group can be measured by their performance at the activities in the experiment, i.e. brainstorming, project planning, causal analysis. Each of these activities will have their own performance metrics associated with them. For example, brainstormings are often evaluated using the number of ideas generated ([Hilliges 07, Buisine 12]) and project planning activities can be compared by evaluating the project flowchart that is produced. These metrics are useful, but they

do not provide a complete picture of performance. It is difficult to say that one brainstorming's affinity diagram is better than another: the categorization of ideas that occurs when building an affinity diagram from subjective results is difficult to assess, and the resulting diagram might be less useful than the spirited debate and exchange of information that might have occurred during its construction. This is why objective 2 will often overlap with objective 3.

The purpose of objective 3 is to understand the quality of co-located collaboration in the interactive space. When discussing the related work in Chapter 2, I highlighted three important aspects of collaboration: communication, coordination, and interpretation. Coordination and interpretation can be approached qualitatively, using general observations, but these processes involve making assumptions about the users' internal cognition. With coordination, the exact process through which participants align themselves into natural roles or enforce organizational roles is often not directly expressed by the participants and is subject to interpretation on the part of the observer. Shared interpretation is especially important particularly with regard to the process of common objectification, where participants assign and share meaning through the construction of shared artifacts. Again, this is difficult to trace throughout a meeting, and it would involve making assumptions regarding how the participants interpret shared ideas and reach conclusions.

In contrast to coordination and interpretation, verbal and nonverbal communication is easily observable and can be measured during experimentation. In Chapter 2, I described communication as a means through which coordination is managed and interpretations are shared in group settings. Measuring communication provides an entry point to a deeper understanding of collaboration and group dynamics that occur during an activity. There also exists a simple model for ideal group communication: Woolley et al. note that equal of participation from all participants is correlated with high performance scores across a number of activities [Woolley 10]. *Equity of participation* provides an objective and quantifiable measure for effective group dynamics during activities. It has previously been used to analyze group dynamics in collocated collaboration [Buisine 12, Martínez 11, Marshall 08].

Table 6.1 shows the different metrics and data gathering tools that are used in the three user evaluations to respond to the three objectives.

6.1.2 Brainstorming Evaluation

Brainstorming sessions are group creativity activities which allow a team to explore the solution space of a problem. This experiment was designed to compare the results of brainstorming sessions conducted in the control condition, on a conventional table with Post-its and pens, to brainstorming sessions on the TATIN

Table 6.1: The different metrics and data gathering tools used in the user evaluations to meet the objectives.

	Objective 1	Objective 2	Objective 3
Evaluation 1	Subjective questionnaire	<i>(brainstorming)</i> Post-it generation count, subjective questionnaire	Equality of communication metric, subjective questionnaire
Evaluation 2	Subjective questionnaire	<i>(project planning)</i> Time to complete activity, duration of resulting project plan, subjective questionnaire	Subjective questionnaire
Case study	Subjective questionnaire	<i>(causal analysis)</i> Post-it generation count	General observation

table, an interactive tabletop system. The experiments were designed for the observation of changes in group dynamics, the quality of the performance at the task at hand, and the subjective evaluation of the testers regarding the user interface in general.

6.1.2.1 Participants

These experiments included a total of 48 participants divided into 8 groups of 6 people.¹ The participants can be divided into two categories: 34 engineering students aged 20 to 25 years and 14 non-students aged 24 to 50 years. These two categories were spread as evenly as possible throughout all groups. All participants spoke french as a common language, and therefore the experiments were conducted in French.

6.1.2.2 Environment and Apparatus

Because the experiments were conducted over a three-year period to help inform the design of the interactive space, the implementation from this experiment differs from the version presented in the previous chapter. This implementation also includes two additional functions designed for increasing the creative user experience: the web searches of images during the brainstorming, and the ability of a user to slide Post-its under their virtual keyboard. The ability to hide ideas under the keyboard assists with evaluation apprehension, allowing users to hide ideas until they are ready to share with the group.

¹In reality, the evaluation was actually conducted with 9 groups in total, for the purpose of removing one group from analysis if this system experienced unrecoverable crashes, if the video recording setup failed, etc.



Figure 6.1.1: BrainTouch, v0.9.

At the time of the evaluation, the interactive space used only the interactive tabletop and a preliminary version of the brainstorming application, BrainTouch v0.9 (Figure 6.1.1). This version of BrainTouch uses six virtual keyboards that were distributed around the tabletop. Each user can enter text into virtual keyboard, which appear in a text area immediately above the keyboard. At this point, they may generate either a text Post-it by pressing the return key, or an image Post-it (referred to as a Polaroid) by pressing the button "FLICKR." Upon pressing this button, the application uses the entered text to search for images using the Flickr® web-service. After a moment, a series of fifteen images corresponding to the text appears in front of the user. By tapping on one of these images, the user can generate the image Post-it (or, if the user finds no images suitable, the user can close the image selection window.) Post-its and Polaroids can be moved, re-sized, and reoriented with standard one-finger or two-finger drag gestures. Though this version of BrainTouch did not use a physics engine to control the behavior of Post-its and Polaroids, they could still be thrown with basic inertial effects. Post-its and Polaroids can also be deleted by dragging them to one of the trash icons placed at the four corners of the table. The grouping technique discussed in Chapter 3 was not developed at this time; instead, users can group Post-its using a lasso technique, by dragging a finger around a collection of Post-its or Polaroid. Groups of Post-it can be de-grouped with the same gesture. Adding a label to a group is done by simply, creating a new Post-it note, expanding it, and placing

it over the group. In BrainTouch v0.9, the system does not recognize this as a labeling, but this is just a spatial positioning that is visually understood by the user.

For the control condition, the same tabletop was used, covered with a sheet of kraft paper, to preserve as many environmental conditions as possible. Blocks of Post-it notes and pens were distributed to all participants so that they may write or sketch their ideas. The participants were instructed to label and circle groups by drawing directly on the kraft paper.

6.1.2.3 Experimental Design

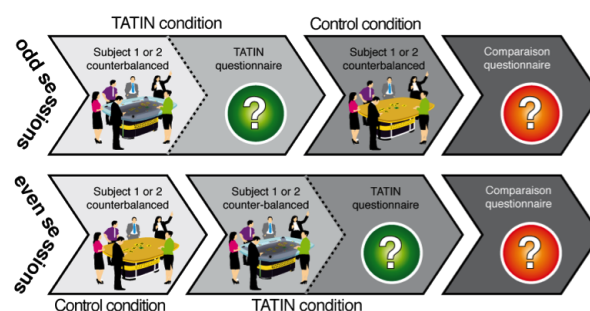


Figure 6.1.2: Experimental design with counterbalancing of control and experimental conditions and brainstorming topics.

Each of the eight experiments lasted 3 hours and was divided into two phases, one for the conventional Post-it session (control condition) and one for the interactive tabletop session (TATIN-PIC condition). The brainstorming topics were the design of a shared calendar for a family and the design of a Swiss Army knife for the twenty-first century. To reduce bias in the observations, the sessions were counterbalanced with the brainstorming topics and the brainstorming apparatus (Figure 6.1.2). Three different questionnaires are used to gather data on (1) the participants and their level of expertise with multitouch surfaces, (2) their subjective evaluation of the interactive tabletop, and (3) a subjective comparison of the two conditions with regard to collaboration and group dynamics. The experimental design also included a general introduction to the experiment, and an introduction to the brainstorming technique, with emphasis on the rules of brainstorming as provided by Alex Osborn who first wrote about the method in [Osborn 57]. These rules include withholding criticism, focusing on the quantity of ideas rather than the quality, welcoming unusual ideas and combining or extending the ideas of others. These rules were also posted on the walls of the room to serve as a reminder to participants during the session.

6.1.2.4 Activity

Immediately before the activity began on the interactive tabletop, a brief three-minute demonstration of the interactive tabletop presented the functionality of the system to the participants. The brainstorming sessions were organized into three separate phases:

- 8 minutes of individual idea generation: users write ideas one at a time, using only one Post-it or one Polaroid per idea.
- 10 minutes of pooling of ideas: each participant presents their ideas to the group, one at a time.
- 12 minutes of categorization of ideas: the group conducts a semantic cleaning (deletion of doubles) and then a semantic categorization using an affinity diagram. Every Post-it must belong to a group and every group must have a label.

One experimental moderator monitored the time for the participants, and issued 2-minute warnings when the end of a phase was approaching.

Indicators of strong performance in this activity would be a high production of ideas from the group and equitable participation from group members. According to several social science studies of brainstorming sessions, one of the most detrimental factors to brainstorming is *social loafing* [Hilliges 07, Buisine 12]. Social loafing occurs when a participant is less involved in the brainstorming activity, producing less ideas and less opinions regarding how ideas may be categorized. Social loafing can occur for a number of reasons, notably evaluation apprehension (fear that their ideas will be criticized by group members) and free-riding (relying on the production yield from other users to supplement their own).

6.1.2.5 Data gathering

The experiments were filmed with a setup consisting of three cameras recording from three different perspectives, with a microphone to record audio from the meeting, and a video screen capture of the interactive tabletop in the TATIN-PIC condition (Figure 6.1.3). The videos were analyzed using a coding scheme for verbal and nonverbal communication. These are in part inspired by the coding scheme proposed in Buisine et al. [Buisine 12].

The coding scheme comprises the following five categories of verbal communication:

- Opinions: When a user expresses an idea or opinion to the group.



Figure 6.1.3: A brainstorming session during the control condition (left) and a brainstorming session during the TATIN-PIC condition (right).

- Questions: Questions can either indicate a request for action (e.g. "Would you pass that Post-it note to me?").
- Request for information (e.g. "What do you think of calling this category 'sharing' "?).
- Responses: Speech that serves toward answering a particular question.
- Request for help: This category is used to classify requests for clarification with either the activity of brainstorming itself or instructions of how to use with interactive tabletop. Such requests can be addressed to the moderator of the session or the teammates.
- Off-task talk: Any dialogue outside of the brainstorming activity and the interactive tabletop.

Opinions, questions and responses are what make the core of meaningful exchanges in the brainstorming phases.

The coding scheme also tracked non-verbal communication with the five following behaviors (discussed in Chapter 2 on communication):

- Handling gestures: Gestures where one performs an operation on an object that changes the state of that object. This could be manipulating a Post-it (virtual or tangible) to move and reoriented it, or writing on the Post-it with a pen or an onscreen keyboard.

- Deictic gestures: Also referred to as pointing gesture, these are frequently done with hands, fingers and gaze to identify either real or imaginary people, objects, and directions
- Adaptor gestures: Often unintentional, adaptor gestures can be identified as self-touching, such as scratching an or crossing arms, or object-oriented, such as playing with a pencil.
- Metaphoric gestures: Such gestures are used to illustrate speech by forming an analogy between a concept and a particular action or pose done with the body. An example would be of a speaker describing their layout of a house and drawing the shape of the rooms with their finger in the air.
- Punctuator gestures: Punctuators are the only one of the five types of gestures that must accompany some form of verbal communication and can be performed by either the speaker or the listener. Punctuators will help carry the rhythm of the conversation. Examples of these gestures are nodding the head after hearing each item of a list, or open palms and raised eyebrows at the end of an assertion to indicate turn-taking (also known as beats and batons, respectively).

It is important to note that because nonverbal communication is defined as any gesture, pose, or gaze made by the body, a subject will never "stop" gesturing, and is always doing one of these five gestures. From the sixteen brainstorming sessions, approximately four hours of traditional brainstorming and four hours of interactive tabletop brainstorming in total were collected. Four coders were recruited and trained to use the coding scheme. Each coder was given the same 30-minute brainstorming session to code, and the two coders with the highest inter-judge agreement score were retained (Cronbach's alpha, 0.70). These coders then annotated four of the eight sessions, transcribing all speech, marking each time a Post-it note was created, and categorizing communication according to the coding scheme for 24 different participants. The raw data is composed of occurrences and duration for each observed communication criteria.

The comparative subjective questionnaire included 32 questions divided into five sections concerning environment, communication, collaboration, creativity, and motivation. The following variables were collected in the form of 7-point Likert scales: ease, effectiveness, and enjoyability of brainstorming in each environment; ease, effectiveness, enjoyability of communication in each condition; ease, effectiveness, and enjoyability of coordination in each condition; creativity in each of the environments; and the user's adherence to the following five statements regarding their motivation "I wanted to do well", "I was invested in the results", "I tried to do my best", "I would like to know how well I performed", "I would like to

know how well others performed”. Furthermore, the questionnaires provide space so that users could make optional qualitative comments.

The subjective questionnaire regarding the interactive system included 32 questions focused on different aspects of the interface and interaction techniques. The following variables were collected in the form of a 7-point Likert scales: learnability of the software; amount of mistakes and errors when using the software; utility of the software; usability, effectiveness, enjoyability, and playfulness of creating virtual Post-it notes; usability, effectiveness, enjoyability, and playfulness of manipulating virtual Post-it notes; usability, effectiveness, enjoyability and utility of the virtual keyboards; usability, effectiveness, enjoyability and utility of the throwing gesture with Post-it notes; usability, effectiveness, enjoyability, utility, and creativeness of using the images (Polaroids) in the brainstorming; overall promise of the system. The questionnaires provided space so that users could make optional qualitative comments.

Likert scale results of the questionnaire were analyzed quantitatively and comments were analyzed qualitatively.

6.1.2.6 Results and Analysis

Performance: Performance in brainstorming activities can be evaluated by measuring the participation levels of the group. I measured participation using speech times, nonverbal handling gestures, idea generation, and questions from the subjective questionnaire evaluating user’s motivation and involvement.

For the four control brainstormings and four experimental brainstormings with complete verbal and nonverbal analysis, I calculated the speaking time, and occurrences of handling gesture events for each participant in each condition. For these two criteria, I also calculated the inequity index I for each participant using the formula found in [Buisine 12],

$$I = \left| \frac{1}{N} - \frac{O_i}{\sum_{i=1}^N O_i} \right| \times 100 ,$$

where N is the number of participants in the group, and O_i is the number of contributing events for each individual in the group. This inequity index assumes that the ideal scenario is one where all participants contributed equally ($1/N$). A number close to 0 indicates nearly complete equity in participation.

Table 6.2 contains the results of this analysis. Mean times in both conditions are approximately 3 minutes for each participant (which is to be expected for a 30-minute brainstorming where the first 8 minutes are dedicated to individual work). Though a higher number of manipulations occurs in the control condition, the two conditions are similar in each of these four criteria and no statistically significant comparison can be made between the two conditions.

Table 6.2: The mean (m) and standard deviation (SD) for speaking time (in seconds), speaking inequity and handling gesture occurrences and handling gesture inequity for all participants in each of the conditions are calculated. The results for each of the criteria are also evaluated using an unpaired Student's t-test, which is presented in the last four columns: degrees of freedom (DOF), F-values (F), corresponding p-value, and statistical significance (sig).

	Control		TATIN-PIC		DOF	F	p-value	sig
	m	SD	m	SD				
Speaking time (seconds)	186.1	105.0	189.1	103.1	46	0.103	$p=0.92$	no
Speaking inequity (I)	6.2	3.7	6.1	4.9	46	-0.012	$p=0.99$	no
Handling gestures (occurrences)	58	43.7	42	25.6	46	-1.6	$p=0.12$	no
Handling gesture inequity (I)	3.2	2.4	3.4	2.4	46	1.49	$p=0.78$	no

The comparative subjective questionnaire asked participants to rate on a 7-point Likert scale the applicability of the five personal statements regarding their involvement and motivation for each of the conditions. The mean and standard deviation of these results are presented in Table 6.3. Each of these statements are rated to be more applicable in describing the user's personal experience in the TATIN-PIC condition, and t-tests indicate a statistically significant difference between the two conditions.

Table 6.3: Results from the 7-point Likert scale questions regarding the participant's motivation and involvement, and the results of a paired Student's t-test for the five statements.

	Control		TATIN-PIC		DOF	F	p value	sig
	m	SD	m	SD				
"I was motivated to do well"	6	0.7	6.4	0.7	46	-4.29	$p<0.01$	yes
"I was invested in the results"	5.5	1.1	5.9	1.1	45	-3.14	$p<0.01$	yes
"I tried to do my best"	6	0.9	6.4	0.7	47	-2.86	$p<0.01$	yes
"I would like to know how well I performed"	5.2	1.8	5.8	1.8	45	-3.34	$p<0.01$	yes
"I would like to know how other performed"	5	1.9	5.4	1.9	45	-3.93	$p<0.01$	yes

Groups in the control condition produced more ideas than groups in the TATIN-PIC condition ($\bar{x}_c = 67.3$, $\sigma_c = 3.14$; $\bar{x}_T = 51.6$, $\sigma_T = 3.16$; $t(46) = 5.38$, $p <$

0.01). In order to better visualize this result, each participant in every session was ranked from 1 (lowest production count) to 6 (highest production count). The average idea production for the eight first-ranked participants in the control condition and the eight first-ranked participants in the TATIN-PIC condition was calculated. Identical calculations were made for the participants of every rank. These results are illustrated in Figure 6.1.4. According to the t-tests used to compare the participants of each rank, there was a statistically significant difference for all ranks except for the participants who produced the most (rank 6).

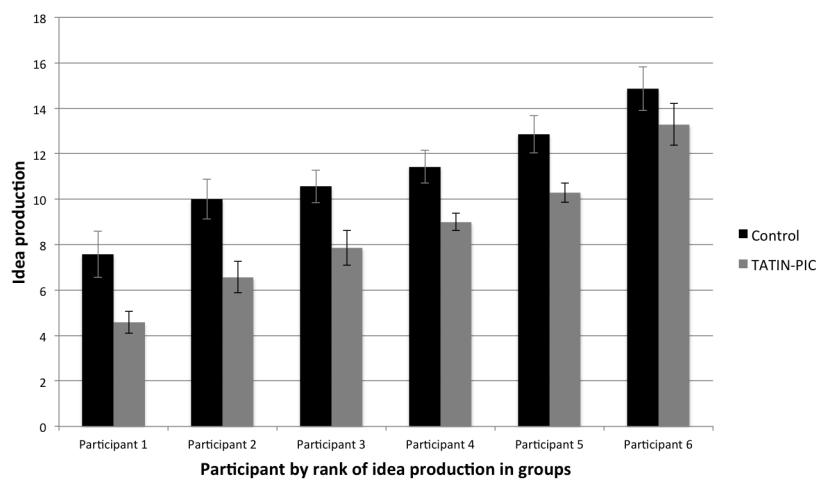


Figure 6.1.4: Idea production for the least-productive participants to the most-productive participants for each of the conditions.

Group dynamics: Because certain aspects of collaboration and group dynamics were used to measure the group performance, there is a fair amount of categorical overlap between this section and the previous one. Still, this section presents additional information regarding the participants' subjective evaluations of the ease, effectiveness, and enjoyability for communication and group coordination in both conditions on a 7-point Likert scale in the comparative questionnaire (Table 6.4). Statistically significant differences occur for the *effectiveness of coordination* (rated higher in the control condition) and the *enjoyability of coordination* (rated higher in the TATIN-PIC condition).

Subjective analysis of the environment: The comparative questionnaire also asked users to rate the environment regarding usability, effectiveness, enjoyability, playfulness, and creativity (Table 6.5). Statistically significant differences occur between the control condition and the experimental condition for the usability and playfulness, indicating that the pen-and-paper brainstorming is much easier to use, while the interactive tabletop is much more playful. The effectiveness, enjoyability, and creativity of the two conditions were rated to be equivalent.

Table 6.4: Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.

		Control		TATIN		DOF	F	p value	sig
		<i>m</i>	SD	<i>m</i>	SD				
Communication	ease	5.9	1.1	5.4	1.4	47	1.64	$p=0.11$	no
	effectiveness	5.8	1.0	5.5	1.3	46	1.02	$p=0.31$	no
	enjoyability	5.7	1.2	5.8	1.0	46	-0.62	$p=0.54$	no
Coordination	ease	5.3	1.2	4.9	1.6	47	1.19	$p=0.24$	no
	effectiveness	5.6	1.0	5.1	1.2	46	2.19	$p<0.05$	yes
	enjoyability	5.4	1.2	5.8	1.2	47	-2.23	$p<0.05$	yes

Table 6.5: Results from the comparative subjective evaluation on criteria for the overall interactive space, with corresponding paired Student's t-test between the two conditions.

	Control		TATIN		DOF	F	p value	sig
	<i>m</i>	SD	<i>m</i>	SD				
Usability	6.1	0.9	5.0	1.3	45	4.3	$p<0.01$	yes
Effectiveness	5.7	0.9	5.5	1.1	47	1.1	$p=0.29$	no
Enjoyability	5.5	1.0	5.9	1.3	47	-1.6	$p=0.12$	no
Playfulness	4.7	1.4	6.3	0.9	47	-6.7	$p<0.01$	yes
Creativity	5.4	1.2	5.3	1.1	47	1.0	$p=0.34$	no

After the TATIN-PIC condition, participants were also asked to complete a subjective questionnaire which included 32 7-point Likert scale questions on various aspects of the interactive space. For space reasons, I only include the top 5 rated items and the bottom 5 rated items in Figure 6.1.5 and Figure 6.1.6, respectively. The complete list of subjective usability criteria is provided in Appendix D.

The throwing gesture for passing Post-it notes among participants was rated to be highly useful and enjoyable, and manipulating Post-its notes (moving, re-orienting, and resizing) was also evaluated to be relatively playful. The highly ranked utility of the keyboard is to be expected, as it is the virtual component with which all Post-it notes and Polaroids are created. The learnability of the system also ranked highly, which is interesting considering that the experimental protocol only allowed for two minutes of practice before the beginning of the activity. This is a testament to the simplicity of direct manipulation techniques as well as the straightforward application design of BrainTouch v0.9.

The lower ranked usability issues call into question the two pieces of functionality that were included in the application for the purpose of encouraging creativity. From the perspective of the users, the ability to hide items under the keyboard and the use of images do not have an impact on creativity. The low-ranked utility and enjoyability of hiding Post-its under the keyboard indicate that another approach should be taken for providing optional privacy during idea generation, and the low-ranked effectiveness of creating images indicate that there may be usability issues present in how this feature was implemented.

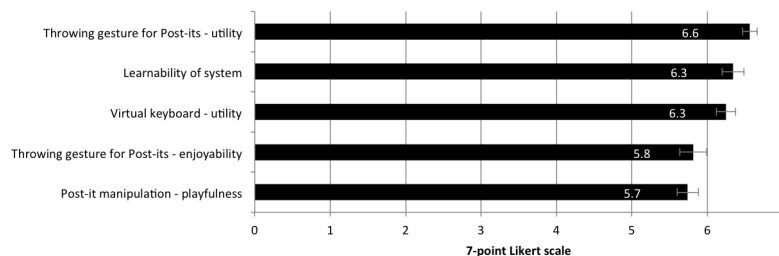


Figure 6.1.5: Top 5 ranked items from the general usability of the BrainTouch v0.9 application.

6.1.2.7 Discussion

The design of such an experimental protocol is difficult because of the radically different nature of the media used in the two conditions. The design of BrainTouch v0.9 is purposely conservative to respect its traditional counterpart. Still, it may be worth highlighting the three largest differences in terms functionality between the

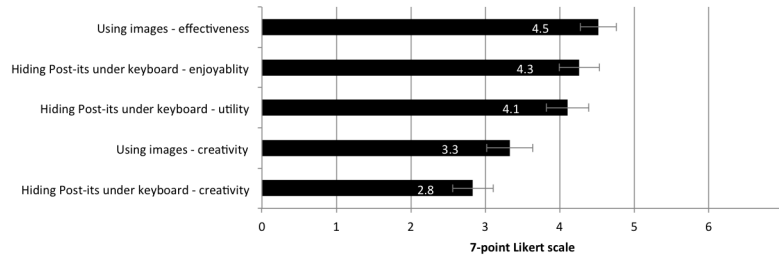


Figure 6.1.6: Bottom 5 ranked items from the general usability of the BrainTouch v0.9 application.

two conditions was the use of virtual keyboards instead of pens, the ability to add images from a web search (which still can be compared to the user's ability to draw images on Post-it notes) and the ability to hide Post-it notes under the keyboard (which could not be done easily in the control condition). The control condition and the TATIN-PIC condition were comparable in terms of basic functionality and the use of metaphors, but due to the radically different nature of media used, a significant difference in group dynamics was hypothesized.

Therefore, perhaps the most striking result from the experiments is the similarity between the two conditions particularly with regard to speech time, speech inequity, handling gesture occurrences, and handling gesture inequity. This is encouraging in the sense that the new media was not rejected by the participants. Almost paradoxically, participants produced less ideas but reported higher levels of involvement and motivation in TATIN-PIC condition. Similarly, coordination was rated to be less effective but more enjoyable in the TATIN-PIC condition. These relationships can be explained by other subjective criteria used in the evaluation of the interface: the comparative questionnaire also revealed that the TATIN-PIC condition was considered to have a higher level of playfulness but a lower level of overall usability.

The playfulness can increase the motivation, involvement and enjoyability of the participants, but the effectiveness of coordination and the production of ideas can be blocked by issues of usability. According to qualitative feedback in the subjective questionnaires, one of the most problematic issues with the application for the interactive tabletop was the use of the virtual keyboards. Text input on virtual keyboards is still very much an open question in user interface research [Hinrichs 07, Findlater 11], and the most advanced commercial applications require high resolution touch input devices, predictive text suggestion, automated correction, coherent system-wide copy-and-paste metaphors and text selection interaction techniques to perform well, which the experimental prototype did not include. Therefore, this was a frequently cited problem in the qualitative com-

ments of the questionnaire.

This is supported by other comparable experiments in the literature. In Buisine et al.'s four-person experiments [Buisine 12], a decrease in idea generation from the paper-and-pen condition to the interactive tabletop condition was also observed, as was an increase in subjectively evaluated motivation. Moreover, as in our case, the interactive tabletop was rated lower in terms of general usability, and higher in terms of playfulness. In their experimentation, virtual keyboards were also used for text input. In contrast to these results, Hilliges et al.'s two-person brainstorming sessions [Hilliges 07] reported no statistically significant difference in idea generation between conditions. In their experiments, the participants wrote the Post-it notes by hand using pen-based interaction techniques. Despite these similar results, Hilliges et al. do mention that the stylus interaction still had usability issues, as the users could not rest their hands on the surface of the tabletop while they were writing. Due to the high number of confounding variables between these experiments, isolating the effects of text entry on idea production in groups is difficult, but from the qualitative feedback, results from similar experiments and the observations made during the experiments, it is our interpretation that difficulties with virtual keyboard impacted idea production.

The experiments also reveal how difficult it is to design for creativity. Overall, creativity was rated to be the same in the two condition, even though the TATIN-PIC condition included two additional pieces of functionality specifically designed to augment creativity: hiding Post-its on the keyboard to reduce evaluation apprehension and producing Polaroids with images from web search. Both of these pieces of functionality were evaluated to have no impact on the user's creativity. Hiding Post-it's under the keyboard was seen to have questionable utility and relatively low levels of enjoyability. Creating images was rated not to be very effective. Some of the more positive aspects of the system was the utility and enjoyability of throwing Post-its and the playfulness of using touch gestures to manipulate Post-its. Moreover, the participants found the system easy to learn, even though the training session was relatively short.

