

# A Pattern Language for Interactive Tabletops in Collaborative Workspaces

Diploma Thesis at the  
Media Computing Group  
Prof. Dr. Jan Borchers  
Computer Science Department  
RWTH Aachen University



by  
Johannes Christian Remy

Thesis advisor:  
Prof. Dr. Jan Borchers

Second examiner:  
Prof. Dr. Martina Ziefle

Registration date: Oct 16th, 2009  
Submission date: May 3rd, 2010



I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

*Aachen, May 3rd, 2010*  
*Johannes Christian Remy*





# Contents

<b>Abstract</b>	<b>xiii</b>
<b>Überblick</b>	<b>xv</b>
<b>Acknowledgements</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Design Patterns</b>	<b>5</b>
2.1 History of Design Patterns . . . . .	5
2.2 Design Patterns in Computer Science . . . . .	6
2.2.1 Software Engineering . . . . .	6
2.2.2 Human-Computer Interaction . . . . .	9
2.3 HCI Design Patterns . . . . .	10
2.4 Advantages of HCI Design Patterns . . . . .	13
<b>3 Related Work</b>	<b>17</b>
3.1 Tabletop Guidelines . . . . .	17
3.1.1 Collaboration . . . . .	18

---

3.1.2	Hardware . . . . .	19
3.1.3	Studies and Observations . . . . .	21
3.2	HCI Design Pattern Languages . . . . .	22
3.2.1	Related HCI Design Patterns . . . . .	22
3.2.2	User Interface Patterns . . . . .	23
<b>4</b>	<b>The Pattern Development Process</b>	<b>27</b>
4.1	Identifying Problems . . . . .	28
4.2	Initial Pattern Language . . . . .	29
4.3	Feedback and Refining . . . . .	32
4.4	Evaluation . . . . .	35
4.4.1	First Iteration . . . . .	37
4.4.2	Second Iteration . . . . .	39
<b>5</b>	<b>The Pattern Language</b>	<b>47</b>
5.1	(01) ROUND TABLE . . . . .	50
5.2	(02) TILTED TABLE . . . . .	52
5.3	(03) LARGE COLLABORATION TABLE . . . . .	54
5.4	(04) ERGONOMIC HEIGHT . . . . .	56
5.5	(05) NARROW SUBSTRUCTURE . . . . .	58
5.6	(10) DESKTOP ORIENTATION . . . . .	60
5.7	(11) USER IDENTIFICATION . . . . .	62
5.8	(12) ZOOMABLE INTERFACE . . . . .	64

---

5.9	(20) HAND GESTURES . . . . .	66
5.10	(21) PRIVATE SPACE . . . . .	68
5.11	(22) HIGH PRECISION INPUT . . . . .	70
5.12	(23) REPLACE PHYSICAL PAPERWORK . . . . .	72
5.13	(24) BALANCED PARTICIPATION . . . . .	74
5.14	(30) PEN INPUT DEVICE . . . . .	76
5.15	(31) PHYSICAL KEYBOARD . . . . .	78
5.16	(32) ON-SCREEN KEYBOARD . . . . .	80
5.17	(33) INPUT TANGIBLES . . . . .	82
5.18	(34) EMBEDDING ELECTRONIC DEVICES . . . . .	84
5.19	(40) EXTENDING REACHABILITY . . . . .	86
5.20	(41) PHYSICAL OBJECT STORAGE BIN . . . . .	88
5.21	(43) DYNAMIC KEYBOARD RELABELING . . . . .	90
5.22	(44) DODGE OBSTACLES . . . . .	92
<b>6</b>	<b>Conclusions and Future Work</b>	<b>95</b>
6.1	Summary and Contributions . . . . .	95
6.2	Future Work . . . . .	96
<b>A</b>	<b>Online Resources</b>	<b>99</b>
	<b>Bibliography</b>	<b>101</b>
	<b>Index</b>	<b>113</b>



# List of Figures

1.1	Microsoft Surface™ . [© 2010, Microsoft®]	2
2.1	Pattern (243) SITTING WALL, part 1. [Alexander et al., 1977]	7
2.2	Pattern (243) SITTING WALL, part 2. [Alexander et al., 1977]	8
3.1	Taxonomy of fundamental hardware capabilities. [Wigdor, at ITS 2009]	20
4.1	Initial graph of 13 patterns.	31
4.2	Graph of 23 patterns before evaluation.	34
5.1	Tabletop design pattern language graph.	49
5.2	The Reactable. [© 2008, Reactable Systems, Inc.]	50
5.3	A flexible tiltable tabletop. [Monash University, 2009]	52
5.4	The Strukttable. [© 2002-2010, Strukt GmbH]	54
5.5	A table with a suitable height for the user. [University of Calgary, 2006]	56

---

5.6	<i>Aachener Frieden</i> , a tabletop for an exhibit. [RWTH Aachen University, 2009] . . . . .	58
5.7	<i>DiamondSpin</i> toolkit. [Shen et al., 2004] . . . . .	60
5.8	A fingerprint scanner. [© 2010, The Associated Press] . . . . .	62
5.9	<i>DTLens</i> toolkit. [Forlines and Shen, 2005] . . . . .	64
5.10	<i>Flicking</i> gesture on an interactive tabletop. [Sato et al., 2008] . . . . .	66
5.11	Students reaching over an interactive tabletop. [Zhang and Takatsuka, 2007] . . . . .	68
5.12	<i>Dual finger stretch</i> technique for high precision. [Benko et al., 2006] . . . . .	70
5.13	Apple iPad. [© 2010, Apple Computer, Inc.] . . . . .	72
5.14	Students playing on an interactive tabletop. [Piper et al., 2006] . . . . .	74
5.15	A pen input device on an interactive tabletop. [Brandl et al., 2008] . . . . .	76
5.16	Keyboards and mice on an interactive tabletop. [Hartmann et al., 2009] . . . . .	78
5.17	Screenshot of <i>BubbleType</i> . [Hinrichs et al., 2008] . . . . .	80
5.18	Tangibles on the <i>Reactable</i> . [© 2008, Reactable Systems, Inc.] . . . . .	82
5.19	An interactive surface receiving files from a cell phone placed on it. [Wilson and Sarin, 2007] . . . . .	84
5.20	Roulette chips and a rake. [© 2009, flickr® from Yahoo!, Inc., user AndreJenny.com] . . . . .	86

---

5.21 SMART Board™. [© 2009, SMART Technologies ULC] . . . . .	88
5.22 User pressing the control key on a <i>SLAP</i> keyboard. [Weiss et al., 2009] . . . . .	90
5.23 User-drawn path menus. [Leithinger and Haller, 2007] . . . . .	92





# Abstract

For almost three decades, the domain of interactive tabletops has been an important topic in Human-Computer Interaction (HCI) research, yielding a great variety of contributions. Recently, the commercial market's interest in tabletops has started to grow, thus making them more and more publicly available. They especially offer new opportunities to collaborative workspaces where they can fully unfold their advantages in application. However, the increasing number of different systems also makes it difficult to keep track of available solutions provided by researchers and the industry. Hence, it is crucial for hardware designers, software developers, and customers to know the possibilities, problems, and potential of interactive tabletops. Yet, there exist hardly any guidelines or standards to help building, refining, and evaluating an interactive tabletop system.

This diploma thesis addresses this lack of information by proposing an HCI design pattern language for interactive tabletops. Design patterns provide working solutions to recurring problems in tabletop design in a consistent, established, and easy-to-read format. The patterns were gathered by analyzing problems, identifying conflicting forces, and assembling existing solutions that remedy the problems while balancing the forces. Using the iterative approach of pattern writing, we incorporated feedback from the tabletop and pattern community from the early stages until the thorough evaluation in the end. Due to the pattern approach with a problem-oriented domain analysis, statement of conflicting forces, and inclusion of approved solution examples, every pattern captures an important essence of design experience from the tabletop community. As a result, the pattern language does not only provide domain experts with a useful means for creating new or informing the design on existing systems, it also enables novices to familiarize themselves with the vocabulary and increase their expertise in this area. In the future, we want to encourage tabletop experts to complement, improve, and extend the proposed pattern language, thus, providing a starting point for the community to share design experience, ease novices the entry into the domain, and create a body of sustainable knowledge for interactive tabletops.



# Überblick

Interaktive Tische sind seit annähernd drei Jahrzehnten ein wichtiger Teilbereich des Forschungsgebiets Mensch-Maschine-Interaktion (engl. Human-Computer Interaction, HCI), der eine Vielzahl unterschiedlichster Beiträge verzeichnet. Das Interesse des kommerziellen Marktes an diesen Tischen hat unlängst zugenommen, wodurch sie einer größeren Öffentlichkeit zur Verfügung stehen. Sie eröffnen insbesondere für gemeinschaftliches Arbeiten neue Möglichkeiten, da dort ihre Vorteile vollauf zur Entfaltung kommen. Die steigende Anzahl verschiedener Systeme erschwert es jedoch, den Überblick über alle verfügbare Lösungen zu behalten, die Forschung und Industrie anbieten. Trotzdem existieren nahezu keinerlei Richtlinien oder Standards, die beim Konstruieren, Weiterentwickeln und Evaluieren eines interaktiven Tisches helfen.

Die vorliegende Diplomarbeit bietet eine Lösung für das Fehlen dieser Richtlinien an, indem sie eine HCI Entwurfsmustersprache für interaktive Tische vorstellt. Entwurfsmuster bieten funktionierende Lösungen für wiederkehrende Probleme in der Entwicklung solcher Tische an, in einem einheitlichen, bewährten und einfach zu lesenden Format. Diese Muster wurden durch eine Analyse der Probleme, Erkennung der gegeneinander wirkenden Zwänge bzw. Kräfte und Sammeln existierender Lösungen, welche die Probleme beseitigen und zugleich für einen Ausgleich der wirkenden Kräfte sorgen, erarbeitet. Wir nutzen den iterativen Ansatz, um die Resonanz Beteiligter aus den Bereichen der interaktiven Tische und Entwurfsmuster vom Beginn bis zur ausführlichen Auswertung dieser Arbeit einzubeziehen. Durch die Herangehensweise der Entwurfsmuster mit einer problemorientierten Bereichsanalyse, Feststellung der widerstreitenden Kräfte und die Erwähnung bewährter Lösungsbeispiele, erfasst jedes Entwurfsmuster eine wichtige Kernaussage für die Gestaltung interaktiver Tische aus der Erfahrung der betreffenden Gemeinschaft. Daher ist die Entwurfsmustersprache nicht nur eine wertvolle Hilfestellung für Experten beim Entwurf neuer oder der Weiterentwicklung bestehender Systeme, sondern ermöglicht auch Einsteigern den Zugang zum spezifischen Vokabular dieses Bereichs und eine Vertiefung ihres Fachwissens. Wir beabsichtigen, in naher Zukunft Experten dafür zu gewinnen, die vorgestellte Entwurfsmustersprache zu ergänzen, zu verbessern und zu erweitern. Dadurch bieten wir der Gemeinschaft einen Ausgangspunkt an, um Entwicklungserfahrung auszutauschen, Einsteigern den Zugang zu erleichtern und einen nachhaltigen Fundus gesammelten Wissens über interaktive Tische zu erstellen.



# Acknowledgements

First of all, I want to thank Prof. Dr. Jan Borchers for introducing me to the field of HCI, giving me the opportunity to write this thesis at his chair, and providing me with useful help and support. A special thanks goes to my supervisor Malte Weiß, who supported me throughout the last months with a tremendous amount of valuable feedback to make this work possible. I would also like to thank Prof. Dr. Martina Ziefle for being my second examiner, always showing interest for my work, and giving helpful feedback.

Thanks to everyone who provided assistance, from reviewing the thesis over mentioning insightful thoughts to supporting me with help of any kind: Simon Völker, Lena Walter, Jan Mußler, Hans-Christian Ebke, Mike Kremer, Markus Rinnen, Deidre Könsgen, the folks from the Media Computing Group, and all those I forgot.

Furthermore, I would like to thank Christian Kohls for the helpful reviews of the tabletop pattern language in the shepherding process.

Last but not least, thanks to my family; my sisters Susanne and Monika and my parents. Their support, encouragement, and motivation over the last years made not only this thesis, but also my whole studies possible.



# Chapter 1

## Introduction

*“A journey of a thousand miles begins with a single step.”*

—Lao-tzu

Everyday tasks, such as paperwork in an office or collaborative work in a meeting, are often situated around a table as the center of the workspace. We place books or sheets of paper on the table to have them quickly at hand in a discussion, use it as a work space when taking notes, or consider it a mere storage for things like coffee cups, keys, or cell phones. However, the increasing proliferation of technology leads to a digitalization of most of our data, thus moving our workspaces from tables to computers.

Collaborative workspaces are usually table-centered

For many single-user tasks, computers have completely superseded the traditional, non-digital environments, but in collaborative workspaces tables are still omnipresent. Besides the aforementioned traits, conventional tables have a number of advantages over electronic devices: They support multi-user interaction, they are intuitive and predictable, and they are non-technological in appearance, which comforts people with a low trust in technology.

Interactive tabletops bridge the gap between these two areas by offering a large display, combined with touch input that allows for intuitive manipulation, while preserving the

spatial characteristics of usual desk environments (Figure 1.1). Due to their direct-touch input, they convey a natural mapping of interaction on a surface that works both as input and output. Contrary to desktop computers, the technology is mostly invisible, hidden inside the table's sub-structure. This puts the users and their tasks at the center of collaboration instead of technology and users dealing with it.



**Figure 1.1:** The Microsoft Surface, an example for an interactive tabletop.

Interactive tabletops:  
long history, huge  
diversification, no  
standards

In 1985, the first paper about a multi-touch tabletop device appeared at a major conference, with Lee et al. [1985] presenting their capacitive interactive surface technology at the Conference on Human Factors in Computing Systems (CHI). Since then, numerous contributions have been made in this field, many of them introducing completely different designs or interaction behaviors. Due to the diversity and alteration, it becomes increasingly difficult to keep track of all publications and products in tabletop design. In order to know the possibilities and avoid common problems when designing new or refining existing tabletop systems, it is important to look at existing systems and available solutions. However, there are almost no comprehensive guidelines or standards in this area.



This diploma thesis presents a Human-Computer Interaction (HCI) design pattern language for interactive tabletops to overcome this lack of information. By identifying recurring problems in tabletop design, gathering approved solutions, and compiling these into the established format of HCI design patterns, we create an easily accessible set of guidelines that captures the community's design experience. Besides the development process of our pattern language, we will outline the importance of standards for interactive tabletops and the advantage of HCI design patterns over traditional guidelines.

While one goal of our work is to aid designers, developers, and engineers improving their existing systems, another is to help avoid common mistakes in early design decisions or circumvent recurring problems when designing new tabletops. Furthermore, the pattern language creates a body of knowledge from the experience of domain experts, making it easier for novices to acquire information about interactive tabletops. As aforementioned, research and the commercial market of interactive tabletops have resulted in a diversity of contributions and products. Thus, we consider our HCI design patterns as a start for the community to share their knowledge rather than a finished set of guidelines, and we want to include tabletop experts to complement and further improve the pattern language in the future.

Pattern benefits for  
the tabletop  
community

In the following, we outline the structure of this thesis and give a brief overview of each chapter's content:

**Chapter 2—"Design Patterns"** . We explain the origin and concept of design patterns in this chapter. Furthermore, we state the definition of an HCI design pattern, as well as its structure and format, and point out its advantages over traditional guidelines.

**Chapter 3—"Related Work"** . The related work consists of two different areas. The first part lists existing guidelines in the field of interactive tabletops and their scope of application. Secondly, we look at other HCI design pattern languages from different topics.

**Chapter 4—"The Pattern Development Process"** . In this chapter, we outline the process of identifying and cre-

ating the tabletop patterns. It contains the general methodology and the format, structure, and tools that were used to create the design patterns. Additionally, we discuss the iterations of the pattern language, which include helpful community feedback in the beginning and a thorough evaluation of the final set.

**Chapter 5—“The Pattern Language”**. This chapter contains the main part of this thesis, the tabletop pattern language. It is preceded by the pattern graph and an explanation of the pattern categories.

**Chapter 6—“Conclusions and Future Work”**. In the final chapter, a summary of the contents concludes the presented work in this thesis. Finally, we give an outlook of the next steps to improve the pattern language, incorporate the community, and ensure the sustainability of this contribution.

## Chapter 2

# Design Patterns

*“In examinations, the foolish ask questions the wise cannot answer.”*

—Oscar Wilde

### 2.1 History of Design Patterns

The concept of design patterns was originally proposed in the domain of architecture. Dissatisfaction about the concept of modern architecture in the 1960s and 1970s led Alexander et al. to search for formal guidelines and standards that illustrate good design principles. Their book *A Pattern Language* [1977] provides these guidelines, in the format of design patterns, which they define as follows:

*A Pattern Language*  
by Alexander et al.

“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”

[Alexander et al., 1977, p. x]

Design patterns in  
Ancient Rome and  
renaissance

The idea to capture successful design into a consistent format and share the knowledge with other domain experts has a long history in architecture. In the era of renaissance master builders, Francesco di Giorgio (1439–1501) compiled design solutions in his manuscript *Trattati di architettura*, which introduces many of the key elements of design patterns that also appear in Alexander’s work. Di Giorgio’s approach was heavily inspired by the Roman architect Marcus Vitruvius Pollio (ca. 70 BC–23 BC), who wrote *De architectura*, today referred to as *The Ten Books of Architecture*. Vitruvius collected important aspects of Roman architecture and aimed to empower the builders of the Roman emperor with rules and guidelines for new constructions. He also intended to provide a body of knowledge for interested readers from other domains than architecture and construction engineering [Kruft, 2004].

Quality Without a  
Name

In contrast to these early approaches similar to design patterns, Alexander’s language targets not only experts, but also empowers non-professionals with basic design knowledge of architecture. Furthermore, Alexander argues that buildings with a good design convey a specific *Quality Without a Name*, or in short *QWAN*, that design patterns are able to capture [1979]. According to Alexander, there is no way to measure the presence or amount of *QWAN* in a construction. He describes it as the positive feeling and joy people experience when living in such an environment. An example pattern is depicted in Figures 2.1 and 2.2, a detailed discussion of the format follows in section 2.3—“HCI Design Patterns”.

## 2.2 Design Patterns in Computer Science

### 2.2.1 Software Engineering

In the 1980s, researchers started to build up on the idea of design patterns and transferred them to the realm of computer science. At the OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) 1987 conference, Beck and Cunningham [1987] presented a set of

## 243 SITTING WALL \*\*



. . . if all is well, the outdoor areas are largely made up of positive spaces—POSITIVE OUTDOOR SPACES (106); in some fashion you have marked boundaries between gardens and streets, between terraces and gardens, between outdoor rooms and terraces, between play areas and gardens—GREEN STREETS (51), PEDESTRIAN STREET (100), HALF-HIDDEN GARDEN (111), HIERARCHY OF OPEN SPACE (114), PATH SHAPE (121), ACTIVITY POCKETS (124), PRIVATE TERRACE ON THE STREET (140), OUTDOOR ROOM (163), OPENING TO THE STREET (165), GALLERY SURROUND (166), GARDEN GROWING WILD (172). With this pattern, you can help these natural boundaries take on their proper character, by building walls, just low enough to sit on, and high enough to mark the boundaries.

If you have also marked the places where it makes sense to build seats—SEAT SPOTS (241), FRONT DOOR BENCH (242)—you can kill two birds with one stone by using the walls as seats which help enclose the outdoor space wherever its positive character is weakest.



**In many places walls and fences between outdoor spaces are too high; but no boundary at all does injustice to the subtlety of the divisions between the spaces.**

Consider, for example, a garden on a quiet street. At least somewhere along the edge between the two there is a need for a seam, a place which unites the two, but does so without breaking down the fact that they are separate places. If there is a high wall or a hedge, then the people in the garden have no way of being connected to the street; the people in the street have no way of being connected to the garden. But if there is no barrier at all—then the division between the two is hard to maintain. Stray dogs can wander in and out at will; it is even uncomfortable to sit in the garden, because it is essentially like sitting in the street.

Figure 2.1: Pattern (243) SITTING WALL, part 1. [Alexander et al., 1977]

*The problem can only be solved by a kind of barrier which functions as a barrier which separates, and as a seam which joins, at the same time.*

A low wall or balustrade, just at the right height for sitting, is perfect. It creates a barrier which separates. But because it invites people to sit on it—invites them to sit first with their legs on one side, then with their legs on top, then to swivel round still further to the other side, or to sit astride it—it also functions as a seam, which makes a positive connection between the two places.

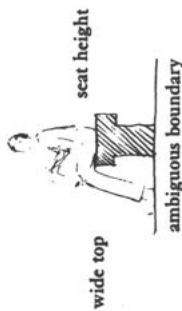
Examples: A low wall with the children's sandbox on one side, connecting path on the other; low wall at the front of the garden, connecting the house to the public path; a sitting wall that is a retaining wall, with plants on one side, where people can sit close to the flowers and eat their lunch.

Ruskin describes a sitting wall he experienced:

Last summer I was lodging for a little while in a cottage in the country, and in front of my low window there were, first, some beds of daisies, then a row of gooseberry and currant bushes, and then a low wall about three feet above the ground, covered with stone-cress. Outside, a corn-field, with its green ears glistening in the sun, and a field path through it, just past the garden gate. From my window I could see every peasant of the village who passed that way, with basket on arm for market, or spade on shoulder for field. When I was inclined for society, I could lean over my wall, and talk to anybody; when I was inclined for science, I could botanize all along the top of my wall—there were four species of stone-cress alone growing on it; and when I was inclined for exercise, I could jump over my wall, backwards and forwards. That's the sort of fence to have in a Christian country; not a thing which you can't walk inside of without making yourself look like a wild beast, nor look at out of your window in the morning without expecting to see somebody impaled upon it in the night. (John Ruskin, *The Two Paths*, New York: Everyman's Library, 1907, p. 203.)

Therefore:

Surround any natural outdoor area, and make minor boundaries between outdoor areas with low walls, about 16 inches high, and wide enough to sit on, at least 12 inches wide.



Place the walls to coincide with natural seat spots, so that extra benches are not necessary—SEAT SPOTS (241); make them of brick or tile, if possible—SOFT TILE AND BRICK (248); if they separate two areas of slightly different height, pierce them with holes to make them balustrades—ORNAMENT (249). Where they are in the sun, and can be large enough, plant flowers in them or against them—RAISED FLOWERS (245). . . .