This experiment was formative to the design of the interactive system. The results directly informed the future implementation efforts and led to an overhaul of the interaction techniques regarding text input and pseudo-physical metaphors. Text input was approached from a hardware and software perspective. First, the amount of cameras for detecting touch input was increased from 2 to 4, essentially doubling the resolution of the touch input device. Second, new image processing software and camera drivers were installed, greatly decreasing the delay between between a user's physical touch and a software touch event. Third, an editing bar was added along the top edge of the virtual keyboard, allowing users to create Post-its by dragging text off the bar and onto the canvas and modify Post-its by

dragging them off the canvas and onto the editing bar. The editing bar also uses text selection and cursor placement interaction techniques to give users a greater control of text input.

Because of the positive response from the participants regarding the enjoyability and utility of the inertial throwing gesture for Post-its, more pseudo-physical metaphors were added to the design of the project-centered applications. The user interface was redesigned to have its components adhere to the effects of a 2-D physics engine, which granted much more expressiveness in how a designer could implement pseudo-physical metaphors. For example, one could increase the inertia of the Post-its to allow them to be thrown further and faster, but also allow them to collide and stick to the border of the tabletop or other components, such as portals. Similarly, they no longer slide under the keyboard, but rather collide with it. This all makes it easier to put Post-it notes in the editing bar, as they sometimes had a tendency to be “thrown” under the keyboard if they were released too early.

6.1.3 Project planning

Brainstorming has emerged as the prototypical activity with which experimental collaborative groupware systems are tested (see [Buisine 12, Geyer 11a, Hilliges 07]). As an activity, it provides a valuable benchmark and point of comparison between different collaborative groupware in the community. It naturally shifts the focus on equitable participation, and amplifies social constructs such as social loafing and evaluation apprehension. However, it is limited in its complexity and does little to convey the range of potential activity that could be completed in collaborative interactive spaces. For this reason, a second evaluation was conducted using a project planning task in a between-group comparative evaluation.

The procedure of this evaluation is designed to resemble a real-world project planning scenario. In this scenario, senior management has requested that a project manager present a written report and timetable for the launch of a new multi-departmental project to develop an all-terrain vehicle. The project manager schedules a meeting with the entire project team, which includes at least one influential member from the human resources department, the research and development department, the industrial engineering department and the financial department. The purpose of this meeting is to gather the team so they may collectively discuss coordination and planning constraints while attempting to optimize the flow of the project as much as possible. Instead of starting from zero (which rarely occurs in company workflows), they use as a departure point the information from a previous multi-departmental project, which needs to be completely restructured for this current project. During the meeting, the brown paper method with PERT analysis is used to construct a pen and paper version of the

project flowchart. After the meeting is finished, the project manager might stay in the room with his laptop and transcribe the flowchart into project management software, such as OmniPlan or Microsoft Project. Alternatively, the project manager might simply take pictures of the paper flowchart and bring these pictures back to his desktop computer in his office for this task. The project manager will additionally want to generate a written report, presenting the general information of the project and integrating any notes that he has taken from the meeting. These two items are circulated for review by the team, and eventually returned to senior management. In this scenario, these are considered to be the resulting artifacts of the collaborative work.

This evaluation will run this scenario in two environmental conditions in which teams must generate a Microsoft Project file and a brief written report of the project. Due to the time-intensive task at hand, the evaluation was split into two 3-hour sessions. The actual nature of the collaborative work will be different in each condition, as each environment will require focusing on different kinds of subtasks, but the output of the collaborative work is design to be the same.

6.1.3.1 Participants

This experiment included a total of 20 participants divided into 4 groups of 5 people. The participants were all engineering students aged 21 to 31 years, with a median age of 23. There were ten males and ten females (two females per group, except for one group in the experimental condition with four females). There was no statistically significant difference between the groups regarding prior knowledge of multitouch interfaces or prior knowledge of project planning applications. All participants belong to a university-level course on project management. This is perhaps less ideal than having an actual project manager and team, but is more ideal than using a random sampling of university students for two reasons: first, the teams have an appropriate level of familiarity with each other as they have already worked together on weekly class projects for four weeks prior to the experimentation, and second, the participants share a standardized (though elementary) base of knowledge regarding project planning.

6.1.3.2 Environment and Apparatus

The implementation of the experimental condition (TATIN-PIC condition) uses the interactive tabletop and the interactive whiteboard running the version of BrainTouch and PlanTouch presented in the previous chapter. The environment also includes a paperboard with permanent markers and a whiteboard with dry-erase markers.

The traditional environment (control condition) is similar to that observed in

the project planning activity in Chapter 1. Prior to the beginning of the session, participants will attach kraft paper to the wall and to the tabletop. Post-it notes and permanent markers are provided, as well as a whiteboard and dry-erase makers.

At the start of both conditions, participants are informed that they are allowed to use any personal devices or personal items that they may have brought with them. As the students all belong to the same course, they each have their own textbooks and notes which can be used for reference. Some participants might bring their own laptops, which will allow them to use software such as Microsoft Project (which has been distributed for free to the university students and recommended in the context of the course). These are representative of physical resources that are brought to group meetings based on the personal preferences of attendees, and therefore are not restricted.

At the end of the second session, the participants were asked to submit a brief report which presents the project plan, and to submit a Microsoft Project file of the project. For this purpose, in the TATIN-PIC condition and control condition, the environment also provided participants with a desktop computer with the necessary software pre-installed. This is presented as an option, in the case that participants do not have the required software on their personal computers. In the traditional environment, there is a computer terminal with a over-head LCD monitor. In the experimental environment, a similar computer is located a few feet away from the interactive whiteboard. It is less likely to be used in the experimental condition, because the Microsoft Project file is generated automatically, but it provides the users with the option of opening and viewing the Microsoft Project file on a provided computer before submitting it.

Using personal devices or the provided desktop computer requires synchronization with the Dropbox system that the TATIN-PIC environment uses for distribution and output. These two options (either the use of the desktop or personal devices synchronized with Dropbox) were offered to participants and set up at the beginning of the second planning session.

Participants in both conditions will have the design brief detailing the nature of the activity distributed to them in paper, but participants in the TATIN-PIC condition also have the possibility of accessing a digital copy, accessible through each participant's file browser on the tabletop. This design brief, provided in Appendix D, is heavily based on a project planning exercise that can be found in [Gidel 07].

6.1.3.3 Experimental Design

Due to the length of the activity, the experiment was divided into two 180-minute sessions which took place with an interval of two weeks (Table 6.6). The general

Table 6.6: Experimental Design over two sessions.

Session	Control	TATIN-PIC	Sub-activity	Date
1	Group A	Group B	Task Identification, Project Flowchart, PERT analysis	D+0
1	Group C	Group D	Task Identification, Project Flowchart, PERT analysis	D+7
2	Group A	Group B	Planning optimization, Report generation	D+14
2	Group C	Group D	Planning optimization, Report generation	D+21

procedure is pictured in Figure 6.1.7 but activity times may vary depending on the productivity of the groups. In the first session, the two groups shared the same general introduction before they entered their respective environments, where they completed the first personal questionnaire, and signed image release consent forms. After the activity is finished, they each completed a questionnaire evaluating different aspects of the environment. The second session was similar, with a general introduction and reorientation at the beginning, and a second post-activity questionnaire.

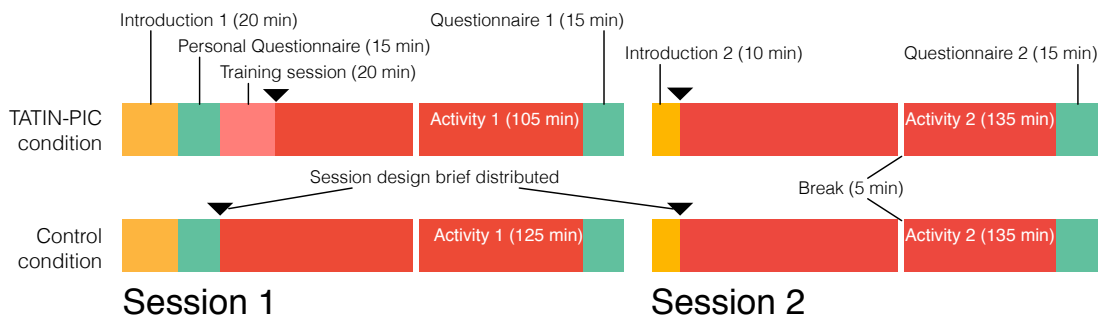


Figure 6.1.7: General procedure for planning evaluation. Note that activity times in practice may vary as some groups may finish earlier.

6.1.3.4 Activity

At the beginning of the first activity session, participants volunteered for one of five roles: project manager (moderator), human resources director, R&D engineer, industrial engineer, and financial director. Once the roles are established the design briefs are distributed to all participants. According to the observations in

Chapter 1, the activity is divided into five main tasks: task identification, project flowchart, PERT analysis, planning optimization, and report generation. These are completed using different but similar approaches depending on the environment. Due to the complex nature of the activity, they also serve as checkpoints so that some form of structure and guidance can be provided for their work. Upon completing one of these tasks, an experimental facilitator will review and validate their work. The purpose of this intervention is not to make sure that the teams have completed the work without any errors, but to confirm that the participants have completed enough work to be able to proceed with the next task.

The activity segment in each session begins with the distribution of a design brief, which describes the scenario and the tasks that the group must complete.

1) *Task identification*: The design brief provides an unordered list of 8 project phases, 24 tasks, and 14 subtasks which must be assigned to different departments. Each of the tasks and subtasks must be written on separate (physical or virtual) Post-it notes and assigned to different departments. This is done in the control condition by each participant collecting the Post-its which belong to their departmental role. In the TATIN-PIC condition, participants must tap on their tasks to highlight them in their personal color. (After this is complete, the groups receive an additional document which presents the hierarchical structure of all phases, tasks, and subtasks, and provides the duration of each task or subtask).



Figure 6.1.8: Control condition (left) and TATIN-PIC condition (right). This image is taken after the task identification sub-activity is complete and the project flowchart has begun. Participants in the control condition are using the vertical board to layout tasks, while the participants of the TATIN-PIC condition must group tasks into phases before importing them into the flowchart.

2) *Project flowchart*: The phases, tasks, and subtasks are used in the construction of the project flowchart in the style of the brown paper method. This requires

that the group also create milestones, which occur after each phase, in the project flowchart. In the control condition, participants can layout the flowchart with colored Post-its and pens depending on their personal preference, though convention dictates yellow Post-its for tasks and pink for milestones, with phase names written directly on the paper in black. In the TATIN-PIC condition, the project structure of phases, tasks, and subtasks is defined by using the hierarchical grouping technique on the tabletop and then importing the groups into a flowchart. The tasks will appear organized by phase in a project flowchart on the interactive board, at which time the users can restructure the sequence of tasks and phases in the flowchart. Groups in the TATIN-PIC condition must also explicitly specify the dependency relationships between tasks by drawing arrows between the tasks to define their successors and predecessors.

3) *PERT analysis*: The project flowchart is used for a PERT analysis (or critical path analysis). In the control condition, this is done by using the task's duration and successor/predecessor relationships for the calculation by hand of the earliest start date, earliest end date, latest start date, latest end date, free float, and total float of each task. In the TATIN-PIC condition, participants must enter in the duration of each task, and the free float and total float is automatically updated and calculated with each modification.

This activity provides a complete project flowchart with the critical path identified, thus concluding the activity of the first session. In both cases, participants are provided with a gantt chart which lists the phases, tasks, and subtasks of the project, and are asked to pencil in the boxes to show the sequence of tasks and their dependencies. This is submitted as an intermediary artifact for the project planning session.

At the start of the second session, a new design brief is given to the participants, with additional information regarding how the current project should be optimized. For example, in the design brief the senior management has approved that the project be reduced from 8 phases to 5 phases, removing three milestones. After reading the design brief, the groups may perform the two following activities (planning optimization and report generation) sequentially or in parallel. The design brief discusses the written report first, prompting groups to at least complete a general outline of the project report before engaging in planning optimization.

4) *Planning optimization*: This involves restructuring the flowchart and its dependencies, leading to an optimized workflow. The participants are informed in the design brief that, ideally, the project should be reduced from 24 months to 18 months, and that senior management has approved that the project be reduced from 8 phases to 5 phases, removing three milestones and allowing for a greater number of tasks to be run in parallel. Participants must discuss and remove any redundant or unnecessary tasks, or loosen dependency relationships to allow more

tasks to be run at the same time.

5) *Report generation*: The report should include a text document which provides a general presentation of the project, its goals, and the role of each of the department in the project. The report also should include a Microsoft Project file of the optimized project. In the control condition, the project flowchart must be reproduced manually in Microsoft Project, while in the experimental condition, it may be generated automatically.

6.1.3.5 Data gathering

The sessions in the control condition were filmed with a setup consisting of three cameras recording from three different perspectives with a microphone to record audio from the meeting. The sessions in the experimental condition were filmed with a setup of four cameras, including one directly overhead the interactive tabletop, and one facing the whiteboard, and a microphone to record audio from the meeting. They will be analyzed for completion times for each activity and as well as to identify any usability issues which occur in the environment. Photographs were also taken at random intervals throughout the sessions.

The subjective questionnaire after the first session included 26 questions with an additional 8 questions for the participants in the TATIN-PIC condition. The first ten questions focused on the subjective evaluation of the following 8 variables on a 7-point Likert scale: level of involvement during the method; general usability of the brown paper method for constructing the flowchart and performing the PERT analysis; ease of collaboration while using this method; ease of generating the deliverables, ease of coordination, ease of time management, group productivity, and confidence in the results. Two opened ended questions asked participants to describe the advantages and disadvantages of using the brown paper method and PERT analysis. The second set of sixteen 7-point Likert scale questions focused on the use of specific components (or interaction techniques in the case of the TATIN-PIC condition). The criteria included the usability, effectiveness, enjoyability, and playfulness of creating Post-its on the tabletop, the usability, effectiveness, enjoyability, and playfulness of manipulating Post-its on the tabletop, the usability, effectiveness, enjoyability, and playfulness of creating hierarchical groups and the usability, effectiveness, enjoyability, and playfulness of performing the PERT analysis. The additional eight 7-point Likert scale questions in the case of the TATIN-PIC condition was the learnability of the system, the amount of errors and mistakes made when using the system, the utility of the system, the table surface size and shape, the table surface height, the board surface size and shape, the board surface height, and the future potential of such a system.

The subjective questionnaire after the second session included 25 questions. The first eight questions of the second questionnaire were identical to the first eight

questions of the first questionnaire, with the same two open-ended questions asking participants to describe the advantages and disadvantages of the method. The second set of twelve 7-point Likert scale questions focused on the use of specific components (or interaction techniques in the case of the TATIN-PIC condition). The criteria included the usability, effectiveness, enjoyability, and playfulness of manipulating the phases in the flowchart, the usability, effectiveness, enjoyability, and playfulness of manipulating Post-its on the board, and the usability, effectiveness, enjoyability, and playfulness of creating dependency relationships for the flowchart. The additional three questions were asked in both conditions regarding the usability, effectiveness, and enjoyability of returning to the work after a period of two weeks.

6.1.3.6 Results

Performance: The two metrics used in evaluating performance are the duration of each sub-activity in each task and the duration of the project plan submitted by the groups. The duration of each sub-activity for the two conditions is presented in Figure 6.1.9. Estimating exact times is difficult because the boundaries between sub-activities may sometimes be unclear. This occurs, for example, when three participants begin optimizing the project planning while two participants continue the project report outline. To a certain extent, Figure 6.1.9 simply indicates which of the tasks are more labor intensive in each condition. Building the actual project flowchart (sub-activity 2) is more labor intensive in the experimental condition because participants must link Post-its and explicitly create the successor and predecessor dependencies, while in the control condition they are implicit and derived from the spatial layout of the Post-it notes.

Once the network of tasks is defined, performing the PERT analysis is trivial in the TATIN-PIC condition, but must be done by hand in the control condition. Though the duration of sub-activities highlights the difference between the two conditions, drawing conclusions which would identify one method as being more effective than another is difficult, due to tradeoffs between effectiveness and efficiency. For example, one interpretation of the data could be that the TATIN-PIC condition is more effective for planning optimization (sub-activity 4) because participants spent more time engaged in the sub-activity. Another interpretation is that the TATIN-PIC condition is less efficient for planning optimization because participants took more time to complete the sub-activity. If this is the case, then the subjective evaluation of the usability of the interactive space would also identify problems with the efficiency of the interface or problems with the time management of the group. This will be addressed in the subjective evaluation of the system below.

In order to have some measure of how well the groups performed, figure 6.1.10

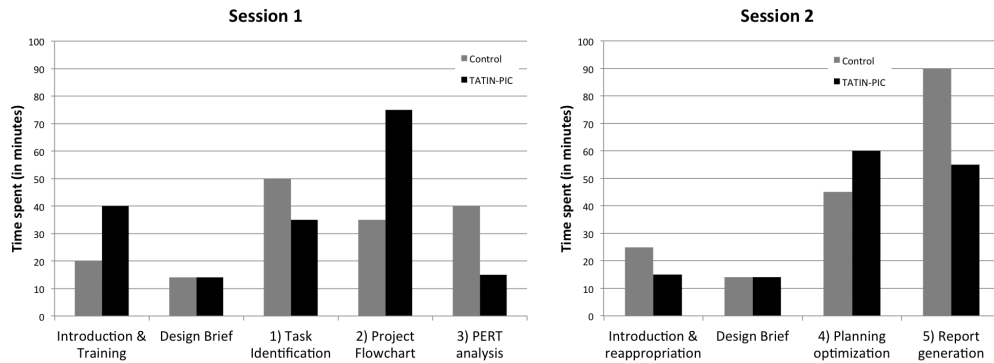


Figure 6.1.9: The duration of each sub-activity in the control and TATIN-PIC conditions.

presents the accuracy of the project duration of the flowchart at the end of session 1 (before optimization) and session 2 (after optimization) in both conditions. The accuracy of the project duration was calculated by comparing the resulting project plan durations to the ideal project durations: 110 weeks for the first session and 78 weeks for the second session. Groups B and D in the TATIN-PIC condition were consistently more accurate in the output of their work, but because there were only two groups for each condition, no statistical analysis could be conducted.

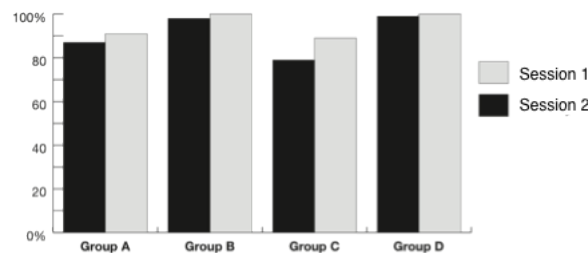


Figure 6.1.10: Accuracy of resulting project plan in each flowchart for each group in each section.

With regard to the resulting project flowcharts, its worth highlighting the difference in media that each condition affords. Figure 6.1.11 shows the resulting flowcharts from the control condition and the experimental condition. The flowcharts of the control condition are highly customizable and the use of colors and pens allows the group to create an artifact which is colorful and personal. There are, however, clear drawbacks to the flowcharts in the control condition. For example, it is particularly difficult to create links between tasks in the flowchart. Here participants in Group C (control) have created arrows on green Post-its which

can be positioned to express dependency relationship when they occur between two tasks that are not already positioned side-by-side. These arrows are difficult to manage as the Post-its repeatably change positions. They are also difficult to interpret for observers outside the group, and they are abandoned entirely through the extensive planning optimization phase. In the TATIN-PIC condition these are more easily maintained, because they automatically follow Post-its when they are displaced.

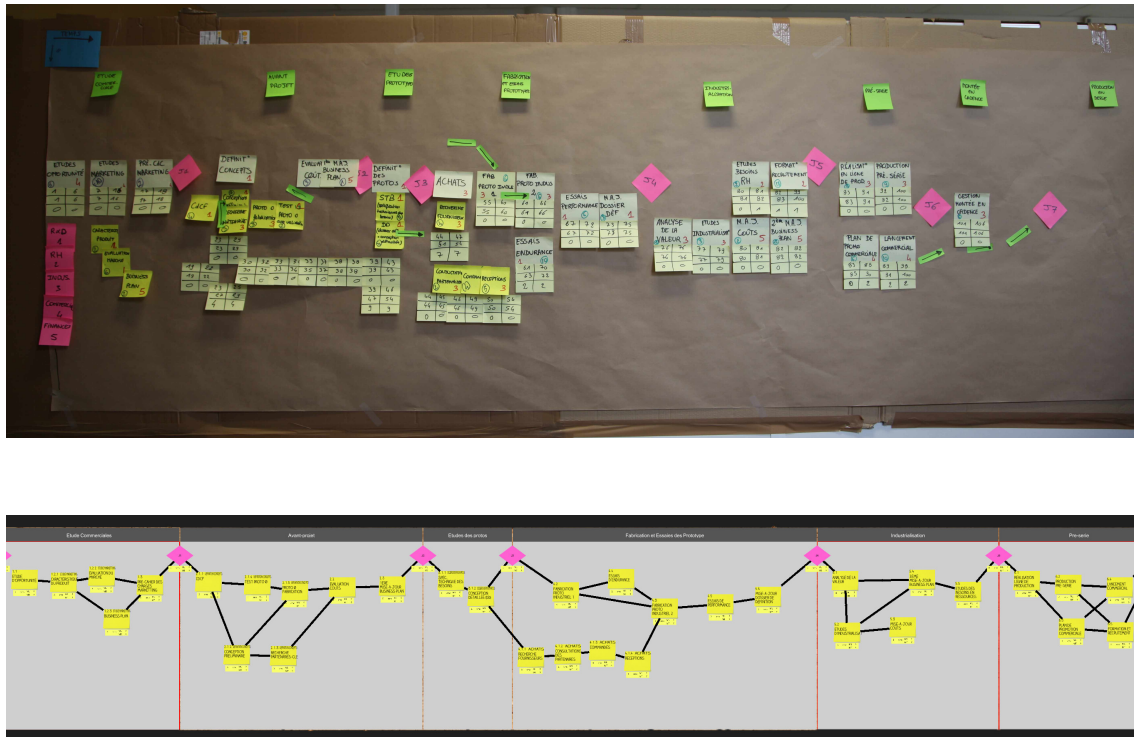


Figure 6.1.11: Flowcharts from Group C, control condition, top and Group B, TATIN-PIC condition, bottom.

Group dynamics: Table 6.7 and Table 6.8 show the results from the subjective questionnaire regarding the criteria which addresses the group dynamics in the first and second session. In the first session, participants reported no statistically significant difference between the control condition and the experimental condition for the participant's personal involvement in the group work, the productivity of the group, their personal confidence in the results, the ease of collaboration, and the ease of coordination. There is a marginally significant difference which points to better time management in the control condition than the TATIN-PIC condition. Qualitative feedback indicates there were issues with the efficiency and the learnability of the system in the TATIN-PIC condition: "I had the impression

of losing time when we were creating the Post-its (typing mistakes, keyboard errors, ...)” (P1, group B); “We spent a great deal of time learning and getting familiar with the system” (P4, group B).

Results for the second session indicated a statistically significant difference and higher levels of productivity and time management in the TATIN-PIC condition than the control condition. Planning optimization, sub-activity 4 in the second session, is difficult in the control condition, because the PERT analysis and the links between Post-its breakdown after repeated restructurization of the flowchart. This forces participants to maintain an electronic version of the project in conjunction with the restructurization of the paper project flowchart. This is mentioned as one of the weakest points in the method in the qualitative comments: “We must always keep updating the MS Project file to do the calculations” (P6, group A). Figure 6.1.12 shows this dynamic in Group C, which involves two participants coordinating with each other while other participants observe. Commenting on group coordination in Group C, one participant writes “... sometimes people are not very involved” (P2, group C). Groups in the TATIN-PIC condition remain engaged in the activity throughout the process of restructurization and this represents one of the most advantageous aspects of the system according to the qualitative feedback from the participants: “The total float is automatically calculated” (P1, group D); “The group interacts well and is cohesive” (P3, group D); “Easy input from other group members, because we can quickly modify and remodify the flowchart” (P4, group B).

Table 6.7: Results from the comparative subjective evaluation on criteria for the group dynamics in the interactive space for the first session, with corresponding paired Student’s t-test between the two conditions.

	Control		TATIN		DOF	F	p value	sig
	m	SD	m	SD				
Personal involvement	5.9	0.6	5.6	1.2	18	0.73	$p=0.48$	no
Group productivity	6.2	0.6	5.8	1.1	18	0.97	$p=0.34$	no
Confidence in results	6.5	0.5	6.5	0.5	18	0	$p=1$	no
Ease of collaboration	5.7	0.7	5.1	1.2	18	1.38	$p=0.19$	no
Group coordination	5.5	1.0	6	0.7	17	-1.27	$p=0.21$	no
Time management	5.2	1.0	4	1.6	18	1.96	$p=0.07$	marginal

General usability:

The subjective questionnaires also focused on specific interaction techniques of the brown paper method, so as to uncover any usability issues with the two conditions. Table 6.9 and Table 6.10 presents the results from the subjective questionnaire from the first session and the second session respectively, as well as the

Table 6.8: Results from the comparative subjective evaluation on criteria for the group dynamics in the interactive space for the second session, with corresponding paired Student's t-test between the two conditions.

	Control		TATIN		DOF	F	<i>p</i> value	sig
	<i>m</i>	SD	<i>m</i>	SD				
Personal involvement	5.2	0.8	5.5	1.0	17	-0.66	$p=0.51$	no
Group productivity	5.5	0.8	6.4	0.7	18	-2.59	$p<0.05$	yes
Confidence in results	6.2	0.6	6.6	0.7	18	-1.34	$p=0.20$	no
Ease of collaboration	5.8	0.6	5.7	0.8	18	0.30	$p=0.76$	no
Group coordination	5.1	0.7	5.6	1.1	18	-1.21	$p=0.24$	no
Time management	4.8	1.1	6.3	0.7	18	-3.59	$p<0.01$	yes

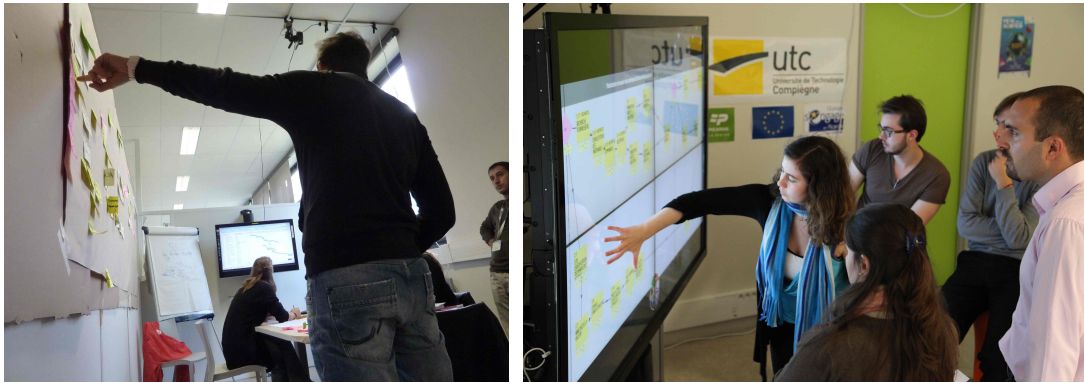


Figure 6.1.12: A participant in Group C restructuring the project flowchart, and communicating changes to another participant who updates an electronic version (left) and participants in Group D discussing the planning optimization (right).

statistical comparison between the two conditions. With regard to the actions which take place on the tabletop, Table 6.9 provides insight concerning the creation of Post-its and the creation of groups. Of all the functionality provided by the interactive space, Post-it and group creation are the lowest ranked functionality in the interactive space in terms of ease of use. When discussing time management above, one participant already cited the inefficiency of the keyboard to be a problem. One participant explains that “writing Post-its is difficult because of the sensitivity of the tabletop” (P5, group D) and another says they cannot rest their fingers on the keyboard and must lift their fingers too high for the keyboard to be intuitive to use (P1, group B). This is because the laser light plane used to detect fingers is positioned several millimeters above the surface, and therefore the touch input is very sensitive and erroneous text entry is possible. Comments regarding the ease of manipulating groups reveal that the method of grouping requires considerable explanation and practice to understand (P1, group B; P2, group D). Once learned, however, the method was “intuitive” (P4, group D). This technique did receive high ranking and solicited comments regarding its enjoyability and playfulness: “On the interactive surface, creating a hierarchy of Post-its becomes a game” (P1, group D).

Selecting Post-its by tapping in order to highlight them in the user’s color was used during the task identification procedure. This technique made use of the user-specific identification functionality proposed in Chapter 3. The questionnaires did not include questions directly related to this piece of functionality, but from observations of the video, identification misses rarely occurred, and when they did, they were immediately corrected. This illustrated that the distance heuristic is a simple and effective means for user-identification.

Creating Post-its was evaluated to be generally easier to do in the control condition. Creating Post-its and groups in the traditional environment was not free from criticism either, as writing them all by hand is “labor-intensive” (P2, group C) and produces problem with legibility: “once we try to change something with a pen, they become unreadable” (P1, group A). Grouping Post-its was ranked highly in terms of enjoyability and playfulness (similar to the experimental condition) because of the use of colors of pens and Post-it notes and space to layout and visualize the groups. Feedback from participants explained that they would have enjoyed even more colors and more space to layout the notes (P1 and P4, group A).

When considering the interactions which take place on the interactive board display, the results point to several advantages of the interactive system. PERT calculations were rated to be more enjoyable and easier to perform. Creating and using links to express dependency relationship was rated to be easier, more effective, and more enjoyable in the TATIN-PIC condition. Moreover, returning

Table 6.9: Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.

		Control		TATIN		DOF	F	p value	sig
		m	SD	m	SD				
Creating or modifying Post-its on the tabletop	ease	6.4	0.8	4.5	1.4	18	3.61	$p < 0.01$	yes
	effectiveness	5.5	1.8	5	1.6	18	0.67	$p = 0.51$	no
	enjoyability	6.1	1.1	5.4	1.4	18	1.23	$p = 0.23$	no
	playfulness	6	1.1	5.8	1.2	18	0.39	$p = 0.70$	no
Manipulating Post-its on the tabletop	ease	5.1	2.1	5.5	1.1	18	-0.54	$p = 0.60$	no
	effectiveness	5.8	1.2	5.5	1.2	18	0.61	$p = 0.55$	no
	enjoyability	5.7	1.5	5.7	0.8	18	0	$p = 1$	no
	playfulness	5.5	1.5	6.2	0.6	18	-1.35	$p = 0.20$	no
Grouping Post-its	ease	4.8	1.2	4.2	1.5	18	0.99	$p = 0.34$	no
	effectiveness	5.9	0.9	5.1	1.4	18	1.45	$p = 0.16$	no
	enjoyability	5.9	1.0	5.9	1.4	18	0	$p = 1.00$	no
	playfulness	5.8	0.8	6.2	0.8	18	-1.13	$p = 0.27$	no
Performing PERT analysis	ease	4.1	1.5	6.3	0.7	18	-4.17	$p < 0.01$	yes
	effectiveness	5.7	1.8	6.3	1.1	17	-0.96	$p = 0.37$	no
	enjoyability	5.1	1.1	6.4	0.8	17	-2.96	$p < 0.05$	yes
	playfulness	5.2	1.7	5.9	1.2	17	-1.01	$p = 0.33$	no

to the work after two weeks was also judged to be easier, more effective and more enjoyable. Generating the final deliverable was also rated to be easier in the experimental condition.

Table 6.10: Subjective evaluation of communication and coordination on a 7-point Likert scale, with corresponding paired Student's t-test between the two conditions.

		Control		TATIN		DOF	F	p value	sig
		<i>m</i>	SD	<i>m</i>	SD				
Manipulating phases in flowchart	ease	5.3	1.6	5.4	1.2	17	-0.27	$p=0.79$	no
	effectiveness	6	0.9	6.2	1.0	16	-0.43	$p=0.67$	no
	enjoyability	5	1.3	5.6	1.0	17	-1.13	$p=0.28$	no
	playfulness	5	1.7	6.2	0.9	17	-1.92	$p=0.08$	no
Manipulating Post-its in the flowchart	ease	5	2.1	5	0.8	18	0	$p=1$	no
	effectiveness	5.2	1.5	5.3	1.3	18	-0.16	$p=0.88$	no
	enjoyability	4.9	1.3	5.3	1.3	18	-0.70	$p=0.49$	no
	playfulness	5.3	1.6	5.8	1.0	18	-0.84	$p=0.41$	no
Dependency relationships (links) in flowchart	ease	3.6	1.9	5.1	1.4	17	-1.98	$p=0.07$	marginal
	effectiveness	3.8	1.3	5.3	1.9	16	-1.98	$p=0.06$	marginal
	enjoyability	3.9	0.8	5.2	1.5	16	-2.25	$p<0.05$	yes
	playfulness	4.4	1.5	5.7	1.3	15	-1.82	$p=0.10$	no
Reappropriation of work after two weeks	ease	3.7	1.4	5.4	1.3	18	-2.67	$p<0.05$	yes
	effectiveness	4.7	1.3	6.1	0.7	18	-2.89	$p<0.05$	yes
	enjoyability	4.2	1.0	6.1	0.7	18	-4.73	$p<0.01$	yes
Report generation	ease	4.8	1.1	6.4	0.8	18	-3.58	$p<0.01$	yes

6.1.3.7 Discussion

These evaluations demonstrated that there is a considerable amount of overhead in initiating the brown paper method and PERT analysis, regardless of the environmental conditions. When comparing the two sessions, the TATIN-PIC condition does not provide any clear advantage in terms of facilitating the set-up of this method. It compounds the already labor intensive process with a wide range of additional interaction techniques which are necessary but difficult to learn. Participants in the control condition are free to build, organize and customize their flowchart with less training and less constraints, and they spread this information across more surface area, using color and spatial arrangements as a valuable tool to visually communicate relationships between different components.

By the end of the first session, however, this dynamic is reversed. Once the

project flowchart is established in the TATIN-PIC condition, the overhead in maintaining its structure diminishes greatly. The flowchart in the TATIN-PIC condition retains its structural soundness and consistency throughout multiple changes and modifications, while the flowchart in the control condition seems to resist these changes. For example, moving one Post-it out of order during the planning optimization negates the laborious calculations of the PERT analysis which identifies the critical path. This lead to one group sticking blank Post-its over the all the original Post-its to hide the now defunct PERT calculations. Additionally, the dependency relationships that were easy to understand from their implied spatial configuration, are more difficult to interpret two weeks later, while participants in the TATIN-PIC condition benefit from the explicitly defined links between Post-its that they created in the first session. The Post-its in control condition also lose their adhesive after time, forcing the groups to recreate the task on a fresh Post-it or tape them to the board. This leads us to the primary conclusion of this evaluation: creating a complex digital artifact such as a project flowchart, is easier in traditional environments than in interactive space, but maintaining the consistency of the artifact through multiple manipulations and rendering it persistent is much easier in the interactive space than in their traditional counterparts.

What this conclusion indicates is that there is still work that remains regarding the creation of data on the interactive tabletop. The virtual keyboards (which were redesigned since the last session) and the grouping technique both garnered some negative feedback. Without the tactile feedback and the ability to rest fingers on the keyboard that is provide by physical keyboards, virtual keyboards will always be at a certain disadvantage. One possible solution would be take inspiration from other interactive tabletops which incorporate physical keyboards in mixed reality setting, providing the benefits of physicality and virtual text input.

The main problem with the grouping technique is that it was designed for the context of brainstorming and inherently benefits from a bottom-up approach to heirarchization. For example, a group must be created using two Post-its before it is labeled. This is counter-intuitive to how participants were hierarchizing information in the project planning session. Participants would want to start first with an empty labeled group, representing a phase, and then add tasks to this group, which could not be done. Once the participants understood the bottom-up reasoning, the participants had less difficulty using the grouping technique, but this functionality should still be added to the grouping technique. Participants also contributed some helpful advice to make the grouping technique easier to use. For example, the length to which participants were required to pull a Post-it away from the group (in order to remove an item from the group) was too long and should be shortened.

6.1.4 Causal analysis

After a within-subjects evaluation for brainstorming and a between-group evaluation for project planning, we were interested in evaluating users in a real world scenario. The third evaluation is a case study focused on the causal analysis session. It is also meant to serve as a counterpart to the traditional causal analysis activity observed in Chapter 1.

6.1.4.1 Context

This case study was completed in collaboration with academic professors of a course on Value Engineering at the Université de Technologie de Compiègne. The Value Engineering methodology for product or service design is similar to preliminary design. It emphasizes the use of creative and collaborative group sessions to explore and evaluate alternate solutions to a problem, and therefore typically includes a causal analysis session. The motivation behind this evaluation was to provide a group of students with the training to use the applications in the interactive environment, and then to provide them with the option of using the system for their own personal project work, so that a real world example of the environment in use could be observed.

6.1.4.2 Procedure

Two groups of five spent three sessions using the environment for brainstorming, causal analysis, and functional analysis (which uses a FAST diagram structurally similar to the cause and effect diagram). Technical staff was available to provide advice and in-situ training throughout these sessions. Following these three sessions, the group met for an additional six sessions to work on a project proposal in the context of their course using the value engineering methods. Many of the meetings used only the basic features of the TATIN-PIC environment, while the group focused their attention on the non-interactive whiteboard and the paperboard. When the group identified the need to complete a specific activity which the TATIN-PIC platform supported, the applications were used by their own volition.

This case study focuses on one such session involving the causal analysis activity performed by one of the groups, and analyzed using the activity theory checklist. A brief one-page questionnaire was also distributed to both groups to gather data on the subjective evaluation of the portal interaction technique.

The objective of the group's class project was to use value engineering methods to propose a new approach to the operations of the university's food services department. The objective of the causal analysis session was to identify the reasons why a student would not take advantages of the current food services (i.e. why they would skip a meal or eat poorly) and the consequences of this event.

The session was recorded from two different angles focusing on the interactive whiteboard and the interactive tabletop. The students used the causal analysis application presented in the previous chapter. Upon the completion of their project, a brief subjective questionnaire was distributed to the students, to gather their thoughts on the portal transfer interaction technique, which had not been used in previous evaluations.

6.1.4.3 Results & Discussion

The activity theory checklist divides analysis into four perspectives: means & ends, environment, learning/cognition/articulation, and development. The causal analysis session is similar to the previously observed session in terms of means and ends, though there exists two key differences. First, the students already have practiced the causal analysis method once before, and second, the subject of causal analysis is a topic of their own choosing, and not one that is assigned to them, meaning they have already had time to reflect on the topic. This is possibly why they created a great number of Post-its: 16 causes and 13 consequences in 40 minutes, compared to the 7 causes and 4 consequences in 90 minutes in the traditional condition. It is important to understand how well the environment managed the creation and manipulation of such a large causal analysis in a shorter period of time.

The environment is the same that is presented in the implementation in Chapter 5. It is the only evaluation that uses the tabletop and the board simultaneously, as opposed to tabletop-only condition in evaluation 1 and the sequential use of the tabletop and the whiteboard presented in evaluation 2. This dynamic is afforded by the portal interaction technique, which allows users to transfer Post-its from the tabletop to the interactive board. Post-its could be created on either surface, but according to the subjective questionnaires participants unanimously favored Post-it creation on the tabletop, simply because “it is easier to use a keyboard on a horizontal surface than a vertical surface” (P4, group 2).

This difference in environment led to a stark difference between the two observations regarding the natural roles that emerge in collaboration. In the traditional environment, ideas were spoken verbally and sequentially, and were filtered by the scribe who created nearly all the Post-its and coordinated their transfer to the moderator. In the experimental environment, Post-its were primarily created by three members of the team, who used the portals to coordinate directly with the moderator. In terms of the shareability of the interface, the portals were effective in providing team access points to the work that was being conducted at the board. Whereas after 40 minutes participants seated furthest away from the board in the traditional environment became increasingly passive, participants in the TATIN-PIC environment remained engaged throughout the exercise. Similar

to the traditional environment, the vertical display was manipulated primarily by the moderator, but both environments still supported the natural and temporary transfer of the moderator role: when one participant had a better understanding of an idea, they would temporarily swap roles with the moderator.

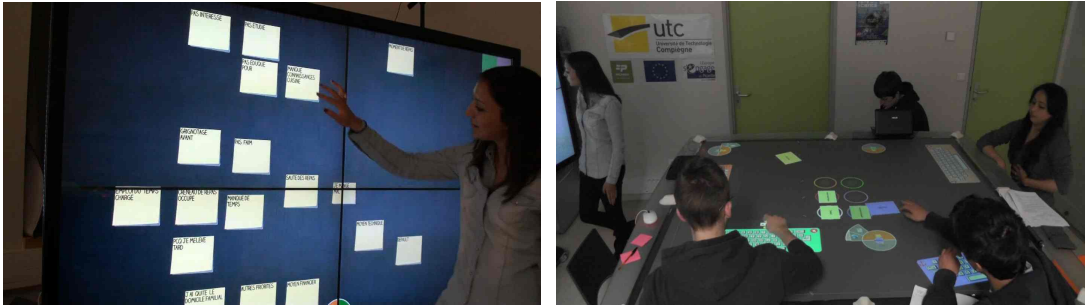


Figure 6.1.13: A group performing causal analysis. The group (right) creates causes and effects as the moderator (left) structures the input from the group.

With regards to learning, cognition and articulation, the participants have already learned how to use the technical aspects of the environment, and no training was required. This is similar to the observation in Chapter 1, in the sense that participants were already familiar with the use of pens and Post-its. The use of space to convey meaningful relationships is also similar in the two conditions. On the interactive board, links were not created between the Post-its until that section of the diagram was complete, similar to the use of links in the traditional environment. Creating links between events signified that the discussion regarding the event was finished, and that participants were ready to turn their attention to a different area of the diagram. Links in the TATIN-PIC environment also had the added benefit of helping to maintain the organization of the diagram, as moving a parent Post-it also changed the position of its children. This was important especially considering the high number of Post-its the moderator was required to manage.

The dynamic between the tabletop and the whiteboard was also acknowledged in the qualitative feedback with the questionnaires: “the tabletop was used so that all could express their ideas” (P3, group 1) and “the board was helpful for organizing the diagram and providing a global view of the work that is being completed” (P5, group 2).

With regard to development in the interactive space, this case study only observed 40 minutes out of many hours of practice and training in the interactive room. Participants were already relatively comfortable in the environment, and would occasionally even use peripheral functionality such as performing web searches or opening PDF using their personal menus.

6.1.5 Conclusion

I will now address the conclusions with regard to the evaluation objectives in reverse order, beginning with the third objective and finishing with the first.

6.1.5.1 Objective 3: Group Dynamics

The third objective of the evaluations was to measure and understand the impact that the interactive space has on communication and group dynamics. I find that *when the interactive space and its applications were designed to maintain the work-flows and the qualities of shareability that are present in traditional environment, group dynamics in the interactive spaces were similar to those inside traditional spaces*. The main findings that result in this conclusion are:

- There was no measurable difference in the equitable distribution of handling gestures or verbal communication in the brainstorming session.
- There was no measurable difference in the subjectively rated ease, effectiveness and enjoyability of communication in the brainstorming session.
- There was no measurable difference in the subjectively rated involvement of participants in the project planning session.

One finding that conflicts with these three results, is the higher level of subjectively evaluated involvement that was found in the interactive condition of the brainstorming session. It is difficult to interpret this results because it did not lead to an increase in talk time, occurrences of handling gestures or idea production. It is perhaps attributable to the general impression of the playfulness of the interactive space, as discussed below.

It is worth noting that at some aspects of the design of the interactive space are very different than its traditional counterpart. The two best examples of this is the automated PERT calculations and dependency network of task (project planning) and the use of the portal interaction (causal analysis). These interaction techniques respectively represent a careful and conscious decision to augment the interpretation and coordination capacities of teams. The automated PERT calculations and dependency networks are successful in increasing subjectively evaluated group productivity and time management. The portal interaction technique eliminates the need for a scribe and allows all participants to participate in the causal analysis equitably. Therefore, I also find *that group dynamics can be augmented in subtle ways through the inclusion of certain interaction techniques* which augment interpretation and coordination. One conflicting piece of evidence with this conclusion, however, is that the creativity techniques that were included in brainstorming (hiding Post-its under the keyboard and creating images) were not found

to be so helpful in augmenting creativity. This was perhaps because of usability issues in their design.

6.1.5.2 Objective 2: Preliminary design performance

The second objective of the evaluations was to measure and understand the impact that the interactive space had on the group performance for the completion of preliminary design activities. This objective encapsulates the first, but because of issues with the experimental design and performance metrics used, it is difficult to extrapolate conclusions beyond those presented above. The evaluations showed that idea production was lower in the brainstorming sessions in the interactive space. The resulting project plan was more correct in the project planning session in the interactive space than in the traditional environment, but with only two groups in each condition, it is difficult to extrapolate conclusions. The case study of causal analysis was extremely productive, but with no real point of comparison, conclusions are again difficult to extrapolate.

Still, there are statistically significant results that highlight the effectiveness of certain sub-activities in interactive project planning sessions such as the enjoyability, ease, and effectiveness of returning to the working environment after two weeks, and the overall usability of the automated report generation and distribution. Therefore, because of these results and because group dynamics (in objective 3) were either equal or better, I find *that interactive spaces support the completion of preliminary design activities as good as or better than their traditional counterpart*. There are some aforementioned threats to the validity of this conclusion, due to the nature of these controlled experiments. These experimentations also focused a great deal on the collaborative, project-centered layer of the interactive space. The participant-centered layer was used, but was always tangential to the objectives of the activities in the experimental design. These findings do motivate the need for deeper ethnographic observations of interactive spaces, in the context of real working environments, which I leave to future work.

6.1.5.3 Objective 1: General usability

The first objective of the evaluation was to evaluate the overall usability of the interactive space. This objective encapsulates the previous two but also leads to some conclusions of its own. It reveals some information concerning the effectiveness of certain interaction techniques that were proposed in Chapter 3. As mentioned in the discussion of the project planning evaluation, the grouping technique served its purpose, but was difficult to learn because it did not allow for top-down hierarchization of groups. Such functionality should be included in future iterations and therefore, I proposed a usability guideline for grouping items

which specifies they should also *support top-down and bottom-up hierarchization*.

The portal interaction technique was also well received and led to a more effective group dynamic in the causal analysis case study, allowing for more involvement from the participants seated around the tabletop. Future work might include a more focused study of this technique comparing it to other space-reaching or cursor-extension multi-surface interaction techniques. However, if the conclusions from Jamil et al.'s comparison of direct manipulation and space-reaching techniques [Jamil 11] in the context of interactive tabletops may be extended to multi-surface interaction, than space-reaching techniques could potentially lack the feedthrough afforded by direct manipulation techniques and be disruptive to group focus.

User-specific identification was also briefly featured in the project planning evaluation, showing promise as a easy to use and easy to implement user identification technique. Still, this was used when participants were at their most immobile, so a more focused evaluation of this interaction technique would also be helpful.

One remaining obstacle in interactive spaces is text entry. The TATIN-PIC interactive space uses virtual keyboards – instead of pen or stylus based input which might have provided a more natural method of text entry – so that the system may exploit the textual content of user input, without having to rely on optical character recognition algorithms. I suspected that usability issues with the virtual keyboard lead to a decrease in idea production in the brainstorming sessions. I added functionality (text selection, cursor placement) to mitigate these usability issues, but in the following project planning evaluation, I still found that Post-it creation was rated significantly low. This is problematic because it is the basis of the metaphor used in all three applications. Still participants were able to slowly adapt to virtual keyboard over time, showing some benefit to practice and training.

6.2 Multi-agent infrastructure: comparison with related work

This section presents several different implementations of interactive space infrastructure in related work in order to compare them with our multi-agent infrastructure. Each subsection presents the infrastructure, discusses the advantages and disadvantages, and how they compare to the TATIN-PIC multi-agent system. When necessary, I also reference the requirements of the infrastructure established in Chapter 4.

6.2.1 iRos middleware

The first propositions for DUI systems were notably event-driven, such as the iRos middleware from the iRoom project [Johanson 02b]. Inspired by the Linda language, iRos is an implementation of the distributed shared memory paradigm. Events (user or inter-device) are expressed as tuples (name, type, value, etc.) and added to a central repository called the Event Heap. The Event Heap manages concurrent access to events granting processes on different devices the ability to connect and to selectively access events using basic pattern matching fields. The Event Heap also includes several features which extend the basic model of tuple-space, such as adding expiration timestamps so that an unclaimed tuple/event may be removed after a given amount of time. A second sub-system in the iRos middleware, the Data Heap, provides a centralized repository for managing data accessibility for multiple processes. The Data Heap allows processes to request access to certain types of data, as well as requesting the format which it is to be received. For example, if a process requests an image stored in the Data Heap, it may also specify the file format, if it can only handle JPEG images. It can also be used to store persistent state information for the devices in the system. A third sub-system, iCrafter, provides service-oriented functionality, including a user interface generator where devices can subscribe to advertised services, allowing iCrafter to supply either custom-designed or generic interfaces to the devices.

The disadvantage of this infrastructure is that user or inter-device events must first pass through the remote Event Heap before they reach their final destination, requiring two passes through the network. Though this has the potential to create some latency issues for certain low-level real-time input devices, it would still be appropriate for the high-level multi-surface interaction techniques used in the TATIN-PIC environment. On the other hand, the authors highlight many advantages of their approach, reporting that they benefited enormously from having a system which is modular and distributed (G1), especially when it comes to rapid prototyping (R1) and flexible network connection (R2). Another advantage of the infrastructure is that it affords a great deal of flexibility for programmers. For example, though the infrastructure is at its core inherently event-driven, it has the advantage of also supporting some aspects of data-oriented programming and service-oriented programming, through the Data Heap and iCrafter frameworks respectively.

The multi-agent system used in the TATIN-PIC interactive space enjoys similar advantages in terms of distributed modularity and fault tolerance. If a single agent crashes, the services of the agent will not be available to the detriment of other agents in its sub-organization, but these effects will be limited to the agent's sub-organization. For example, even if the most central agent, the `PrimaryDeviceAgent`, experiences a fatal error, the sub-organization responsible for

persistence will be able to save the state of the projectModel before the device is restarted. This level modularity is dependent upon the implementation. JADE uses a container metaphor which runs on each device to coordinate agent activity, and therefore crashes will often require restarting a device to relaunch the container and all agents in that container. The personal assistant and its staff agents, implemented in OMAS, are more flexible allowing separate agents to be relaunched individually.

6.2.2 BEACH

The BEACH (Basic Environment for Active Collaboration with Hypermedia) [Tandler 04] framework was developed using Smalltalk for the purpose of the i-LAND ubiquitous computing environment. It uses a holistic approach to modeling the needs of collaborative interactive spaces by dividing its architectural components into a “grid” of four levels of abstraction (core layer, model layer, generic layer, and task layer) and five basic concerns (interaction model, environment model, user-interface model, application model and data model). The use of the grid architecture provides a clear separation of concerns and varying levels of abstraction. The purpose of this design is to facilitate the extensibility of its distributed applications, as developers may follow this grid to abstract and share information relevant to collaboration when incorporating new and heterogeneous devices into the architecture at develop-time.

At its core level of abstraction, it provides an object-oriented mechanism for distributing data in the environment, through the use of a shared object space running on a central server. Once objects are initially copied onto a client device, they maintain their synchronicity through coordinated direct messaging. This makes the implementation of persistence trivial, because all objects are located on one device. One of the most interesting features of this infrastructure is its use of an environment model, which explicitly captures and distributes information regarding the physical aspects of the devices in the room. This facilitates the development multi-device interaction strategies.

Though the authors do highlight the flexible network communication when clients connect and disconnect from the server at run-time, they do not discuss the stability and restartability of the central server itself. This could potentially be an issue when prototyping applications for the room at runtime (R1). Another disadvantage of this framework is the centralized server (similar to iRos) which could be a potential bottleneck in the network messages. This is identified by the authors of shared substance (see next section) when discussing the feasibility of implementing a shared, distributed canvas for rendering across surfaces [Gjerlufsen 11]. The authors of the ZOIL framework [Jetter 12b] (see section after next) also point out that the success of UI design was never evaluated by users or discussed in greater

detail.

6.2.3 Shared substance

Shared substance is a middleware for multi-surface applications used in the WILD room [Gjerlufsen 11]. The middleware is entirely data-oriented, and relies on an explicit separation of the data of an application and the behavior of the data. Building a multi-surface application first requires encapsulating the data in an application tree. For example, branches of the application tree can be dedicated to the data of objects, windows, cursors, or specific input devices such as cameras. Shared substance also provides functionality for distributing this data to other clients located in the interactive space through either the *replication* or the *mounting* of a subtree. Replication allows a data subtree to be stored as a local copy on the device, introducing concurrency issues with the master copy while providing fast access to its local copy. Mounting provides access to the subtree over the network (similar to Java's remote method invocation), which increases latency but eliminates the need for synchronization.

Finally, behavior is added to the data nodes in the subtree through dynamically adding and removing *facets*. When a specific facet is attached to a node, it can supply functionality such as the rendering of objects on a display, or the issuing of cursor events which may originate from pointing instruments and the 3D tracking from the cameras. Similarly, the aforementioned replication and mounting functionality is implemented through facets.

The advantage of this approach is that it provides a powerful and coherent abstraction for a distributed application model, which grants the programmer explicit language-level control of data-sharing functionality. A developer can make choices between mounting and replication strategies depending on the nature of the data, either prioritizing access speed or multi-device data coordination. Moreover, as the authors indicate, this is information that the middleware can and should not hide, because local and distributed data are inherently different. These decisions regarding data sharing strategies can even be modified at run-time, as all behaviors (facets) can be dynamically added and removed. The flexibility afforded by facets is also critical when considering the (R1) rapid prototyping of the user interface in the room, as GUI behaviors can be added and removed with little difficulty.