Figure 2.2: Pattern (243) SITTING WALL, part 2. [Alexander et al., 1977]

five patterns and their impressions from using them in field work. In the following years, several pattern languages in the domain of computer science were proposed, focusing on the field of software engineering in particular.

The most influential collection of design patterns was published by the “Gang of Four”<sup>1</sup> [Gamma et al., 1994], containing 23 patterns for problems developers encounter frequently while programming. Their work has become the definitive book for many programmers and a series of workshops, e.g., the PLoP (Pattern Languages of Programming) conference series, and is widely known and accepted inside the software engineering community. However, it is targeted towards domain experts only, and differs a lot compared to the architectural design pattern format by Alexander. He did not agree with their proposals, the way how his ideas were adapted to another domain, and the direction the software engineering patterns were heading, as he clearly stated in his keynote at OOPSLA [1996].

*Gang of Four* book  
and mixed reception

### 2.2.2 Human-Computer Interaction

Not only software engineers intended to transfer the idea of design patterns to their domain in search for guidelines and standards. In *The Design of Everyday Things* by Norman [1990], which is a reference work for every HCI designer, he mentions the vividness of Alexander’s pattern language for the particular domain of architecture. However, there are significant differences for design patterns in HCI. E.g., buildings are static structures, therefore Alexander uses a spatial categorization for his pattern, from large towns to small constructions. As the term HCI suggests, users are interacting with technology in this field, and user interfaces are often dynamic, introducing a time dimension in addition to the two or three dimensions of space.

Time in design  
patterns

Considering these issues, the *Common Ground* is one of the first comprehensive sets of HCI design patterns and the most influential work to establish the concept in HCI. This

---

<sup>1</sup>“Gang of Four” is an expression introduced and accepted by the community, mainly due to the long and ambiguous title of the book.

Interdisciplinary  
pattern languages

pattern collection about user interface design by Tidwell was proposed at the PLoP '98 conference and, in an extended version, published as a book in [2005]. However, the first book on HCI design patterns appeared in [2001] by Borchers, introducing three pattern languages for interactive exhibits, the idea of interdisciplinary pattern languages to complement each other, and HCI design patterns as the link between engineering and application domain patterns. An in-depth overview of the most important HCI design pattern languages follows in chapter 3—"Related Work".

## 2.3 HCI Design Patterns

HCI design pattern  
definition

The definition of what exactly a design pattern is varies across the several domains. Alexander's definition for the realm of architecture is stated at the beginning of this chapter. The HCI community conducted several workshops at the CHI conference series on this topic and agreed on the following:

"An HCI design pattern captures the essence  
of a successful solution to a recurring usability  
problem in interactive systems."

[Griffiths et al., 2000]

Furthermore, the elements that every pattern should contain were listed. We will now look into the details of a typical HCI design pattern's elements, based on the results of the CHI workshop.

The following explanation of the design pattern format is primarily oriented towards the elements present in the languages by Alexander et al. [1977] and Borchers [2001]. Therefore, to see an actual example of the elements and how they look in a pattern, refer to Figures 2.1 and 2.2.

Convey the pattern's  
solution

**Name.** The function of this element is to give a very short idea about the solution that the pattern proposes, and should consist of two to four words. Choosing the right



name can be a difficult process; a wrong name can result in a complete misunderstanding of the pattern. Alexander and Borchers distinguish their pattern titles from section headings by writing them in small caps, like WINDOW PLACE or INCREMENTAL REVEALING. Additionally, most languages assign a unique identifier (ID) to every pattern, consisting of digits, an alphabetical collection prefix, or a combination of both.

**Ranking.** To illustrate the usefulness and applicability, every pattern usually has a ranking, sometimes also referred to as “confidence”. E.g., Alexander uses zero to two stars, whereby zero stars denote a solution that is currently the best available, but there are probably better solutions possible and further investigation can improve this pattern. On the contrary, a two-star pattern provides a sustainable and likely ideal solution for the specific problem.

The solution’s sustainability

**Illustration.** While the name conveys the solution idea to the reader, the illustration serves a more general purpose. It works as “sensitizing example”, a picture that shows a situation where this pattern might be applied to or has been implemented already. Instead of using a photograph, the illustration can also be replaced with a series of photos or even a movie. However, it is not recommended to use a sketched picture or a diagram, since this interferes with the diagram element (see below).

Sensitizing example

**Context.** Not every pattern is applicable in every situation. The context part contains use cases in which the pattern might be interesting for the reader and lists requirements that need to be fulfilled before it is valid. Unless it is a top-level pattern without ancestors, the context also mentions other patterns that depict some of these requirements or situations.

Requirements with pattern links

**Problem statement.** Printed in bold, the problem statement, as its title suggests, names the problems leading to the pattern’s solution. At first glance, this might be a part easy to find, but identifying a good problem statement is both difficult and critical. One of the core ideas Alexander proposed are the *conflicting forces*, which are problems that

Conflicting forces

induce certain solutions, but compete with each other. The solution proposed in the pattern has to balance these conflicting forces, and the better the identified forces are, the better is the solution and therefore the pattern. This part is a central one in an HCI design pattern and many argue that the forces are the most important element of a pattern. The bold-printed problem statement in Alexander's pattern SITTING WALL, depicted in Figure 2.1, shows an example for conflicting forces.

Detailed forces and examples

**Rationale.** The rationale, also called the "problem description", is the biggest section of a pattern. It elaborates on the conflicting forces mentioned in the problem statement, names other, less striking forces, and points to examples for the solution of this pattern. Many pattern languages, e.g., most contributions to the PLoP conferences, divide this into two parts: a list of forces and examples. However, by combining these elements, the examples can be used to explain the forces, which is particularly useful for novice readers. The examples also show empirical evidences both for the forces and the solution, increasing the validity and resulting in a higher credibility of the pattern for domain experts. Borchers calls this the "inductive approach" in his book [2001], due to the examples stated first and inducing a general solution for the problem.

General instruction how to solve or avoid the problem

**Solution.** Introduced with the keyword "therefore", the bold-printed solution statement provides the reader with a clear instruction how to apply the pattern. It is broad enough to fit to all scenarios mentioned in the context, specific such that the designer knows what to do without much further reading, and generative in a way that it does not supply a blue print of an existing solution. A good solution statement is short, precise, and inspires the reader to solve a problem using the pattern, but creating an actual implementation of the solution that is even better than the examples listed in the pattern.

Sketched image of the solution

**Diagram.** In contrast to the illustration, the diagram conveys the solution idea, but without being too specific. To complement this creativity-supporting fashion, the diagram is often hand-drawn [Alexander et al., 1977], [Borchers, 2001].

**References.** The last part of the pattern are the references, going more into detail about the specific application scenario and unveiling possible further problems one might encounter. Similar to the context, the references name other patterns; these bidirectional links distinguish a pattern language from a pattern collection.

Links to other  
patterns

## 2.4 Advantages of HCI Design Patterns

Traditional guidelines come in many different shapes, with different purposes and target audiences, e.g., the *Seven Principles of Usability* by Norman [1990], the *Eight Golden Rules* of interface design by Shneiderman [1997], or the *Macintosh Human Interface Guidelines* by Apple Computer Inc. [1993]. Some less extensive guidelines addressing the tabletop domain are mentioned in chapter 3—“Related Work”. HCI design patterns add up to this diversity of guidelines by collecting the design experience and creating a body of knowledge, and they bring a lot of advantages to the table that distinguish them from usual guidelines and make them more valuable. In this section, we outline the most important characteristics of pattern languages and their quality in particular for the domain of HCI.

**Lingua franca.** One important goal of Alexander’s patterns is to help non-expert readers acquiring basic knowledge about architecture, such that clients of buildings can express their wishes, needs, and demands for the construction to the architect or builders. Erickson [2000] carries this over to HCI, where the vocabulary itself often handicaps communication between designers, developers, and customers. Design patterns provide novices with an easily accessible expertise in a language they understand, which empowers them with a basic terminology of the specific area. Therefore, pattern languages help to realize the concept of “participatory design”, where users and experts work together on a product in close contact, which benefits both sides: the users can give valuable feedback and contribute to the design process, while domain experts create results that better suit the expectations of the users.

Provide users with a  
vocabulary, enhance  
participatory design

Offer relevant  
advices suitable to  
the scenario

**Solutions in context.** Traditional guidelines offer help for specific problems, or give general instructions how to implement or not implement a solution. However, in a collection of hundreds of useful guidelines, only a few might be relevant to the designer's problems, while many of the other advices do not fit in a particular scenario. Pattern languages—and this is what distinguishes them from pattern collections and qualifies them as “language”—provide solutions in context, i.e., they state the relevant scenarios of a pattern where the mentioned problems may appear and its solution is suitable [Dearden and Finlay, 2006]. Alexander emphasizes the importance of a pattern language's connectivity: “In this network, the links between the patterns are almost as much a part of the language as the patterns themselves” [Alexander, 1979, p. 314]. Browsing and selecting HCI design patterns by following the links in its context and reference parts creates a subset of applicable patterns, matching the need of the designer for her product.

Narrative language  
and clear  
arrangement

**Readability.** Unlike scientific papers or most books providing expert knowledge, patterns support an easier language, since they are supposed to be written in prose [Alexander et al., 1977, p. xli]. This makes design patterns both more interesting to read, enhances comprehensibility, and captures the reader, increasing the value of the language. In addition to the narrative speech, the consistent format and structuring through typography used in many design patterns, e.g., those of Alexander and Borchers, makes it easy to skim the language and search for relevant patterns.

Different scale of  
application

**Different abstraction levels.** Alexander's architectural pattern language starts with large-scale patterns for the layout of regions and towns on top, continues with places, large buildings, and communities, and ends with small constructions, apartments, and single rooms. This scope of application in terms of spatial dimension is also visible in HCI design pattern languages, though on another level. E.g., the pattern language for interactive exhibits [Borchers, 2001] ranges from patterns about the general arrangement, such as ATTRACT–ENGAGE–DELIVER, up to patterns for specific interface problems such as a recommendation for

a DYNAMIC DESCRIPTOR. Guidelines or style guides often cover one particular area rather than a broad spectrum; and if they do, these different areas are not as seamlessly connected as in a pattern language.

**Generativity.** Dearden and Finlay [2006] support the thoughts of Alexander et al. [1977], who stresses that designers produce results with a high quality, a harmonious concept, and more than the sum of the single patterns combined. By applying design patterns in an early stage of designing a concept, or even use them to improve an existing project, the generative structure of a pattern language—in contrast to a pattern collection or guidelines without connectivity—creates good design with, as Alexander calls it, *quality without a name*. Besides this understanding of generativity, there is another interpretation among HCI researchers: design patterns do not propose blueprint solutions, but rather inspire for new ways to avoid or circumvent a problem [Dearden and Finlay, 2006].

Create good design,  
inspire the designer

**Teaching.** Many studies and examples show that design patterns are more suitable to teach usability than traditional guidelines, heuristics, or standards. One of the most recent studies by Koukouletsos et al. [2009] compared the results of students after short design sessions, where the different groups were assigned to read guidelines or design patterns, respectively. In conclusion, students using the design patterns produced better results and solutions with higher usability than the other group, which agrees to numerous former evaluations of pattern languages. It is worth mentioning that in all these studies even a small number of design patterns was sufficient to produce a significant difference, therefore also unfinished or small pattern languages have a high value for teaching usability.

Convey better  
understanding of  
usability

**Examples.** Every pattern lists examples from the real world, showing applied samples for the solution. As aforementioned, examples are used to explain the forces and convey the general problem this pattern addresses, and they serve as a rationale for the problem statement, introducing novices to the specific scenario. These examples are also important to give evidence that the solution has

Samples for validity,  
rationale, and  
understanding

been applied successful in the real world. In general, three samples are necessary to give a pattern a certain credibility. However, it is not always possible to find three examples, especially in domains that are less commercialized, where research papers may be the most important contributions to mention.

Further advantages of HCI design patterns over traditional guidelines have been mentioned, e.g., the ability for designers to “think outside the toolkit” [Tidwell, 2005], the “organizational memory” and “technical lexicon” a pattern language creates [Dearden and Finlay, 2006], or the possibility to combine pattern languages from different domains to create an interdisciplinary design pattern framework [Borchers, 2001].

Despite all these advantages, pattern languages are still less known than guidelines, heuristics, or standards. In architecture, patterns were disregarded and rarely applied. One reason that is often used to explain this is that architects did not believe that every reader and non-expert could get a reasonable level of expertise by reading one book, and they did not like the idea of a world full of architects, interfering with their work, either. Since most HCI designers prefer including users in the early stages of design, this problem does not apply for this realm. However, to create a pattern language that benefits the community, all these above aspects have to be taken into consideration, and both professionals and non-expert readers have to be convinced of its value before the work becomes a successful contribution.

## Chapter 3

# Related Work

*“Learning is remembering what you are  
interested in.”*

—Richard Saul Wurman

The related work to look at is divided into two different parts. In the first section, existing guidelines and taxonomies in tabletop research are presented. The second part of related work names important HCI design pattern languages that show the successful application of the design pattern concept to HCI topics other than tabletops.

### 3.1 Tabletop Guidelines

The need for standards in the community of interactive tabletops has been stated on multiple occasions by several researchers. Some contributions started with set of guidelines for specific areas of tabletop design, e.g., social aspects of collaboration or taxonomies focusing on hardware of interactive tabletops.

### 3.1.1 Collaboration

Due to the large work space that interactive tabletops offer, they are particularly useful in collaborative environments. Therefore, many researchers have focused on this area in recent years and presented their observations. Summarizing many of these contributions, Scott et al. [2003] elaborate on co-located collaboration on interactive tabletops and state eight guidelines that designers should consider when building tabletop systems:

1. Support interpersonal interaction.
2. Support fluid transitions between activities.
3. Support transitions between personal and group work.
4. Support transitions between tabletop collaboration and external work.
5. Support the use of physical objects.
6. Provide shared access to physical and digital objects.
7. Consideration for the appropriate arrangement of users.
8. Support simultaneous user actions.

Sensitize for problems, but partially obsolete

These guidelines help to focus on the problems in tabletop design and list various examples from research in HCI and CSCW (computer supported cooperative work), and they are the most comprehensive approach to create a body of knowledge for tabletop designers. In contrast to design patterns, they lack clear and detailed instructions for designers, and address only expert readers, giving an overview of research problems and solutions, as well as suggesting fields that need further investigation. Additionally, some of the statements in the paper have become obsolete, e.g., the eighth rule outlining the demand for simultaneous input—basically every recent tabletop system supports multi-touch.



In a more recent paper five years later, Wallace and Scott [2008] give a more general overview of tabletop issues. They choose three important aspects of tabletop design (“software interface, physical form, and connectedness”) and outline the implications resulting from five different contexts (“social and cultural, activity, temporal, ecological, and motivational”) that designers have to consider. While their observations are less obsolete, more general, and cover an even broader area in tabletop design, this is also the biggest disadvantage: they are vague and difficult to apply. There are hardly any concrete instructions, which makes it hard even for experts to improve their design and come to a decision based on this paper. A short summary of this paper could be “consider the tasks and users before building your tabletop”—which once more supports the idea of design patterns, since they always offer solutions in context.

Good coverage, but no clear instructions

### 3.1.2 Hardware

Focusing on 3D tabletop interaction, which includes all systems with two-dimensional surfaces extended by a virtual environment supporting a third dimension, Grossman and Wigdor [2007] present a taxonomy. Their goal is to categorize the available systems to give designers an overview of blank spots in the taxonomy and show practitioners the possibilities of tabletop interaction on existing systems. In a presentation at the ITS (Interactive Tabletops and Surfaces) conference 2009 by the second author of the aforementioned paper, an extended taxonomy included a more general collection of different tabletop system properties, see Figure 3.1.

Taxonomy on hardware capabilities to categorize existing systems

A categorization of hardware seems to be difficult though, since the progression of technology emerges new systems constantly, rendering those taxonomies incomplete. This is one of the reasons why our patterns presented in this work will not include basic hardware design patterns. Additionally, a taxonomy serves a completely different purpose than the pattern approach. It does not consider problems or forces that lead to solutions or inspire for new systems, instead it offers an overview of existing tabletop design.

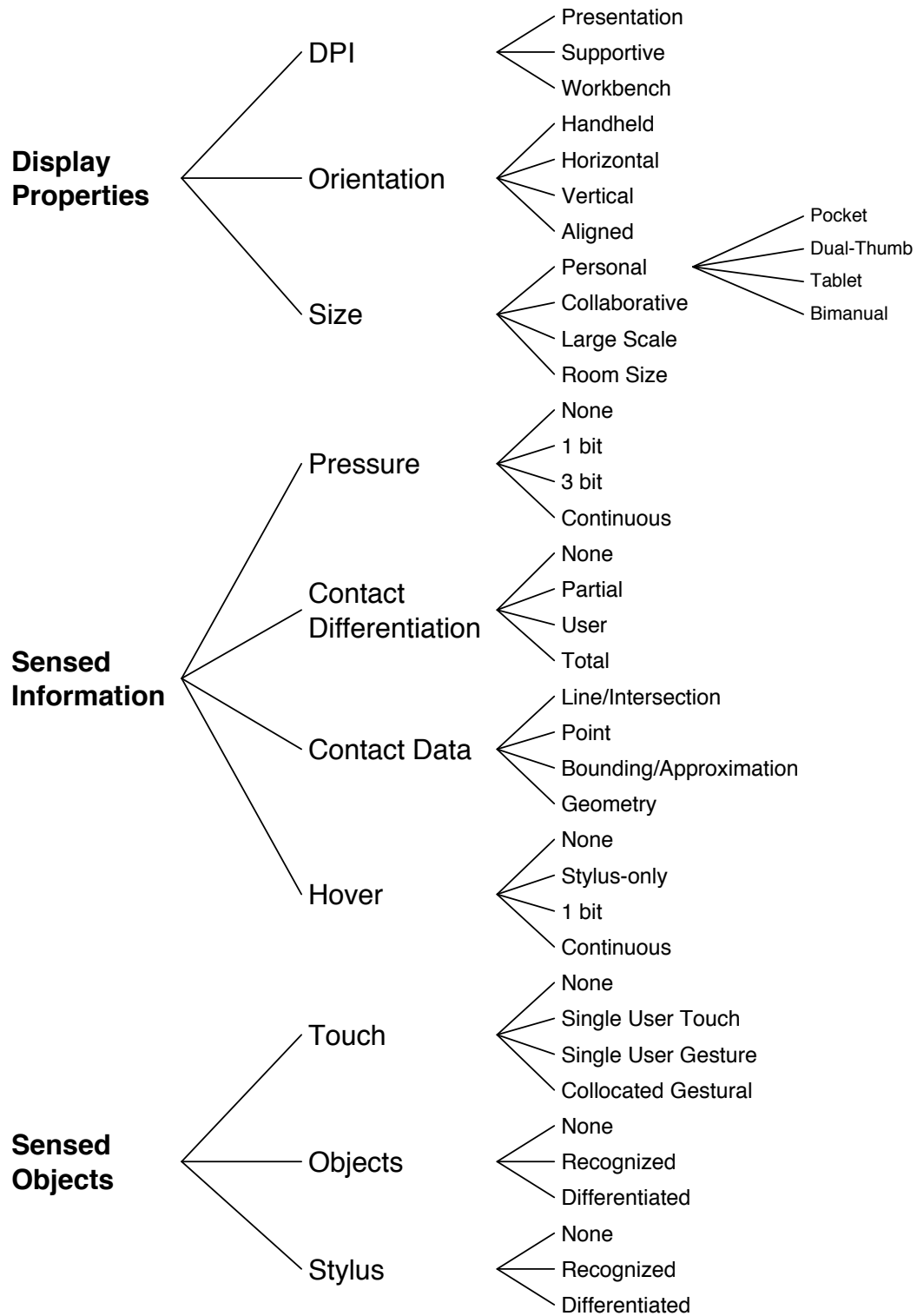


Figure 3.1: Taxonomy of fundamental hardware capabilities. [Wigdor, at ITS 2009]

### 3.1.3 Studies and Observations

Ryall et al. [2006] compared different existing tabletop systems, observed participants using them, and stated design considerations based on their experiences. Some of these observations are formulated as concrete instructions and give precise hints how to improve tabletops, e.g., they outline that existing desktop interfaces are designed for the use of mouse and need to be changed for the use of finger touch input. Other observations such as “users gravitate toward single-finger interaction” help understanding the problem, but are less precise to guide designers, developers, or engineers on system design and how to overcome these problems.

Two distinct long-term studies revealed informal guidelines similar to the above, with the goal to identify problems and direct the research into areas that demand further investigation. In the first study by Wigdor et al. [2007], one knowledge worker used a tilted interactive surface for 13 months, while in the second observation by Morris et al. [2008] eight participants worked on a horizontal system in addition to their desktop computer for one month.

Collected experiences stated as design considerations

Both studies give instructions on aspects to consider when designing an interactive tabletop or developing software for it, but those are hardly applicable to general systems. They do not offer any context on the specific application scenarios, user environments, or tasks for each finding, which is an important issue in tabletop design, as former studies outlined.

The need for standards has also been expressed by commercial tabletop developers, directed to HCI researchers [Wigdor and Morrison, 2010]. At CHI conferences in 2009 and 2010, the demand for standards among applications and frameworks was addressed in a Special Interest Group, but without any significant results yet. However, it shows the need for guidelines in both research and industry and motivates to increase the efforts for creating a tabletop pattern language.

The search for standards in conference meetings

## 3.2 HCI Design Pattern Languages

### 3.2.1 Related HCI Design Patterns

Design patterns for gestures illustrate basic concepts

As of today, there is no HCI design pattern language that provides general guidelines for the design of interactive tabletops. Saffer [2008] proposes a collection of 21 patterns for the use of gestures in the design of interactive systems, whereby twelve of these patterns are targeting touch interaction and nine patterns illustrate gestures initiated without touching a surface. The pattern language mentions several examples, ranging from small touch screen devices such as digital cameras, kitchen equipment with a touch screen, cell phones, and occasionally also tabletop devices, but only commercially available products and no solutions from research that did not make it into the industry yet. With its simple, almost superficial instructions, it is easily accessible even for novices, but probably less useful for professional designers that have already a decent level of experience. Therefore the real value of this pattern language is introducing the non-expert audience to the possibilities of gestures on touch screens, rather than establishing standards for the community.

Interactive exhibit design patterns, partially applicable for tabletops

A language targeting novices as well as professionals, is the HCI design pattern language for interactive exhibits by Borchers, one of three presented pattern languages in his book [2001]. While his 17 patterns mention no interactive tabletops, some of the patterns are definitely viable for this domain, like *INVISIBLE HARDWARE* or *AUGMENTED REALITY*. If a tabletop designer is building a device for an interactive exhibit, the whole language gives useful insights and outlines important aspects to consider, but it references no examples from interactive tabletops and therefore lacks validity for this area.

Media spaces design patterns with different scope, though useful hints

Another set of 19 design patterns by the same author illustrate important HCI aspects to consider when rebuilding a room into an augmented workspace, furnished for meetings with movable walls, tables, chairs, and digital equipment [Borchers, 2009]. Patterns like *FLEXIBLE FURNITURE* from this language could be transferred to a language for in-

teractive tabletops, which are even mentioned in this work by the pattern INTERACTIVE TABLES. However, the focus of the pattern language is equipping a room with suitable furniture to enhance the working experience, therefore it cannot deal with the problems in tabletop design in this scope.

### 3.2.2 User Interface Patterns

As aforementioned, one of the first comprehensive and most elaborated pattern languages for interface design by Tidwell [2005] was created in 1998 and refined in many iterations. In its current version, it consists of 94 design patterns, and covers not only desktop user interfaces, but also web design problems and solutions. Because of its constant iterations and improvements, this pattern language still is probably one of the most valuable, and it shows the sustainability that pattern languages provides. Unlike traditional guidelines, it offers precise and clear instructions with examples, and it has managed to stay valid after 12 years, a time period with many changes in the appearance of user interface design.

Sustainable patterns  
for interface design  
by Tidwell

While Tidwell mainly aims at patterns for desktop application user interfaces, there are several pattern languages for user interface design focusing on web design. Within one year, two books appeared that provided design patterns for web sites: *The Design of Sites* by [van Duyne et al., 2006, 1st edition in 2002] containing 107 patterns in 13 categories and *Web Usability* by Graham [2003] with a collection of 79 patterns. Both pattern languages are closer to Alexander's format than, e.g., the "Gang of Four" book or even Tidwell's patterns. However, Graham's work contains less pictures, more text-oriented descriptions, and less examples, whereas van Duyne et al. heavily modified the style of Alexander while preserving the general format and contents. *The Design of Sites* also provides small, letter-sized handouts for each pattern, giving a brief overview of the problem and solution. A recent addition to the diversity of pattern language books on web design by Vora [2009] consists of 104 design patterns, in a layout oriented more towards the software engineering design pattern format.

Three different books  
on web design  
patterns

A publication of a book with design patterns has its disadvantages: a new edition of the book has to be created even for small updates, the creation process of a book is more complicated than, e.g., a paper, and a book of patterns is harder to browse than digital material. Therefore, web sites for design patterns have been created even before the first books on HCI design patterns were published.

Well-known websites  
for collaborative  
design patterns

In 2000, van Welie started the *Interaction Design Patterns Library* [H:Welie2000], a web site where HCI researchers could submit and discuss design patterns. As of April 2010, the patterns are still being updated, currently counting 131 in three categories (“User needs”, “Application needs”, and “Context of design”) and further sub-categories. Similar approaches are the *Yahoo! Design Pattern Library* [H:Yahoo!2005], the *User Interface Design Patterns Library* [H:Toxboe2008], and the *Infragistics Quince UX Design Patterns Explorer* [H:Infragistics2009]. The latter web site focuses more on the discussion part than on the urge of identifying new patterns. Its main goal is to share experiences of developers that applied design patterns to their work and rate the pattern, its help for the design process, and the value for the resulting design.

Incorporate existing  
patterns in a new  
language

All these design pattern libraries are admittedly no light at the end of the tunnel for the tabletop community on the search for guidelines and standards. However, they are important to look at for two reasons: First, they illustrate that the design pattern approach can be a successful solution to create a sustainable body of knowledge, even for a domain that changes dramatically over time and where new problems continue to emerge. Second, as the mentioned examples of the pattern languages by Saffer and Borchers show, existing patterns from other languages can be altered, updated, and incorporated to other pattern libraries. Once the first standardized software frameworks have been established in the domain of interactive tabletops, one can browse these pattern languages, choose those patterns that fit to the interaction on touch devices, and create a new pattern language for user interfaces on interactive tabletops.

The HCI design patterns community is in a continuous exchange and offers several web sites that keep track of new developments in this field. In the following, we list four

of the most important web sites that offer not one specific pattern language, but rather a compilation of the available work in the field of design patterns:

Pattern library web sites

**The Interaction Patterns Home Page** is a web site that is frequently updated and mentions several contributions about design patterns (including, but not limited to HCI) [H:Erickson1998].

**The HCI Patterns Home Page** lists important news about HCI design patterns [H:Borchers1999].

**The Pattern Gallery** offers an overview of comprehensive pattern languages published [H:Fincher2000].

**The Hillside Group Design Patterns Library**, a large web site which contains publications from the PLoP conference series. It focuses mainly on software engineering, but also includes some HCI design patterns and is the most active source on design patterns in research [H:Hillside1997].





## Chapter 4

# The Pattern Development Process

*“If you don’t make mistakes, you aren’t really trying.”*

—Coleman Hawkins

Guidelines, heuristics, and standards give experts a fundamental overview of the current problems and the solutions in a particular domain, but they can also aid novices to acquire comprehensive knowledge. In this chapter, we elaborate on the creation process of our tabletop pattern language to offer such guidelines, and give a detailed overview of the iterations that were conducted to improve the design patterns and issued the final version.

We started with an extensive literature review, searching for existing approaches to guidelines, but also different kind of tabletop systems, frameworks, and studies about interaction and collaboration. Designing the concept for an interactive tabletop system requires a lot of know-how from multiple disciplines, e.g., basic architectural knowledge for the table construction, understanding technological details of the display and camera mechanism, software engineering skills for the tracking algorithm, and considering the social aspects of users working at the table. Therefore, the literature review did not only cover papers focus-

Literature review  
across multiple  
disciplines

ing on tabletops, but was also more oriented towards different areas such as ergonomics of hardware, mobile touch devices, or software frameworks, relevant for tabletops as well.

## 4.1 Identifying Problems

Problem-oriented approach

While creating a database of knowledge about interactive tabletops, we chose an problem-oriented approach to gather a starting point for our HCI design patterns. Therefore, we compiled the following list of problems in tabletop interaction that designers, developers, and practitioners have to deal with (in no particular order):

- **Text input**, e.g., lack of haptic feedback
- **Text interaction**, manipulating text on the tabletop
- **Exhaustion, fatigue, and ergonomics**, e.g., height, size, form factor, material, arrangement, technical approach
- **Reachability**, on large tables
- **Sufficient space**, on small tables
- **View direction**, multiple angles of view
- **Unfamiliarity with gestures**, since new gestures are unintuitive
- **Occlusion**, by hands and objects
- **Obstacles**, e.g., objects on the table
- **Loss of precision**, compared to PCs
- **Social awareness**, see the input of others, and his actions as well (in remote scenarios)
- **Standardized software framework**, as there is no common toolkit yet
- **Resolution**, related to size

- **Data retrieval**, access and interact with large amounts of data
- **Customization**, every user has its own preferences for the physical and virtual setup
- **Augmented Reality**, for third dimension applications on tabletops
- **Transitions**, between interactions, physical objects, and users
- **Multi-pointer handling**, desktop operating systems and applications only support single pointers
- **Digital paper**, can tabletops replace paper
- **Turn-taking**, equity of participation
- **Orientation**, horizontal vs. vertical
- **Diversity of input**, e.g., pen, tangibles, single touch, multi-touch, hand shape
- **Tracking techniques**, e.g., capacitive, resistive, frustrated total internal reflection, diffused illumination

With this approach, we ensure that the HCI design patterns offer “solutions” instead of “features”, i.e., the patterns solve actual problems. Alexander [1979] names the identification of problems as one of the important steps that are necessary to discover design patterns.

## 4.2 Initial Pattern Language

As aforementioned and as visible from the list of problems, interactive tabletops is a large research domain with several different disciplines from related areas. To keep the start of our pattern language focused enough and to avoid a low quality of design patterns, we narrowed our scope down. We aim to create a comprehensive pattern language for tabletops, therefore we decided to exclude subjects that would move our language too far away from sustainable,

valuable, and generally applicable design patterns. Therefore, we only considered tabletop interaction issues in collaborative workspaces, and neglected areas that are too specific, too large to cover, or not sustainable, e.g., tracking techniques or Augmented Reality.

Create pattern *stubs*

The process of pattern creation requires many iterations, before an HCI design pattern is in a valuable state and can be seen as finished. Therefore we started early in the process with pattern writing by creating pattern *stubs*, small patterns containing only a problem statement and the solution. We consider the following three steps necessary to be completed before we compiled a stub into a real HCI design pattern, similar to the concepts and requirements proposed by Alexander:

1. Name an *invariant solution*.
2. Create a *problem statement*.
3. Identify clear *conflicting forces* that induce problems and lead to the proposed solution.

The third step in particular is the one that distinguishes patterns from other guidelines, but it is also the most difficult step in pattern writing. Our database contains more than 250 papers about interactive tabletops, each of them offering probably multiple solutions. In the first section of this chapter we presented a list of problems in tabletop interaction. However, there are many problems that remain unsolved, and many solutions that are either not invariant or more a “feature”, in terms of a pattern definition for a design solution. After careful consideration and discussions about the quality of the conflicting forces, we finished with a set of initial 13 patterns (Figure 4.1). As format for the patterns, we chose the same layout as Borchers in his HCI design pattern languages for interactive exhibits [2001] and augmented workspaces [2009].

Examples for validity of HCI design patterns

Among the design pattern community, the *Rule of Three* has established, which stands for the number of examples a pattern needs to mention to be valid. For the validity of our patterns, this was of utmost importance, since there is

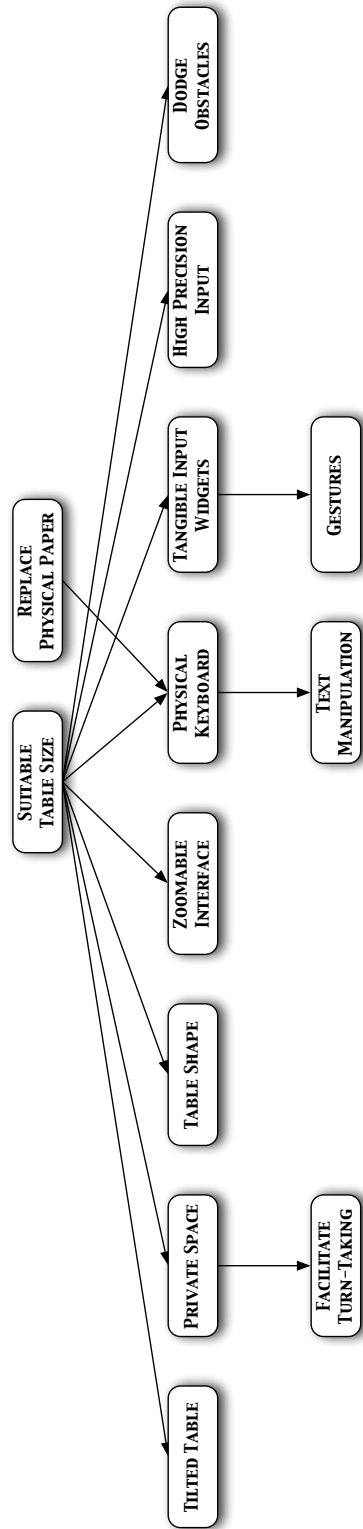


Figure 4.1: Initial graph of 13 patterns.

no other way to measure the quality of a pattern language. A workshop at CHI 2000 agreed on the examples as sufficient justification for an identified HCI design pattern:

“Therefore, the *Examples* section of a pattern should include brief summaries of results from representative user studies, with references to the original publications. This information will satisfy the demands of scientific rigour of the professional and researcher, but also show the amateur that the suggested solutions have been tested empirically.”

[Borchers, 2000]

Three examples in  
tabletop design hard  
to find

No exact number was stated by the HCI design pattern professionals in this workshop, but the number of three examples has been applied by many pattern authors, and therefore all the 13 initial patterns had listed three or more examples. However, tabletops are rarely commercially available and only in a limited variety of different designs. The majority of solutions to the identified problems mentioned above has been proposed by researchers, which makes it hard to find three examples for a solution that share the same pattern. As a result, there are five patterns in our final pattern language that reference to less than three examples, but still remain in the collection. They were evaluated as patterns with a sufficient quality despite their lack of applied examples, more on that in section 4.4—“Evaluation”.

### 4.3 Feedback and Refining

Feedback from  
tabletop design  
experts

We presented the concept of HCI design patterns for interactive tabletops at the ITS 2009 conference in a poster session [Remy et al., 2009] to gather feedback from experts and practitioners of the tabletop community. Besides the pattern language graph depicted in Figure 4.1, we provided the pattern DODGE OBSTACLES as a sample, to outline the structure, layout, and content of a pattern. In the following, we list the most important points mentioned by the audience:

- Define the user group more exactly
- Ensure that the pattern language is accessible for non-experts as well as professionals (avoid genre-specific jargon where possible)
- Use simple English rather than a complex language
- Consider carefully whether patterns are hardware-dependent
- Create more basic design patterns
- Attach example implementation if available
- Publish as web site
- Let the community contribute patterns (some even mentioned patterns they would write themselves to add them to the language)

It should be noted that literally no one was familiar to the concept of HCI design patterns. However, following a brief introduction to the design pattern idea with the help of the sample pattern, they agreed on the concept and its potential to overcome the lack of guidelines in tabletop design. We refined our pattern language based on the feedback received. This includes not only the aforementioned feedback from the conference, but also other additional suggestions from tabletop designers and design pattern experts.

Improving patterns  
based on feedback

In the process of pattern writing, there are no clear distinguishable iterations. The patterns rather changed in a constant cycle of small iterations, as soon as new evidence or other suggestions for changes appear. To keep track of these changes, we switched our pattern repository to a web site (linked on the [Tabletop Patterns Web Site](http://tabletoppatterns.org/)<sup>1</sup>), based on the [Tiki Wiki](http://info.tikiwiki.org/tiki-index.php)<sup>2</sup> software. We slightly modified the layout, such that every page created with a certain template mimics the design pattern style automatically and provides a consistent format for all patterns. Additionally, the built-in system that keeps track of all previous versions provides us

<sup>1</sup><http://tabletoppatterns.org/>

<sup>2</sup><http://info.tikiwiki.org/tiki-index.php>

with a history of changes, offering the possibility to see the changes made over time.

23 patterns ready for evaluation

In addition to the changes to the existing patterns, we identified ten new design patterns during the refinement that were added to the language. We finished with 23 HCI design patterns for tabletops (see Figure 4.2), which now had to be evaluated to ensure the validity and quality of the patterns.

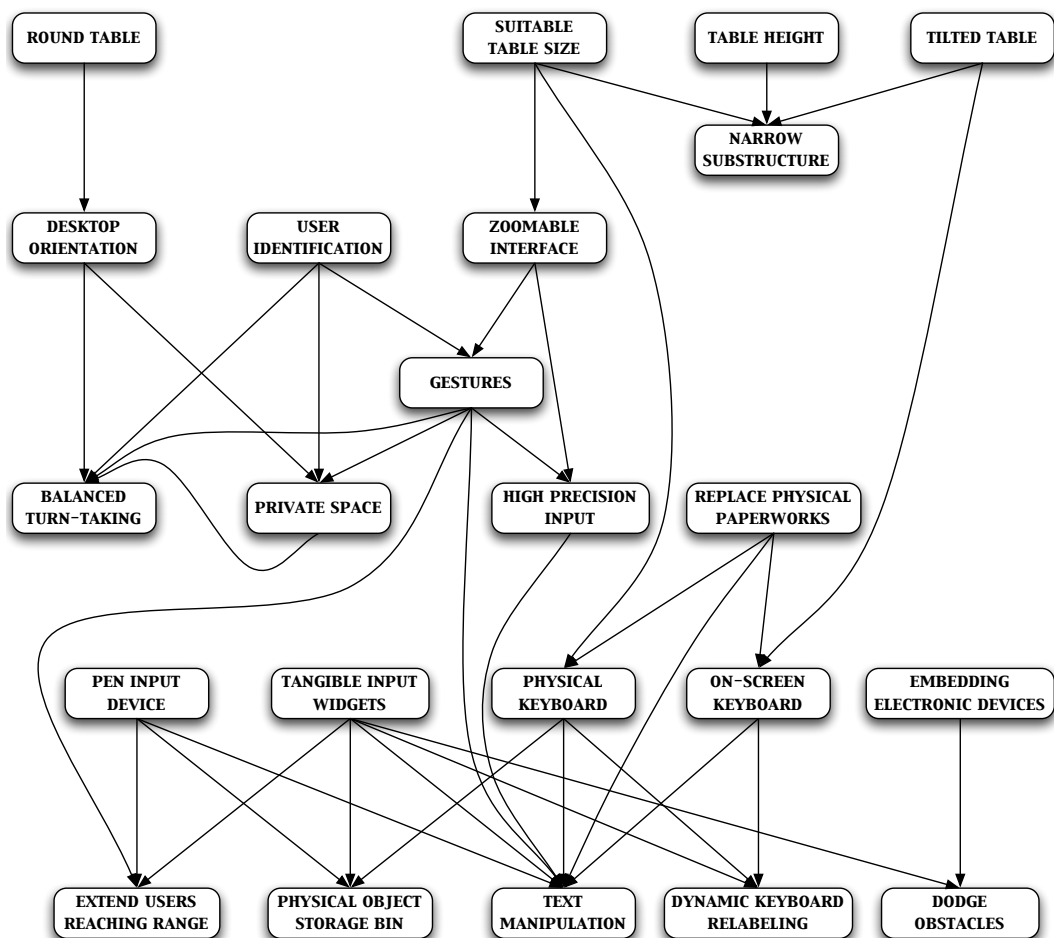


Figure 4.2: Graph of 23 patterns before evaluation.



## 4.4 Evaluation

Identifying HCI design patterns is a difficult process, and not always reveals useful design patterns that match the original idea. Alexander states that discovering the invariant, which is necessary for a design pattern, “is at least as hard as anything in theoretical physics” [Alexander, 1979, p. 261]. Griffiths and Pemberton [1999] observed a similar experience: while teaching HCI design patterns and letting students identify design patterns, many students created so called “anti-patterns” that captured bad design and stated rules of how not to do something, rather than providing positive instructions and useful solutions. She emphasized that the students understood the concept of design patterns, but it was difficult for them to identify real-world examples of good design and compile them into patterns. Therefore, it is important to evaluate design patterns, ensure that they capture an invariant, and improve their quality.

Difficulty of identifying design patterns

To evaluate pattern languages, there is an established process in the design patterns community, the *writers' workshop* [Gabriel, 2002]. Every participating design pattern author submits a pattern language prior to the workshop, so all workshop participants can prepare their feedback on the patterns. At the workshop, each of the proposed design patterns is then reviewed by a group of eight to ten other authors, in an alternating manner. To achieve the best possible feedback for the pattern author, every writers' workshop follows an exact procedure, summarized as follows:

Writers' workshop

1. The author welcomes the participants, cites a sample sentence of his or her pattern to remind of its core idea, and steps back to become a silent listener to the discussion.
2. One moderator summarizes the pattern.
3. Suggestions for things to keep (form and content).
4. Discussion about improvements (form and content).
5. Sandwich: summarize positive points.

6. Author joins the discussion and asks clarifying questions in a non-defending manner.
7. Before moving to the next pattern, talk about something unrelated.

This schedule has proven to issue outstanding results for the pattern author. As aforementioned, several pattern languages are reviewed in a writers' workshop instead of one single contribution. Therefore, writers' workshops are usually conducted in larger meetings of design pattern experts, on a regular basis. One of these occasions is the PLoP conference series, which hosts several writers' workshops to give design pattern authors the opportunity to present their patterns to the community and receive valuable feedback. In the scope of this thesis though, a writers' workshop was not feasible.

Shepherding as  
evaluation

However, in preparation of the writers' workshop, there is a guided evaluation process, which is called "shepherding". This is an established approach in the design pattern community and used, e.g., in the submission process of the PLoP conferences to help pattern authors improving the quality of their submissions. This shepherding, opposed to the writers' workshop, is done by a single reviewer assigned to every author. While the writers' workshops conducted at the PLoP conferences consist of authors from several disciplines with different expertise, the shepherd assigned is usually both an expert in design pattern writing and in the target domain of the design patterns.

Our pattern language was reviewed by Christian Kohls from SMART Technologies, who currently works on the product design of interactive tabletops, and who also has experience in writing and reviewing design patterns. He reviewed the pattern language for both form and content to an extent similar to those of a writers' workshop, thus revealing important, comprehensive, and useful feedback for our design patterns. In the following, we outline the feedback we received, by summarizing his suggestions and listing the changes we carried out. note that the evaluation was separated into two different steps, each of them with a different scope.

### 4.4.1 First Iteration

In the first iteration, the suggestions were kept on a larger scale, reviewing the format of the patterns and mentioning points to improve for the content in general. The feedback consisted of six annotations, a brief summary of these points:

Six general annotations to our pattern language

1. Change order and put examples after solution.
2. Elaborate on some forces more precisely, separate forces and examples.
3. Improve the context parts, include requirements and scenarios.
4. Include application areas, both in context and if necessary in rationale.
5. Try to stick to one solution for each pattern, different solutions belong to different patterns.
6. If a solution (or an example for a solution) has drawbacks, mention them.

In the following, we document the most important changes, referring to each of the shepherd's annotations.

**1. (Examples after solution.)** After careful consideration, we decided against changing the order. We reason that the inductive approach used by Alexander and seconded by Borchers is especially useful for the tabletop patterns, since tabletops are unfamiliar to many readers. The examples help to illustrate the forces, as they give the reader an idea about the design of interactive tabletops, making it easier to understand the problems. Additionally, since the solutions are mainly presented in research papers rather than industry products, they are no actual real-world examples. By mentioning the examples, we can generalize them to the final solution statement, conveying both novice readers and professionals a more useful instruction. In the second iteration, the shepherd mentions his approval for our approach and seconds that it is more suitable for our topic than other design pattern approaches that separate forces and examples.