Shared substance's complete commitment to data-oriented programming and its application tree data structure, as well as its implementation in a custom Python-based language, might be a deterrent for programmers who are accustomed to object-oriented programming. It is unclear how tolerant such a system might be of disconnection and network failures (R1), or how easily persistence can be implemented to save the global state of applications (R3). Presumably, a persistence manager can be implemented as a facet and attached to the application subtree

of a device, though some coordination mechanism would be required if multiple persistence managers exist on different devices. Another possible issue involves user-specific multimodal fusion. This could potentially be done by introducing application sub-trees dedicated to each participant's user model and the sharing these sub-trees across devices. Ultimately, these are dependent on the architectural patterns that designers employ when constructing a distributed application model using shared substance. The coherence and simplicity of shared substance's distributed and data-oriented application model merits further exploration of this kind.

The conceptualization of the multi-agent system in TATIN-PIC is an architectural pattern which is prescribed for multi-surface environment. It directly addresses persistence issues and, because of the multi-agent system toolkits that are used, it is tolerant of network failures. However, when comparing the TATIN-PIC architecture and shared substance, we could benefit from making data sharing more explicit. Many of the message exchanges of the agents are dedicated to coordinating different kinds of serialized objects (replication) and providing access to remote data or behaviors (mounting). GUI designers are not necessarily aware of the actions of the agents, and therefore data can be hidden behind multiple messages passed by agents. This is useful for the purpose of abstraction, but still hides the actual sharing of data. This motivates the design of specific agent types such as replicator agents or mounter agents who connect to data share agents, to provide access to serialized data using these explicit sharing strategies.

6.2.4 ZOIL

ZOIL (Zoomable Object Information Landscape) is a new software framework for interactive spaces recently presented in 2012 by [Jetter 12b]. It is open-source and written in C# and Microsoft Windows Presentation Foundation (WPF), allowing it to interface with the helpful gesture-based APIs of the Microsoft Surface SDK. Structurally, it is similar to BEACH, in the sense that there exists an event-driven centralized server providing the synchronization of all objects, but instead of providing automatically generated user interfaces from a centralized service as is done with iRos (iCrafter), it provides a helpful framework for defining these user interfaces using declarative definitions written in XAML. This reduces the overhead of building GUIs for devices and maintains a similar look and feel across applications.

Beyond the declarative programming used for defining GUIs, ZOIL has several interesting attributes that make it an attractive multi-device software framework. For example, transparent persistence of data is automatically handled in all client-side application models, requiring no additional code as is the case with BEACH and Shared Substance. However, what makes ZOIL the most unique is its approach

to its post-WIMP user interface. Instead of relying on a desktop metaphor, it uses ZUI (Zoomable User Interface) which is a UI which employs semantic zooming of its components. This allows users to expand components on the interface to reveal more information, details or controls on those components through a smooth and gesture-based transition. This adds depth to the information landscape, while still preserving the user's ability to spatially organized content along the x and y axis. Other devices in the environment can share the same information landscape, providing different views on the zoomable landscape, allowing users, for example, to keep an overview on an interactive board, while performing detailed zooms on a tabletop.

As mentioned earlier, in terms on infrastructure, it is similar to previously mentioned solutions. The centralized servers could be problematic in terms of latency and fault tolerance (though locally-maintained persistence mitigates the effects of the latter). It does provide an excellent example of how an infrastructure and framework for an interactive space, can provide coherent and environment-wide post-WIMP GUIs.

6.2.5 Service-oriented infrastructures

Of notable mention is service-oriented infrastructure, which is geared to providing “serendipitous interoperability” for multi-device interaction [Edwards 02, Kindberg 02]. One seminal example of this kind of infrastructure is Edward et al.'s Recombinant Computing, where a server with a small set of core functionality allows for the advertisement of discoverable services on a network. Once a client subscribes to a service, the server transfers relevant data and “mobile code” to this device, providing dynamic multi-device applications with ad hoc interoperability [Edwards 02]. *one.world* is another example of service-oriented infrastructure for pervasive computing [Grimm 04]. It uses a centralized service for tuple storage and provides a core set of services that are much more involved than the examples in Recombinant Computing, encompassing data storage, persistence through checkpointing, and remote interactions. Another more recent example is the multi-device framework 3MF for ad hoc multimodal interaction [Kaufmann 12].

Service-oriented architectures represent a successful software architectural style and their implementation and design patterns are a continued area of research, especially as mobile devices become more commonplace. The focus of our research in the TATIN-PIC architecture has excluded device discovery, as well as the ad hoc creation of interoperable functionality and the dynamic construction of user interfaces afforded by service-oriented architecture. These are all considerations that are worth investigating when moving forward.

6.2.6 Discussion

These solutions, whether object-oriented, data-oriented, or service-oriented, demonstrate a range of possibilities and emphasize different objectives of distributed infrastructure than those presented by TATIN-PIC (Table). Their biggest difference with the TATIN-PIC multi-agent system is that they each proposed custom-built toolkits, language extensions or software development frameworks in order to implement multi-device interaction. Therefore, they do not meet either one or both of the goals of platform toolkit independence for user interfaces (D1) or distributed infrastructure (D2). Of course, the tradeoff is that they provide organized and powerful libraries (often times open-source [Gjerlufsen 11, Jetter 12b]) that can significantly facilitate the work of software developers. But this restricts certain classes of platforms, or forces the use of wrappers or generic HTML web interfaces.

One issue that we have not taken into consideration is device discovery and serendipitous interoperability, a topic considered by all aforementioned infrastructure except ZOIL. Like ZOIL, we consider that all network IP addresses are known at run-time, which allows the different devices to communicate with each other. There are many ways for overcoming this limitation, and such issues have been previously addressed in the literature on multi-agent system. For example, FIPA provides specifications and standards so that a generic agent discovery service may be implemented [fip]. This could allow agents of different platforms to locate each other, as is done in [Ratsimor 04]. There also exist mobile agents [Nwana 96] which can leave one platform and enter another, providing a helpful abstraction for serendipitous interoperability. Multi-agent based solutions can even go further than serendipitous interoperability and provide serendipitous intelligence. This idea, more frequently referred to as emergence in multi-agent societies, has been explored by [Jamont 10] in the monitoring of complex physical systems.

Another issue that is revealed when examining this related work is that during the presentation of these frameworks, both shared substance and ZOIL provide useful qualitative data from workshops with software developers or class projects with computer science students where their frameworks were taught and used for the prototyping of multi-surface applications. These provide an evaluation from the perspective of the software developer as a user, and are useful for identifying the advantages and disadvantages of each approach.

There were eight student projects in the TATIN-PIC project, but seven of these focused primarily on local single-device applications which required only limited interaction with agents. Teaching and encouraging the use of the Application Manager and Application Supporter agents, for example, could have been interesting to provide student projects with the possibility of developing multi-surface applications. Instead, these agents were prebuilt by the TATIN-PIC team and provided to them. On the other hand, we were able to capitalize on the modular-

Table 6.11: Comparison of different approaches to distributed user interfaces.

Name	Distribution strategy	D1: Interface toolkit independence	D2: Infrastructure toolkit independence	R1: Rapid prototyping	R2: Flexible network connections	R3: Persistence	Other comments
iRos Middleware [Johanson 02a, Johanson 02b]	Event-driven tuplespace (EventHeap)	no	no	no	limited (uses a centralized server)	yes	Includes DataHeap for sharing of data, and iCrafter for interface generation
BEACH [Tandler 04]	Centralized server of shared objects	yes	no	no	limited (uses a centralized server)	yes	Details of implementation somewhat limited
Shared Substance [Gjerlufsen 11]	Data-oriented programming allows for easy replication or remote-access of application data	yes	no	yes	yes	not addressed	Data-oriented programming style proves to be extremely suitable for distributed user interfaces. Work remains to be done on architectural patterns.
ZOIL [Jetter 12b]	Centralized server of shared objects	no	no	no	limited (uses a centralized server)	yes	More focused on frontend (user interface) than backend (distributed objects/data),
TATIN-PIC <i>this thesis</i>	Message passing through software agents	yes	Any FIPA compliant multi-agent toolkit	yes, though somewhat limited	yes	yes	Architectural pattern rather than programming library.

ity of our infrastructure by a separation of concerns that allowed students to focus solely on the development of local applications, that were easily integration into the ecosystem of the TATIN-PIC environment. One additional student project was taught how to communicate with the Workbench Agent. These students developed the tablet-based Post-it application for the Android OS and were able to easily send and receive messages with the WorkbenchAgent on the interactive tabletop to create, delete or modify Post-it notes. This demonstrates the advantage of such a modular and distributed approach, as well as the platform independence used in our infrastructure. I leave more in-depth evaluations from the perspective of developers as a prospect for future work.

Conclusion and future work

Revisiting research goals

This thesis is a study of interactive spaces for co-located collaboration in preliminary design activities. In the introduction of this thesis, I identified two distinct research goals that have mutually influenced each other. The first is the design of an interactive space which could augment co-located collaboration for the preliminary design phase of engineering projects. The second is the design of a distributed infrastructure to support said interactive space.

The first research goal is approached from a human-computer interaction perspective and therefore leverages many user-centered design methods. Observations of users engaged in preliminary design activities are analyzed with the activity theory checklist in order to uncover how groups utilize traditional spaces for completing preliminary design activities. A literature review is conducted which revolves around four domains of interactive space: personal interaction, social interaction, organizational workflows, and physical environment. This is used to extract relevant design guidelines concerning these domains, and also to introduce the design principle of shareability.

An interactive space is proposed with special attention to the shareability of its physical configuration and its input devices. Its information landscape is presented which uses two multi-surface user interface layers to maintain a separation of concerns between project-centered activities and participant-centered activities. Several interaction techniques are also designed for grouping virtual components, transferring data across surfaces and across workflows, and performing user-specific multimodal interaction with personal assistant agents.

User evaluations are conducted, including a within-subjects comparative evaluation of brainstorming, a between-subjects comparative evaluation of project planning, and a case study of students performing causal analysis for a personal project. These evaluations find that interactive spaces do support the collaborative preliminary design activity because they either preserve or augment the group dynamics and collaboration styles of groups in preliminary design. Interactive spaces also provide well-received functionality for saving and returning to previous work and

for facilitating the distribution of electronic documents to all participants after a meeting is complete. Several of the proposed interaction techniques were also tested within the context of larger applications (e.g. grouping technique in project planning, portal technique in causal analysis). They performed relatively well, though some improvements are suggested in the case of the grouping technique. The general usability evaluation of the interactive space also revealed that text entry is still very much an open question in our proposed interactive space.

The second research goal is approached from a distributed computing perspective. The interactive space designed in the first question poses certain challenges with regard to its distributed infrastructure. Multi-agent societies are particularly well-suited for the task, providing a natural abstraction which leverages the inherent modularity and distribution afforded by agent-oriented programming.

The multi-agent society is designed using the Gaia methodology, providing three sub-organizations for managing projects and applications, personal user information and interaction, and persistence. Through the use of established organization rules in Gaia, the multi-agent system remains open and extensible for further additions and modifications.

This multi-agent infrastructure is fully implemented and used throughout the user evaluations of the interactive space. During the presentation of the implementation, the agent message content language is presented which allows different agents to trigger user-specific actions in the interactive space inside applications which use the Post-it note metaphor. With this common content language established, new agents can be developed and introduced at run-time that capitalize on this established model of communication to also perform similar actions.

Contributions

The first main contribution of this thesis is an extensive presentation of an interactive space, which at its core uses a configuration not yet fully explored in previous literature: a large multitouch tabletop and a large multitouch interactive board display. Due to its distribution of entry points and access points for teamwork, it is an appropriate tool for preliminary design activities, as well as other activities which require similar qualities of shareability.

Related to this contribution, is the abstraction used in the information landscape of the interactive space to provide a separation of concerns between project-oriented work and participant-oriented work, as well as several of the interaction techniques. These have already been reused in the continued development of the TATIN-PIC project, to introduce new applications for other preliminary design activities, such as a functional analysis application and a general flowchart application.

To a certain extent, the evaluation of this interactive space also serves as a

contribution in and of itself. It seeks to go further than some other evaluations of interactive spaces by using comparative evaluations, increasing the number of participants used in groups, and does so within the context of activities beyond the prototypical brainstorming scenario.

The second main contribution of this thesis is an implementation plan for a multi-agent infrastructure for the distributed computing environment which effectively accommodates a wide range of platforms and devices in concerted interaction. Because it uses agent-oriented programming and the Gaia methodology, the infrastructure can be applied to a variety of interactive spaces. This approach is different than others in the related work, because it does not prescribe the use of any specific user interface toolkits or distributed infrastructure toolkits (though it does imposing a message-driven content language), providing freedom and flexibility to the designers of the interactive systems.

Future work

There are many possible avenues of future work. Some have been previously mentioned in the conclusion of the usability evaluations. For example, the grouping technique requires some extension to allow for top-down hierarchization in conjunction with its bottom-up approach. Text entry should be directly considered either by providing better support for personal predictive text auto-completion and better support for copying and pasting, or by providing physical wireless keyboards that can be used with the tabletop in mixed-reality interaction.

One issue with the evaluations is that I did not provide user-testing of the multimodal interaction techniques and the personal assistant agent. This is because the implementation still requires more attention so that it can effectively and robustly support co-located multi-user speech recognition. The current implementation requires much user-specific acoustic model training and calibration to set up, and in the context of the user evaluations, the five or six participants would also require some time to become acclimated with their personal assistant agents. Therefore, future work might include using more sophisticated voice recognition software (which does not require as much training of user's acoustic models) and an easier set-up. New experimental protocols and evaluations could also be designed. These might compare the multimodal personal assistant agent approach, to a multitouch-only approach, or to the multimodal oracle agent approach (presented in Chapter 3).

This thesis begins by considering groups that are using traditional tools to perform traditional activities of preliminary design. This thesis advocates replacing the traditional tools with new ones which can better perform the traditional activities of preliminary design. In reality, and according to activity theory, the introduction of these new tools is irrevocably changing the preliminary design ac-

tivities that it seeks to support. For the purpose of this thesis, the preliminary design activities were held constant and used as a point of comparison. But interactive spaces can provide new approaches and alternative workflows within the preliminary design methodology. With the design of an interactive space complete, future work can better explore the relationship between groups, interactive spaces, and new preliminary design methodologies, specifically through longitudinal evaluations. This work is already being undertaken by Andrea Guerra in his PhD thesis at the Université de Technologie de Compiègne.

Finally, instead of co-located collaboration in interactive spaces for preliminary design, future work may cross the semantic Rubicon and consider co-located collaboration in *intelligent environments* for preliminary design. The semantic Rubicon, as defined by Kindberg and Fox, is “the division between system and user for high-level decision-making or physical-world semantics processing” [Kindberg 02]. The focus on the thesis has been to augment the decision-making capacities of co-located collaborative teams, allowing the users to better make connections and obtain insight in the information they are creating and manipulating. However, as the project database contains more and more information through continued use of the environment, artificial intelligence and knowledge engineering solutions will be increasingly relevant for managing connections and relations in the data.

To this end, much of the foundation is set: for example, the multi-agent infrastructure can be extended to provide distributed intelligence, the artificial intelligence of the personal assistant agents can also be improved. Moreover, the TATIN-PIC project has already conducted significant research in building ontologies for preliminary design and for the actions of users involved in the process [Joiron 13b, Joiron 13a]. If such functionality is introduced into the interactive space, with proper consideration given to the existing domains of interactive spaces (personal interaction, social communication, organizational workflows, and the physical environment) the results would be extremely beneficial for collaborative environments.

Reflection

When I began working on the subject of interactive spaces three years ago, my initial approach was to ask myself the following: *What would be the optimal interactive space for supporting co-located collaboration?* This led me to consider a great number of different approaches taken by related work, especially in terms of input devices, such as styluses, pointing wands, tangibles, mixed-reality objects, in-the-air gestures, etc, as well as possible display sizes, configurations and form factors. I eventually realized it is difficult comparing something like Beaudouin-Lafon et al. ’s massive 32-display WILD room [Beaudouin-Lafon 11] to another thing such as Geyer et al. ’s elegant mixed-reality AffinityTable [Geyer 11a]. The

physical structure of the two rooms suggest something entirely different even before the devices are turned on. Perhaps no matter how well the interaction techniques or input devices are designed, the AffinityTable may never feel right for the exploration of large data sets, while the WILD room would be too spread out to maintain a focused and heated group brainstorming session. There are multiple “optimal” interactive spaces, depending on the nature of the activity that space wishes to support.

This is what motivated me to refocus my efforts on specific preliminary design activities such as project planning and causal analysis (an opportunity afforded to me within the context of the project on which I was working) as well as to consider interactive spaces in terms of their shareability in Chapter 3. Thinking about interactive spaces in terms of their access points and entry points reveals a great deal about the kind of co-located collaboration that the environment can support. The TATIN-PIC dual-surface multi-touch approach is an excellent platform for supporting co-located collaboration in preliminary design and I look forward to continued research and experimentation in this and other interactive spaces.

I also benefited enormously from the expertise of researchers at the Université de Technologie de Compiègne regarding multi-agent systems. Agent-oriented programming is a natural fit for distributed systems, and I have only touched upon a fraction of its potential in the context of interactive spaces and intelligent environments. I hope what I have contributed here can be of use to future practitioners of interactive space infrastructure, and look forward to a variety of possible extensions and modifications that may come from continuing research in this domain.

My research conducted for this thesis has been inspired by several disciplines and has included a wide range of topics within human-computer interaction, such as computer-supported cooperative work, ubiquitous computing, and software engineering. I hope the use of theory, design, technological implementation, and evaluation in my approach has provided a clear and helpful understanding of co-located collaboration in interactive spaces and I hope my contribution will assist in the application of these environments in the context of preliminary design activities in the future.

Appendix

Appendix A

Preliminary Design Project Ontology

This appendix provides the preliminary design project ontology that was developed in collaboration with the MIS laboratory at Université de Picardie Jules Verne. Figure A.1 refers to collective actions, such as brainstorming and risk analysis, that are undertaken by the project team during sessions. Figure A.2 describes the idea of documents that are used in the TATIN-PIC. These ontologies extend the high-level ontology DOLCE-CORE [Gangemi 02] and were developed using the OntoSpec method. They are presented in greater detail in [Joirion 13b].

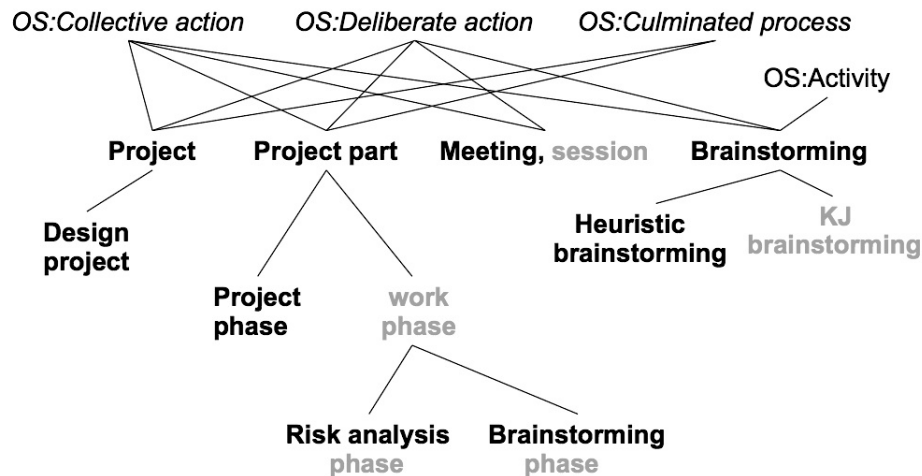


Figure A.1: Ontology describing collective actions that are undertaken by the team.

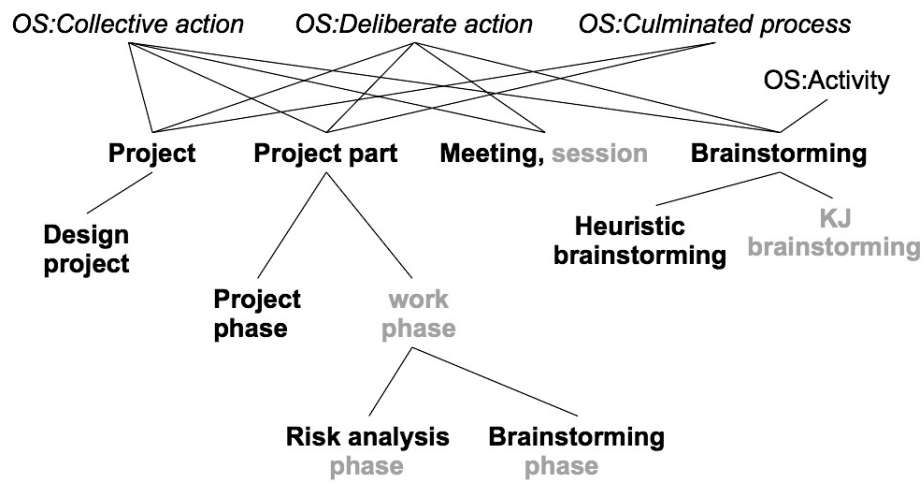


Figure A.2: Ontology describing documents used in TATIN-PIC.

Appendix B

Multi-level grouping technique

This section presents the entirety of the grouping technique through illustrations.

- Figure B.1 shows the creation of a group.
- Figure B.2 shows how items and labels are added to a group.
- Figure B.3 illustrates how groups can be contracted and expanded through a tap gesture, so that they may be more space efficient.
- Figure B.4 shows how a subgroup may be added to a group.
- Figure B.5 shows how an expansion of a subgroup may be performed.
- Figure B.6 illustrates how an item or a subgroup may be removed.
- Figure B.7 demonstrates what happens to the label of a group if it has no more members.

These images were originally developed by Atman Kendira for a patent application. The Université de Technologie de Compiègne currently holds a patent for this interaction technique.

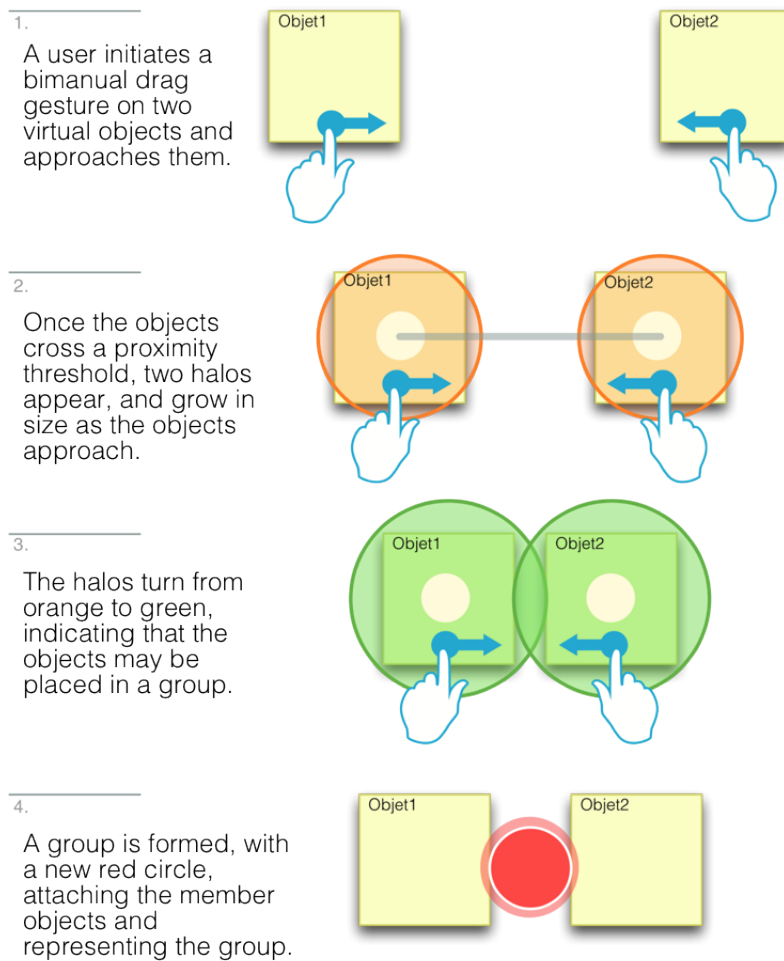


Figure B.1: Creation of a group of objects.

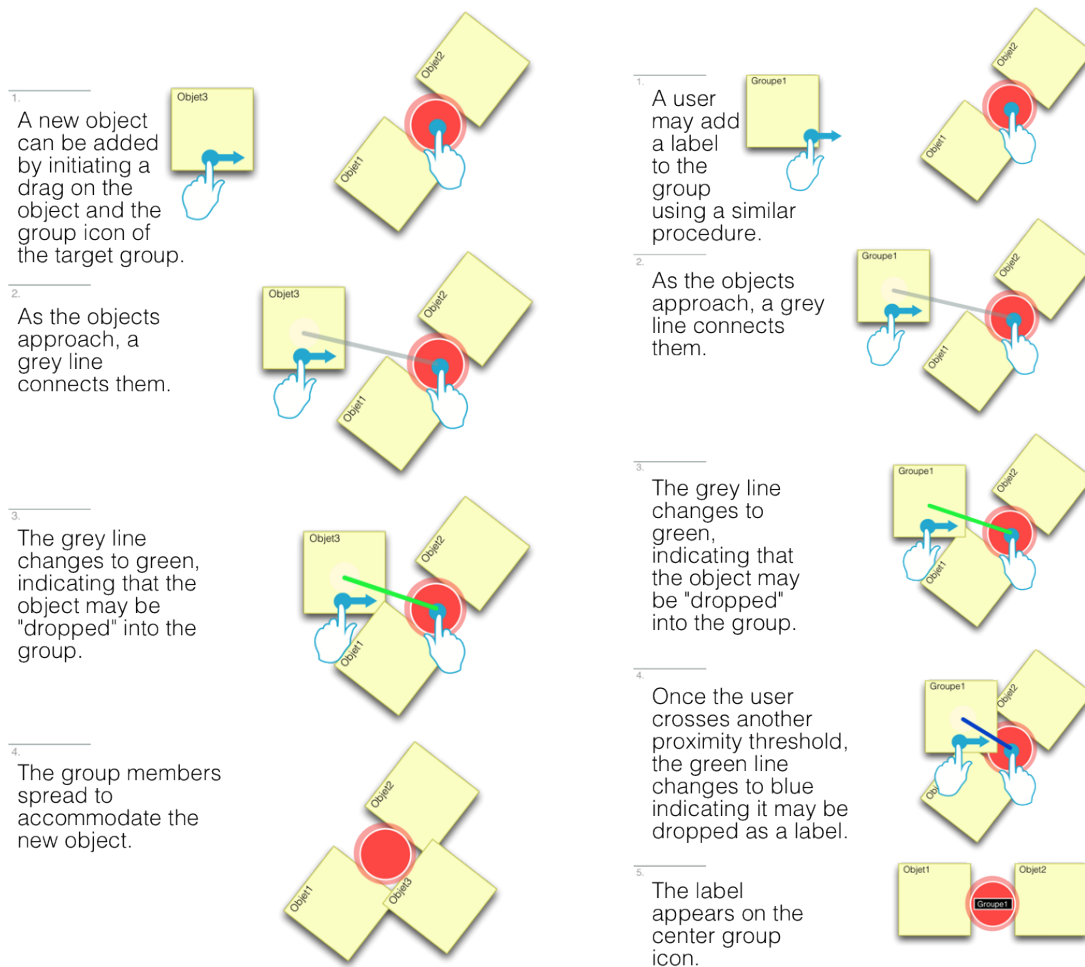


Figure B.2: Adding an object or a label to a group.