Stayed with inductive approach

Phrased forces more precisely	<p><b>2. (Elaborate forces, separate forces and examples.)</b> We reviewed all forces statements once again, in both the problem statement and rationale, and stated them more precisely. In some patterns, such as ROUND TABLE, the forces were sufficient, but in other patterns, e.g., TILTED TABLE, we added a long paragraph, explaining the forces. As aforementioned, we decided against separating the forces and examples, for specific reasons.</p>
New context written for all 23 patterns	<p><b>3. (Improve context.)</b> The context of every of the 23 patterns was written completely new, to reflect the scenario in which the pattern was applicable. During this iteration, many new links between patterns appeared that we had overlooked before; likewise, some existing links between patterns were removed. The context and reference parts of these patterns was updated and the pattern graph changed to match the new connections between the patterns.</p>
Reflect the use case scenario	<p><b>4. (Include application area.)</b> In addition to the context part, the rationale was updated in many patterns to match the application scenarios required for the pattern.</p>
Changed pattern SUITABLE TABLE SIZE	<p><b>5. (One solution per pattern.)</b> This point, though being a generally useful annotation, was particularly targeting the pattern SUITABLE TABLE SIZE. The pattern was no real pattern, but more a problem analysis without solution. We renamed the pattern to LARGE COLLABORATION TABLE. Additionally, a new solution statement was created and the rationale part rewritten, giving the designer instructions to apply a size according to the number of projected users and their tasks.</p>
Outlined disadvantages of solution examples where necessary	<p><b>6. (Mention drawbacks.)</b> We changed the pattern USER IDENTIFICATION as suggested to reflect the drawbacks of the example solutions. In general, this annotation is a valid point, and we updated some other examples in other patterns as well to reflect the disadvantages, such as in ON-SCREEN KEYBOARD or PHYSICAL KEYBOARD. However, the mentioning of drawbacks of a solution has to be handled with care. If a pattern names more disadvantages than advantages of a proposed solution, it becomes an anti-pattern, i.e., an advice of how not to do something, instead</p>

of being a useful instruction for good design. Balancing the conflicting forces is the central part of a pattern's solution, which is obviously often a trade-off and should be mentioned, but only to a certain extent.

In addition to all these changes reflecting the suggested improvements, we changed other pattern elements that appeared to be not precise enough, and renamed the patterns `EXTEND USERS REACHING RANGE` to `EXTENDING REACHABILITY` and `TABLE HEIGHT` to `ERGONOMIC HEIGHT`.

#### 4.4.2 Second Iteration

The second iteration of the pattern language mentioned improvements specifically for each pattern. In the following, we summarize the criticism first, and outline our changes according to the annotations. Note that the shepherd considered some patterns as finished and did not mention any annotations; these patterns are not listed here. Basically, every pattern was changed to a certain extent, but in some iterations the changes were not significant—the most important changes are documented here.

Second iteration:  
targeting patterns  
specifically

##### (01) ROUND TABLE

**Feedback:** Remove the “if applicable” from the solution, improve context to distinguish between other solutions for collaboration tabletops, and group these pattern with `ROUND TABLE` and `TILTED TABLE` semantically. Furthermore, consider the loss of screen real estate when using round projections, and the arrangement of tables in a room and its implications, e.g., tables on wheels.

**Changes:** We removed the phrase “if applicable” and slightly changed the solution statement, and the context was updated to reflect the situation in which this pattern is applicable. Additionally, we added a reference to `ERGONOMIC HEIGHT` as suggested, and the disadvantage of round projections due to the cut-off of pixels that decreases the number of pixels in the projection. We did not follow his suggestion to add the tables on wheels though, since

Slightly changed  
solution and context

this would blur the focus of the pattern. The pattern FLEXIBLE FURNITURE from Borchers [2009] could be added as a reference, if we extend our language by links to other pattern languages later on.

#### (02) TILTED TABLE

**Feedback:** Add the user's preference for specific table angles.

Added user's preference for a certain angle

**Changes:** The forces from the study in the third paragraph were elaborated to reflect the angle preferences more clearly. Furthermore, we added a reference to ERGONOMIC HEIGHT.

#### (03) LARGE COLLABORATION TABLE

**Feedback:** The context does not properly introduce the situation in which this pattern is applicable. Also, the pattern should reflect the dependency between number of users, tasks, and table size.

Give a more specific instruction for the solution

**Changes:** Sample application scenarios were added to the context, to give an idea whether the pattern is applicable or not in a specific use case. Likewise, we updated the solution, as suggested, to give better instructions of the table size that the designer should choose, and changed the rationale for a better connection between the examples and the solutions. Additionally, we added the patterns ERGONOMIC HEIGHT and DESKTOP ORIENTATION in the references part.

#### (04) ERGONOMIC HEIGHT

**Feedback:** Do not mention a specific height value in the solution—this should be kept more general—but rather advise the use of a prototype to find the suitable height. This is probably not a two-star pattern since it lacks evidence.

Reduced to one star and added prototyping

**Changes:** We reduced the pattern ranking from two stars to one star due to the shepherd's justification, on which we agreed. This pattern lacks evidence, as mentioned by him, but nevertheless, we think it is an important aspect for interactive tabletops and necessary for good design, and

therefore qualifies as pattern. The solution was updated by adding the prototyping advice—also to the rationale part—and removing the concrete value from the evidence referenced. Furthermore, we removed the reference to NARROW SUBSTRUCTURE, but put it on the same level in the language graph and added the three first patterns of the language to this pattern’s context part.

#### (05) NARROW SUBSTRUCTURE

**Feedback:** Change the illustration, and focus less on prototyping, but more on actual examples and instructions for the designer.

**Changes:** A new illustration was added to the design pattern that better suits its application. To give an example for the application of mirrors, we added a reference to a paper that elaborated on the use of mirrors for an increase of the tabletop user’s legroom. Additionally, the context was updated accordingly to the aforementioned changes of the other ergonomics patterns. The shepherd mentioned the possibility of creating a new pattern called PROTOTYPING, which we did not follow. While we could imagine such a pattern, it would not suit the tabletop language as such. We could imagine including a prototyping pattern in the future from another pattern language, though. The importance of prototyping was mentioned by the shepherd already in the pattern ERGONOMIC HEIGHT, where he even suggested to add this to the solution. Therefore we kept the general prototyping advice, but slightly modified it to stress that it is one possible way to achieve the best result.

New illustration,  
added reference

#### (10) DESKTOP ORIENTATION

**Feedback:** Change this to a two-star pattern, and add LARGE COLLABORATION TABLE to the context, since it is applicable there, too. The manual versus automatic rotation of interface elements needs further investigation in this pattern, and more evidence of user preference.

**Changes:** The context and ranking were updated as suggested. We added two more references to studies in research about the preference of users regarding automatic orientation versus manual rotation of interface elements. In

Elaborated on  
manual versus  
automatic orientation

addition to the huge changes in the rationale part, the solution was slightly altered.

### (20) GESTURES

**Feedback:** The use of hints is not a good solution, gestures should be intuitive from the start, and not be overused. If possible, add a table with available gestures on tabletops.

Added the demand for intuitivity and a reference to a list of gestures

**Changes:** We mentioned in the rationale part of the pattern that the use of gestures should be limited, and they should be intuitive. While we agree that hints are not a good solution for gestures in the first place, it is not always possible to avoid them. An intuitive gesture is simply a gesture that the user is familiar with, but for new gestures this is sometimes hard to realize. The general thoughts on that were added to the rationale part of this pattern. Additionally, we changed the name from GESTURES to HAND GESTURES to be slightly more specific about the solution, and exclude, e.g., gestures with tangibles. However, we admit that the pattern is still very general, and the amount of gestures is overwhelming. We added another reference to a paper that lists a number of relevant gestures for tabletop devices, in response to the shepherd's demand for a table of available gestures and to increase the validity of the pattern.

### (22) HIGH PRECISION INPUT

**Feedback:** Mention the example of text input on the Apple iPhone [H:Apple2007], where a small popup window as a magnifying glass increases precision.

Apple iPhone example

**Changes:** The aforementioned example was added.

### (23) REPLACE PHYSICAL PAPERWORK

**Feedback:** The rationale part offers only problems and solutions, but no real forces that require these solutions. Some of the hints given are too unspecific, too general, and questionable. Try to clearly identify the forces that make users switch from physical paperwork to interactive tabletops.

**Changes:** During the second iteration, this pattern was the one that received the largest amount of changes. It was



completely edited two times: first, according to the shepherd's annotations. We were not satisfied with this version though, as it still lacked the real feeling of an invariant that is necessary for a pattern. While changing the pattern TEXT MANIPULATION, it occurred to us that the forces of both questionable patterns combined would make one reasonable pattern. Therefore, the pattern TEXT MANIPULATION was removed from the language, and this pattern REPLACE PHYSICAL PAPERWORK was almost rewritten from scratch. We added new references, stated clear and valid forces, and supplied instructions that lead the designer towards a good design, rather than giving rough, unspecific, and questionable hints.

Completely rewritten and enriched by pattern TEXT MANIPULATION

#### (24) BALANCED TURN TAKING

**Feedback:** Turn taking is only an issue in single touch environments, and the pattern should reflect that.

**Changes:** This is a good example that illustrates the importance of a correct design pattern name. The term "turn taking" suggests that we refer to real turns, e.g., by physical constraints as on a single-touch tabletop or in a game. However, as the examples in our patterns show, turn taking is an issue even on large multi-touch walls. With a few changes to the pattern and changing the name to BALANCED PARTICIPATION, the real forces of this patterns are now conveyed more clearly to the reader.

Renamed pattern to BALANCED PARTICIPATION

#### (32) ON-SCREEN KEYBOARD

**Feedback:** Experimental keyboard layouts have disadvantages over the default layouts, they should be used with care. Mention other layouts than the experimental ones. Furthermore, the solution suggests to implement keys as big as possible—this is probably not ergonomic. Additionally, think about a pattern HANDWRITING RECOGNITION.

**Changes:** The additional layouts were mentioned, and the careful consideration of the use of experimental layouts was added as well. We changed the solution such that it recommends to use keys sized accordingly those of physical keyboards. His idea for a HANDWRITING RECOGNITION pattern was added to our list of possible patterns in

Mentioned additional layouts and ergonomic size

the future. As of yet, we do not see enough empirical evidence and consider on-screen keyboards as one of the more promising solutions. However, we mentioned handwriting on a related subject in the pattern PEN INPUT DEVICE.

### (33) TANGIBLE INPUT WIDGETS

**Feedback:** Add “learning environments” to the context.

Learning environments as application

**Changes:** As suggested, we updated the context. Additionally, the pattern name was changes to INPUT TANGIBLES, since this is a more appropriate term, and more frequently used in the tabletop community.

### (34) EMBEDDING ELECTRONIC DEVICES

**Feedback:** Mention the need of sharing data among users and the advantages of working on the shared data on the tabletop.

Added the need to share data

**Changes:** We updated the problem statement to reflect the user’s demand of sharing data. Additionally, the second paragraph mentions the advantages of a tabletop to work collaboratively on shared documents.

### (42) TEXT MANIPULATION

**Feedback:** This is a weak pattern with no real solution yet. Furthermore, it provides no concrete problem statement and no forces. However, there are important aspects mentioned, but there is no real invariant in its current formulation.

Pattern removed

**Changes:** As mentioned in the discussion of the pattern REPLACE PHYSICAL PAPERWORK, this pattern was removed from the language. Its ideas were partially transferred to the other pattern to achieve a better quality of this design pattern.

### (43) DYNAMIC KEYBOARD RELABELING

**Feedback:** Add the instruction to implement a visual highlighting for a keystroke.

**Changes:** The advice, to highlight a key that a user has just pressed, was added. Furthermore, we added suggestions for other keyboard layouts, such as different languages. This moves the pattern further away from being only a “quasi-mode” pattern, referring to its examples and the illustration.

Highlighting and  
further layouts

This concludes the pattern evaluation. The final version of our pattern language is presented in the next chapter.



## Chapter 5

# The Pattern Language

*“... in a healthy society there will be as many pattern languages as there are people—even though less languages are shared and similar.”*

—Christopher Alexander

This chapter contains the final tabletop pattern language. The patterns can be separated into different categories. We added a color coding to these categories and assigned every pattern within the same category an ID with an identical first digit, between zero for top-level patterns and four for patterns at the bottom of the language graph (Figure 5.1):

Unique color and leading digit for different categories

- **Blue:** Ergonomics patterns
- **Green:** Interface patterns
- **Yellow:** Usability and specific collaboration patterns
- **Red:** Extending input patterns
- **Orange:** Special scenario patterns

These categories emphasize the scope of application for the design patterns. While ergonomics patterns deal with construction issues and offer guidance in the early stages of

designing an interactive tabletop, patterns further down in the language have a smaller scope, e.g., extending the input is easy to realize, yet very effective, and can be applied to an already finished tabletop, too.

Each arrow in the language graph of Figure 5.1 corresponds to the connections between the patterns. The context part in each pattern mentions incoming links from other patterns, likewise the references part lists patterns further down in the language that are also connected to each other in the graph. Therefore, the reader can start with any pattern, choose other patterns from its context and references parts that suit a specific situation, and create a subset of patterns applicable to a specific project.

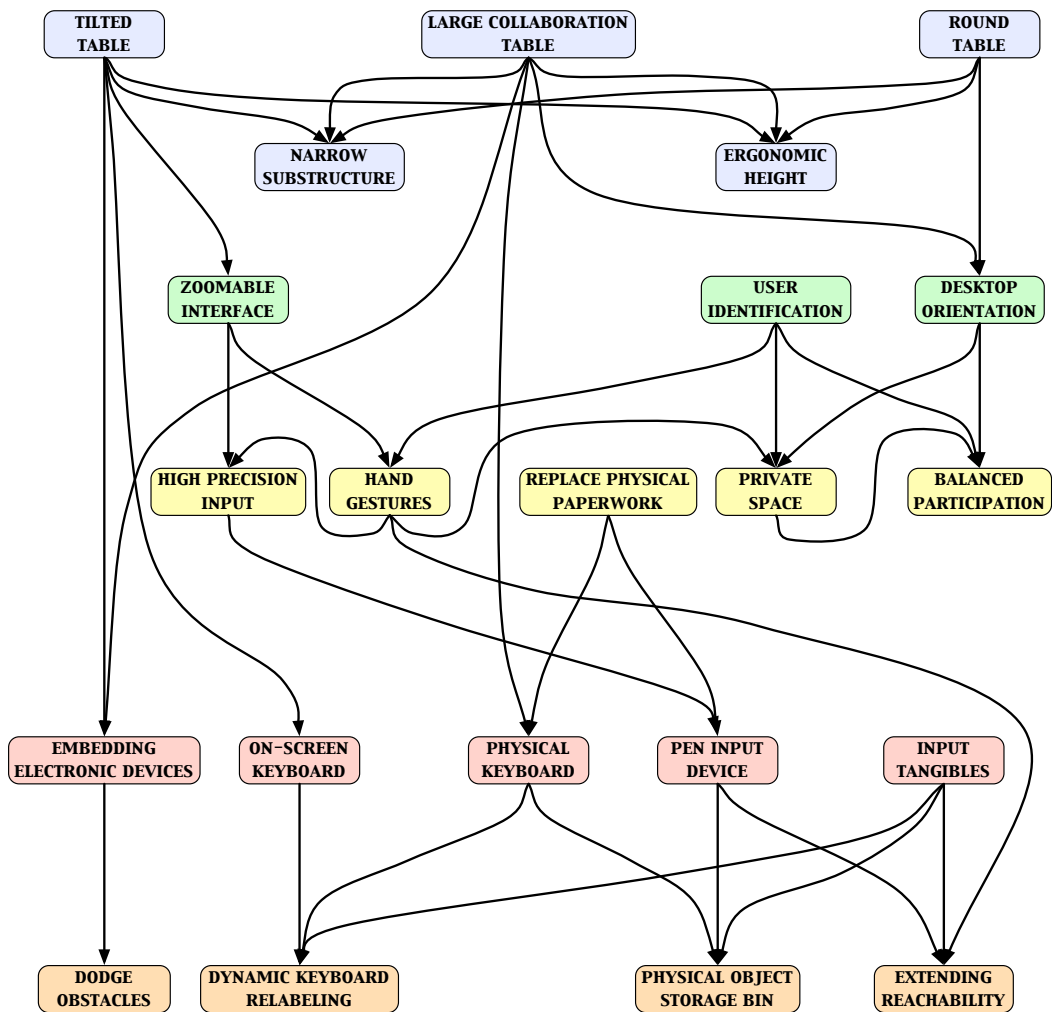


Figure 5.1: Tabletop design pattern language graph.

## (01) ROUND TABLE \*



Figure 5.2: The Reactable.

... you just started building the concept for your tabletop and are looking into basic form factor decisions. Your tabletop will not be used for single-user tasks like office work or text reviews, but rather serve as the center of a team meeting or as an interactive tabletop for an exhibit. All users at the table are supposed to interact on the same level, without privileging a specific user.



**In a collaborative environment, many participants work around a table simultaneously, but everybody wants to have the windows and the table aligned towards herself.**

On a rectangular tabletop there are often positions for collaborators where they feel uncomfortable, especially if there are more than four users. E.g., sometimes participants join the discussion and have to sit at a corner or at the edge of the table from where it is hard to reach the other side of the table. A circular tabletop overcomes this issue, since the table looks the same from every position. Every collaborator has the same amount of table space in their area of reach, and if the windows are aligned and distributed in a circular way, no one has a superior look on the desktop (see *DESKTOP ORIENTATION* ).

One of the best-known circular tabletops is the *Reactable* [H:Reactable2009]. Its main application is a music demonstration where users can place tangibles on the table and interact with them to emit unique sounds. The *Reactable* encourages participants to mix different sounds created with their tangible objects in hand, thus creating a concert of ambient music. Due to the round table, every collaborator is at the same level and no one is implicitly acting as a principal musician or orchestra leader.

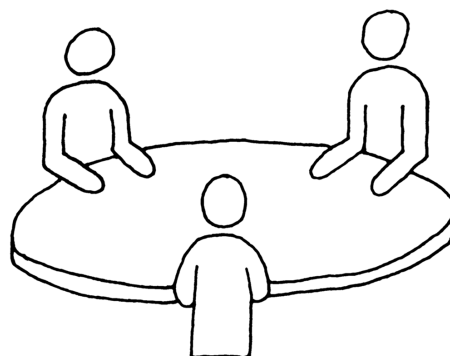


In the same area of producing music, the *SoundScape* renderer built by Bredies et al. [2008] is also a round tabletop, though the interaction is completely different. The *SoundScape* renderer is located in the center of a circular room, such that the display mimics the spatial setup of the environment. Users can then directly manipulate sound sources indicated by spots on the table, arrange them as they want and change their volume independently. A small set of HAND GESTURES allows for more input variety.

Another example for a round table is given by Koike et al. [2007], with the addition that it is also rotatable, which was the main focus of this research. The possibility to easily rotate the tabletop by putting wheels beneath the table is another advantage of a round table; in their work, a round rotary surface was mounted on a larger rectangular table, which was sufficient for their purpose of user tests. Since some of the advantages are not true anymore in this setup, e.g., there are still corners that may decrease the reach distance for some users, this is not a good example for a round table, unless you want to switch between round and rectangular environments for some reasons in your project. Another disadvantage that is clearly visible in this solution applies to most round tabletop setups: unless you are using a special projector, a non-rectangular image is simply created by not displaying the pixels in the corners of the screen. Hence, a round tabletop is either reduced in screen real-estate due to less pixels or it is more expensive, when using multiple projectors or a special device that displays a round image.

Therefore:

**Build a round table or a rectangular table with round corners.**



When building the hardware, choose an ERGONOMIC HEIGHT such that users will sit or stand comfortable at your table. Implement a DESKTOP ORIENTATION algorithm for your displayed widgets to ensure the effect of your table shape can fully unfold and benefit the users...

## (02) TILTED TABLE \*



Figure 5.3: A flexible, tiltable tabletop display.

... you are in the early stages of designing a tabletop that will only be used by a single user. Tasks do not include many large objects on a table, and the user will make extensive use of direct touch, e.g., for layout or design scenarios.



**Working for a longer period of time on a multi-touch wall can be exhausting. On the other hand, staring down on a horizontal surface can induce neck pain.**

There are many arguments for both horizontal and vertical displays. A board is usually used for presenting your drawings to an audience, like in a classroom or lecture hall. But also for a list of milestones, design guidelines, or project notes users put a board on their wall, even in a single-user environment. One reason why users do this, instead of putting the information on a sheet of paper on their desk, is that you cannot accidentally occlude them with books, cups, notes, or other obstacles on your table. It is also more comfortable to read information at eye level instead of looking down all the time. On the other hand, in some situations a vertical surface is not suitable for some tasks. Consider text input, which can be done in different ways. E.g., a pen is the usual device for whiteboards, and mimics the input metaphor of chalk on blackboards. This kind of input originally did not allow to put your hand on the surface, as it would smudge the chalk. It is also an uncomfortable position for your hand while writing on a wall, opposed to a sheet of paper on a horizontal surface. Another, more preferable text input is a keyboard, like an ON-SCREEN KEYBOARD or a PHYSICAL KEYBOARD. Both are almost unusable in a vertical alignment, especially the physical keyboard, which first would have to be mounted on the surface.

In a study that compared different angles of tabletops, Inkpen et al. [2005] explored that there is no clear preference of users for one of both settings. Another observation of knowledge workers from the tabletop domain by Morris et al. [2008] stated

informal guidelines about the setup of interactive surfaces, including the advice to support a display that can be tilted.

A study named Tilted Tabletops [Muller-Tomfelde et al., 2008] examined the effect on collaboration tasks for groups at tabletops with different angles. The experiment with 78 participants revealed that users in general preferred the tilted setup over horizontally or vertically aligned surfaces. Quite astonishing in this study seems the fact that higher tilt angles were preferred over a light tilt even for writing tasks. This maybe due to similarities with whiteboards where users put down short handwritten notes. However, it should be kept in mind that higher angles permit placing physical objects on the tabletop, which was not considered in this specific user test. An adjustable tilt angle solves this conflict; users that prefer higher angles can turn the tabletop if there are no objects placed upon that could fall off the table.

In single user setups [Wigdor et al., 2007] and remote collaboration tasks [Ishii and Kobayashi, 1992], tilted displays clearly outperform horizontal and vertical multi-touch displays. For collaborative displays, this can sometimes result in awkward arrangements for the users. This can be avoided by choosing a LARGE COLLABORATION TABLE, such that every user can look on the tilted table from a comfortable angle. Additionally, the tilting support should be optional and not be mounted with a fixed angle. This offers users also an alternation of the setup to avoid neck pain from looking down on a horizontal surface or arm fatigue on a multi-touch wall, when the users have to use the display for a lot of text input.

Therefore:

**Use a tilted table in an angle most comfortable to the user. If possible, make the angle flexible to account for better collaboration or placement of physical objects.**



Since physical objects would probably fall down from the tilted table, proper use of an ON-SCREEN KEYBOARD should be considered. Your tabletop will probably be used by a single user, therefore you might offer personalization of screen data by EMBEDDING ELECTRONIC DEVICES. Other important aspects are a NARROW SUBSTRUCTURE and an ERGONOMIC HEIGHT...

### (03) LARGE COLLABORATION TABLE \*



Figure 5.4: Struktable 70 inch multi-touch table.

... your tabletop will be used in a collaborative workspace where a large number of users interacts with the surface simultaneously. You have yet to decide which hardware you will use, but you already know the estimated amount of users and possibly also the projected tasks.



**Every user wants sufficient space for comfortable interaction, but you do not want to arise interaction issues like objects being out of reach. The tabletop should also work as a ubiquitous furniture and not as an object spatially dividing the participants and invoking a virtual distance.**

If a tabletop is too small, it raises many problems like insufficient screen space, restricting wide-ranging gestures, and visibility issues as the number of collaborators increases. E.g., in a meeting, everyone needs enough space to take down notes, or on an exhibit tabletop a large number of users should be able to explore the media together. On the other hand, if the table is sized too big, the practitioners cannot reach the entire area and you are facing reachability issues. You can always solve these problems with other techniques, such as *HAND GESTURES*, but you will lose your ubiquitous feeling that the tabletop just supports the discussion. It will more become the center of the meeting, and technical details should not become the topic of a meeting in general, unless it was the original intention. Large tables can also create a distance feeling among users, such that they feel spatially divided by it and they behave more passive during the discussion.

Multi-touch tables can have many different sizes, from devices sized as small as a tablet to huge walls, as large as cinema screens. In most cases, your choice is

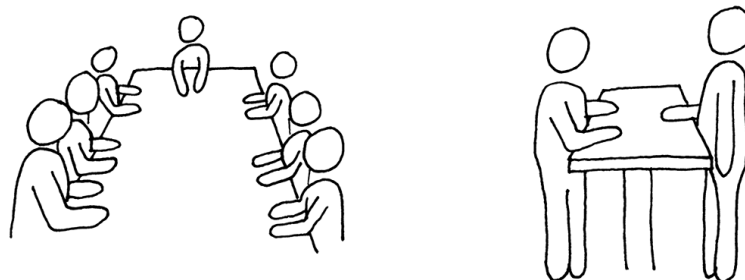
obviously not easy to undo later on, so it should be well-justified. Elaborate the projected number of users, prototype basic tasks with the minimum and maximum number of practitioners to get the best size for your table. For the same number of people, different tasks may require different table sizes; e.g., working with four people on one list of items requires a smaller table compared to four people working on four separate graphical tasks. Sometimes, combining multiple tables can also suffice the need for space [Ryall et al., 2004].

Inkpen et al. [2005] conducted an extensive study about how the tabletop setup affects the collaboration, with display size one of four factors that they evaluated. They conclude that a display should be small enough to fit into the user's view, and large enough to give each user sufficient space to interact with one another. In comparison to the larger display this setup resulted in an even more distribution of touched areas.

A design space of tabletop hardware, evaluated by Grossman and Wigdor [2007], also stresses the importance of the display size selected for the tabletop. In a similar taxonomy, focusing rather on the collaboration aspect on multi-user display devices than the hardware, Terrenghi et al. [2009] emphasize the impact on the display's size on collaboration. While not limited to interactive tabletops, they list different sizes of devices and the implications for the interaction between users and the device that should be considered in designing such systems.

Therefore:

**Build a tabletop that offers enough space for all users and their expected tasks. The size should be equal to a traditional non-interactive table where users would work in a similar scenario.**



The larger the table, the more functionality is needed for a reasonable display and tracking quality, but you should still consider a NARROW SUBSTRUCTURE and an ERGONOMIC HEIGHT. You also can take advantage of your available space by putting additional input objects on the table, e.g., EMBEDDING ELECTRONIC DEVICES like cell phones or laptops, or a PHYSICAL KEYBOARD, when you expect a lot of text input...

## (04) ERGONOMIC HEIGHT \*



Figure 5.5: A table with a suitable height for the user.

... you have started to design your tabletop and you are in the very beginning of choosing the hardware specifications. Depending on the target audience and tasks, your system has a different setup, e.g., a *LARGE COLLABORATION TABLE* or a *ROUND TABLE* for a multi-user scenario, or a *TILTED TABLE* for a single-user environment. You also know whether the users are sitting or standing at the table, and whether you are designing for children or adults.



**The tabletop should have a surface with a lot of functionality that depends on a lot of technical requirements. With all the technology underneath the table, it may be too high for users to interact comfortably.**

While designing your table, you might have several restrictions to the ergonomic design. If you have a rather large table with projector and cameras beneath it, you need a reasonable size for the projected image to fit the table measurements. On the contrary, if your table is too small, then you do not have a lot of space next to all the elements in the substructure, and while keeping a *NARROW SUBSTRUCTURE*, you may tend to stack things up as much as possible. In both situations the height of your final tabletop has to be higher. But this possibly raises another issue depending on your scenario, e.g., it might be impossible for children sitting at your table to interact properly. On the contrary, if the tabletop is used in an exhibit with users walking around, it needs to be much higher than just the height dedicated by your technical limitations. Before building the basic hardware, you have to consider the application of your tabletop, e.g., by observing users in a scenario on a prototype table.

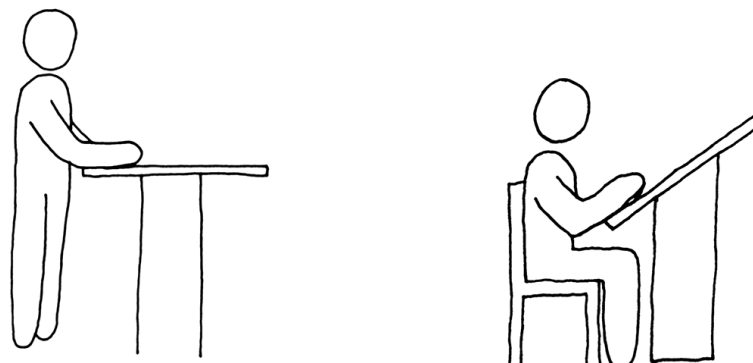
One of the drawbacks of direct touch input on tabletops is the fatigue of the users' arms. However, the intensity of exhaustion that users feel when using touch in-

teraction is connected to the ergonomics of the table. Compare this to a desktop PC, where most users usually sit on a chair at a table with the keyboard, mouse, and display. To make this situation more comfortable, users arrange these devices such that their arms are placed in a natural way and it is not exhausting to work on a PC. Additionally, the chair is often height adjustable and tables have a uniform standard height. Of course you can choose this height for your tabletop, too. ISO 9241-5 [1998] recommends 72 cm for ergonomic work space furniture, but keep in mind that this is for a scenario where the user sits at your table. The recommended height for standing at a work place with a monitor is 118 cm, however this is probably too high for touch interaction and needs further investigation.

In a long-term study, where a tabletop was designed for one particular user who did all his digital work on the multi-touch table, Wigdor et al. [2007] built a table that let the user feel comfortable in both sitting and standing position. A good way to avoid that your construction will be too high or too low is a preliminary user test with a prototype, which can easily be done by taking an office table and chair, adjust its height and simulate some touchscreen tasks with pieces of paper that have to be sorted. Be sure that you check for both sitting and standing scenario, in case of sitting on a chair the chair should be the same that is used later on (e.g., height adjustable) and your table should have the same NARROW SUBSTRUCTURE as your finished tabletop has to allow for sufficient legroom.

Therefore:

**Consider the target user group and the working environment for your tabletop carefully and choose a suitable table height. If the users are children or they are intended to sit at the table, choose a small table height. To get the best results, build early prototypes and emulate interaction tasks.**



This is a basic pattern with no further references within this language.

## (05) NARROW SUBSTRUCTURE \*\*



**Figure 5.6:** An exhibit tabletop (*Aachener Frieden*) with a tapered substructure to avoid visitors hitting the structure with their knees.

... you are designing a system for a multi-user environment, either a LARGE COLLABORATION TABLE or a ROUND TABLE that allows to sit or stand comfortably. For a single-user environment, you might have chosen a TILTED TABLE.



**Tables with top-projection have the occlusion problem, while a tabletop with a projector beneath the table does not leave much space for the legs of the user.**

Multi-touch detection can be realized in many different ways. Both the detection of input and the display output might need devices that have to be placed in a reasonable distance on top or beneath the table. While top projection raises occlusion problems, which are especially distracting in multi-user environments, projectors and cameras beneath the table take a lot of space. Whenever users sit down at a table, there should be enough space for their legs. But also while standing at table-tops, this is an important issue, as users should not hit the substructure with their knees.

You can solve these problems without big changes in your hardware design which you created by using a mirror, as proposed by Masoodian et al. [2007] in their paper for an environment to share documents on a tabletop. Projectors with a high projection angle, e.g., short-throw projectors, further improve the legroom since they enable even higher angles. Also consider user tests with prototypes to find out which substructure construction allows which range for the user's arms on the table. These prototypes can be really low-cost, made of wood or cheap metal, where you let the user grasp objects on the table and see if they touch the substructure with their knees. As long as users can reach the whole surface without hitting the substructure with their knees, the tabletop is small enough.

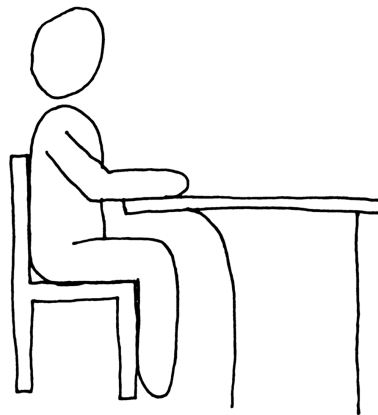


---

Some basic guidelines for legroom can be found in several standards published by authorities. E.g., according to the ISO 9241-5 [1998], the depth should be 60 cm to 80 cm in 12 cm height, measured from the floor, and 20 cm in 67 cm height, respectively. The latter requirement can be acceptable at 20 cm in 62 cm height, if the structure does not meet the first requirements. However, always keep in mind the scenario you will use the table in; e.g., children need smaller tables than adults, changing your blueprint considerably. Additionally, in a multi-user environment legroom might not be an issue as big as in a single-user task, since on every table users try to avoid hitting the legs of other users, thus keeping a larger distance than in a scenario where they are working at a table alone.

Therefore:

**Make preliminary user tests with your hardware to check if the users have enough space for their legs and can sit comfortably at your tabletop. At all edges where you expect users to sit or stand, use hardware that allows for a narrow substructure, such as mirrors or special cameras and projectors.**



This is a basic pattern with no further references within this language.

**(10) DESKTOP ORIENTATION \*\***

**Figure 5.7:** Windows on a rotary table aligned to the border.

... multiple users will interact with your LARGE COLLABORATION TABLE simultaneously, which might be a ROUND TABLE that does not favor anybody. As you are developing the software framework, you still keep equilibrium of participation among the several users in mind, knowing that it is not possible for all users to look at the screen from the same direction.



**Multiple users interact with multiple desktop widgets from different points of view. Viewing content, especially reading text upside down or even sideways can be hard and makes the user uncomfortable while using the tabletop.**

On vertical displays such as traditional desktop computers, there is always a clear direction for all widgets, immediately visible for text being displayed. In a collaborative environment, participants are usually standing or sitting around multi-touch tables, hence everyone is looking on the table from a different angle. E.g., imagine a task where four users sort small note widgets with short text passages on it. It is very hard to read upside down or even sideways, thus the user with the most text elements oriented towards her is in advantage. However, the system often does not know which user is looking at which elements on the surface. In addition, aligning everything to one side of the screen as on a traditional PC gives one user an unfair advantage over the others and does neglect the possibilities of technology in today's systems, where dynamic and independent orientation can easily be implemented.

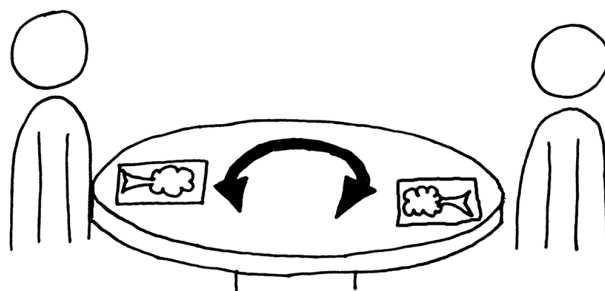
The *DiamondSpin* toolkit [Shen et al., 2004] enhances a *DiamondTouch* table with a circular workspace projection. It offers an easy way to orientate and align desktop elements according to the viewpoint of the users, thus improving the interaction experience of all collaborators. It is also possible to rotate the whole table, e.g., to switch tasks between two users, or orient all windows towards one single user, a technique they call “magnetizing”. However, in their implementation there was no independent orientation of widgets, i.e., it is not possible to orient an element towards a side of the table that is farther away than an other element.

Another similar approach was done by Koike et al. [2007], focusing on browsing large amounts of data, in their specific example huge picture collections. The pictures are being displayed in a circle, thus aligning all elements automatically to the nearest edge.

Both examples share the same disadvantage: Automatic orientation is not always what users may want. Kruger et al. [2004] second this in their extensive user study that investigates the orientation issue. Results clearly suggest that viewpoint correction has to be resolved dependent on the tasks and user’s demands. A recent technique which allows manual orientation regardless of the user’s position at the table was proposed by Dragicevic and Shi [2009]. With the help of HAND GESTURES, users can create their own workspace, automatically including and aligning a set of specified documents selected beforehand.

Therefore:

**Enable free rotation of interface elements on a basic system level in your tabletop setup. Support automatic orientation if you have knowledge of the user’s position at the table, but also include manual orientation functionality.**



To achieve equity of the users’ amount of participation, ensure **BALANCED PARTICIPATION** in a meeting or on an exhibit tabletop. Also think about implementing **PRIVATE SPACE** areas to let users protect their documents, e.g., private notes or other separated work...

## (11) USER IDENTIFICATION \*



**Figure 5.8:** User identification with a fingerprint scanner.

... your tabletop will be used in a setup where multiple users work simultaneously at a tabletop. Either on different tasks at the same time or together on a task where every user produces a semantically distinct input. E.g., in a game where users own certain virtual objects, or in a meeting where they put down personal notes.



**In a collaborative environment, multiple users work at the tabletop together. But often it is critical for the system to know which user produced which input.**

When multiple users are working simultaneously at a tabletop, using only their bare hands for direct touch input, it is not easily possible to tell which input belongs to which user. Electronic devices, such as mice, keyboards, and pens can have unique identifiers to be separated from each other. Unless your system uses an incredible high input resolution to detect finger prints - which is unlikely to be realized on a large tabletop surface with the current state of technology - there is no simple way to assign each finger to a person. You could put a small fingerprint scanner in one of the corners of your table, but then you would need overhead cameras to keep track of the fingers after they are lifted from the table. A similar result could also be achieved by identifying users via an electronic device they are carrying, e.g., their cell phone or a dongle.

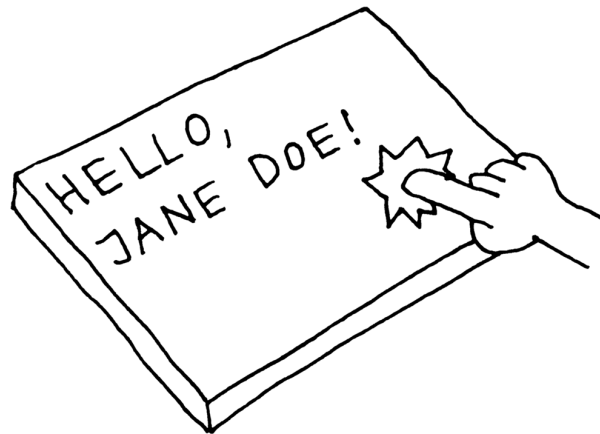
The *DiamondTouch* [Dietz and Leigh, 2001] is a capacitive tabletop, where an array of antennas is used to transmit small electrical charges from a transmitter beneath the table through the user's finger, further to their body, and to the chair and finally to the receiver. This enables user identification since the system can tell from which chair the input was received. However, obviously the imagination of sitting on an electric chair can produce a very uncomfortable feeling for the users. It is also a difficult and time-consuming procedure to connect all the wires, and the connection is lost as soon as users get up or even change their seats.

Mohamed et al. [2006] use gestures from pen input to identify users. Their recognition algorithm is able to tell on which side of the table the user is located, only by pressure and angle information and without the need for overhead camera detection. The downside of this method is obviously that you need a pen, which eliminates the advantage of direct touch. Consider a tabletop game, where user identification is crucial, but you do not necessarily have enough pens and direct touch input is possibly preferred, e.g., in a card game.

Recent research came up with an idea that gets close to the initial fingerprint idea, but does not rely on high resolution: Schmidt and Gellersen [2009] implemented a hand shape detection mechanism. With high robustness and an error rate of only 0.5% this approach is a realistic alternative that does not depend on techniques as capacitive touch or pen input like the aforementioned solutions.

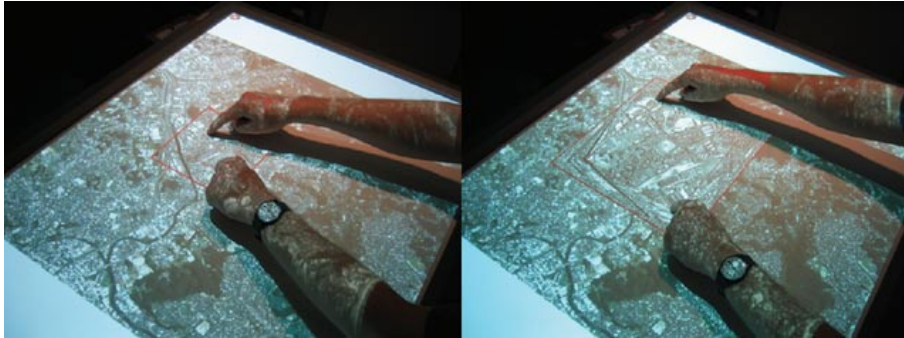
Therefore:

**Let the table distinct the input from different users. Identify the user either by hardware or via software mechanisms and assign the interface elements to the user that he or she is working with.**



With identification of users, PRIVATE SPACE and BALANCED PARTICIPATION can be supported more easily. Also, personalized HAND GESTURES offer a distinct system response for different users with various additional options...

## (12) ZOOMABLE INTERFACE \*



**Figure 5.9:** A user performing actions with the *DTLens* toolkit on a tabletop, zooming into a map.

... although your tabletop has a suitable size, you sometimes may encounter space issues. E.g., when a large number of participants is working simultaneously on creating a large mind map out of brainstorming results, or on a TILTED TABLE in a single-user environment.



**A tabletop should be as large as possible such that not every interaction is limited by the physical edges of the table. But sometimes the desktop elements being displayed on the tabletop are so space-demanding that it is impossible to show them all at once.**

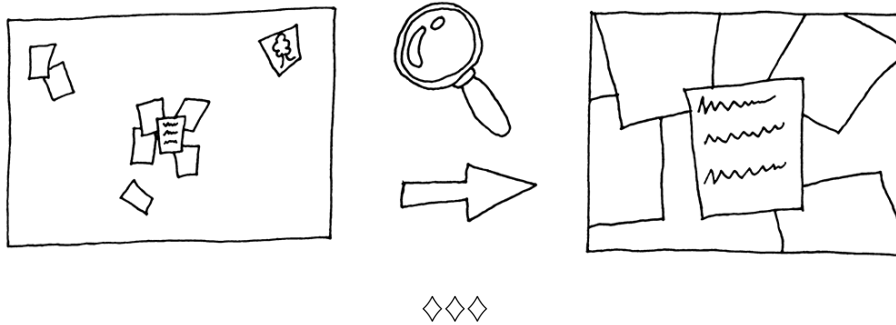
The idea of zoomable interfaces has often been mentioned in research, with one early concrete example proposed by Perlin and Fox [1993] in their *Pad* interface. They strive to make interaction more natural by using geographic analogies for navigation. The interface is one flat two-dimensional space, where the user can zoom in and out to see more details or get a broader overview, respectively. Raskin built up on this approach and described *ZoomWorld* [Raskin, 2000], imagining a novel interface without file names and other descriptors, just content where the user can browse intuitively using her real-world experience on navigating through large amounts of data.

With multi-touch tables and their natural input mechanism, it is possible to create interfaces that mimic the behaviour of *Pad* and *ZoomWorld* while ensuring a maximum amount of usability. The *DTLens* toolkit by Forlines and Shen [2005] allows users to zoom on large data files, e.g., geological maps or pictures of space with high resolution. The two-handed zoom gesture is an example of the advantages that tabletops offer over traditional desktop computers for zoomable interfaces to realize these ideas that have been around for a long time.

The most sophisticated example of a zoomable interface can be experienced on a multi-touch device that is much smaller than a tabletop: Apple's iPhone [H:Apple2007]. The built-in web browser allows to visit websites that are designed to be viewed on large desktop displays. The user can zoom out to see the website as a whole, and with zooming into the content it is possible to click on tiny links or enter text in small text input areas. When building your tabletop think about adopting these ideas in order to have sufficient space at the end.

Therefore:

**Let the user not only zoom into the content of your applications, but also into the interface elements itself.**



To ease navigating through your tabletop environment and its widgets, HAND GESTURES are commonly used and especially applicable to your zoomable interface. Also consider implementing HIGH PRECISION INPUT methods in your tabletop software framework...

**(20) HAND GESTURES \*\***

**Figure 5.10:** A *flicking* gesture that maps real-world interaction to virtual objects.

... the tabletop software framework is being developed, with possibly USER IDENTIFICATION and a ZOOMABLE INTERFACE. You are considering the users' tasks and realize that there is a variety of different commands beyond your limited number of buttons or similar interface elements to issue these commands.