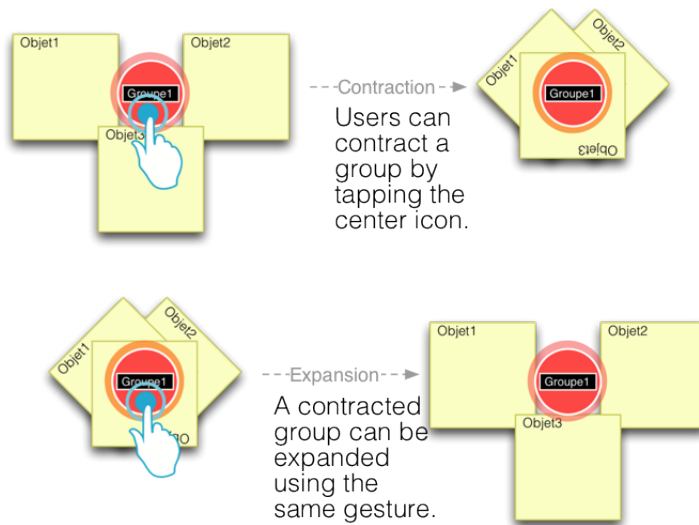


Figure B.3: Expansion and contraction of groups.

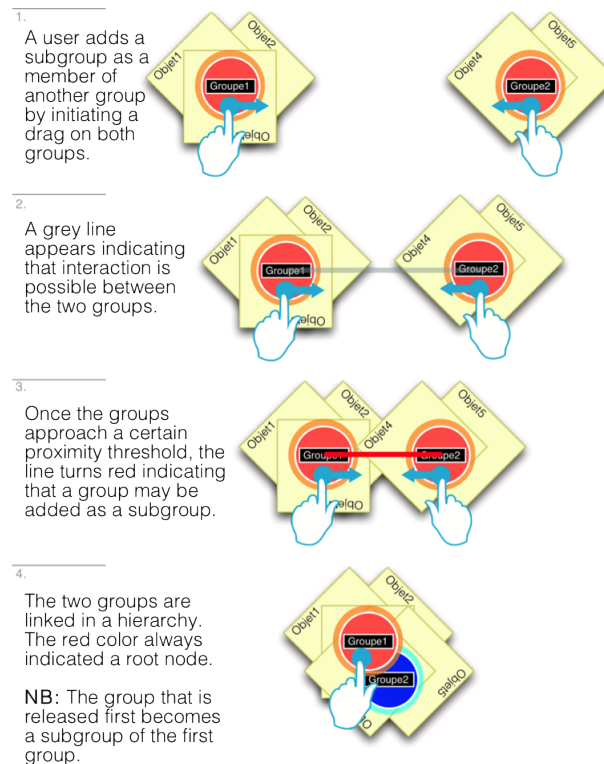


Figure B.4: Adding a subgroup.

1. Subgroups maintain their original functionality; items and labels can still be added and removed.

In this example, a user expands two levels of a group, in order to better access the subgroup.

2.

3.

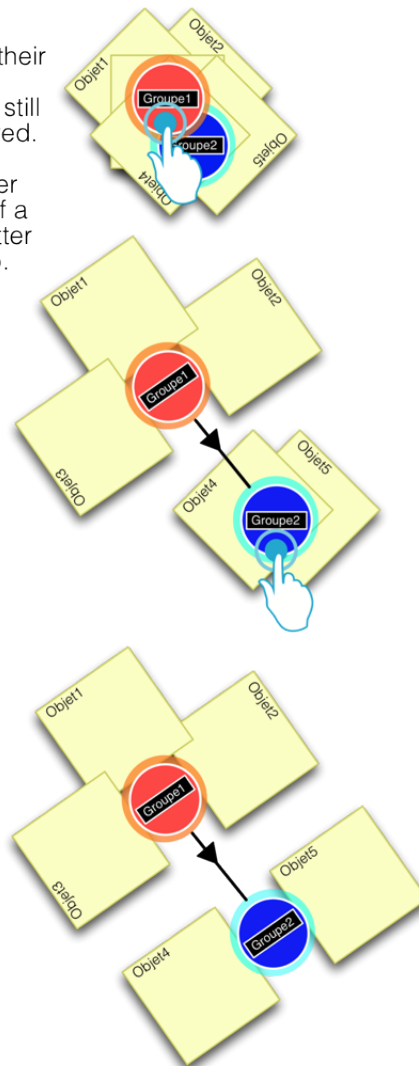


Figure B.5: Expanding groups and subgroups.

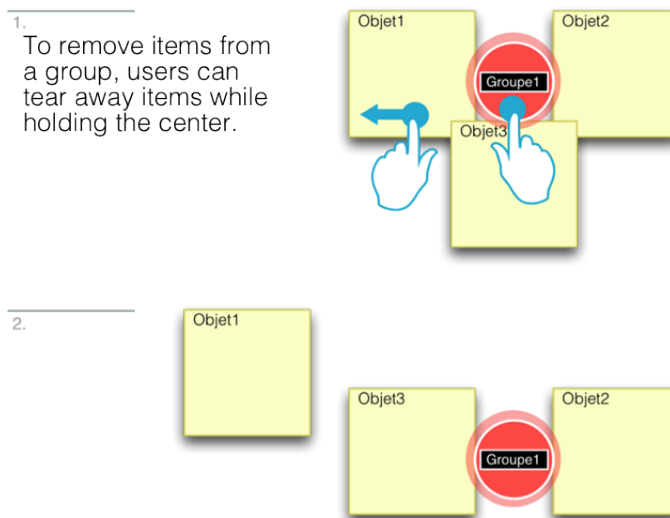


Figure B.6: Removing an object from a group. This gesture is illustrated here with an item, but also applies to subgroups.

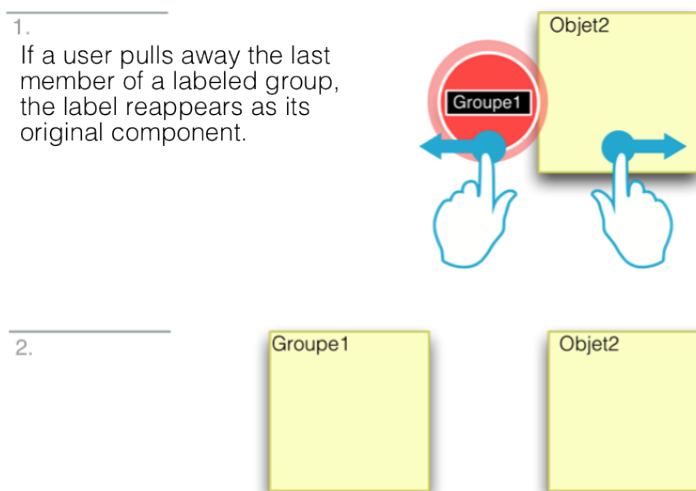


Figure B.7: Removing the last object from a group.

Appendix C

Models of Gaia analysis

This appendix includes all models which were used in the Gaia analysis of the system. This includes the role models, the interaction models, and the service model.

C.1 Role schemas

The section provides the complete list of role schema for the roles in the TATIN-PIC multi-agent society.

- Figure C.1 provides the role schema for the PROJECTMANAGEMENT role.
- Figure C.2 provides the role schema for the PROJECTSUPPORT role.
- Figure C.3 provides the role schema for the APPLICATIONMANAGEMENT role.
- Figure C.6 provides the role schema for the APPLICATIONSUPPORT role.
- Figure C.5 provides the role schema for the PERSISTENCEMANAGEMENT role.
- Figure C.6 provides the role schema for the PERSISTENCESUPPORT role.
- Figure C.7 provides the role schema for the USERIDENTIFICATIONONSHARED-SURFACE role.
- Figure C.8 provides the role schema for the USERPRIMARYSHAREDDEVICE-SUPPORT role.

- Figure C.9 provides the role schema for the USERPERSONALDEVICESTUPPORT role.
- Figure C.10 provides the role schema for the USERPERSONALASSISTANT role.

Role Schema: PROJECTMANAGEMENT	
Description:	
This role involves supporting a project which is created or opened on a primary shared surface and enabling the other project support roles in the environment by granting access them to project data, and informing them of application switches, and new documents that are created.	
Protocols and Activities:	
AwaitActivitySwitch, InformOthersOfActivitySwitch, AwaitTransferRequest, TransferDocument, TransferListOfDocumentNames, <u>AwaitDocumentCreation</u> , InformOthersOfDocumentCreation	
Permissions:	
reads	supplied <i>masterProjectDocumentList</i> // List of all documents of a project
changes	supplied <i>masterActiveDocument</i> // Link to the local active document (scene and model)
Responsibilities	
Liveness:	
PROJECTMANAGEMENT = (ManageCurrentActivity ManageResourceAccess ManageDocumentCreation)	
MANAGECURRENTACTIVITY = (AwaitActivitySwitch. InformOthersOfActivitySwitch) ^ω	
MANAGERESOURCEACCESS = (AwaitTransferRequest. (TransferDocument TransferListOfDocumentNames)) ^ω	
MANAGEDOCUMENTCREATION = (<u>AwaitDocumentCreation</u> . InformOthersOfDocumentCreation) ^ω	
Safety:	
true	

Figure C.1: PROJECTMANAGEMENT role schema.

Role Schema: PROJECTSUPPORT[i], i = total_ancillary_shared_devices							
Description: This roles involves supporting an ancillary shared device in the environment by managing local access to project data, and keeping the device in sync with the activity of the primary shared device.							
Protocols and Activities: RequestListOfDocumentsNames, AwaitListOfDocumentNames, RequestDocument, AwaitActivitySwitch, SwitchActivity, AwaitNewDocumentInfo, <u>AddDocumentToList</u>							
Permissions: <table><tr><td>changes</td><td>supplied localProjectDocumentList[i]</td><td>// List of all documents of a project</td></tr><tr><td></td><td>supplied activeDocument[i]</td><td>// Link to the local active document (scene and model)</td></tr></table>		changes	supplied localProjectDocumentList[i]	// List of all documents of a project		supplied activeDocument[i]	// Link to the local active document (scene and model)
changes	supplied localProjectDocumentList[i]	// List of all documents of a project					
	supplied activeDocument[i]	// Link to the local active document (scene and model)					
Responsibilities Liveness: PROJECTSUPPORT[i] = (RequestListOfDocumentNames. AwaitListOfDocumentNames. (MaintainActivity ManageDocuments) ^ω MAINTAINACTIVITY = (AwaitActivitySwitch. [RequestDocument]. SwitchActivity) MANAGEDOCUMENTS = (AwaitNewDocumentInfo. <u>AddDocumentToList</u>) Safety: activeDocument = nil => RequestDocument							

Figure C.2: PROJECTSUPPORT role schema.

Role Schema: APPLICATIONMANAGEMENT[k], $k = total_applications$		
Description: Because applications run across multiple devices, coordination is required between devices to manage user interface components. The application manager role can trigger and respond to application-level events on other devices. The application manager role is also responsible for <i>importing</i> another document into the current document. (e.g. importing brainstorming document into project planning)		
Protocols and Activities: AwaitImportRequest, ImportDocumentDataIntoCurrentDocument, InformSupporters, AwaitInterdeviceAction, TriggerInterdeviceAction, SuspendActivity, <u>Sleep</u> , ResumeActivity, ConductActivity		
Permissions:		
generates	<i>applicationEvent</i>	// Any data used for interdevice activity
changes	supplied <i>masterActiveDocument</i>	// Link to the local active document (scene and model)
reads	supplied <i>masterActiveDocument.applicationType</i>	// Describes the application which opens this document
Responsibilities		
Liveness:		
APPLICATIONMANAGEMENT[k] = (SuspendActivity. <u>Sleep</u> ^ω . ResumeActivity. ConductActivity ^ω) ^ω		
CONDUCTACTIVITY = (ManageImportRequests ^ω ManageInterdeviceAction ^ω)		
MANAGEIMPORTREQUESTS = (<u>AwaitImportRequest</u> . <u>ImportDocumentDataIntoCurrentDocument</u> . InformSupporters)		
MANAGEINTERDEVICEACTION = (AwaitInterdeviceAction TriggerInterdeviceAction.)		
Safety:		
<i>masterActiveDocument.applicationType</i> = k		

Figure C.3: APPLICATIONMANAGEMENT role schema.

Role Schema: APPLICATIONSUPPORT[i][k], $i = total_ancillary_shared_devices$, $k = total_applications$		
Description: The application support role can trigger and respond to application-level events on other devices. (e.g. inter-device portal interaction technique)		
Protocols and Activities: AwaitInterdeviceAction, TriggerInterdeviceAction, AwaitImportAction, UpdateLocalModel, SuspendActivity, <u>Sleep</u> , ResumeActivity, ConductActivity		
Permissions:		
generates	<i>applicationEvent</i>	// Any data used for interdevice functionality
changes	supplied <i>activeDocument</i> [i]	// Link to the local active document (scene and model)
reads	supplied <i>activeDocument</i> [i]. <i>applicationType</i>	// Describes the application which opens this document
Responsibilities		
Liveness: APPLICATIONSUPPORT[i][k] = (SuspendActivity. <u>Sleep</u> ^{ω} . ResumeActivity. ConductActivity) ^{ω} CONDUCTACTIVITY = (AwaitInterdeviceAction ^{ω} TriggerInterdeviceAction ^{ω} (AwaitImportAction ^{ω} . UpdateLocalModel))		
Safety: <i>activeDocument</i> [i]. <i>applicationType</i> = k		

Figure C.4: APPLICATIONSUPPORT role schema.

Role Schema: PERSISTENCEMANAGEMENT	
Description: This role is responsible for storing the current project information, including all shared document data on other devices, into a persistent database or file structure.	
Protocols and Activities: <u>AwaitSaveTick</u> , <u>SaveProjectDocuments</u> , <u>AwaitLoadProjectRequest</u> , <u>LoadProject</u> , <u>AwaitSupporterProjectData</u> , <u>SynchronizeSupporterProjectData</u>	
Permissions:	
reads	supplied <i>masterProjectDocumentList</i> // List of all project document files
changes	supplied <i>projectDatabase</i> // Database of every project and all files
generates	<i>projectDocumentDatabaseEntry</i> // A database entry representing the current project
Responsibilities	
Liveness: $\text{PERSISTENCEMANAGEMENT} = (\text{ManageSave} \parallel \text{ManageLoad} \parallel \text{ManageSupporters})^\omega$ $\text{MANAGESAVE} = (\text{AwaitSaveTick}, \text{SaveProjectDocuments})$ $\text{MANAGELoad} = (\text{AwaitLoadProjectRequest}, \text{LoadProject})$ $\text{MANAGESUPPORTERS} = (\text{AwaitSupporterProjectData}, \text{SynchronizeSupporterProjectData})$	
Safety: true	

Figure C.5: PERSISTENCEMANAGEMENT role schema.

Role Schema: PERSISTENCESUPPORT[<i>i</i>], <i>i</i> = <i>total_ancillary_shared_devices</i>	
Description: This role is responsible for coordinating with the Persistent Manager Role and sending local document information to be synchronized and rendered persistent.	
Protocols and Activities: <u>AwaitSynchronizeTick</u> , <u>SendLocalDocumentsToPersistentManager</u>	
Permissions:	
reads	supplied <i>activeDocument[i]</i> // Link to the local active document (scene and model)
Responsibilities	
Liveness: $\text{PERSISTENCESUPPORT} = (\text{AwaitSynchronizeTick}, \text{SendLocalDocumentsToPersistentManager})^\omega$	
Safety: true	

Figure C.6: PERSISTENCESUPPORT role schema.

Role Schema: USERIDENTIFICATIONONSHARED SURFACE	
Description:	Takes <i>interactionEvents</i> that occur on a user-agnostic shared surface, identifies their owner, and transfers them to the corresponding user's local agent (e.g. on the primary shared surface, this would be the agent with the USERPRIMARYSHAREDDEVICESTUPPORT role.)
Protocols and Activities:	AwaitUserEvent, <u>IdentifyOwner</u> , TransmitUserEventToLocalUserAgent
Permissions:	reads supplied <i>interactionEvent</i> // An interaction event on a shared surface
Responsibilities	
Liveness:	USERIDENTIFICATIONONSHARED SURFACE = (AwaitUserEvent. <u>IdentifyOwner</u> . TransmitUserEventToLocalUserAgent) ^ω
Safety:	true

Figure C.7: USERIDENTIFICATIONONSHARED SURFACE role schema.

Role Schema: USERPRIMARYSHAREDDEVICESUPPORT[p], $p = total_participants$		
Description:		
This role serves as link between the active document on a shared device and the user's personal access point on the that shared device. This role is responsible for executing actions on the information resources on this device which come from other roles. Requests to modify the local document model or scene, may come from other agents serving the same user.		
Protocols and Activities:		
AwaitDistalModifyDocumentRequest, ModifyDocumentModel, AwaitLocalUserEvent, InformOtherUserAgentsOfUserAction		
Permissions:		
reads	supplied $userID[p]$	// ID of the user so it may coordinate with other roles
changes	supplied $masterActiveDocument$	// Link to the local active document (scene and model)
	supplied $userModel[p]$	// Any relevant user-specific information (e.g. selected items, previous actions)
generates	$userEvent$	
Responsibilities		
Liveness:		
USERSHAREDDEVICESUPPORT[p] = (ManageDocumentEvents ManageUserSpecificEvents) $^{\omega}$		
MANAGEDOCUMENTEVENTS = (AwaitExternalModifyModelRequest. ModifyDocumentModel)		
MANAGEUSERSPECIFICEVENT = (AwaitLocalUserEvent. InformOtherUserAgentsOfUserAction)		
Safety:		
true		

Figure C.8: USERPRIMARYSHAREDDEVICESUPPORT role schema.

Role Schema: USERPERSONALDEVICESUPPORT[p], $p = total_participants$		
Description:		
If a user has a personal device, then this role is responsible for the coordination (with the user's other agents on other devices) of the user's information resources across all devices. It may follow the user's actions in the environment or it request specific modifications of the project document on the primary shared device.		
Protocols and Activities:		
AwaitKnowledgeOfDistalUserAction, <u>UpdateModel</u> , <u>AwaitKnowledgeOfLocalUserAction</u> , TransferRequestToOtherUserAgents		
Permissions:		
reads	supplied $userID[p]$	// ID of the user so it may coordinate with other roles
changes	supplied $userPersonalDeviceModel[p]$	// Any relevant user-specific information (e.g. selected items, previous actions)
generates	$userEvent$	// A user event on a shared surface
Responsibilities		
Liveness:		
$USERPERSONALDEVICESUPPORT[q] = (ManageDistalEvents \parallel ManageLocalEvents)^\omega$ $MANAGEDISTALEVENTS = (AwaitKnowledgeOfDistalUserAction. \underline{UpdateModel})$ $MANAGELocalevents = (\underline{AwaitKnowledgeOfLocalUserAction}.$ $\underline{TransferRequestToOtherUserAgents})$		
Safety:		
true		

Figure C.9: USERPERSONALDEVICESUPPORT role schema.

Role Schema: USERPERSONALASSISTANT[r], $p = total_participants$		
Description:		
This role is responsible for all verbal exchanges with the user, including information requests, and multimodal fusion.		
Protocols and Activities:		
AwaitVocalRequest, ProcessNaturalLanguageRequest, TransferRequestToOtherUserAgents, AwaitConfirmation, <u>CommunicateConfirmationToUser</u> , AwaitKnowledgeOfDistalUserAction, UpdateModel		
Permissions:		
reads	supplied $userID[p]$	// ID of the user so it may coordinate with other roles
changes	supplied $userPAModel[p]$	// Any relevant user-specific information (e. g. selected items, previous actions)
generates	$userEvent$	// An interaction event
Responsibilities		
Liveness:		
USERPERSONALASSISTANT[r] = (ProcessVocal ManageDistalEvents) ^ω		
MANAGEDISTALEVENTS = (AwaitKnowledgeOfDistalUserAction. <u>UpdateModel</u>)		
PROCESSVOCAL = (AwaitVocalRequest. ProcessNaturalLanguageRequest.		
TransferRequestToOtherUserAgent. AwaitConfirmation. [<u>CommunicateConfirmationToUser</u>])		
Safety:		
true		

Figure C.10: USERPERSONALASSISTANT role schema.

C.2 Interaction Models

The legend of the interaction model is described in Figure 4.3.3.

- Figure 4.3.4 illustrates the exchange between the PROJECTMANAGEMENT role and the PROJECTSUPPORT role in order to synchronize project lists.
- Figure 4.3.5 illustrates the exchange between the PROJECTMANAGEMENT role, PROJECTSUPPORT role, APPLICATIONMANAGEMENT role and the APPLICATIONSUPPORT role during activity switches.
- Figure C.4 illustrates the exchange between the APPLICATIONMANAGEMENT role and the APPLICATIONSUPPORT role when one document is imported into an existing document.
- Figure C.5 illustrates the exchange, at an application-level, of modifications that occurs to the active document between the APPLICATIONMANAGEMENT and APPLICATIONSUPPORT roles. Note that their interaction pat-

terns, and roles, are similar. Its possible to capture this using only one role, but we use two different roles to emphasize the different information resources that the roles have access to (*masterActiveDocument* and *activeDocument*).

- Figure C.6 illustrates the exchange between the PERSISTENCEMANAGER and PERSISTENCESUPPORT, when the latter sends the former a copy of its local document model for storage.
- Figure C.7 illustrates how *interactionEvents* are passed from the UserIdentificationOnSharedSurface role to the appropriate USERPRIMARYSHARED-DEVICESUPPORT role, which then generates a *userEvent* and transfers this to other roles serving the same user.
- Figure C.8 illustrates the exchanges which are initiated by USERPERSONALDEVICE role and the UserPersonalAssistant role.

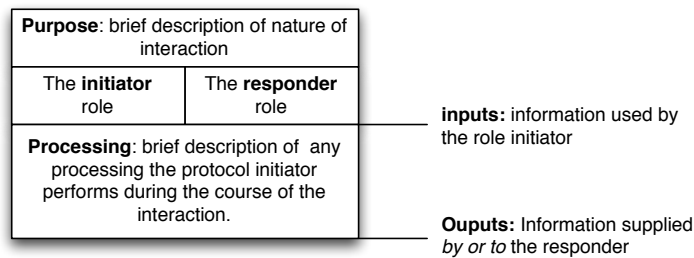


Figure C.1: The interaction model legend.

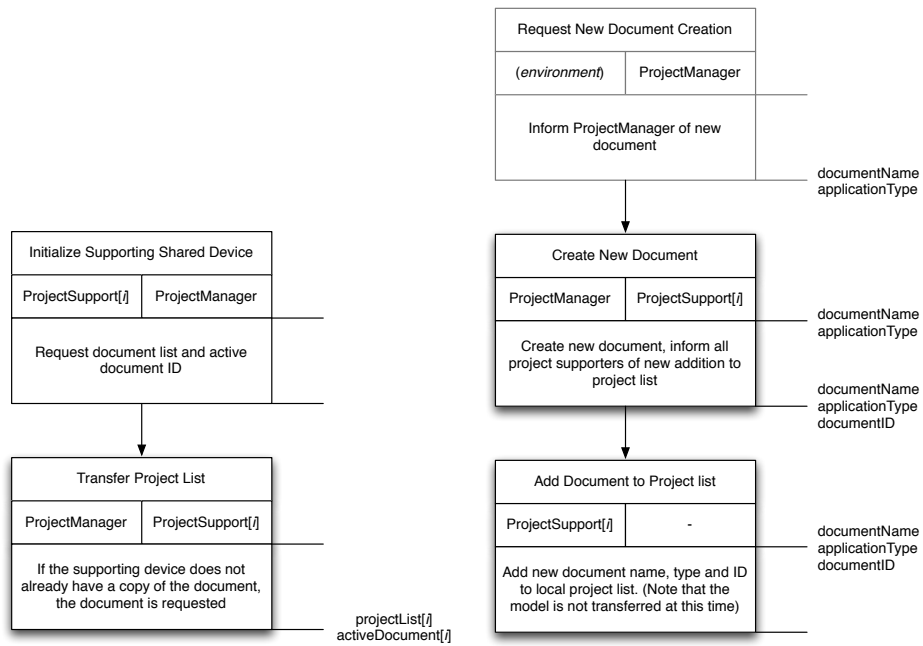


Figure C.2: Interaction model associated with PROJECTMANAGER and PROJECT-SUPPORT roles.

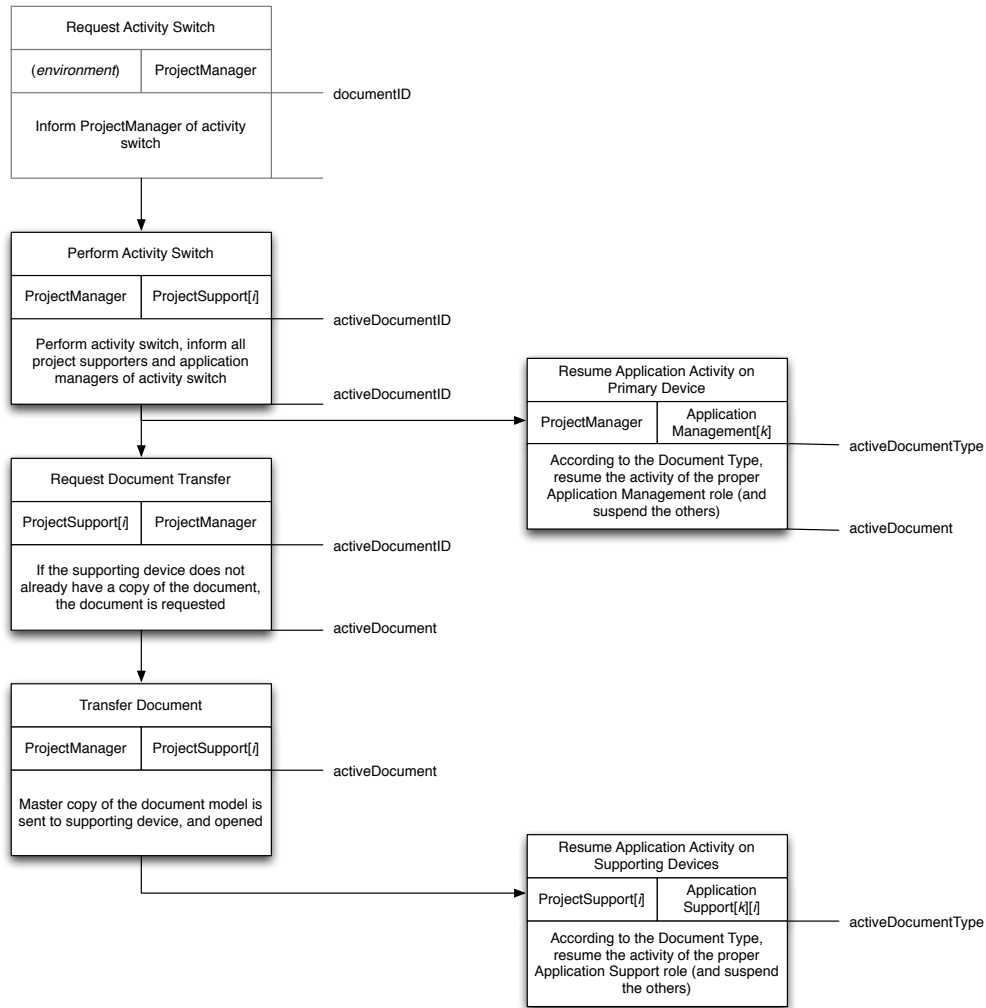


Figure C.3: Interaction model associated with an activity switch, involving the PROJECTMANAGER, PROJECTSUPPORT, APPLICATIONMANAGER and APPLICATIONSUPPORT roles.