**There can only be a certain number of buttons on the screen without overcrowding the desktop and render it and therefore make it unusable. But as the applications grow in functionality, so does the set of input commands.**

While there are hardly any gestures commonly known on desktop computers, multi-touch devices with direct input afford the development and use of gestures. The most well-known gesture is probably the two-finger zoom gesture, which was not invented, but introduced to a broad public by the Apple iPhone [H:Apple2007]. This zoom gesture is often used on tabletops as well, but since tables are larger devices, users tend to use two hands instead of just two fingers from one hand. This example shows that gestures are extremely useful, they can extend tabletop interaction in innovative ways, and they are easy to learn due to their natural behavior. But there are also some drawbacks when introducing new gestures. Without visual hints, most users will not use them, unless they were told to do so. And especially the zoom gesture shows that they are ambiguous in the way they are done: One user might use two fingers of one hand, another user may use one finger of each hand, and some users may even use more than two fingers. It is up to the designer to decide whether these different gestures result in different or in the same procedures; this should be a decision made with careful consideration.

The former mentioned zoom gestures were also used in a paper by Tse et al. [2006], but extended by speech input and even other gestures, application-specific. Two different user scenarios were created, two users collaborating on a geographic map (*Google Maps*) and playing a real-time strategy video game (*Warcraft III*). E.g., when



browsing the map one finger is used to move the map, the complete hand issues a 3D tilt up, whereas a five finger touch results in a 3D tilt down. In the other task, the strategy game, there are actions often necessary like selecting a number of units. While putting two sides of the hand down on the table, the space in between is marked and the units in the area are selected.

Wu et al. [2006] elaborate on the development of new gestures with design guidelines for the process of defining gestures. They say it is important to make sure the gesture registration is distinct from any other input done, the recognition should be variable and not too selective to allow errors in reproducing the gesture, and gestures should be short, easy, and partly reused, to ease the learning process for the users.

To get an overview of useful gestures that have been invented so far, take a look at the work by Wobbrock et al. [2009], who created a taxonomy for gestures. They also conducted a user study with user-defined gestures, where in all 1080 gestures were evaluated for 27 different commands, leading to interesting results. When choosing gestures for your application, consider these important aspects: gestures should be as intuitive as possible, or even better if gestures have been used before in other applications for issuing the same command. If the gesture is a newly invented one, provide small hints without being annoying; however, if too many hints are necessary, the gesture is clearly unintuitive and should be replaced by another. Most important, gestures should be consistent among different systems. By adopting established gestures for your tasks, you can reduce the learning effort, enhance usability, and contribute to build up new standards.

Therefore:

**Extend the number of input commands by implementing a technique to detect gestures. The gestures should be robust, unambiguous, and easy to learn, or even better, already be commonly used, like the two-finger zoom. Do not overuse gestures though, and offer novice users hints for unfamiliar gestures, e.g., when using the application for the first time.**



Gestures can be used for EXTENDING REACHABILITY or invoking PRIVATE SPACE. There are also examples of gestures to offer HIGH PRECISION INPUT...

## (21) PRIVATE SPACE \*



**Figure 5.11:** Students at a tabletop reaching into the private space of each other.

... you have multiple users working on their own private data, e.g., personal documents or in a game. With techniques as *DESKTOP ORIENTATION* or *USER IDENTIFICATION* the users' data is being personalized, and *HAND GESTURES* extend the input variety on your tabletop, but you are still facing privacy issues while sharing data among users.



**Users want to use the whole table, but sometimes privacy becomes an issue when users want to interact with displayed elements without sharing them with other users.**

In collaborative workspaces users have personalized data, e.g., their folder of documents, cards in a game, or piles of photos. When multiple users bring their data to the same tabletop where they are working collaboratively, they do this for a reason. They want to share documents, work together on a task or just play a game. But while all this data is available to all the users, it raises the problem of undesired access in specific situations. The situation also avoids undesired access, e.g., that someone steals a document from someone else. Consider a scenario where two users are sorting pictures, one user applies a zoom gesture to enlarge a picture to look at some details, and accidentally the whole screen is covered by that single picture, hindering the others from interacting with the tabletop.

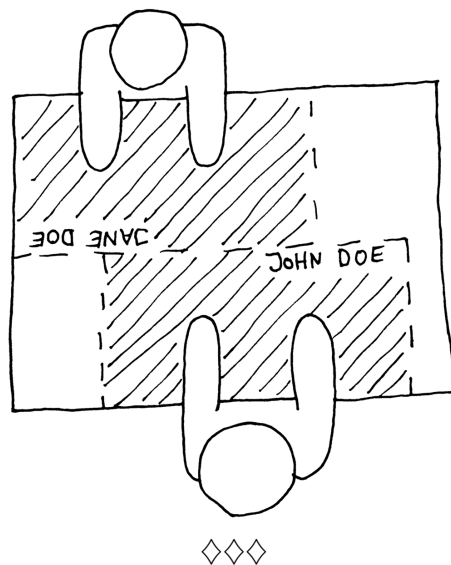
Tse et al. [2004] explored the interaction interferences of co-located collaborators in everyday's work tasks. They proposed a set of informal design guidelines for application designers to account for territoriality in multi-user interaction. If the tasks are semantically separated, they should be spatially separated from each other as

well. Pop-up windows and other widgets should not be centered on the screen like on usual desktop systems, but stay inside the input area of the user who is working on that particular task. They also advice to create a private space as well as a group space to distinguish between the different modes of tasks.

A study of territoriality based on a trading-card game experience by Pinelle et al. [2009] revealed that automatic protection mechanisms are preferred by the users. They implemented a user control level that changes accordingly to the distance of the participants: If the user's input focus is farther away from their home area, the control level is lower and other users can steal items from the users private space as long as their control level is higher. If a user is working in her private space, the control level is at maximum so that no other user can steal or even distract the user during her work. In this particular application, the automated mechanism was preferred by the users and also showed better results in the quantitative study than the user-controlled mechanism. However, depending on the specific task of your application, a combined privacy implementation of both automatic and user-controlled may be more suitable for your needs.

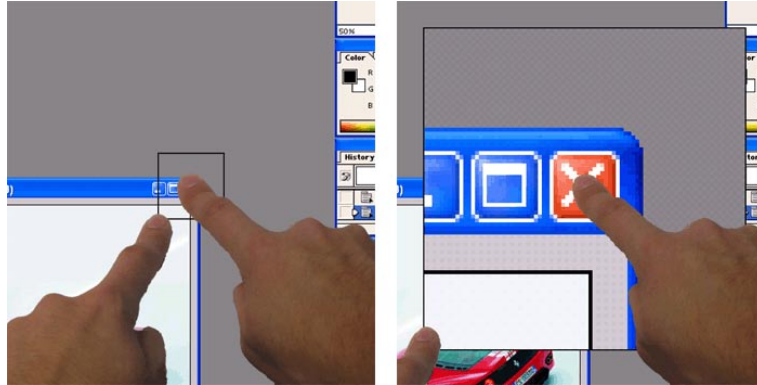
Therefore:

**Offer an area of private space that allows users to protect their data against undesired access. This can be implicitly or explicitly implemented, by having a lock/unlock button or a designated area that does not allow interaction from other users than the documents' owner.**



With users having control over data, make sure that there is a **BALANCED PARTICIPATION** among the collaborators and no one can block others from interacting with the tabletop, controlling the discussion...

## (22) HIGH PRECISION INPUT \*



**Figure 5.12:** *Dual finger stretch* technique for high precision input.

...your tabletop is dealing with large amounts of data, and you may have implemented HAND GESTURES or a ZOOMABLE INTERFACE already to cope with the problems of navigating through the content. Although your interface is designed properly for multi-touch purposes, sometimes users may encounter difficulties when trying to hit buttons or content.



**The touch interaction needs to be precise and avoid mistakes, but it is hard to get the exact pixel that was hit by a certain finger.**

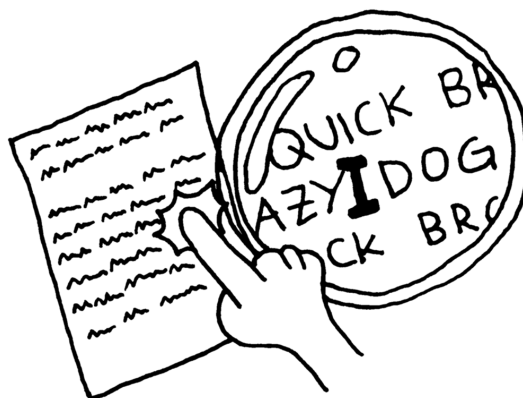
Direct touch and its natural behavior is one of the biggest advantages of multi-touch tabletops. But although the resolution is high and the tracking mechanism is quite exact, it is hard to tell which exact point the user hit, since the shape of a finger covers a larger area of the surface than just one pixel. In tabletop research, this is often referred to as the “fat finger problem”. Although to a different extent, it applies to almost every task on a tabletop. In a scenario where you want to point out a location or a route on a geographical map, you want to hit certain locations like roads, towns, or sights as exactly as possible. You can always zoom in and out, but sometimes this is not possible in your task, e.g., when you want to draw a line alongside a long road on a map, you probably do not want to zoom in, because you might not see your target anymore on the surface. A similar problem applies to text input, where you want to hit the small space between two letters, and by zooming in you will lose the context of the surrounding text when it gets off the screen. The Apple iPhone [H:Apple2007] overcomes this issue by displaying a small magnifying glass above the finger, such that the user is able to tell exactly which letter he just hit. A more general example is depicted in the sample illustration above, where a user encountered problems to hit the desktop interface elements that were build for a traditional PC and are now too small to hit on an interactive tabletop.

Benko et al. [2006] elaborate on this situation and offer five different techniques to overcome this problem. User studies shows that the best of their solutions is a dual finger stretch technique, where one finger is used as the selector and a second finger is placed nearby the selector, moving away and thereby zooming in to the area, making it easier to aim for the first finger. Four other techniques were presented: two with a cursor speed control (one with a zoom-like gesture, one with a pie-menu) and two solutions which did not perform very well in the user tests, being the dual finger midpoint technique (where the midpoint of two fingers is the cursor) and the offset technique (where the cursor has a slight offset to the real touch input point).

Another more recent paper by Olwal et al. [2008] introduces two different HAND GESTURES to solve the precision problem, rubbing and tapping, which zoom into the interface. They combined these gestures in several distinct ways, e.g., one finger to zoom in with the rubbing gesture and the other finger doing the selection. In conclusion the tapping gesture as zoom-in with the finger release point being the selection performed well, offering a very short and easy method to increase precision with only one finger. Nevertheless, the user tests also suggest that an additional click to confirm the selection is more robust without sacrificing too much time.

Therefore:

**Offer input methods to point to exact coordinates on the tabletop. Use gestures, tangibles, or implemented algorithms for interaction to allow the user to precisely select specific points on the surface.**



High precision input extends the variety of applications on your tabletop. One other way to achieve this, is a PEN INPUT DEVICE...

## (23) REPLACE PHYSICAL PAPERWORK \*



Figure 5.13: The Apple iPad strives to replace the traditional newspaper.

... you are building a tabletop that primarily be used in office scenarios like team meetings or as an extension for everyday's paperwork tasks. Its focus is mainly actively reading text, which involves editing or taking notes.



**Paper offers many advantages, such as comfortable reading, sharing, and annotating. However, books and paper sheets occlude the tabletop surface, which limits the available interaction area. Additionally, for tasks such as laying out text, collaboratively working on the same paragraph, or browsing through indexed pages, digital files are superior to physical paper.**

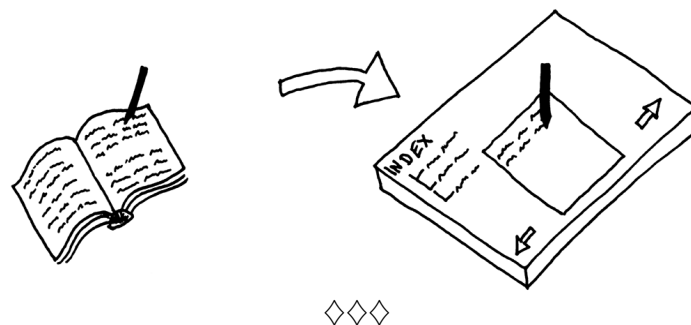
There are many arguments for keeping physical paperwork, but ebooks and other digital texts are becoming more and more popular. We are receiving a lot of information in a digital form, e.g., scientific papers, social networks, or online news services. However, the current implementations to read text on desktop computers are still not user-friendly enough to convince users to throw away their books. On the other hand, on tabletops you do not have a large area for books and sheets as on your desk for your desktop computer. The goal should be not just to replace paper, but to make use of the advantages of tabletops over paper and desktop computers [Terrenghi et al., 2007]. E.g., when searching a particular text passage, you flip the pages of a book or look into the index. On a desktop computer, you either look the word up in the index, too, or use the search function. On a tabletop, you can use intuitive gestures to mimic the flipping pages functionality, as well as have the search and index functions from desktop computers. Especially making annotations and highlighting text is more natural with direct input than relative input like the mouse. You can even add a pen to your tabletop, to make it work the exact same way like annotating a physical book. The advantage then again is that you can search later on for all the highlighted text passages.

With the Apple iPad [H:Apple2010], the press and many publishers hope to bridge the gap between physical paper and digital text. Due to its low weight and size, it mimics the look and feel of a book, while offering many of the advantages of today's technology. Unlike the single-user centered desktop computers, tabletops support multiple users that can work simultaneously on multiple files. Additionally, sharing files is even easier than on the iPad, which you have to hand over to someone like a book. On a tabletop, you can just flick the document to the other user with a touch, and edit it collaboratively on the huge work space, as Masoodian et al. [2007] proposed: They modified a desktop text processing application to allow for collaborative editing, where multiple users could simultaneously work on the same document using personalized HAND GESTURES. These ideas support the thoughts of Sellen and Harper [2003], who point out that it is important to change the work practice before users replace paper with digital reading material.

An example for the use of digital paper is presented in a study by Piper and Hollan [2009], where students received sheets of paper for a task in both a digital and a physical form. The results indicated that the digital material encouraged students more to try out since "undo" was offered. Especially a graph drawing task is hard to undo on paper, whereas the digital drawing allows for even more options than just redraw, e.g., change the path by a certain amount into the direction of an axis. The big advantage here is that users have unlimited trials; when writing a text, you can write words, delete, write again, and so on. When doing this on paper, you waste many sheets and probably decide to keep some thoughts in your head rather than writing them down.

Therefore:

**Implement paperwork applications that support tasks by mapping real-world interactions to digital text, such as flipping pages, marking pages, and visible feedback of the available pages left over. Make use of the advantages of digital text, e.g., collaborative editing, quick sharing, undo, and dynamic layouts.**



If you expect a lot of text input and your tabletop offers a lot of space, put a PHYSICAL KEYBOARD on the table. With a PEN INPUT DEVICE, you can allow users to make handwritten annotations...

## (24) BALANCED PARTICIPATION \*



**Figure 5.14:** The *SIDES* game is a turn-based game for children on a tabletop. After each turn, the user presses a “proceed” button to give the turn to the next player.

... you designed a tabletop for collaborative tasks such as a game or an exhibit, where users are meant to interact not only with the tabletop, but also with other participants. The input can be distinguished by USER IDENTIFICATION and users have their own PRIVATE SPACE, with elements aligned towards them using DESKTOP ORIENTATION.



**In a collaborative environment, multiple users should contribute to the interaction. Occasionally, some users take the lead and hinder other, more calm users take part in the discussion, thus possibly creating unbalanced results.**

There are many different situations for multiple users in a collaborative workspace: working simultaneously or in distinct turns, interacting on the same element or completely different desktop widgets, contributing to the same task or doing another project on the same table. In most situations, each collaborator should get roughly the same amount of interaction time at the tabletop. Imagine a brainstorming session or a discussion about new design ideas. Usually, some people tend to take the lead in these situations, no matter whether unintentionally or on purpose. Other users stay in the background, they speak up only when being asked or if there is a break in a conversation, which does not happen very often, e.g., in brainstorming sessions.

An insightful experiment was conducted in Helsinki, where a large multi-touch wall was installed in a central city location [Peltonen et al., 2008]. Several obser-

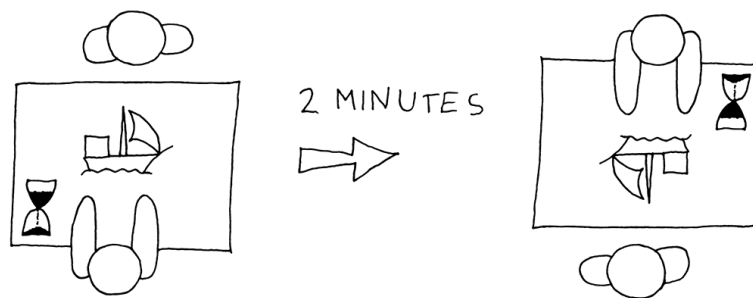


variations were made how distinct multi-user interaction can be, and we can outline how the setup of the multi-touch display can influence the turn taking. E.g., although the surface was large enough that several users could interact simultaneously, users lined up at some point and started interacting one after each other. In another situation, one user used the zoom gesture to enlarge a photo to the whole screen. Another user, who was exploring the wall at the same time, was distracted by this photo on his side of the wall, immediately stopped the interaction, and left the scene. Both examples outline how important it is to make use of the whole table as a space to invite users. In the first example, some invitation screen could have encouraged users to start interacting simultaneously, while in the second example the zoom gesture should not be able to overlap other interaction areas that are active at this time.

Marshall et al. [2008] evaluated how mouse versus finger input and single-touch versus multi-touch influenced the equity of participation. Their findings indicate that touch interfaces and especially multi-touch interfaces lead to more equity in interaction. However, the verbal participation showed no significant change, although the subjective perception of the users contradicted to these points, as they perceived more equity. A possible explanation might be that even though the amount of verbal participation does not increase quite much, the contribution of more silent collaborators can be huge. Nevertheless, the study also suggests that a larger table might lead to different results, hence make sure to choose a suitable table size for your tabletop setup.

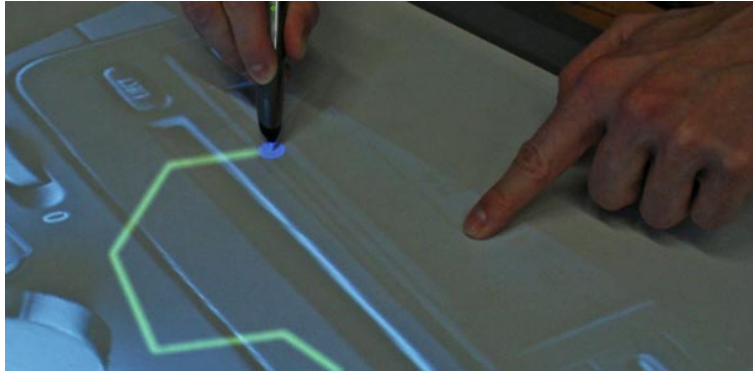
Therefore:

**Set up the table such that every collaborator can contribute in the same way to the task. Your software should not allow anyone to occupy the table and hinder interaction by the others. If applicable to your task, implement timers that show the tabletop interaction time, e.g., in a design discussion to balance the contribution over all participants.**



This is a basic pattern with no further references within this language.

### (30) PEN INPUT DEVICE \*\*



**Figure 5.15:** A user combines pen and finger input while working on a tabletop.

... you have built a tabletop for educational purposes or office tasks, e.g., for presentations or to design discussions in a team, or to REPLACE PHYSICAL PAPERWORK with the use of HIGH PRECISION INPUT. Now you think about further input devices you can add to your setup for these specific tasks.



**Tabletops offer natural input with direct touch, but for some tasks more precision or multi-functional input devices with different modes and states would be useful.**

A digital pen combines two advantages of tabletop interaction: while it mimics the natural input of drawing or writing with a pen on a piece of paper, it also carries the option to embed micro-technology for further interaction possibilities. A well-known pen for tabletops is the ANOTO [H:Anoto2000] pen. A small camera inside the pen identifies the position on the surface, which is covered by a so called *ANOTO pattern*, a paper with small printed dots on it that store unique location information. The pen's camera tracks these positions while the user is writing and sends the position of information via Bluetooth to the tabletop system.

The uPen by Bi et al. [2006] aims at combining a laser pointer with a pen. A camera detects the laser spot to get the current pointing position, thus the pen can be used from any distance. It supports simultaneous multi-user input, USER IDENTIFICATION, and mouse emulation by two buttons mounted on it, mapped to right-click and left-click. *Ortholumen* [Piazza and Fjeld, 2007] uses a similar technique, although it uses light from LEDs instead of a laser and a polarization filter on the tracking camera for higher sensitivity and precision. The camera, in their implementation located beneath the table, detects not only the position pointed on the surface, but also its shape and size. From that information it calculates the angle

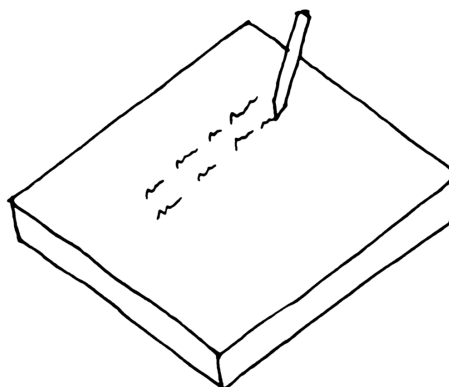
and height in which the device is pointing. And as a result, adding more degrees of freedom to the input.

Pen input can be used in many other ways to extend the input. Mohamed et al. [2006] combined pen input with gestures to detect on which edge of the table a user is located. Their algorithm needs no camera, it can track the user just by pressure and angle of the pen, with vanishingly little error rate in their user tests.

Pen input does not only replace the input, additionally it extends touch input. In an extensive user study, Brandl et al. [2008] discovered that bi-manual input with both direct touch and pen simultaneously is superior to bi-manual direct touch input in some situations. The user performed different tasks, one particular example was a drawing application with a drawing canvas on the right and a huge settings and toolbox pane on the left. Selecting menus and creating a free-hand drawing are two distinct semantic tasks. These are easier to separate for our brain if they are executed in a more distinguished way, such as one by direct touch and the other with a digital pen. For different application tasks they used three setups, one with two pens, one with bi-manual touch, and one with touch and pen combined. The results suggest that “speed, accuracy, and user preference” of the pen and touch setup were superior to the others.

Therefore:

**Support a digital pen as an additional input device. The pen should at least be easily trackable by your system without any errors, and it can be enhanced with special functionality to enrich the input variety, e.g., for drawing applications or text input.**



With novel interaction techniques, the pen can be used for many different tasks, such as EXTENDING REACHABILITY. When the pen is not used, it can be placed in a PHYSICAL OBJECT STORAGE BIN so it does not get lost...

### (31) PHYSICAL KEYBOARD \*



**Figure 5.16:** Wireless keyboards on a tabletop.

... you have built a **LARGE COLLABORATION TABLE** for an office environment, e.g., to **REPLACE PHYSICAL PAPERWORK** in an administration, or as a meeting tabletop device. You expect a lot of text input and you are looking for solutions such that users will not lose time taking notes compared to someone using a laptop or a sheet of paper.



**Users often need text input on a tabletop. However, a displayed virtual keyboard on the tabletop lacks tactile feedback of the keys.**

Text input is necessary in many different use cases. On an interactive tabletop in an exhibit, you probably do not need text, maybe only for putting in the user's name or a comment; an **ON-SCREEN KEYBOARD** is sufficient for this scenario. However, for large text passages, e.g., in a meeting where one person is putting brainstorming notes into a mind map, or in administration office tasks with a lot of paperwork, such a virtual keyboard has many downsides. It lacks haptic feedback, i.e., the user does not feel where she is typing and whether she hits the center of a certain key or in between two keys, thus producing more typing errors.

Hinrichs et al. [2007] examined different approaches of text input and identified their benefits and drawbacks. A physical keyboard is one of the fastest input devices for text, it is familiar from the traditional desktop setup, and novice-user friendly. However, due to its size it reduces the visible screen space on the tabletop. Additionally, the transition between an external input device and the touch surface can interrupt the user's task focus. Nevertheless, depending on the setup the authors rate a physical keyboard as best input method for most systems.

Summarizing their tabletop research experience and drawing conclusions from their observations, Ryall et al. [2006] also suggest that a wireless keyboard is the

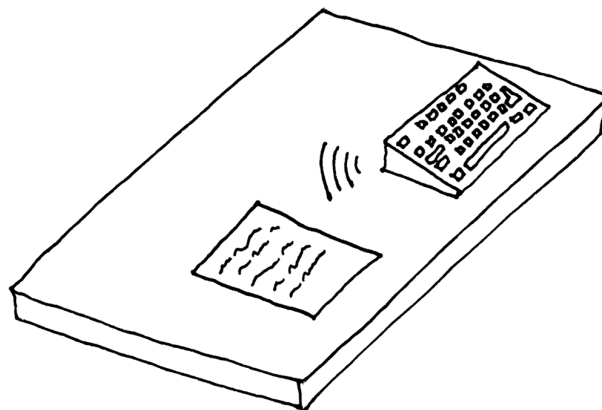
best input solution for text. They point out that typical tabletop tasks should minimize the need for text input and focus on other, more tabletop-suited tasks.

In a more recent paper, Hartmann et al. [2009] used wireless keyboards on a tabletop. To overcome the occlusion problem from the keyboard-occupied space, the tabletop was larger than other usual tabletop systems. This introduced another common tabletop problem: users could not reach the whole table. They added wireless mice to solve this problem, additionally to the keyboards. Although this seems to degenerate the tabletop to a simple horizontal output display, since mouse and keyboard input seems to make the direct touch input obsolete, informal user tests showed that users still switched between physical device input and direct touch input. This once more emphasizes the observation that physical input devices do not replace, but supplement touch input.

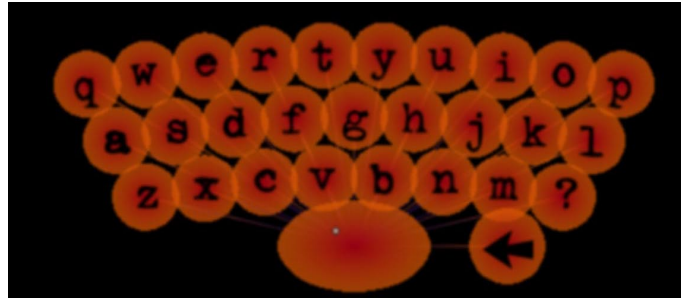
While we propose in this pattern to offer a keyboard for text input, the use of mice should be avoided by choosing the right table size. If it is too small to offer an additional keyboard without sacrificing too much space, other solutions such as INPUT TANGIBLES may be more appropriate.

Therefore:

**Put a wireless keyboard on the tabletop to allow the user a familiar way of text input with haptic feedback. The keyboard should be small compared to the table size and easy to move around to minimize occlusion issues.**



Buy a special keyboard, e.g., with small displays or transparent keys, or modify your available keyboard to support DYNAMIC KEYBOARD RELABELING. After finishing your tasks, the keyboard wastes a lot of space on the screen, so put it in a PHYSICAL OBJECT STORAGE BIN...

**(32) ON-SCREEN KEYBOARD \***

**Figure 5.17:** *BubbleType* on-screen keyboard for text input on a tabletop.

... your tabletop is used in scenarios like interactive games, at exhibits, or in meetings where it serves more as a presentation table than for a lot of paperwork, and thus only needs limited amount of text input. In these occasions your screen space becomes limited, with cups, notes, and electronic devices like cell phones on the table, or it is even a TILTED TABLE, where you cannot put objects on.



**A physical keyboard may offer a familiar text input environment, but it clutters the space with its occlusive appearance. On a very limited amount of available screen space, you have to enable text input sometimes in order to store textual information.**

Text entry is necessary in almost every imaginary scenario for an interactive tabletop. In exhibits you may want to allow comments, in meetings participants want to take notes, and in a game you want to put your name in before you start playing. But as Hinrichs et al. [2007] elaborated, there are situations in which the realization of text input becomes a difficult problem. E.g., if your space is limited, as on a rather small device, or in the case you have a TILTED TABLE or even a vertical wall instead of a horizontal tabletop surface, you cannot place input objects on the surface. Also, if your task demands only for small amount of input such as short annotations, it is a tedious waste of time to get a PHYSICAL KEYBOARD, typing in your text, and then putting it away again. Sometimes even a keyboard is not necessary, you may even allow hand-written annotations, which can easily be extended by drawings to illustrate the note.

These interfaces do not need to mimic the layout of a physical keyboard, instead they should make proper use of the dynamic behavior by displaying only necessary input elements. E.g., if the input focus is on a numeric-only text input area, such as a calculator application. Therefore, a small number keypad is sufficient. On the other hand, those number keypads as well as the function keys are rarely used in a pure text-writing environment and can be hidden. Other examples are, e.g.,

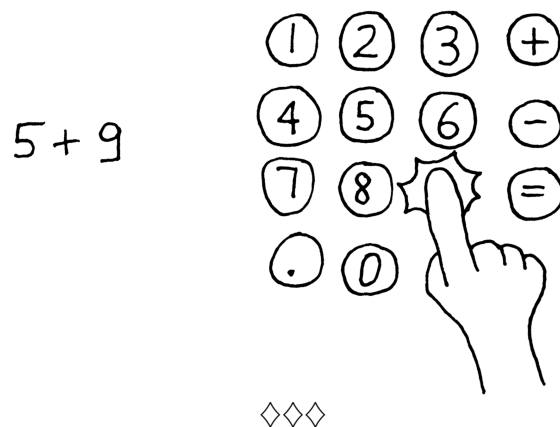
graphical software where only hotkeys are needed, applications that limit input to function keys, or only copy, paste, edit keys in a photo viewing program.

Another particular aspect of on-screen keyboards is the ability to try completely different new layouts, even those that are not easily realizable on physical devices. User tests that compare these new layouts with traditional ones could lead to completely new input devices. Hirche et al. [2008] introduced a keyboard with 14 keys (one for each finger except the little fingers and forefingers, which has two keys), where a small shift downwards or upwards for each finger hit another letter. The order of letters on the keys was arranged according to the frequency of use in the English language.

However, keep in mind that these layouts are only useful if users are faster than on the default "QWERTY" layout. Changing this is not recommended in scenarios such as exhibits or for other tabletop use cases where users expect familiar keyboards. Hinrichs et al. [2008] proposed such different layouts, e.g., keys in a circle around the hand, or a single line of keys. But BubbleType, the final suggestion of their paper, is again a traditional "QWERTY" keyboard layout. It makes use of the tabletop display by offering a prediction system: After typing a specific letter, the next most likely letter is highlighted to decrease the possibility of a typing error.

Therefore:

**Implement an on-screen keyboard that uses the advantages of the tabletop's direct touch nature. The keys should be as big as on traditional keyboards and easy to hit, offer visual and auditory feedback to ease the interaction. If the input is limited to numbers or some other subset of keys, display only these to save space.**



An on-screen keyboard offers superior, previously unknown possibilities to the user's working experience, such as DYNAMIC KEYBOARD RELABELING, which makes excellent use of the tabletop display for your tasks...

**(33) INPUT TANGIBLES \*\***

**Figure 5.18:** The *Reactable* tabletop with tangibles.

... if you build a tabletop for environments like exhibits or for gaming purposes, you probably want novel and unfamiliar input devices. Also, if your tabletop is used in a learning environment, traditional desktop PC input devices might not suffice your demands and the user can benefit from additional devices, e.g., to allow simultaneous input from multiple users.



**The direct touch input of a tabletop does not always suffice the needs for intuitive interaction and tactile feedback. External devices like keyboards draw the user's attention away from the tabletop and hinder fluent interaction.**

Ishii [2008] illuminates the advantages of tangible user interfaces (TUIs), and Fishkin [2004] proposed a taxonomy of TUIs. There are many different kinds of TUIs, but they all share the same solutions to common problems, such as the lack of tactile feedback or the limited input due to the flat tabletop surface. While they add tactile feedback, they do not separate visually from the tabletop since they are directly mounted on the surface. Additionally, they allow direct control of the tangible objects' virtual representation by manipulating a physical widget.

An example of the possibilities of tangibles is Photohelix [Hilliges et al., 2007], a knob that can be twisted to scroll through a circular arrangement of pictures. The system design supports bimanual photo editing with the Photohelix, such as holding a knob in the left hand and an ANOTO pen in the right hand. Specific gestures for particular viewing and editing options facilitate handling of large collections.

SLAP widgets [Weiss et al., 2009] combine the advantages of the tangibles' idea with the tabletop output projected on the screen. The transparent objects allow

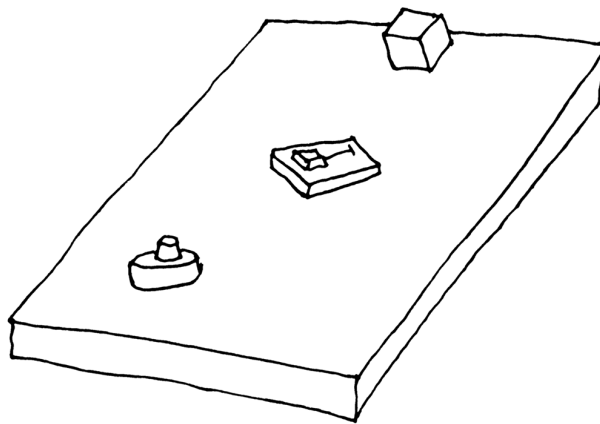


input metaphors such as rotating a knob mounted on the screen, while the table is still visible beneath it and thus the interaction result is immediately visible without drawing off the user's attention from the screen. This also adds new possibilities for text input. The SLAP keyboard offers a customizable keyboard layout: when pressing the shift or the control key, the keyboard layout changes to reflect the projected key or action when pressed.

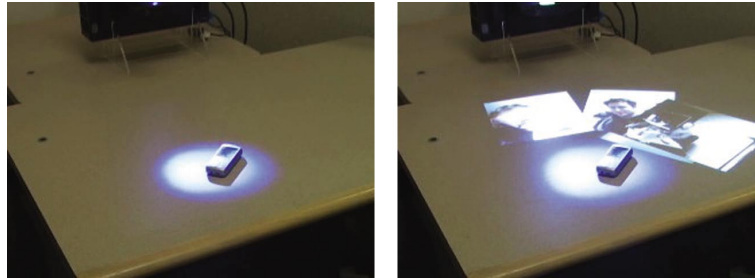
Another example is the *Reactable* [Jorda et al., 2007], where different tangible widgets represent different synthesizer instruments on the table. By moving, rotating, and turning those objects the user can change aspects like volume, tone, pitch, and various other settings. Multiple tangibles can be arranged relatively to each other to generate synergy effects. During interaction the user gets continuous and immediate feedback, both visual and acoustic.

Therefore:

**Offer tangibles to enhance interaction and increase input variety. Keep the tangibles unobtrusive and provide clear and intuitive transitions towards traditional direct touch tabletop input so that users do not feel distracted by the additional devices.**



Tangibles can have displays or just be transparent to support DYNAMIC KEYBOARD RELABELING. Especially when you have a large number of tangibles, a PHYSICAL OBJECT STORAGE BIN can keep the tabletop tidy and your objects do not get lost...

**(34) EMBEDDING ELECTRONIC DEVICES \***

**Figure 5.19:** A cell phone sharing its data with an interactive tabletop.

... your tabletop is built and you have implemented a basic software framework. But you realize that users need their data while working on the interactive tabletop, or you even want the users to use their data, e.g., in a meeting at a **LARGE COLLABORATION TABLE**, in an exhibit, or if your **TILTED TABLE** is used as an information device in public.



**Tabletops offer large areas to work on personal data or share it with others, e.g., to work collaboratively on it, but those files are often on electronic devices such as cell phones or notebooks.**

We carry a lot of data with us, like documents on our laptop, pictures, or contact data on our cell phone, some more data on a pen drive or other devices. When you work with a tabletop, you deposit these devices somewhere; maybe in a **PHYSICAL OBJECT STORAGE BIN**. Then you grasp a keyboard to type some text, or a digital pen to paint something, or some tangibles for other special input. But one might ask: why not combine all these actions to enhance your tabletop experience? Instead of putting your laptop or your cell phone away, place it on the tabletop and use it as input device as well as use the data on it. Former devices which were recognized as obstacles and hindered proper interaction, should rather be recognized as tools in order to improve the user's experience.

There are numerous examples for this, e.g., with your laptop you could create a small synchronized text input area on a **LARGE COLLABORATION TABLE** to take notes, or an information screen that tells the user the way to a sight and store the direction on her cell phone. Other scenarios could be, e.g., a tabletop game where participants can save their game on a pen drive, or a meeting where workers bring in sales figures from different company departments to compare them on the tabletop and collaboratively create one single chart for a presentation.

Rekimoto et al. [1999] suggest to connect your electronic devices to the tabletop environment to create a "spatially continuous work space". The laptop keyboard can

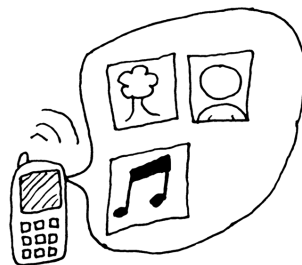
be used as text input device, documents are shared, and the laptop display compensates for the occluded part of the tabletop surface. The devices are recognized via a camera mounted on top of the surface that searches for unique pattern markers attached. An application example shows a large area map on the tabletop and upon pointing to sights they are displayed on the laptop's screen.

Eight years later, connecting these devices has become incredibly easy, since almost every portable device supports Wireless network and Bluetooth. Based upon the latter, Wilson and Sarin proposed *BlueTable* [2007], a tabletop system which uses object detection via a camera on top of the surface just like Rekimoto, but now detection by shape was used and without the need to attach markers to it. When a known device like a cell phone is recognized, a connection is automatically established via Bluetooth handshaking protocol. If it is successful, recent files on the cell phone like pictures are automatically displayed around the phone on the surface. *Spilling* by Olsen et al. [2007] uses a different technique for detecting a hand-held: conductive feet that mimic finger touches for the tabletop allow not only recognition of the device, but also its orientation. Thus the device itself can be used as a tangible, e.g., turning an application window according to the position of the device.

All the previous approaches need either markers to be physically attached to the device or its shape stored in a database, although the shape is not unique. Kray et al. [2008] use markers like Rekimoto, but not physical ones. They use a cell phone's display to show a pattern which is identified by the camera above the surface. Using multiple cell phones with unique markers, they created an application to exchange photos on a tabletop surface.

Therefore:

**Ease the transfer of data between the tabletop and other electronic devices. Support seamless connectivity to share data and expand the tabletop's input variety.**



To avoid that these devices occlude important information on the screen, techniques such as DODGE OBSTACLES are helpful...

**(40) EXTENDING REACHABILITY \***

**Figure 5.20:** A roulette rake mimics a physical example of reaching extension.

... you have built a tabletop system with a large surface that supports HAND GESTURES, and you possibly added INPUT TANGIBLES or a PEN INPUT DEVICE. The tabletop offers a lot of space for a specific number of users, but sometimes there are not as many collaborators as expected, and there are unreachable areas on the table for some users.



**Tabletops should be as large as possible, even if many users are working on it, it should offer enough space. But the larger the table, the further away are certain interface elements and other widgets on the table, and users cannot reach them anymore.**

On traditional desktop computers, the area of reach is not an issue, since the mouse is a relative device and even if direct input is supported, e.g., on recent multi-touch enabled laptops, the surface is rather small. Tabletops are usually large surfaces, and especially if you create a collaborative environment with a rather huge display, there will likely be a scenario where users cannot reach elements they want to interact with. Many studies suggest that users then ask other participants to pass the elements, but as Zhang and Takatsuka [2007] observed, this is not always the case. Their user tests included tasks with time pressure, an important aspect to be considered for meetings.

A study by Toney and Thomas [2006] named important aspects for the reachability problem on tabletops. Different regions serve as different areas for work practice: The area close to the user is the working space, farther away but still reachable is the storage space for documents that will be used later on, and there is the area that is not reachable without any additional technique or walking around the table.

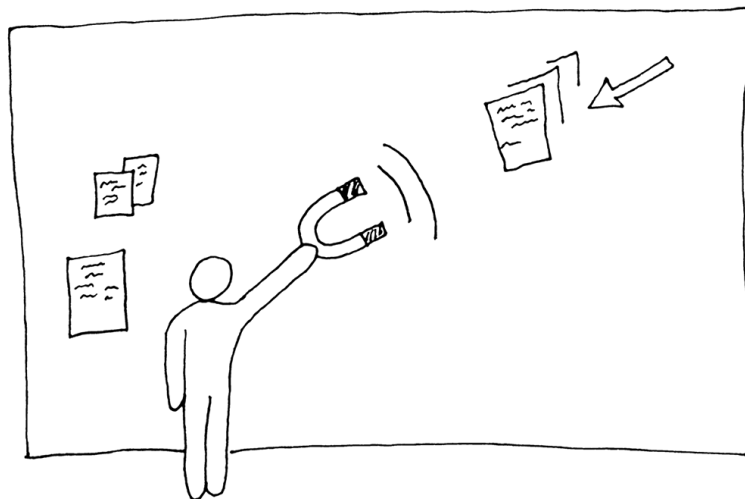
They also mention the difference between sitting and standing, as the reach area obviously increases significantly when the users are standing at the table.

Nacenta et al. [2005] compared six different reaching techniques on a tabletop in two different conditions, one with targets in user's hand's reach and one farther away and out of reach. The best technique was a radar, a small map that represents the whole table in miniature. Other solutions to be considered are sling shot and pantograph; in both approaches a pen stroke of a small distance is mapped to a longer movement alongside the stroke. The difference is that the sling shot gesture has an initial backward movement in the opposite direction, whereas the pantograph gesture has a pen stroke into the direction of the desired object.

A recently developed solution is the I-Grabber [Abednego et al., 2009], which uses a bi-manual gesture for grasping unreachable objects. Two finger touching the table subsequently in 20 cm distance of each other initialize the I-Grabber, within direction to the second touch. By moving this finger farther away from the other, the I-Grabber extends its length by five times the finger movement. Releasing the second finger selects the object which is located beneath the hook in front of the I-Grabber at the same time.

Therefore:

**Offer the user the ability to reach elements that are on the other side of the tabletop without having to get up from her place.**



This is a basic pattern with no further references within this language.

### (41) PHYSICAL OBJECT STORAGE BIN \*



**Figure 5.21:** An interactive SMART Board with storage for input devices.

... you have physical objects on your tabletop, e.g., a PEN INPUT DEVICE, a PHYSICAL KEYBOARD, INPUT TANGIBLES, a cell phone, or a laptop. These objects are used to extend your input variety in some occasions, but sometimes they just occlude the surface and you want to put them away.



**To extend the input variety, electronic devices, tangibles, and other objects can be connected to the table. But those objects clutter the screen and when not being used, they easily get lost.**

There are many physical objects to be used on tabletops, e.g., custom self-made tangibles, digital pens, or even desktop PC devices like a mouse or a keyboard. Even if you want your users to use nothing but their bare hands, you may have electronic devices or just everyday objects with you, e.g., a notebook, a cell phone, your keys, or just a glass of water. You do not want to place these objects on the table. A notebook wastes too much space, keys could damage the surface, and a cup of coffee may even be dangerous for the technology inside. On the other hand, you do not want to put these items far away, consider the notebook to configure the tabletop which has to stay close to it, or a cell phone or keys which you might forget if put too far away.

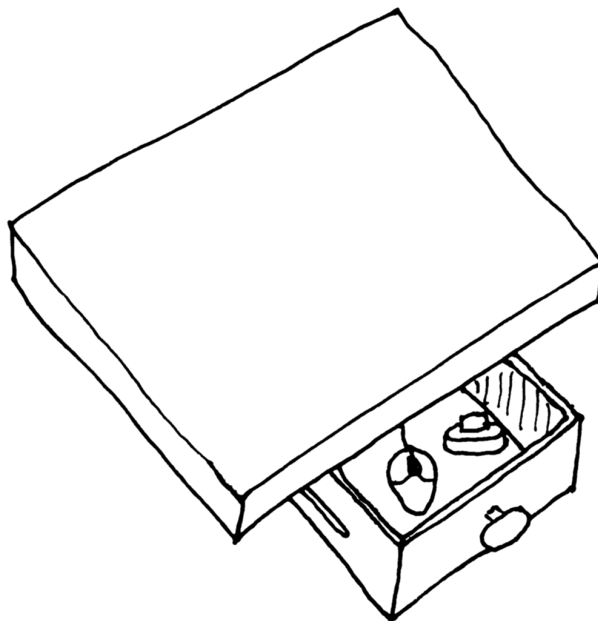
On every whiteboard or blackboard, there are storage bins for pens or pieces of chalk, respectively. It has become a ubiquitous element on whiteboards, which we

use without thinking about - and everyone knows where to look for chalk or pens on such boards, just take a look at the bottom of it. Most tabletops usually do not offer this room for items. You can see people putting big cases on the table when putting tangibles on it, and without some specific place for pens you will often hear the question "Have you seen my pen? Where is it? Where did you put it?". To avoid this, create a storage bin beneath or next to the table. It should be clearly visible and not hidden, but not too big on an edge where users are supposed to sit, since it would increase their distance to the table unnecessarily.