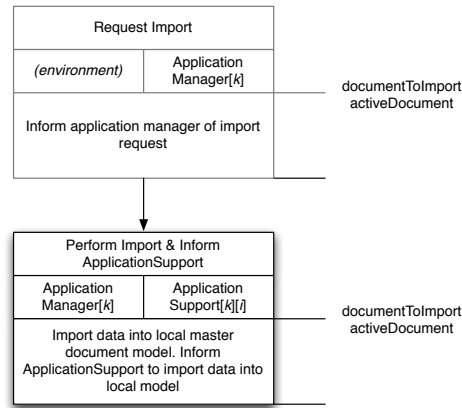


Figure C.4: Interaction model associated with APPLICATIONMANAGER and APPLICATIONSUPPORT roles regarding document importation.

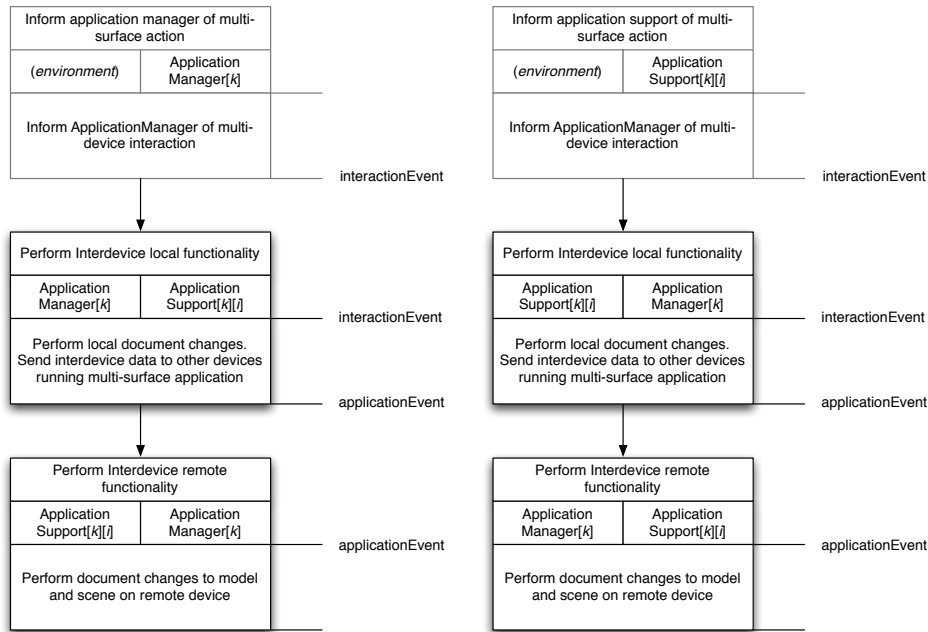


Figure C.5: Interaction model associated with APPLICATIONMANAGER and APPLICATIONSUPPORT roles regarding inter-device functionality.

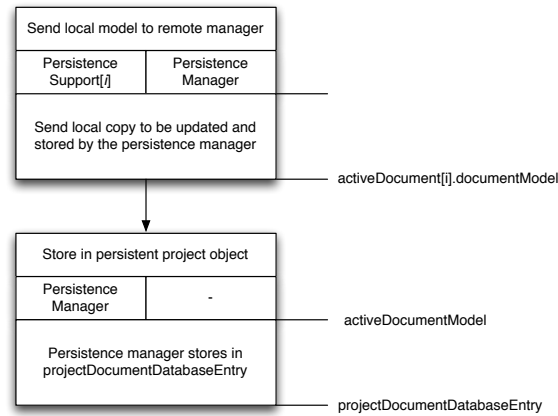


Figure C.6: Interaction model associated with PERSISTENCEMANAGER and PERSISTENCESUPPORT roles.

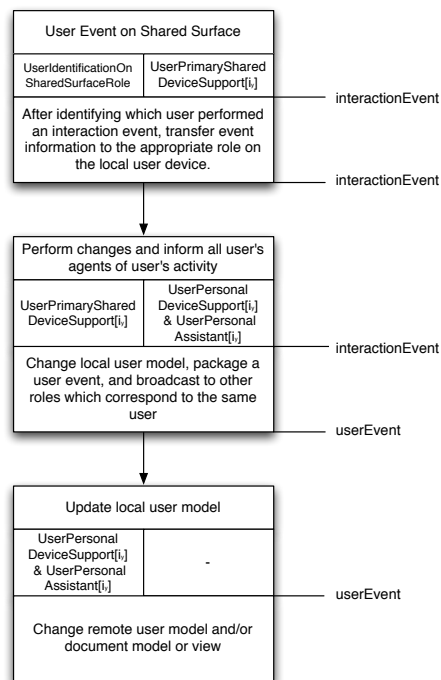


Figure C.7: Interaction model associated with USERIDENTIFICATIONONSHARED-SURFACE and USERPRIMARYSHAREDDEVICESUPPORT role.

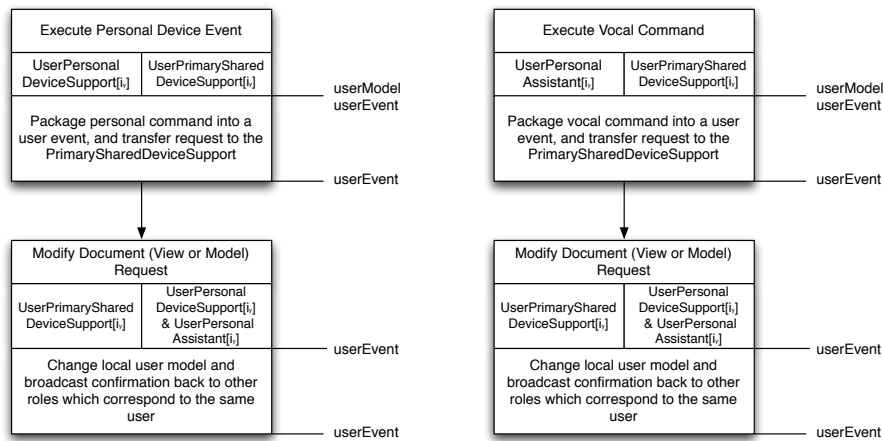


Figure C.8: Interaction model associated with USERPERSONALDEVICESUPPORT role and USERPERSONALASSISTANT role.

C.3 Service Model

This section presents the service model of the TATIN-PIC multi-agent infrastructure.

- Table C.1 presents the service related to the project, application, and persistence functionalities of the infrastructure.
- Table C.2 presents the services related to the sub-organization of a user's agents.

Table C.1: The services model for the Project and Persistence sub-organizations.

Services	Inputs	Outputs	Pre-conditions	Post-conditions
Open an existing project from database	<i>projectName</i>	<i>masterProject-DocumentList</i>	true	true
Initialize supporting device	-	<i>localProject-DocumentList</i>	true	$\forall i, localProjectDocumentList[i].documentName = masterProjectDocumentList[i].documentName$
Create a new document	<i>documentName, applicationType</i>	<i>documentID</i>	true	document is added to the <i>masterProjectList</i> and all <i>localProjectList</i>
Auto-save project	-	<i>projectDatabaseEntry</i>	true	true
Switch activity (open document)	<i>documentID</i>	-	new <i>documentID</i> \neq current <i>documentID</i>	<i>activeDocument</i> and <i>masterActiveDocument</i> have new <i>documentID</i> ; application change if necessary
Change application	<i>appType</i>	-	new <i>appType</i> \neq current <i>appType</i>	true
Perform multi-surface interaction	<i>interfaceEvent</i>	<i>interdeviceEvent</i>	true	true
Import document into current document	<i>documentID, master-ActiveDocument</i>	-	imported <i>documentID</i> \neq current <i>documentID</i>	true

Table C.2: The services model for the user focused sub-organization.

Services	Inputs	Outputs	Pre-conditions	Post-conditions
Inform all user agents of user action on primary device	<i>interactionEvent</i>	<i>userEvent</i>	true	<i>localUserModel</i> on user's personal device and user's personal assistant are updated
Process distal user request on primary device	<i>userEvent</i> , <i>masterActiveDocument</i>	-	true	Workbench agent updates <i>masterActiveDocument</i> model
Execute vocal request	<i>localUserModel</i>	<i>userEvent</i>	Voice recognition software places voice text inside the <i>localUserModel</i>	<i>userEvent</i> is transferred to the Workbench Agent, who will return a confirmation
Vocally communicate confirmation to user	<i>userEvent</i>	-	Workbench agent transfers a confirmation from a previous action	Voice synthesis software "speaks" the response generated by the personal assistant agent
Execute personal device request	<i>localUserModel</i>	<i>userEvent</i>	User performs action on personal device application	<i>userEvent</i> is transferred to the Workbench Agent, who will return a confirmation
Confirm completion of action on personal device	<i>userEvent</i>	-	Workbench agent transfers a confirmation from a previous action	Application on personal device is update to reflect changes

Appendix D

Supplementary materials from evaluations

This section includes supplementary material from the evaluations.

- Figure D.1 shows the complete results from the usability criteria questionnaire from the brainstorming evaluations. This was distributed to the participants in the experimental TATIN-PIC condition only.
- The following six pages are from the design brief (written in French) from the first and second sessions of the project planning evaluation. These design briefs were heavily based on a set exercises from the Gidel and Zonghero's *Management du projet, vol. 3* [Gidel 07].
 - The first design brief contains the description of the activity and the deliverables, as well as the list tasks and phases that was distributed for the task identification phase, and also the empty Gantt chart with the task hierarchy.
 - The second design brief contains the description of the activity and the deliverables for the second session.

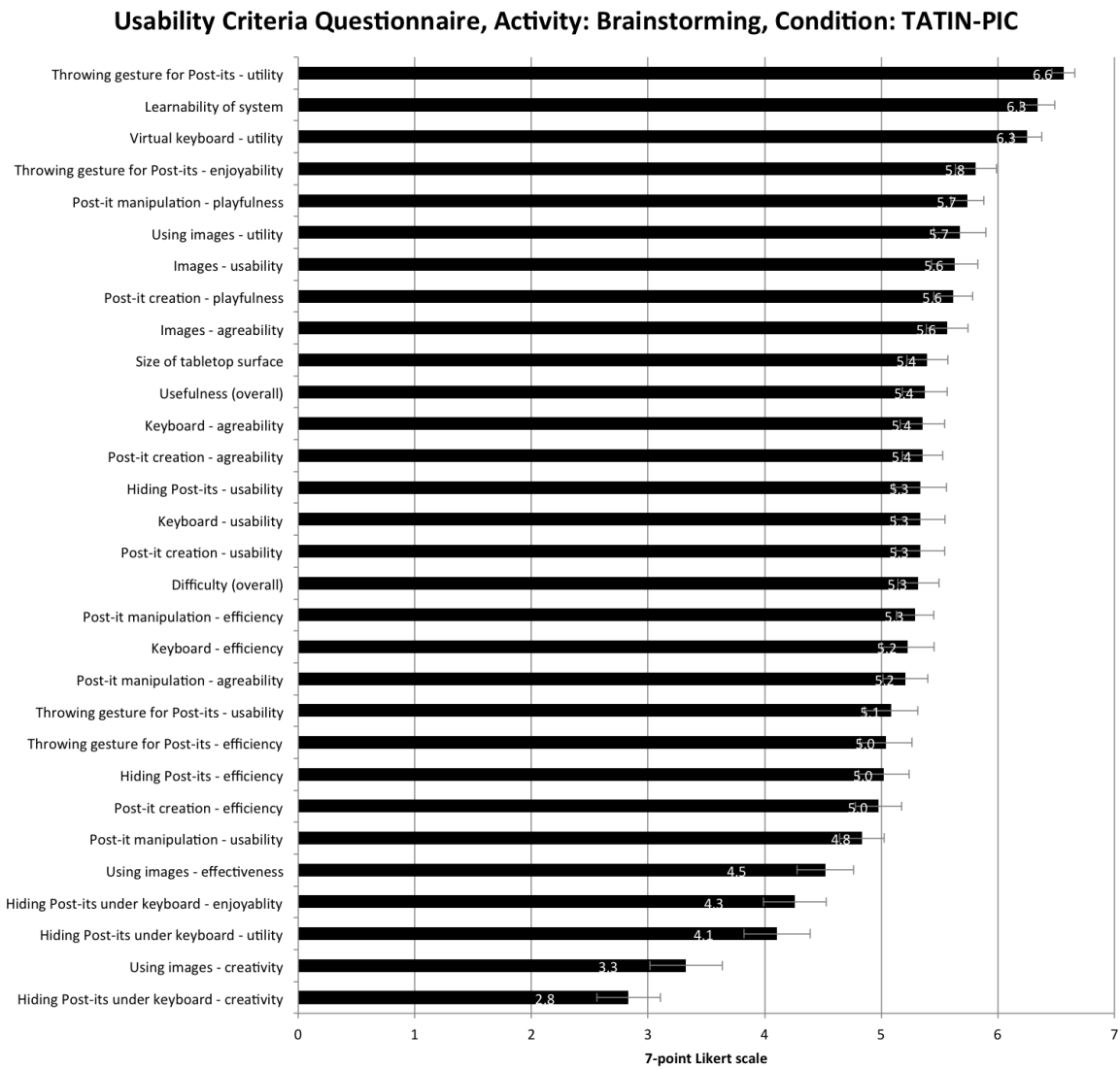


Figure D.1: Complete results from the usability criteria questionnaire from the brainstorming evaluations. This was distributed to the participants in the experimental TATIN-PIC condition only.

Cas EUROTP 1.

Rationalisation d'un processus de développement de nouveau produit : « ingénierie séquentielle ».

ENONCE DU CAS

Le cas EuroTP se fait en trois séances de TD. Ce cas analyse les différences entre les processus d'ingénierie séquentielle et d'ingénierie concurrente. Lors des deux premières séances nous traiterons des aspects processus et planning, tandis que lors de la troisième séance nous traiterons l'aspect économique.

Méthodes de travail en groupe

Cette première partie, ainsi que la suivante, sera faite en utilisant les méthodes de travail en groupe.

Vous constituerez un groupe de travail d'environ 5 personnes.

Le groupe doit disposer d'une grande surface verticale (papier kraft = « brown paper »), de feutres et de « post-it ».

Le groupe désignera un animateur, dont le rôle est d'exprimer les objectifs du groupe, d'organiser la discussion, stopper les digressions inutiles au regard des objectifs, distribuer la parole, résumer, reformuler ce qui est convenu par le groupe. L'animateur se concentre sur le bon fonctionnement et la qualité de la production du groupe.

Le groupe devra s'organiser pour améliorer son efficacité en distribuant, par exemple, les tâches parmi ses membres : l'animation, la gestion des données d'entrée (énoncé), la rédaction des « post-it » sur la table, la mise au propre des résultats obtenus par le groupe, la réalisation du planning Gantt, la gestion du temps par un « gardien du temps », ...

Contexte du cas EuroTP

EuroTP est une entreprise qui fabrique des engins de travaux publics. Depuis quelques années, en prenant appui sur la technologie et le savoir-faire acquis dans son cœur de métier, EuroTP se diversifie et souhaite prendre pied sur de nouveaux marchés. C'est ainsi que l'entreprise produit déjà des engins de manutention de conteneurs et qu'elle projette de développer un Véhicule Tout-Terrain Multifonctions (VTTM) destiné, par exemple, à la protection civile, l'armée ou les forestiers. L'étude de marché sur le concept du VTTM a donné des résultats prometteurs.

Le PDG d'EuroTP constate que le cycle de développement d'un nouveau produit est de l'ordre de 2 ans. Ce délai lui paraît trop long pour des raisons commerciales (délai entre la perception du besoin et la disponibilité du produit) mais aussi économiques (retour d'investissement). D'autre part il pense que le processus de développement actuel ne permet pas d'optimiser le coût des produits, d'où une rentabilité d'exploitation insuffisante pour financer notamment de nouveaux développements et assurer la pérennité de l'entreprise.

Objet du projet EuroTP

Le comité de direction d'EuroTP décide de lancer le projet d'étude, d'industrialisation et de commercialisation du VTTM. Celui-ci sera l'occasion de rationaliser le processus de développement de nouveau produit de l'entreprise en adoptant le principe de l'ingénierie concurrente. Il nomme un chef de projet à la tête de ce projet stratégique de l'entreprise, dont les objectifs sont :

- Produire et commercialiser un VTTM répondant aux besoins du marché identifié,
- Réduire le délai de développement des nouveaux produits à 18 mois maximum,
- Améliorer la rentabilité d'exploitation de 2% par rapport à la celle des produits actuels.
- Améliorer la rentabilité globale du projet permettant l'autofinancement d'autres projets.

Le chef de projet conduira son projet en trois étapes : l'analyse du processus de développement actuel et sa critique (EuroTP1), puis la mise en place de l'organisation du projet VTTM selon le nouveau processus d'ingénierie concurrente (EuroTP2), enfin la comparaison des deux méthodes sous les aspects économique et risques (EuroTP3).

Données (voir annexe)

La liste des tâches et phases et le fond de planning Gantt du processus actuel (séquentiel), après 15 minutes de travail. Ce planning fourni la liste des macro tâches et des tâches ainsi que la durée de celles-ci.

ETAPES**1. Mini-note de clarification**

Sur un paperboard identifier le contexte, le nom du projet, le périmètre (services/fonctions concernées,...), les objectifs (chiffres,...) et les données d'entrée.

2. Lister les fonctions de l'entreprise

Définir dans un premier temps l'organigramme de l'entreprise. Il s'agit d'une PME qui n'est pas organisée par projets et dont l'organigramme est classique.

3. Installer le brown paper (seulement pour le groupe en condition contrôle)**4. Rédiger et répartir les tâches selon la structure de responsabilité de l'entreprise.**

Pour traiter la question 4, il est nécessaire de préparer des « post-it » marqués de chacune des tâches et macro tâches de l'annexe 1. Cette préparation se fait sur la table.

Sur la table placer en colonne sous chaque grande fonction les tâches qui sont réalisées par cette fonction.

5. Regrouper les tâches par macro-tâche

Les « post-it » sont regroupés par phases sur la table puis déplacés sur le papier kraft (tableau) pour traiter la question 5.

6. Définir l'ordonnement

Après avoir enregistré le résultat de la question précédente (par exemple sous la forme d'une photo), utiliser les « post-it » pour construire le processus en les collant sur le papier kraft, indiquer les jalons et les phases.

Le développement est conduit selon 7 phases, une revue de projet (jalon) autorise le démarrage de la phase après avoir validé les résultats de la phase précédente (voir fond de planning en annexe). Le projet est terminé lorsque le jalon « 8 - Lancement de la production en série » est atteint.

Le processus est séquentiel : chaque département (service ou métier) de l'entreprise effectue à son tour les missions qui lui sont confiées avec les compétences dont elle dispose. Elle doit, dans certains cas remettre à jour des documents, car durant le développement d'autres départements affinent ou modifient certaines données qui peuvent remettre en cause les résultats précédents.

Données complémentaires :

- La fabrication ne dispose pas des moyens pour fabriquer les deux prototypes industriels en même temps, en revanche la R & D peut faire les essais en parallèle. Un prototype est destiné aux essais d'endurance, l'autre, aux essais de performance.
- Les tâches de ressources humaines (étude des besoins en ressources, formation/recrutement) et les tâches commerciales, n'interfèrent pas avec les tâches techniques et peuvent donc être faites en parallèle.
- L'autorisation de montée en cadence (jalon 7) n'est donnée que si la ligne de production est prête (présérie achevée) mais également s'il y a suffisamment de commandes (lancement commercial) et si le personnel est recruté et formé.
- Les tâches de formation/recrutement ne sont pas concernées par le jalon 6 : lancement de la présérie.
- La recherche de partenaires clés doit se faire en même temps que la conception préliminaire.

- Pour lancer les tâches d'achat, il faut disposer des spécifications techniques du besoin (STB). La conception détaillée (DD) est nécessaire pour la fabrication des prototypes mais n'est pas nécessaire pour démarrer les tâches d'achat.

7. Calculer la durée et les marges en utilisant les durées fournies lors du TD

Définir le réseau PERT, calculer le délai et le chemin critique du projet selon le processus de développement actuel (séquentiel). Pour construire le réseau PERT, préparer des « post-it » comprenant une grille de 6 cases selon l'exemple ci-dessous. Ne tenez compte, cette fois, que des tâches élémentaires (celles qui sont affectée d'une durée) car se sont les tâches du plus bas niveau qui participent au calcul du planning.

Début au plus tôt DTO	Fin au plus tôt FTO
Début au plus tard DTA	Fin au plus tard FTA
Marge libre	Marge totale

A partir du processus que vous avez défini et des durées des tâches données par le fond de planning, établir le réseau PERT du développement actuel (séquentiel).

Déterminer les dates au plus tôt en parcourant le réseau du début vers la fin. Arrivé à la dernière tâche : $FTA = FTO = \text{délai du projet}$. Déterminer les dates au plus tard en parcourant le réseau en sens inverse. Votre calcul est correct si vous trouvez pour la première tâche : $DTA = 0 = DTO$.

Préciser ensuite, pour chaque tâche, les marges libres et totales. Déterminer le chemin critique.

8. Définir le planning de Gantt du projet selon le processus de développement actuel (séquentiel) sur le support donné en TD.

A l'aide du fond de planning et sur la base du réseau PERT tracer les tâches du planning de Gantt. Positionner judicieusement les tâches en utilisant les marges libres.

9. Analyse.

Quel délai obtenez-vous pour la fin de la montée en cadence ? Commentez le résultat.

ANNEXE 1 : LISTE DE TÂCHES , MACRO-TÂCHES ET PHASES

Phases
Avant-Projet
Etudes Commerciales
Etudes des prototypes
Fabrication et essais des prototypes
Industrialisation
Montée en cadence
Pré-série
Production en série

Tâches
Business plan
Caractéristiques du produit
CdCF (Cahier des Charges Fonctionnels)
Commandes
Conception préliminaire
Consultations des partenaires
DD (Dossier de Définition; Conception détaillée)
Evaluation du marché
Proto 0 fabrication
Réceptions
Recherche Fournisseurs
Recherche partenaires-clés
STB (Spécification Techniques des Besoins)
Tests proto 0 (validation de la conception)

Macro tâches
1ère mise-à-jour Business Plan
2ème mise-à-jour Business Plan
Achats
Analyse de la valeur
Définition Concepts
Définition des prototypes
Essais d'endurance
Essais de performance
Etude d'opportunité
Etude Marketing
Etudes d'industrialisation
Etudes des besoins en ressources humaines
Evaluation coûts
Fabrication proto industriel 1
Fabrication proto industriel 2
Formation en recrutement
Gestion de la montée en cadence
Lancement commercial
Mise-à-jour coûts
Mise-à-jour DD (Dossier de Définition)
Plan de promotion commerciale
Pré-cahier des charges marketing
Production pré-série
Réalisation en ligne de production

PLANNING GANTT DE DEVELOPPEMENT DE PRODUIT (situation actuelle : ingénierie séquentielle)

03/10/12

Cas EUROTP 2.

Rationalisation d'un processus de développement de nouveau produit : « ingénierie concourante ».

ENONCE DU CAS

Le cas EuroTP 2 est la suite du cas EuroTP 1. Après avoir étudié le processus de développement actuel de l'entreprise (séquentiel), le chef de projet doit mettre en œuvre le nouveau processus de développement (ingénierie concourante) qui permettra d'atteindre les objectifs qui lui ont été fixés par le comité de direction.

Cette seconde partie sera faite en utilisant les méthodes de travail en groupe.

Contexte

Le projet VTTM a été lancé à la demande d'un client : La protection civile, qui a un besoin important de ce type de véhicule notamment pour la prévention des risques d'incendie de forêt. Le chef de projet a donc la mission difficile de développer un nouveau produit pour un client extérieur et, à l'occasion de ce projet, de mettre en place un nouveau processus de développement pour son « client interne » : le comité de direction d'EUROTP.

La protection civile a déjà passé une commande de 250 VTTM dont la livraison est échelonnée sur deux ans. Les 25 premières livraisons (10 véhicules de préséries et 15 de montée en cadence) ont été promises dans un délai de 18 mois. Un accord avec la protection civile précise que celle-ci participera activement à la mise au point du cahier des charges fonctionnel, et s'impliquera dans le projet de développement : elle s'est engagée à fournir les chauffeurs pour faire les essais d'endurance et de performance sur le terrain.

Le VTTM devra pouvoir circuler sur la voie publique, il sera donc soumis à une réglementation assez sévère.

L'étude de marché a montré que ce concept répond bien aux besoins de l'armée de terre et des forestiers en particulier dans les zones montagneuses.

Le CP s'est engagé sur les objectifs de la commande de la protection civile en anticipant les résultats attendus du nouveau processus de développement d'EuroTP (délai = 18 mois). Après réflexion et consultation des diverses parties prenantes, le CP pense que, pour atteindre ses objectifs, il doit utiliser les principes de l'ingénierie concourante et adopter une organisation en « task force ». Le CP appliquera le principe de la « conception partagée », c'est-à-dire qu'il intégrera dans son équipe, dès la phase de conception, les compétences de marketing, d'industrialisation, d'achat, d'analyse de la valeur ; il y intégrera également les fournisseurs principaux. Il compte ainsi faire une conception robuste qui optimise la contribution de tous les acteurs du projet.

Le Comité de Direction d'EuroTP est d'accord sur cette stratégie, il donne les pouvoirs et les moyens nécessaires au CP pour conduire le projet, il prend en conséquence les décisions suivantes :

Le projet sera exécuté par une équipe intégrée (en « plateau »).

Le CP rendra compte à un comité de pilotage ayant l'autorité pour décider.

Chaque département fournira au CP les moyens en compétences dont il a besoin. Pendant la période du projet, le CP aura l'autorité fonctionnelle sur son équipe. Chaque membre de l'équipe continuera cependant de profiter du soutien technique et de l'expertise de sa direction d'origine.

Réduction du nombre de points d'arrêt (jalons) de 7 à 4.

En accord avec la protection civile, des négociations sont déjà engagées avec certains fournisseurs de composants clés : VOLVO pour le moteur, ZF pour la boîte de vitesse automatique et SOMA pour les ponts avant et arrière. Des pré-accords sont signés avec ces fournisseurs leur demandant notamment de participer à la conception dans le cadre de l'équipe de projet intégrée qui sera mise en place par EUROTP.

Compte tenu de ces dispositions, l'enveloppe budgétaire pour le coût de développement du projet VTTM est estimé à 1.350 KEuros.

La direction financière pense que les objectifs de rentabilité du PDG seraient assurés si le coût de revient direct (hors amortissement et frais généraux) du produit de série était inférieur ou égal à 54 KEuros.

Données complémentaires nécessaires à la réalisation du cas :

Enoncé du Cas EuroTP 1 ;
Corrigé du cas EuroTP 1.

QUESTIONS

1. Faire la mise à jour de la note de clarification du projet VTTM.

2. Organisation du nouveau projet (A faire sur le paperboard)

Quel est l'OBS du projet VTTM ?

Quelle serait, à votre avis, la constitution idéale du comité de pilotage du projet ?

3. Définir le nouveau processus de développement de projet.

L'objectif est ici d'obtenir un délai de l'ordre de 18 mois, il peut être utile qu'un des équipiers évalue, en temps réel, le délai obtenu (question 4 ci-après) pour guider le groupe pendant la conception du processus.

Appliquer la méthode des « post-it » utilisée pour EuroTP 1.

Les jalons autorisant l'engagement des 4 phases sont les suivants :

1. Marketing et avant-projet (début du processus),
2. Etudes, fabrication et essais des prototypes, industrialisation,
3. Pré-série,
4. Montée en cadence,
5. Série (pour mémoire, ce jalon valide la fin du projet de développement).

Pour optimiser le processus, il faut tenir compte de toutes les tâches.

Les tâches seront faites au plus tôt, simultanément par l'équipe intégrée. Certaines tâches seront éliminées pour éviter les « doublons » et les mises à jour.

La production mettra les moyens pour fabriquer les prototypes en parallèle.

4. Etablir le planning Gantt du nouveau projet (MsProject).

Utiliser un logiciel de gestion de projet pour établir le planning Gantt. L'équipier chargé du planning simule l'évolution du délai selon les options de logique choisies par le groupe.

Quel délai obtenez-vous pour la fin de la montée en cadence ?

Bibliography

- [Ackad 10] Christopher James Ackad, Anthony Collins & Judy Kay. *Switch: exploring the design of application and configuration switching at tabletops*. In ACM International Conference on Interactive Tabletops and Surfaces, ITS '10, page 95–104, 2010.
- [Amabile 83] Teresa Amabile & Teresa M Amabile. The social psychology of creativity, volume 11. Springer-Verlag New York, 1983.
- [Andrews 10] Christopher Andrews, Alex Endert & Chris North. *Space to think: large high-resolution displays for sensemaking*. In Proceedings of the 28th international conference on Human factors in computing systems, CHI '10, page 55–64, New York, NY, USA, 2010.
- [Bachl 11] Stefan Bachl, Martin Tomitsch, Karin Kappel & Thomas Grechenig. *The Effects of Personal Displays and Transfer Techniques on Collaboration Strategies in Multi-touch Based Multi-Display Environments*. Human-Computer Interaction – INTERACT 2011, vol. 6948, pages 373–390, 2011.
- [Bachour 08] Khaled Bachour, Frédéric Kaplan & Pierre Dillenbourg. *Reflect: An Interactive Table for Regulating Face-to-Face Collaborative Learning*. In Pierre Dillenbourg & Marcus Specht, editeurs, Times of Convergence. Technologies Across Learning Contexts, volume 5192 of *Lecture Notes in Computer Science*, pages 39–48. Springer Berlin Heidelberg, 2008.
- [Banavar 00] Guruduth Banavar, James Beck, Eugene Gluzberg, Jonathan Munson, Jeremy Sussman & Deborra Zukowski. *Challenges: an application model for pervasive computing*. In Proceedings of the 6th annual international conference

- on Mobile computing and networking, MobiCom '00, pages 266–274, New York, NY, USA, 2000. ACM.
- [Barab 04] Sasha Barab, Steve Schatz & Rebecca Scheckler. *Using Activity Theory to Conceptualize Online Community and Using Online Community to Conceptualize Activity Theory*. Mind, Culture, and Activity, vol. 11, no. 1, pages 25–47, 2004.
- [Barthès 12] Jean-Paul A. Barthès, Alistair Jones, Atman Kendira, Dominique Lenne, Claude Moulin & Thierry Gidel. *Ambiguity in Multimodal Interaction with Multi-touch Multi-user Graphics Tables*. In Cooperative Design, Visualization, and Engineering, volume 7467 of *Lecture Notes in Computer Science*, pages 1–13, Osaka, Japan, September 2012.
- [Baumer 11] Eric Baumer & Bill Tomlinson. *Comparing activity theory with distributed cognition for video analysis: beyond "kicking the tires"*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, pages 133–142, New York, NY, USA, 2011. ACM.
- [Beaudouin-Lafon 00] Michel Beaudouin-Lafon. *Instrumental interaction: an interaction model for designing post-WIMP user interfaces*. In Proceedings of the SIGCHI conference on Human factors in computing systems, page 446–453, 2000.
- [Beaudouin-Lafon 11] Michel Beaudouin-Lafon. *Lessons learned from the wild room, a multisurface interactive environment*. In 23rd French Speaking Conference on Human-Computer Interaction, page 18, 2011.
- [Bergstrom 07] Tony Bergstrom & Karrie Karahalios. *Seeing more: visualizing audio cues*. In Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II, INTERACT'07, page 29–42, Berlin, Heidelberg, 2007. Springer-Verlag.
- [Biehl 08] Jacob T. Biehl, William T. Baker, Brian P. Bailey, Desney S. Tan, Kori M. Inkpen & Mary Czerwinski. *Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field*

- evaluation for co-located software development*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, page 939–948, New York, NY, USA, 2008. ACM.
- [Bragdon 10] Andrew Bragdon, Arman Uguray, Daniel Wigdor, Stylianos Anagnostopoulos, Robert Zeleznik & Rutledge Feman. *Gesture play: motivating online gesture learning with fun, positive reinforcement and physical metaphors*. In ACM international conference on interactive tabletops and surfaces, page 39–48, 2010.
- [Brandl 07] P. Brandl, M. Haller, M. Hurnaus, V. Lugmayr, J. Oberngruber, C. Oster, C. Schafleitner & M. Billinghamurst. *An Adaptable Rear-Projection Screen Using Digital Pens And Hand Gestures*. In Artificial Reality and Telexistence, 17th International Conference on, pages 49–54, 2007.
- [Brandl 08] Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller & Chia Shen. *Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces*. In Proceedings of the working conference on Advanced visual interfaces, AVI '08, page 154–161, New York, NY, USA, 2008. ACM.
- [Bressolle 98] Marie-Christine Bressolle, Bernard Pavard & Marcel Leroux. *The role of multimodal communication in cooperation: The cases of air traffic control*. In Harry Bunt, Robbert-Jan Beun & Tijn Borghuis, editors, Multimodal Human-Computer Communication, volume 1374 of *Lecture Notes in Computer Science*, pages 326–343. Springer Berlin Heidelberg, 1998.
- [Buisine 12] Stephanie Buisine, Guillaume Besacier, Améziane Aoussat & Frédéric Vernier. *How do interactive tabletop systems influence collaboration?* Computers in Human Behavior, vol. 28, no. 1, pages 49–59, 2012.
- [Burns 12] Chris Burns, Teddy Seyed, Theodore Hellman, Jennifer Ferreira & Frank Maurer. *Towards a Usable API for Constructing Interactive Multi-Surface Systems*. In Proceedings of the Workshop on Infrastructure and Design Challenges of Coupled Visual Interfaces (PPD 2012), Capri, Italy, 2012.

- [Butz 10] Andreas Butz. *User Interfaces and HCI for Ambient Intelligence and Smart Environments*. In Hideyuki Nakashima, Hamid Aghajan & JuanCarlos Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 535–558. Springer US, 2010.
- [Buxton 86a] W. Buxton & B. Myers. *A study in two-handed input*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '86, page 321–326, New York, NY, USA, 1986. ACM.
- [Buxton 86b] William Buxton. *Chunking and phrasing and the design of human-computer dialogues*. In Proceedings of the IFIP World Computer Congress, page 475–480, 1986.
- [Buxton 97] William AS Buxton. *Living in Augmented Reality: Ubiquitous Media and Reactive Environments*. Video-Mediated Communication, page 363, 1997.
- [Cassell 99] J. Cassell, T. Bickmore, M. Billinghamurst, L. Campbell, K. Chang, H. Vilhjálmsón & H. Yan. *Embodiment in conversational interfaces: Rea*. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, CHI '99, page 520–527, New York, NY, USA, 1999. ACM.
- [Christiansen 13] Ellen Christiansen. *Commentary on 'Activity Theory'*. In Mads Soegaard & Rikke Friis Dam, editors, *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. The Interaction Design Foundation, Aarhus, Denmark, 2013.
- [Clayphan 11] Andrew Clayphan, Anthony Collins, Christopher Ackad, Bob Kummerfeld & Judy Kay. *Firestorm: a brainstorming application for collaborative group work at tabletops*. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS '11, page 162–171, New York, NY, USA, 2011. ACM.
- [Collins 07] A. Collins, T. Apter & J. Kay. *Tabletop File System Access: Associative and Hierarchical Approaches*. In *Horizontal Interactive Human-Computer Systems*, 2007. TABLETOP '07. Second Annual IEEE International Workshop on, pages 113–120, October 2007.

- [Collins 08] Anthony Collins. *Tabletop File System and Personal Information Management in Pervasive Computing*. In *Advances in Pervasive Computing: Adjunct Proceedings of the 6th International Conference on Pervasive Computing*, page 160–166. Österreichische Computer Gesellschaft, 2008.
- [Collins 09] Anthony Collins, Anastasia Bezerianos, Gregor McEwan, Markus Rittenbruch, Rainer Wasinger & Judy Kay. *Understanding file access mechanisms for embedded Ubicomp collaboration interfaces*. In *Proceedings of the 11th international conference on Ubiquitous computing, Ubicomp '09*, page 135–144, New York, NY, USA, 2009. ACM.
- [Collins 11a] Anthony Collins. *New dimensions of file access at tabletops: associative and hierarchical; private and shared; individual and collaborative*. PhD thesis, School of Information Technologies, University of Sydney, 2011.
- [Collins 11b] Anthony Collins, Christopher James Ackad, Trent Apted, Paul Sztajer, Peter Ward, Hanley Weng & Judy Kay. *Core functionality and new applications for tabletops and interactive surfaces*. In *UbiComp '11, 13th international conference on Ubiquitous computing*, page 607–608, New York, NY, USA, 2011. ACM Press.
- [Dietz 01] Paul Dietz & Darren Leigh. *DiamondTouch: A Multi-User Touch Technology*. In *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01*, page 219, Orlando, Florida, 2001.
- [Döweling 12] Sebastian Döweling, Benedikt Schmidt & Andreas Göb. *A model for the design of interactive systems based on activity theory*. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, CSCW '12*, page 539–548, New York, NY, USA, 2012. ACM.
- [Edwards 02] W. Keith Edwards, Mark W. Newman, Jana Sedivy, Trevor Smith & Shahram Izadi. *Challenge: recombinant computing and the speakeasy approach*. In *Proceedings of the 8th annual international conference on Mobile computing and networking, MobiCom '02*, page 279–286, New York, NY, USA, 2002. ACM.

- [Ekman 69] Paul Ekman & Wallace. V. Friesen. *The repertoire of non-verbal behavior: categories, origins, and coding*. Semiotica, vol. 1, no. 1, pages 49–98, 1969.
- [Elrod 92] Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier, John Tang & Brent Welch. *Live-board: a large interactive display supporting group meetings, presentations, and remote collaboration*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '92, page 599–607, New York, NY, USA, 1992. ACM.
- [Engeström 99] Yrjö Engeström. *Expansive Visibilization of Work: An Activity-Theoretical Perspective*. Computer Supported Cooperative Work (CSCW), vol. 8, no. 1-2, pages 63–93, 1999.
- [Erman 80] Lee Erman, Frederick Hayes-Roth, Victor Lesser & Raj Reddy. *The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty*. Computing Surveys, vol. 12, no. 2, pages 213–252, 1980.
- [Everitt 06] K. Everitt, Chia Shen, K. Ryall & C. Forlines. *MultiSpace: enabling electronic document micro-mobility in table-centric, multi-device environments*. In Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on, page 8 pp., January 2006.
- [Fernaes 06] Ylva Fernaeus & Jakob Tholander. *Finding design qualities in a tangible programming space*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06, page 447–456, New York, NY, USA, 2006. ACM.
- [Findlater 11] Leah Findlater, Jacob O. Wobbrock & Daniel Wigdor. *Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 2453–2462, New York, NY, USA, 2011. ACM.
- [fip] *FIPA specifications*. <http://www.fipa.org/specifications/index.html>

- [Fitzmaurice 95] George W. Fitzmaurice, Hiroshi Ishii & William A. S. Buxton. *Bricks: laying the foundations for graspable user interfaces*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95, page 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [Fitzmaurice 97] George W. Fitzmaurice & William Buxton. *An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input*. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, CHI '97, page 43–50, New York, NY, USA, 1997. ACM.
- [Fjeld 02] Morten Fjeld, Kristina Lauche, Martin Bichsel, Fred Voorhorst, Helmut Krueger & Matthias Rauterberg. *Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware*. In Computer Supported Cooperative Work (CSCW '02), page 153–180, 2002.
- [Freeman 09] Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris & Daniel Wigdor. *ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures*. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09, page 165, Banff, Alberta, Canada, 2009.
- [Frisch 09] Mathias Frisch, Jens Heydekorn & Raimund Dachsel. *Investigating multi-touch and pen gestures for diagram editing on interactive surfaces*. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, pages 149–156. ACM, 2009.
- [Fuks 05] Hugo Fuks, Alberto Raposo, Marco Gerosa & Carlos Lucena. *Applying the 3C model to groupware development*. International Journal of Cooperative Information Systems, vol. 14, no. 02, pages 299–328, 2005.
- [Gangemi 02] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari & Luc Schneider. *Sweetening Ontologies with DOLCE*. In Asunción Gómez-Pérez & V. Richard Benjamins, editors, Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, volume

- 2473 of *Lecture Notes in Computer Science*, pages 166–181. Springer Berlin Heidelberg, 2002.
- [Geyer 11a] Florian Geyer, Ulrike Pfeil, Jochen Budzinski, Anita Höchtl & Harald Reiterer. *Affinitytable-a hybrid surface for supporting affinity diagramming*. Human-Computer Interaction–INTERACT 2011, pages 477–484, 2011.
- [Geyer 11b] Florian Geyer, Ulrike Pfeil, Anita Höchtl, Jochen Budzinski & Harald Reiterer. *Designing reality-based interfaces for creative group work*. In Proceedings of the 8th ACM conference on Creativity and cognition, C&C ’11, page 165–174, New York, NY, USA, 2011. ACM.
- [Geyer 12] Florian Geyer, Anita Höchtl & Harald Reiterer. *Harnessing the benefits of bimanual and multi-finger input for supporting grouping tasks on interactive tabletops*. In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design, page 496–499, 2012.
- [Gidel 07] T. Gidel & W. Zonghero. Management de projet: Tome 3, etudes de cas et supports de formation. Collection Finance, gestion, management. Hermes Science Publications, 2007.
- [Gjerlufsen 11] Tony Gjerlufsen, Clemens Klokmoose, James Eagan, Clément Pillias & Michel Beaudouin-Lafon. *Shared Substance: Developing Flexible Multi-Surface Applications*. In Proceedings of the ACM Conference on Human Factors in Computing CHI 2011, 2011.
- [Goldin-Meadow 99] Susan Goldin-Meadow. *The role of gesture in communication and thinking*. Trends in Cognitive Science, vol. 3, no. 11, pages 419–429, 1999.
- [Good 84] Michael D. Good, John A. Whiteside, Dennis R. Wixon & Sandra J. Jones. *Building a user-derived interface*. Commun. ACM, vol. 27, no. 10, page 1032–1043, October 1984.
- [Grimm 04] Robert Grimm, Janet Davis, Eric Lemar, Adam Macbeth, Steven Swanson, Thomas Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble & David Wetherall. *System support for pervasive applications*. ACM Trans. Comput. Syst., vol. 22, no. 4, page 421–486, November 2004.

- [Guerra 12] Andrea Luigi Guerra, Thierry Gidel, Enrico Vezzetti & Atman Kendira. *Co-evolution of design methods and CSCWD systems to improve the preliminary design process*. In 18ème Colloque des Sciences de la Conception et de l’Innovation, CONFERE’12, Venise, Italie, July 2012.
- [Guerra 13] Andrea Guerra, Thierry Gidel, Atman Kendira, Enrico Vezzetti & Alistair Jones. *Co-evolution of design tactics and CSCWD systems: Methodological Circulation and the TATIN-PIC platform*. In 19th International Conference on Engineering Design - ICED13, 2013.
- [Guiard 87] Yves Guiard. *Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model*. Journal of motor behavior, vol. 19, page 486–517, 1987.
- [Guimbretière 01a] François Guimbretière. *Fluid Interaction for High Resolution Wall-Size Displays*. PhD thesis, Stanford University, 2001.
- [Guimbretière 01b] François Guimbretière, Maureen Stone & Terry Winograd. *Fluid interaction with high-resolution wall-size displays*. In Proceedings of the 14th annual ACM symposium on User interface software and technology, UIST ’01, page 21–30, New York, NY, USA, 2001. ACM.
- [Gumienny 11] Raja Gumienny, Lutz Gericke, M. Quasthoff, C. Willems & C. Meinel. *Tele-Board: Enabling efficient collaboration in digital design spaces*. In Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on, pages 47–54, 2011.
- [Haller 06] Michael Haller, Peter Brandl, Daniel Leithinger, Jakob Leitner, Thomas Seifried & Mark Billingham. *Shared design space: Sketching ideas using digital pens and a large augmented tabletop setup*. In ICAT 2006, Lecture Notes in Computer Science 4282, page 948–959. Springer Verlag, 2006.
- [Haller 10] Michael Haller, Jakob Leitner, Thomas Seifried, James R. Wallace, Stacey D. Scott, Christoph Richter, Peter Brandl, Adam Gokcezade & Seth Hunter. *The NiCE Discussion*

- Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, page 609–618, New York, NY, USA, 2010. ACM.
- [Hamdan 13] Nur Al-huda Hamdan, Simon Voelker & Jan O. Borchers. *Conceptual framework for surface manager on interactive tabletops.* In CHI '13 Extended Abstracts on Human Factors in Computing Systems, CHI EA '13, page 1527–1532, New York, NY, USA, 2013. ACM.
- [Han 05] Jefferson Y. Han. *Low-cost multi-touch sensing through frustrated total internal reflection.* In Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05, page 115–118. ACM, 2005.
- [Han 06] Jeff Han. *Jeff Han demos his breakthrough touchscreen,* February 2006. http://www.ted.com/talks/jeff_han_demos_his_breakthrough_touchscreen.html
- [Hewitt 77] Carl Hewitt. *Viewing control structures as patterns of passing messages.* Artificial Intelligence, vol. 8, no. 3, pages 323 – 364, 1977.
- [Hilliges 07] Otmar Hilliges, Lucia Terrenghi, Sebastian Boring, David Kim, Hendrik Richter & Andreas Butz. *Designing for collaborative creative problem solving.* Proceedings of the 6th ACM SIGCHI conference on Creativity cognition CC 07, page 137, 2007.
- [Hinckley 94] Ken Hinckley, Randy Pausch, John C. Goble & Neal F. Kassell. *Passive real-world interface props for neurosurgical visualization.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '94, page 452–458, New York, NY, USA, 1994. ACM.
- [Hinrichs 07] U. Hinrichs, M. Hancock, C. Collins & S. Carpendale. *Examination of Text-Entry Methods for Tabletop Displays.* In Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on, pages 105–112, October 2007.

- [Hinrichs 11] Uta Hinrichs & Sheelagh Carpendale. *Gestures in the wild: studying multi-touch gesture sequences on interactive tabletop exhibits*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 3023–3032, New York, NY, USA, 2011. ACM.
- [Höchtel 12] Anita Höchtel, Florian Geyer & Harald Reiterer. *A Comparison of Spatial Grouping Techniques on Interactive Surfaces*. In Mensch & Computer, page 73–82, 2012.
- [Hollan 00] James Hollan, Edwin Hutchins & David Kirsh. *Distributed cognition: toward a new foundation for human-computer interaction research*. ACM Trans. Comput.-Hum. Interact., vol. 7, no. 2, page 174–196, June 2000.
- [Hornecker 07] Eva Hornecker, Paul Marshall & Yvonne Rogers. *From entry to access: how shareability comes about*. In Proceedings of the 2007 conference on Designing pleasurable products and interfaces, DPPI '07, page 328–342, New York, NY, USA, 2007. ACM.
- [Hunter 11] Seth Hunter, Pattie Maes, Stacey Scott & Henry Kaufman. *MemTable: an integrated system for capture and recall of shared histories in group workspaces*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 3305–3314, New York, NY, USA, 2011. ACM.
- [Ingram 12] Amy Ingram, Xiaoyu Wang & William Ribarsky. *Towards the establishment of a framework for intuitive multi-touch interaction design*. In Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12, pages 66–73, New York, NY, USA, 2012. ACM.
- [Isenberg 10] P. Isenberg, D. Fisher, M.R. Morris, K. Inkpen & Mary Czerwinski. *An exploratory study of co-located collaborative visual analytics around a tabletop display*. In Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on, pages 179–186, 2010.
- [Ishikawa 76] K. Ishikawa. *Guide to quality control*. Industrial engineering and technology. Asian Productivity Organization, 1976.

- [Izadi 03] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers & Mia Underwood. *Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media*. In Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST '03, page 159–168, New York, NY, USA, 2003. ACM.
- [Jamil 11] Izdihar Jamil, Kenton O'Hara, Mark Perry, Abhijit Karnik & Sriram Subramanian. *The effects of interaction techniques on talk patterns in collaborative peer learning around interactive tables*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 3043–3052. ACM, 2011.
- [Jamont 07] J. Jamont & M. Occello. *Designing Embedded Collective Systems: The DIAMOND Multiagent Method*. In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on, volume 2, pages 91–94, 2007.
- [Jamont 10] J.-P. Jamont, M. Occello & A. Lagrèze. *A multiagent approach to manage communication in wireless instrumentation systems*. Measurement, vol. 43, no. 4, pages 489 – 503, 2010.
- [Jetter 11] Hans-Christian Jetter, Jens Gerken, Michael Zöllner, Harald Reiterer & Natasa Milic-Frayling. *Materializing the query with facet-streams: a hybrid surface for collaborative search on tabletops*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 3013–3022, New York, NY, USA, 2011. ACM.
- [Jetter 12a] Hans-Christian Jetter, Florian Geyer, Tobias Schwarz & Harald Reiterer. *Blended Interaction-Toward a Framework for the Design of Interactive Spaces*. In Proceedings of Workshop on Designing Collaborative Interactive Spaces AVI2012, Capri, 2012.
- [Jetter 12b] Hans-Christian Jetter, Michael Zöllner, Jens Gerken & Harald Reiterer. *Design and Implementation of Post-WIMP Distributed User Interfaces with ZOIL*. International Journal of Human-Computer Interaction, vol. 28, no. 11, page 737–747, 2012.

- [Johansen 88] Robert Johansen. GroupWare: computer support for business teams. The Free Press, New York, NY, USA, 1988.
- [Johanson 02a] Brad Johanson, Armando Fox & Terry Winograd. *The interactive workspaces project: Experiences with ubiquitous computing rooms*. Pervasive Computing, IEEE, vol. 1, no. 2, page 67–74, 2002.
- [Johanson 02b] Brad Johanson & O. Fox. *The Event Heap: A Coordination Infrastructure for Interactive Workspaces*. In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, WMCSA '02, pages 83–93, 2002.
- [Joiron 13a] C. Joiron, F. Furst, G. Kassel, A. Jones, J.-P. Barthès, C. Moulin & D. Lenne. *Une ontologie pour la conception d'une plateforme collaborative exploitant une table multi-tactile et multi-modale*. In 24èmes Journées francophones d'Ingénierie des Connaissances (IC'2013), Lille, France, July 2013.
- [Joiron 13b] Céline Joiron, Frédéric Fürst, Gilles Kassel, Alistair Jones, Jean-Paul Barthès, Claude Moulin & Dominique Lenne. *An Ontology for Designing a Collaborative Platform Involving a Multi-touch and Multi-user Interactive Table*. In Proceedings of the 5th International Conference of Knowledge Engineering and Ontology Development, KEOD'13, 2013.
- [Jonassen 99] David H. Jonassen & Lucia Rohrer-Murphy. *Activity theory as a framework for designing constructivist learning environments*. Educational Technology Research and Development, vol. 47, no. 1, pages 61–79, 1999.
- [Jones 11a] Alistair Jones, Atman Kendira, Dominique Lenne, Thierry Gidel & Claude Moulin. *The TATIN-PIC project: A multi-modal collaborative work environment for preliminary design*. In Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on, pages 154 –161, June 2011.
- [Jones 11b] Alistair Jones, Dominique Lenne, Atman Kendira, Thierry Gidel, Claude Moulin & Jean-Paul Barthès. *Apprentissage autour d'une table interactive*. In Audrey Serna & Sébastien George, editors, Actes de l'Atelier "IHM avancées pour

- l'apprentissage" - EIAH 2011, page 36–44, Mons, Belgium, May 2011
- [Jones 12a] Alistair Jones, Atman Kendira, Thierry Gidel, Claude Moulin, Dominique Lenne & Jean-Paul Barthès. *Personal Assistant Agents and Multi-agent Middleware for CSCW*. In From the proceedings of the 16th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD), pages 72–79, Wuhan, China, May 2012.
- [Jones 12b] Alistair Jones, Atman Kendira, Thierry Gidel, Claude Moulin, Dominique Lenne, Jean-Paul Barthès & Andrea Luigi Guerra. *Evaluating Collaboration in Table-centric Interactive Spaces*. In Proceedings of the AVI Workshop on Designing Collaborative Interactive Spaces (DCIS 2012), Capri, Italy, May 2012.
- [Jones 12c] Alistair Jones, Atman Kendira, Claude Moulin, Jean-Paul Barthès, Dominique Lenne & Thierry Gidel. *Vocal Interaction in Collocated Cooperative Design*. In IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing (IEEE ICCI*CC), pages 246–252, Kyoto, Japan, August 2012.
- [Jordà 07] Sergi Jordà, Günter Geiger, Marcos Alonso & Martin Kaltenbrunner. *The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces*. In Proceedings of the 1st international conference on Tangible and embedded interaction, page 139–146, 2007.
- [Kaltenbrunner 05] M. Kaltenbrunner, T. Bovermann, R. Bencina & E. Costanza. *TUIO: A protocol for table-top tangible user interfaces*. In Proceeding of the The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, 2005.
- [Kaltenbrunner 07] Martin Kaltenbrunner & Ross Bencina. *reacTIVision: a computer-vision framework for table-based tangible interaction*. In Proceedings of the 1st international conference on Tangible and embedded interaction, page 69–74, 2007.
- [Kaptelinin 99] Victor Kaptelinin, Bonnie A. Nardi & Catriona Macaulay. *Methods & tools: The activity checklist: a tool for repre-*

- senting the \“space\” of context.* interactions, vol. 6, no. 4, page 27–39, July 1999.
- [Kaptelinin 06] V. Kaptelinin & Bonnie A. Nardi. *Acting with technology: Activity theory and interaction design.* Acting With Technology Series. Mit Press, 2006.
- [Kaptelinin 13] Victor Kaptelinin. *Activity theory.* The Interaction Design Foundation, Aarhus, Denmark, 2013. http://www.interaction-design.org/encyclopedia/activity_theory.html
- [Kaufmann 12] Bonifaz Kaufmann, Martin Gratzer & Martin Hitz. *3MF - A Service-Oriented Mobile Multimodal Interaction Framework.* In Workshop on infrastructure and design challenges of coupled display visual, 2012.
- [Kendira 10] Atman Kendira, Alistair Jones, Guillaume Lehoux, Thierry Gidel, Stephanie Buisine & Dominique Lenne. *Project TATIN: creativity and collaboration during a preliminary product design session using an interactive tabletop surface.* In Proceedings of IDMME-Virtual Concept 2010, pages 158–164, October 2010.
- [Kendira 11] Atman Kendira, Thierry Gidel, Alistair Jones, Dominique Lenne, Jean-Paul Barthès & Claude Moulin. *Conducting Preliminary Design around an Interactive Tabletop.* In Proceedings of the 18th International Conference on Engineering Design (ICED11), volume 2, pages 366–376, København, Copenhagen, August 2011.
- [Kendira 13] Atman Kendira, Alistair Jones, Thierry Gidel & Dominique Lenne. *Verbal and Nonverbal Communication for Evaluating Interactive Spaces.* In CHI ’13 Extended Abstracts on Human Factors in Computing Systems, Paris, France, 2013.
- [Kharrufa 10] Ahmed Kharrufa. *Digital Tabletops and Collaborative Learning.* PhD thesis, Newcastle University, UK, 2010.
- [Kindberg 02] Tim Kindberg & Armando Fox. *System Software for Ubiquitous Computing.* IEEE Pervasive Computing, vol. 1, no. 1, page 70–81, January 2002.
- [King 08] John King. *Jeff Han.* TIME, May 2008.

- [Kruger 04] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott & Saul Greenberg. *Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication*. Computer Supported Cooperative Work, vol. 13, no. 5-6, page 501–537, December 2004.
- [Kubicki 11] Sébastien Kubicki, Sophie Lepreux & Christophe Kolski. *Evaluation of an interactive table with tangible objects: Application with children in a classroom*. In Proceedings of the 2nd Workshop on Child Computer Interaction “UI Technologies and Educational Pedagogy”, at CHI, 2011.
- [Kubicki 12] Sébastien Kubicki, Sophie Lepreux & Christophe Kolski. *RFID-driven situation awareness on TangiSense, a table interacting with tangible objects*. Personal Ubiquitous Comput., vol. 16, no. 8, page 1079–1094, December 2012.
- [Kubicki 13] Sebastien Kubicki, Yoann Lebrun, Sophie Lepreux, Emmanuel Adam, Christophe Kolski & René Mandiau. *Simulation in contexts involving an interactive table and tangible objects*. Simulation Modelling Practice and Theory, vol. 31, no. 0, pages 116 – 131, 2013.
- [Kurtenbach 93] Gordon Kurtenbach & William Buxton. *The limits of expert performance using hierarchic marking menus*. In Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, CHI '93, page 482–487, New York, NY, USA, 1993. ACM.
- [Kuutti 95] Kari Kuutti. *Activity Theory as a Potential Framework for Human- Computer Interaction Research*. In Bonnie A. Nardi, editeur, Context and consciousness: Activity Theory and Human-Computer Interaction, page 17–44. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
- [Laufs 10] Uwe Laufs, Christopher Ruff & Jan Zibuschka. *MT4j - A Cross-platform Multi-touch Development Framework*. In ACM EICS 2010, Workshop: Engineering patterns for multi-touch interfaces, 2010.
- [Lebrun 13] Yoann Lebrun, Emmanuel Adama, René Mandiau & Christophe Kolski. *Interaction Between Tangible and Virtual Agents on Interactive Tables: Principles and Case*

- Study*. The 4th International Conference on Ambient Systems, Networks and Technologies, vol. 19, no. 0, pages 32 – 39, 2013.
- [Leithinger 07] Daniel Leithinger & Michael Haller. *Improving Menu Interaction for Cluttered Tabletop Setups with User-Drawn Path Menus*. In Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems TABLETOP07, volume 0, page 121–128, 2007.
- [Leitner 13] Jakob F. Leitner, Florian Perteneder, Can Liu, Christian Rendl & Michael Haller. *Kolibri: tiny and fast gestures for large pen-based surfaces*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, page 1789–1798, New York, NY, USA, 2013. ACM.
- [Lepreux 11] Sophie Lepreux, Sébastien Kubicki, Christophe Kolski & Jean Caelen. *Distributed Interactive Surfaces: A Step Towards the Distribution of Tangible and Virtual Objects*. In Distributed User Interfaces, Human-Computer Interaction Series, pages 133–143. Springer London, 2011.
- [Lhommet 11] Margaux Lhommet, Domitile Lourdeaux & Jean-Paul Barthès. *Never Alone in the Crowd: A Microscopic Crowd Model Based on Emotional Contagion*. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02, WI-IAT '11, page 89–92, Washington, DC, USA, 2011. IEEE Computer Society.
- [Liu 06] Jun Liu, David Pinelle, Samer Sallam, Sriram Subramanian & Carl Gutwin. *TNT: improved rotation and translation on digital tables*. In Proceedings of Graphics Interface 2006, GI '06, page 25–32, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.
- [Mackay 03] Wendy E. Mackay. *The missing link: integrating paper and electronic documents*. In Proceedings of the 15th French-speaking conference on human-computer interaction on 15eme Conference Francophone sur l'Interaction Homme-Machine, IHM 2003, page 1–8, New York, NY, USA, 2003. ACM.

- [MacLeamy 04] Patrick MacLeamy. *Collaboration, Integrated Information, and the Project Lifecycle in Building Design and Construction and Operation*. Rapport technique WP-1202, 2004.
- [Marshall 08] P. Marshall, E. Hornecker, R. Morris, N. Sheep Dalton & Y. Rogers. *When the fingers do the talking: A study of group participation with varying constraints to a tabletop interface*. In Horizontal Interactive Human Computer Systems, 2008. Tabletop 2008. 3rd IEEE International Workshop on, pages 33–40, October 2008.
- [Martínez 11] Roberto Martínez, Anthony Collins, Judy Kay & Kalina Yacef. *Who did what? Who said that?: Collaid: an environment for capturing traces of collaborative learning at the tabletop*. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS '11, page 172–181, New York, NY, USA, 2011. ACM.
- [Masse 00] L. Masse. *Aspects structurels et fonctionnels d'indicateurs gestuels dans l'analyse d'entretiens thérapeutiques*. Revue internationale de psychologie sociale, vol. 13, no. 4, 2000.
- [Mathis 11] Lukas Mathis. *Designed for use: Create usable interfaces for applications and the web*. Pragmatic Bookshelf, 2011.
- [Mauney 10] Dan Mauney, Jonathan Howarth, Andrew Wirtanen & Miranda Capra. *Cultural similarities and differences in user-defined gestures for touchscreen user interfaces*. In CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10, page 4015–4020, New York, NY, USA, 2010. ACM.
- [McNeil 96] David McNeil. *Hand and mind: What gestures reveal about thought*. University of Chicago Press, January 1996.
- [Melchior 09] Jérémie Melchior, Donatien Grolaux, Jean Vanderdonckt & Peter Van Roy. *A toolkit for peer-to-peer distributed user interfaces: concepts, implementation, and applications*. In Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems, EICS '09, page 69–78, New York, NY, USA, 2009. ACM.
- [Moldt 97] Daniel Moldt & Frank Wienberg. *Multi-agent-systems based on coloured Petri nets*. In Pierre Azéma & Gianfranco

- Balbo, editeurs, Application and Theory of Petri Nets 1997, volume 1248 of *Lecture Notes in Computer Science*, pages 82–101. Springer Berlin Heidelberg, 1997.
- [Morris 04a] Meredith Ringel Morris. *Benefits and Challenges of Tabletop Peripheral Displays*. In Proceedings of UbiComp '04 Extended Abstracts, 2004.
- [Morris 04b] Meredith Ringel Morris, Dan Morris & Terry Winograd. *Individual audio channels with single display groupware: effects on communication and task strategy*. In Proceedings of the 2004 ACM conference on Computer supported cooperative work, CSCW '04, page 242–251, New York, NY, USA, 2004. ACM.
- [Morris 06a] M. R Morris, A. Cassanego, A. Paepcke, T. Winograd, A. M Piper & A. Huang. *Mediating Group Dynamics through Tabletop Interface Design*. Computer Graphics and Applications, IEEE, vol. 26, no. 5, pages 65–73, September 2006.
- [Morris 06b] Meredith Morris. *Supporting Effective Interaction with Tabletop Groupware*. PhD thesis, Stanford University, 2006.
- [Morris 06c] Meredith Ringel Morris, Anqi Huang, Andreas Paepcke & Terry Winograd. *Cooperative gestures: multi-user gestural interactions for co-located groupware*. In Proceedings of the SIGCHI conference on Human Factors in computing systems, CHI '06, page 1201–1210, New York, NY, USA, 2006. ACM.
- [Morris 10] Meredith Ringel Morris, Jarrod Lombardo & Daniel Wigdor. *WeSearch: supporting collaborative search and sense-making on a tabletop display*. In Proceedings of the 2010 ACM conference on Computer supported cooperative work, CSCW '10, page 401–410, 2010.
- [Moulin 11] Claude Moulin, Alistair Jones, Jean-Paul Barthès & Dominique Lenne. *Preliminary Design on Multi-touch surfaces Managed by Multi-agent System*. International Journal of Energy, Information and Communications, vol. 2, no. 4, 2011.

- [Mwanza 01] Daisy Mwanza. *Where Theory meets Practice: A Case for an Activity Theory based Methodology to guide Computer System Design*. In Proceedings of INTERACT? 2001: Eighth IFIP TC 13 Conference on Human-Computer Interaction, 2001.
- [Nacenta 07] Miguel A. Nacenta, David Pinelle, Dane Stuckel & Carl Gutwin. *The effects of interaction technique on coordination in tabletop groupware*. In Proceedings of Graphics Interface 2007, GI '07, page 191–198, New York, NY, USA, 2007.
- [Nacenta 09] Miguel A. Nacenta, Carl Gutwin, Dzmitry Aliakseyeu & Sri-ram Subramanian. *There and Back Again: Cross-Display Object Movement in Multi-Display Environments*. Human-Computer Interaction, vol. 24, no. 1-2, pages 170–229, 2009.
- [Nacenta 13] Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang & Per Ola Kristensson. *Memorability of pre-designed and user-defined gesture sets*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, page 1099–1108, New York, NY, USA, 2013. ACM.
- [Nardi 96] Bonnie Nardi. *Studying context: A comparison of activity theory, situated action models, and distributed cognition*. Context and consciousness: Activity theory and human-computer interaction, page 69–102, 1996.
- [Negroponte 97] Nicholas Negroponte. *Software agents*. page 57–66. MIT Press, Cambridge, MA, USA, 1997.
- [Neill 98] ThomasMc Neill, JohnS. Gero & James Warren. *Understanding conceptual electronic design using protocol analysis*. Research in Engineering Design, vol. 10, no. 3, pages 129–140, 1998.
- [Nguyen 05] A. Nguyen & W. Wobcke. *An agent-based approach to dialogue management in personal assistants*. In Proceedings of the 10th international conference on Intelligent user interfaces, page 137–144, 2005.
- [Nielsen 94] Jakob Nielsen. *Enhancing the explanatory power of usability heuristics*. In Proceedings of the SIGCHI Conference

- on Human Factors in Computing Systems, CHI '94, page 152–158, New York, NY, USA, 1994. ACM.
- [Norman 10] Donald A. Norman. *Natural user interfaces are not natural*. interactions, vol. 17, no. 3, pages 6–10, May 2010.
- [Nowacka 13] Diana Nowacka, Karim Ladha, Nils Y Hammerla, Daniel Jackson, Cassim Ladha, Enrico Rukzio & Patrick Olivier. *Touchbugs: actuated tangibles on multi-touch tables*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, page 759–762, 2013.
- [nui 09] *NUI Group Community FAQs*, July 2009. <http://nuigroup.com/faq/>
- [Nunamaker 91] J.F. Nunamaker, Alan R. Dennis, Joseph S. Valacich, Douglas R. Vogel & Joey F. George. *Electronic meeting systems to support group work*. Communications of the ACM, vol. 34, no. 7, pages 40–61, 1991.
- [Nunamaker 97] J.F. Nunamaker, Robert O. Briggs, Daniel D. Mittleman, Douglas R. Vogel & Pierre A. Balthazard. *Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings*. Journal of Management Information Systems, vol. 13, no. 3, pages 163–207, 1997.
- [Nwana 96] Hyacinth S. Nwana. *Software agents: an overview*. The Knowledge Engineering Review, vol. 11, no. 03, page 205–244, 1996.
- [Ogawa 12] Kyohei Ogawa, Yukari Hori, Toshiki Takeuchi, Takuji Narumi, Tomohiro Tanikawa & Michitaka Hirose. *Table talk enhancer: a tabletop system for enhancing and balancing mealtime conversations using utterance rates*. In Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities, CEA '12, page 25–30, New York, NY, USA, 2012. ACM.
- [O'Hara 97] Kenton O'Hara & Abigail Sellen. *A comparison of reading paper and on-line documents*. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, page 335–342, 1997.

- [Olson 92] Gary M. Olson, Judith S. Olson, Mark R. Carter & Marianne Storrøsten. *Small group design meetings: an analysis of collaboration*. Hum.-Comput. Interact., vol. 7, no. 4, page 347–374, December 1992.
- [Osborn 57] A.F. Osborn. *Applied imagination: Principles and procedures of creative thinking*. Charles Scribner’s Sons, 1957.
- [Oviatt 97] Sharon Oviatt, Antonella DeAngeli & Karen Kuhn. *Integration and synchronization of input modes during multimodal human-computer interaction*. In Referring Phenomena in a Multimedia Context and their Computational Treatment, ReferringPhenomena ’97, page 1–13, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.
- [Oviatt 99] Sharon Oviatt. *Ten myths of multimodal interaction*. Commun. ACM, vol. 42, no. 11, page 74–81, November 1999.
- [Parker 05] J. Karen Parker, Regan L. Mandryk & Kori M. Inkpen. *TractorBeam: seamless integration of local and remote pointing for tabletop displays*. In Proceedings of Graphics Interface 2005, GI ’05, page 33–40. ACM, 2005.
- [Pinelle 03] David Pinelle, Carl Gutwin & Saul Greenberg. *Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration*. ACM Transactions on Computer-Human Interaction, vol. 10, no. 4, page 281–311, December 2003.
- [Pinelle 08] D. Pinelle, T. Stach & C. Gutwin. *TableTrays: Temporary, reconfigurable work surfaces for tabletop groupware*. In Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on, pages 41–48, 2008.
- [Pirolli 05] Peter Pirolli & Stuart Card. *The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis*. In Proceedings of International Conference on Intelligence Analysis, volume 5, page 2–4, 2005.
- [Ratsimor 04] Olga Ratsimor, Dipanjan Chakraborty, Anupam Joshi, Timothy Finin & Yelena Yesha. *Service discovery in agent-*

- based pervasive computing environments*. Mobile Network Applications, vol. 9, no. 6, page 679–692, December 2004.
- [Reetz 06] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta & Sriram Subramanian. *Superflick: a natural and efficient technique for long-distance object placement on digital tables*. In Proceedings of Graphics Interface 2006, page 163–170, 2006.
- [Rekimoto 99] Jun Rekimoto & Masanori Saitoh. *Augmented surfaces: a spatially continuous work space for hybrid computing environments*. In Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, CHI '99, page 378–385, New York, NY, USA, 1999. ACM.
- [Rogers 04] Yvonne Rogers & Siân E. Lindley. *Collaborating around vertical and horizontal large interactive displays: which way is best?* Interacting with Computers, vol. 16, no. 6, pages 1133–1152, 2004.
- [Rogers 09] Yvonne Rogers, Youn-kyung Lim, William R. Hazlewood & Paul Marshall. *Equal Opportunities: Do Shareable Interfaces Promote More Group Participation Than Single User Displays?* Human–Computer Interaction, no. 1-2, pages 79–116, 2009.
- [Rogers 11] Y. Rogers, H. Sharp & J. Preece. Interaction design: Beyond human - computer interaction. Wiley, 2011.
- [Roth 10] Volker Roth, Philipp Schmidt & Benjamin Gldenring. *The IR ring: authenticating users' touches on a multi-touch display*. In Proceedings of the 23rd annual ACM symposium on User interface software and technology, UIST '10, page 259–262. ACM, 2010.
- [Schmidt 10] Dominik Schmidt, Ming Ki Chong & Hans Gellersen. *IdLenses: dynamic personal areas on shared surfaces*. In ACM International Conference on Interactive Tabletops and Surfaces, ITS '10, page 131–134, New York, NY, USA, 2010. ACM.
- [Schnelle-Walka 12] Dirk Schnelle-Walka & Sebastian Dweling. *Speech augmented multitouch interaction patterns*. In Proceedings of

- the 16th European Conference on Pattern Languages of Programs, EuroPLoP '11, page 8:1–8:23, New York, NY, USA, 2012. ACM.
- [Schöning 08] Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth & Ulrich von Zadow. *Multi-Touch Surfaces: A Technical Guide*. Rapport technique Technical Report TUM-I0833, Institute for Geoinformatics University of Münster, 2008.
- [Scott 03] Stacey D. Scott, Karen D. Grant & Regan L. Mandryk. *System guidelines for co-located, collaborative work on a tabletop display*. In Proceedings of the 8th conference on European Conference on Computer Supported Cooperative Work, ECSCW'03, page 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.
- [Searle 69] J.R. Searle. Speech acts: An essay in the philosophy of language. Cam: [Verschiedene Aufl.]. Cambridge University Press, 1969.
- [Seyed 12] Teddy Seyed, Chris Burns, Mario Costa Sousa, Frank Maurer & Anthony Tang. *Eliciting usable gestures for multi-display environments*. In Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces, ITS '12, page 41–50, New York, NY, USA, 2012. ACM.
- [Shen 04] Chia Shen, Frédéric D. Vernier, Clifton Forlines & Meredith Ringel. *DiamondSpin: An Extensible Toolkit for Around-the-table Interaction*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, pages 167–174, New York, NY, USA, 2004. ACM.
- [Shen 06] Chia Shen, Kathy Ryall, Clifton Forlines, Alan Esenther, Frederic D Vernier, Katherine Everitt, Mike Wu, Daniel Wigdor, Meredith Ringel Morris, Mark Hancock *et al.* *Informing the design of direct-touch tabletops*. Computer Graphics and Applications, IEEE, vol. 26, no. 5, page 36–46, 2006.

- [Shneiderman 83] B Shneiderman. *Direct Manipulation: A Step Beyond Programming Languages*. IEEE Computer, vol. 16, no. 8, page 57–69, 1983.
- [Shoham 93] Yoav Shoham. *Agent-oriented programming*. Artificial Intelligence, vol. 60, no. 1, page 51–92, March 1993.
- [Smith 80] R. G. Smith. *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*. IEEE Transactions on Computers, vol. c29, no. 12, pages 1104–1113, 1980.
- [Stewart 99] Jason Stewart, Benjamin B Bederson & Allison Druin. *Single display groupware: a model for co-present collaboration*. In Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, page 286–293, 1999.
- [Streitz 99] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin’ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz & Ralf Steinmetz. *i-LAND: an interactive landscape for creativity and innovation*. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, CHI ’99, page 120–127, New York, NY, USA, 1999. ACM.
- [Sulaiman 10] A Sulaiman & P Olivier. *Exploring the Requirements of Tabletop Interfaces for Education*. International Journal of Learning Technology, vol. 5, no. 1, pages 42–61, February 2010.
- [Swaminathan 97] Kishore Swaminathan & Steve Sato. *Interaction design for large displays*. interactions, vol. 4, no. 1, page 15–24, January 1997.
- [Tandler 04] Peter Tandler. *The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments*. J. Syst. Softw., vol. 69, no. 3, pages 267–296, January 2004.
- [tei 13] Tei ’13: Proceedings of the 7th international conference on tangible, embedded and embodied interaction, New York, NY, USA, 2013. ACM.

- [Terrenghi 09] Lucia Terrenghi, Aaron Quigley & Alan Dix. *A taxonomy for and analysis of multi-person-display ecosystems*. Personal Ubiquitous Computing, vol. 13, no. 8, page 583–598, November 2009.
- [Tse 08] Edward Tse, Saul Greenberg, Chia Shen, Clifton Forlines & Ryo Kodama. *Exploring true multi-user multimodal interaction over a digital table*. In Proceedings of the 7th ACM conference on Designing interactive systems - DIS '08, pages 109–118, Cape Town, South Africa, 2008.
- [Turnwald 12] Marc Turnwald, Alexander Nolte & Michael Ksoll. *Easy collaboration on interactive wall-size displays in a user distinction environment*. In Proceedings of the AVI Workshop on Designing Collaborative Interactive Spaces (DCIS '12), 2012.
- [Ullman 04] David G Ullman. *The Ideal Engineering Decision Support System*. Artificial Intelligence, 2004.
- [Vogt 11] Katherine Vogt, Lauren Bradel, Christopher Andrews, Chris North, Alex Endert & Duke Hutchings. *Co-located Collaborative Sensemaking on a Large High-Resolution Display with Multiple Input Devices*. In Human-Computer Interaction – INTERACT 2011, volume 6947 of *Lecture Notes in Computer Science*, pages 589–604. Springer Berlin Heidelberg, 2011.
- [Volda 09] Stephen Volda, Matthew Tobiasz, Julie Stromer, Petra Isenberg & Sheelagh Carpendale. *Getting practical with interactive tabletop displays: designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration*. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09, page 109, Banff, Alberta, Canada, 2009.
- [Wallace 09] James R Wallace, Stacey D Scott, Taryn Stutz, Tricia Enns & Kori Inkpen. *Investigating teamwork and taskwork in single- and multi-display groupware systems*. Personal Ubiquitous Computing, vol. 13, page 569–581, November 2009.
- [Wallace 13] James R. Wallace, Stacey D. Scott & Carolyn G. MacGregor. *Collaborative sensemaking on a digital tabletop and*

- personal tablets: prioritization, comparisons, and tableaux.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, page 3345–3354, New York, NY, USA, 2013.
- [Wang 98] Yanqing Wang, Christine L. MacKenzie, Valerie A. Summers & Kellogg S. Booth. *The structure of object transportation and orientation in human-computer interaction.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '98, page 312–319, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- [Wang 02] Lihui Wang, Weiming Shen, Helen Xie, Joseph Neelamkavil & Ajit Pardasani. *Collaborative conceptual design—state of the art and future trends.* Computer-Aided Design, vol. 34, no. 13, pages 981 – 996, 2002.
- [Wang 06] QianYing Wang, Alberto Battocchi, Ilenia Graziola, Fabio Pianesi, Daniel Tomasini, Massimo Zancanaro & Clifford Nass. *The role of psychological ownership and ownership markers in collaborative working environment.* In Proceedings of the 8th international conference on Multimodal interfaces, ICMI '06, page 225–232, New York, NY, USA, 2006. ACM.
- [Watanabe 07] Nayuko Watanabe, Motoi Washida & Takeo Igarashi. *Bubble clusters: an interface for manipulating spatial aggregation of graphical objects.* In Proceedings of the 20th annual ACM symposium on User interface software and technology, UIST '07, page 173–182, New York, NY, USA, 2007. ACM.
- [Weber 00] Wolfgang G Weber. *Organizational conditions fostering prosocial work orientations in teams.* Innovative Theories, Tools, and Practices in Work and Organizational Psychology. Seattle, WA: Hogrefe and Huber, page 75–96, 2000.
- [Weiser 91] Mark Weiser. *The Computer for the 21st Century.* Scientific American, vol. 265, no. 3, 1991.
- [Wellner 93] Pierre Wellner. *Interacting with paper on the DigitalDesk.* Communications of the ACM, vol. 36, no. 7, page 87–96, 1993.

- [Wigdor 09a] Daniel Wigdor, Joe Fletcher & Gerald Morrison. *Designing user interfaces for multi-touch and gesture devices*. In CHI'09 Extended Abstracts on Human Factors in Computing Systems, page 2755–2758, 2009.
- [Wigdor 09b] Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin & Chia Shen. *WeSpace: the design development and deployment of a walk-up and share multi-surface visual collaboration system*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, page 1237–1246, New York, NY, USA, 2009. ACM.
- [Wigdor 11] Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo & Sarah Williams. *Rock /& rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations*. In Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, page 1581–1590, New York, NY, USA, 2011. ACM.
- [Wobbrock 09] Jacob O Wobbrock, Meredith Ringel Morris & Andrew D Wilson. *User-defined gestures for surface computing*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 1083–1092. ACM, 2009.
- [Wooldridge 00] Michael Wooldridge, NicholasR. Jennings & David Kinny. *The Gaia Methodology for Agent-Oriented Analysis and Design*. Autonomous Agents and Multi-Agent Systems, vol. 3, no. 3, pages 285–312, 2000.
- [Woolley 10] Anita Williams Woolley, Christopher F. Chabris, Alex Pentland, Nada Hashmi & Thomas W. Malone. *Evidence for a Collective Intelligence Factor in the Performance of Human Groups*. Science, vol. 330, no. 6004, pages 686–688, 2010.
- [Wu 03] Mike Wu & Ravin Balakrishnan. *Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays*. In Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST '03, page 193–202. ACM, 2003.
- [Zambonelli 03] Franco Zambonelli, Nicholas R. Jennings & Michael Wooldridge. *Developing multiagent systems: The Gaia*

methodology. ACM Transactions on Software Engineering Methodology, vol. 12, no. 3, page 317–370, 2003.