In scenarios where the user enters a lot of text, a PHYSICAL KEYBOARD can be necessary. In their paper about text input on interactive tabletops, Hinrichs et al. [2007] mention this scenario, and recommend a drawer for the keyboard. When the keyboard is not necessary because the text input task is finished, it does not clutter the surface and waste valuable space. On the other hand, as soon as you need it, the keyboard is quickly available again.

Therefore:

**Mount a small item storage beneath or on the side of the tabletop to store unused physical objects. It should be large enough so that a keyboard or other large input devices fits in, but small enough to not annoy the user who has to sit in front of it.**



This is a basic pattern with no further references within this language.

### (43) DYNAMIC KEYBOARD RELABELING \*

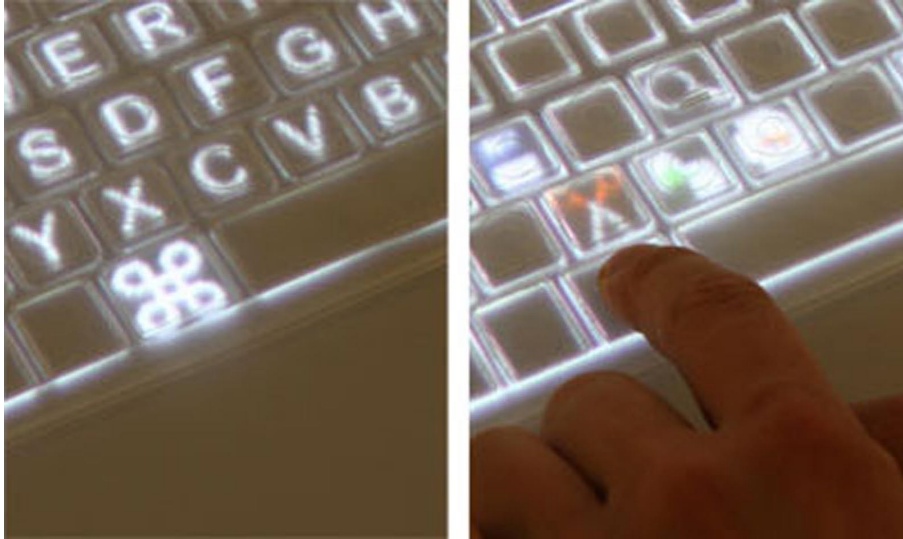


Figure 5.22: *SLAP* keyboard relabeling.

... you offer tabletop text input by either implementing an ON-SCREEN KEYBOARD, putting a PHYSICAL KEYBOARD on the table, or building your own INPUT TANGIBLES. But for your tasks, you are searching for more possible ways to make use of the tabletop display or the objects on the surface, to create a more vivid tabletop environment.



**Keyboards on tabletop should be flexible and unobtrusive, but there is a lack of the haptic feedback of quasi-mode keys like Shift or Control.**

Tabletop keyboards always introduce new problems, dependent on the solution you choose: a physical keyboard is obtrusive, tangibles might be unintuitive, and on-screen keyboards lack haptic feedback. You can compensate all these issues by extending your chosen keyboard layout to change dynamically. This is easily implemented for an ON-SCREEN KEYBOARD, where the labels are just displayed anyway. Beyond the obvious capitals or control keys, there are further options, e.g., a combination of keys could be suggested to the user by highlighting the next key to be pressed. *BubbleQWERTY* [Hinrichs et al., 2008] implemented this technique to avoid spelling mistakes, using a dictionary predicting the next most probable letter upon typing. Another advantage of keyboards with a changeable display is the possibility to highlight the pressed key. On smaller devices this can be a simple, but tremendous assistance for the user, consider the Apple iPhone [H:Apple2007], which enlarges the key pressed by the user. It is even possible to move the finger and look at the enlarged key to ensure that the right key was pressed.

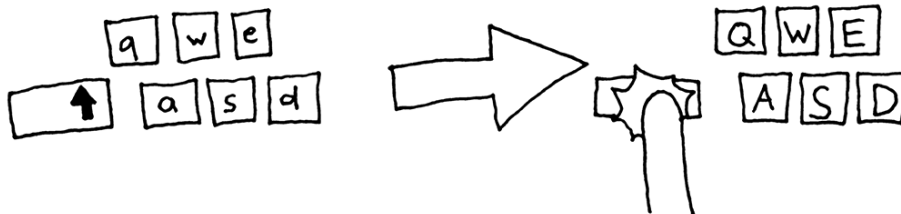


An example for INPUT TANGIBLES with dynamic relabeling of keys is the SLAP keyboard [Weiss et al., 2009], which is a keyboard made of silicon. It overcomes the lack of haptic feedback, but it is less obtrusive than a PHYSICAL KEYBOARD. Since it is transparent, the user can see the displayed items on the tabletop beneath the SLAP keyboard, but the user is able to feel the keys which she is typing. The application goes beyond the mapping for quasi-mode keys, e.g., for the function keys as depicted in the illustration. The user could choose her most familiar localized keyboard layout, e.g., an English, German, or French positioning of keys, or a completely user-defined custom layout.

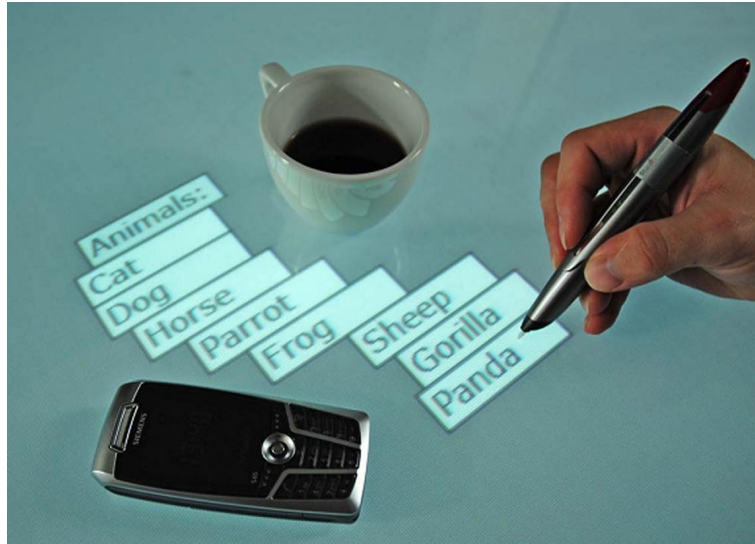
Even when using a PHYSICAL KEYBOARD, it is possible to change the keyboard layout, if the keys have small displays. A commercial available example is the Optimus Maximus keyboard [H:ArtLebedev2008], which consists of 113 OLEDs of 48x48 pixel size. While not being restricted to tabletops since it is a technology on its own, this is also one of its shortcomings: the tabletop display itself usually has no influence on what is being displayed on the keyboard.

Therefore:

**Change the keyboard labels dynamically. Compensate for missing haptic feedback and offer alternative layouts.**



This is a basic pattern with no further references within this language.

**(44) DODGE OBSTACLES \***

**Figure 5.23:** User-drawn path menus alongside a coffee cup on a tabletop.

... your tabletop system is used in everyday working areas like meetings, where users occasionally put certain devices on the tabletop. EMBEDDING ELECTRONIC DEVICES can offer new input as well as data connectivity, if these devices are capable of being combined with your system. But you also have to deal with objects like keys, cups, notes, or books, which occlude valuable screen space.



**The tabletop size is well-adjusted to the content displayed, so screen space is valuable and should not be wasted. But users carry everyday objects and may put them on the table during work, either unintentionally or on purpose.**

Cotting and Gross [2006] introduced a display system where the tabletop detects obstacles on the table and displays data accordingly. Widgets are not displayed as rectangular windows like on traditional desktop UIs, but as *display bubbles* around the obstacles. Every bubble area is surrounded by a boundary area to ensure that the bubbles are not too close to any obstacles and distract the user's view.

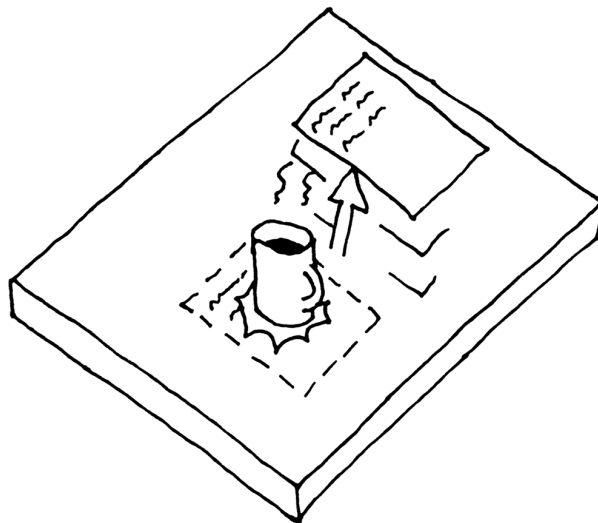
In a more specific approach, Leithinger and Haller [2007] invented user-drawn path menus on a pen-input driven tabletop. When the user requests a context menu, it is not drawn immediately but the first item appears on a pen tip. The system waits for the user to draw a line with the pen and creates the menu alongside the path. The authors implemented the following four different menu creation styles: The fan out menu expands all items simultaneously. The card deck menu with a delayed appearance sequence corresponding to the pen stroke. The pearl string

menu works like the card deck but with inversed order. And finally the trail menu which is similar to the pearl string menu, but if the pen is moved after the input, then the menu follows its trail.

Olwal and Wilson [2008] go one step further. They do not only avoid obstacles, but integrate them into the working area. RFID markers and a camera detect the objects and then contextual information is displayed, surrounding the item. These objects are more than just obstacles, they work almost as INPUT TANGIBLES, although they are like passive everyday objects.

Therefore:

**Embed the obstacles in your screen design. Detect those objects and let the information flow around, either automatically or manually by the user, and if you are aware of the context of those items, adjust the arrangement and content accordingly.**



This is a basic pattern with no further references within this language.



## Chapter 6

# Conclusions and Future Work

*“There’s a way to do it better—find it.”*

—Thomas Edison

### 6.1 Summary and Contributions

In this thesis, we presented an HCI design pattern language for interactive tabletops in collaborative workspaces. Based on the design pattern concept originated in architecture and transferred to many other domains, the lack of guidelines in this field was addressed by gathering the collective wisdom of research and industry and compile them into the established format of HCI design patterns.

Prior to the creation of patterns, a thorough literature review issued a list of problems and solutions in tabletop interaction. We identified conflicting forces and produced an initial set of HCI design patterns, whereby every pattern names examples from existing systems to support the validity, usefulness, and credibility of the proposed solution. Throughout the whole process, we collected feedback from tabletop and HCI design pattern experts to refine our results and increase the value of the pattern language.

Identified problems, conflicting forces, and compiled solutions into HCI design patterns

We extensively documented the last iteration step, an evaluation by an expert in both the HCI design patterns and interactive tabletops domains, which issued the final language of 22 HCI design patterns listed in this thesis. The pattern language, unlike previous approaches for guidelines in this domain, covers a broad area of interactive tabletop design, provides professionals with a comprehensive and sustainable body of knowledge, and aids novices to acquire knowledge about successful tabletop design.

## 6.2 Future Work

Writers' workshop:  
further evaluation

As aforementioned, one established procedure to evaluate, improve, and refine design patterns, is a writers' workshop [Gabriel, 2002]. This approved format is applied to many pattern languages on a regular basis, e.g., in the series of PLoP conference series. We intend to use this process as a further evaluation step and to gain additional feedback.

Extend the language  
by more patterns and  
adding pattern  
languages

Even after the writers' workshop, we will not consider our pattern language as finished, but rather as start for the community to collaboratively create a library of collective wisdom. Therefore we aim to encourage tabletop domain experts to contribute to the pattern language, by improving the existing patterns, but also search for new patterns. We support this process by introducing the HCI design pattern concept to the tabletop community, e.g., on workshops or at conferences in the future, and allow for immediate improvements on a moderated and quality-oriented website, building upon the existing web site used for creating this pattern language.

Furthermore, we search not only for improved or new patterns, but we could also imagine to include other pattern languages in a network of HCI design pattern languages for interactive tabletops. E.g., the pattern language for gestural interfaces by Saffer [2008] complements to our pattern HAND GESTURES, though on a slightly different basis. Likewise other pattern languages could complement areas in tabletop research and industry that have evolved to a complete particular domain on its own, e.g., tangibles or

the different tracking techniques, which still lack standards demanded by designers, developers, and practitioners.

In all these future approaches, incorporating the community is our highest priority, to make sure that the pattern language achieves the best possible quality and suits the specific demand of tabletop design. Therefore, we hope that our language contributes to the need for guidelines and advances research and industry to collaboratively work on a comprehensive body of knowledge for both experts and novices, bringing the community of interactive tabletops closer together.





## Appendix A

# Online Resources

This chapter contains the URLs of online resources that have been referenced in the text.

**Anoto2000** Anoto Group AB: *Anoto - THE PEN* (established 2000, accessed at April 30, 2010)

<http://www.anoto.com/the-pen.aspx>

**Apple2007** Apple Computer, Inc.: *Apple - iPhone* (established 2007, accessed at April 30, 2010)

<http://www.apple.com/iphone/>

**Apple2010** Apple Computer, Inc.: *Apple - iPad* (established 2010, accessed at April 30, 2010)

<http://www.apple.com/ipad/>

**ArtLebedev2008** Art. Lebedev Studio: *Optimus Maximus Keyboard* (established 2008, accessed at April 30, 2010)

<http://www.artlebedev.com/everything/optimus/>

**Borchers1999** Jan O. Borchers: *The HCI Patterns Home Page* (established 1999, accessed at April 30, 2010)

<http://www.hcipatterns.org/>

**Erickson1998** Thomas Erickson: *Interaction Patterns Home Page* (established February 1998, accessed at April 30, 2010)

<http://www.visi.com/~snowfall/InteractionPatterns.html>

**Fincher2000** Sally Fincher: *The Pattern Gallery* (established 2000, accessed at April 30, 2010)

<http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html>

**Infragistics2009** Infragistics, Inc.: *Infragistics Quince UX Design Patterns Explorer* (established 2009, accessed at April 30, 2010)

<http://quince.infragistics.com/>

**Hillside1997** Hillside Group: *Design Patterns Library* (established 1997, accessed at April 30, 2010)

<http://www.hillside.net/patterns/>

**Reactable2009** Reactable Systems: *Reactable* (established 2009, accessed at April 30, 2010)

<http://www.reactable.com/>

**Toxboe2008** Anders Toxboe: *User Interface Design Patterns Library* (established 2008, accessed at April 30, 2010)

<http://ui-patterns.com/>

**Welie2000** Martijn van Welie: *A Pattern Library for Interaction Design* (established 2000, accessed at April 30, 2010)

<http://welie.com/>

**Yahoo!2005** Yahoo!, Inc.: *Yahoo! Design Pattern Library* (established 2005, accessed at April 30, 2010)

<http://developer.yahoo.com/ypatterns/>

## Bibliography

- Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *ITS '09: Proceedings of ACM International Conference on Interactive Tabletops and Surfaces*, pages 61–64, Banff, Alberta, Canada, November 2009.
- Christopher Alexander. *The Timeless Way of Building*. Oxford University Press US, 1979.
- Christopher Alexander. Keynote speech, OOPSLA '96: Conference on object-oriented programs, systems, languages and applications, 1996.
- Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, August 1977.
- Apple Computer Inc. *Macintosh Human Interface Guidelines*. Addison-Wesley Professional, 2nd edition, 1993.
- Kent Beck and Ward Cunningham. Using pattern languages for object-oriented programs. In *OOPSLA '87: Workshop on Specification and Design for Object-Oriented Programming, Systems, Languages and Applications*, 1987.
- Hrvoje Benko, Andrew Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, April 2006.
- Xiaojun Bi, Yuanchun Shi, and Xiaojie Chen. uPen: a smart pen-liked device for facilitating interaction on large displays. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 160–168, 2006.

- Jan Borchers. Interaction design patterns: Twelve theses. In *Position paper on CHI 2000 workshop*, The Hague, Netherlands, April 2000.
- Jan Borchers. *A Pattern Approach to Interaction Design*. John Wiley & Sons, Inc., New York, 2001.
- Jan Borchers. The aachen media space: Design patterns for augmented work environments. In *Designing User Friendly Augmented Work Environments*, pages 261–312. Springer Verlag, 2009.
- Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 154–161, Napoli, Italy, 2008. ACM.
- Katharina Bredies, Nick Alexander Mann, Jens Ahrens, Matthias Geier, Sascha Spors, and Michael Nischt. The multi-touch SoundScape renderer. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 466–469, Napoli, Italy, 2008. ACM.
- Daniel Cotting and Markus Gross. Interactive environment-aware display bubbles. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 245–254, Montreux, Switzerland, 2006. ACM.
- Andy Dearden and Janet Finlay. Pattern languages in HCI: a critical review. *Human-Computer Interaction*, pages 49–102, March 2006.
- Paul Dietz and Darren Leigh. DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, Orlando, Florida, 2001. ACM.
- Pierre Dragicevic and Yuanchun Shi. Visualizing and manipulating automatic document orientation methods using vector fields. In *ITS '09: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 65–68, Banff, Alberta, Canada, 2009. ACM.

- Thomas Erickson. Lingua francas for design: Sacred places and pattern languages. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 357–368, New York City, New York, United States, 2000. ACM.
- Kenneth P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal Ubiquitous Computing*, 8(5):347–358, 2004.
- Clifton Forlines and Chia Shen. DTLens: multi-user tabletop spatial data exploration. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 119–122, Seattle, WA, USA, 2005. ACM.
- Richard P. Gabriel. *Writers' Workshops & the Work of Making Things: Patterns, Poetry...* Pearson Education, 1st edition, 2002.
- Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.
- Ian Graham. *A Pattern Language for Web Usability*. Pearson Education, 1st edition, 2003.
- Richard Griffiths, Lyn Pemberton, Jan Borchers, and Adam Stork. Pattern languages for interaction design: Building momentum. In *CHI '00: Extended abstracts on Human factors in computing systems*, pages 363–363, The Hague, The Netherlands, 2000. ACM.
- Richard N. Griffiths and Lyn Pemberton. Teaching usability design through pattern language. In *CHI '99: Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1999.
- Tovi Grossman and Daniel Wigdor. Going deeper: a taxonomy of 3D on the tabletop. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 137–144, Newport, RI, USA, 2007.
- Björn Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. Augmenting interactive tables with mice & keyboards. In *UIST '09: Proceedings of the*

- 22nd annual ACM symposium on User interface software and technology*, pages 149–152, Victoria, BC, Canada, 2009. ACM.
- Otmar Hilliges, Dominikus Baur, and Andreas Butz. Photohelix: Browsing, sorting and sharing digital photo collections. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 87–94, 2007.
- Uta Hinrichs, Mark Hancock, Christopher Collins, and Sheelagh Carpendale. Examination of text-entry methods for tabletop displays. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 105–112, October 2007.
- Uta Hinrichs, Holly Schmidt, Tobias Isenberga, Mark S. Hancock, and Sheelagh Carpendale. BubbleType: enabling text entry within a walk-up tabletop installation. Research Report 2008-893-06, Department of Computer Science, University of Calgary, Canada, 2008.
- Johannes Hirche, Peter Bomark, Mikael Bauer, and Pawel Solyga. Adaptive interface for text input on large-scale interactive surfaces. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 153–156, October 2008.
- Kori Inkpen, Kirstie Hawkey, Melanie Kellar, and Regan Mandryk. Exploring display factors that influence co-located collaboration: Angle, size, number, and user arrangement. In *HCI '05: In proceedings of HCI International 2005*, 2005.
- International Organization for Standardization. ISO 9241-5, ergonomic requirements for office work with visual display terminals: Workstation layout and postural requirements, 1998.
- Hiroshi Ishii. Tangible bits: Beyond pixels. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv, Bonn, Germany, 2008. ACM.
- Hiroshi Ishii and Minoru Kobayashi. ClearBoard: a seamless medium for shared drawing and conversation with

- eye contact. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 525–532, Monterey, California, United States, 1992. ACM.
- Sergi Jorda, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, Baton Rouge, Louisiana, 2007. ACM.
- Hideki Koike, Shintaro Kajiwar, Kentaro Fukuchi, and Yoichi Sato. Information layout and interaction on virtual and real rotary tables. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 95–102, 2007.
- Kostas Koukouletsos, Babak Khazaei, Andy Dearden, and Mehmet Ozcan. Teaching usability principles with patterns and guidelines. In *Creativity and HCI: From Experience to Design in Education*, pages 159–174. Springer Verlag, 2009.
- Christian Kray, Michael Rohs, Jonathan Hook, and Sven Kratz. Group coordination and negotiation through spatial proximity regions around mobile devices on augmented tabletops. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 1–8, 2008.
- Hanno-Walter Kruft. *Geschichte der Architekturtheorie. Studienausgabe: Von der Antike bis zur Gegenwart*. Verlag C.H. Beck, 5th edition, April 2004.
- Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Saul Greenberg. Roles of orientation in tabletop collaboration: Comprehension, coordination and communication. *Computer Supported Cooperative Work (CSCW)*, 13(5): 501–537, December 2004.
- S. K. Lee, W. Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bulletin*, 16(4):21–25, 1985.
- Daniel Leithinger and Michael Haller. Improving menu interaction for cluttered tabletop setups with user-drawn path menus. In *TABLETOP '07. Second Annual IEEE*

- International Workshop on Horizontal Interactive Human-Computer Systems*, pages 121–128, 2007.
- Paul Marshall, Eva Hornecker, Richard Morris, Nick Sheep Dalton, and Yvonne Rogers. When the fingers do the talking: A study of group participation with varying constraints to a tabletop interface. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 33–40, 2008.
- Masood Masoodian, Sam McKoy, and Bill Rogers. Hands-on sharing: Collaborative document manipulation on a tabletop display using bare hands. In *CHINZ '07: Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction*, pages 25–31, Hamilton, New Zealand, 2007. ACM.
- Khaireel A. Mohamed, S. Haag, Julia Peltason, Frank Dalri, and Thomas Ottmann. Disoriented pen-gestures for identifying users around the tabletop without cameras and motion sensors. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 43–52, Washington, DC, USA, 2006. IEEE Computer Society.
- Merrie Morris, Bernheim A. J Brush, and Brian Meyers. A field study of knowledge workers' use of interactive horizontal displays. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 105–112, 2008.
- Christian Muller-Tomfelde, Anja Wessels, and Claudia Schremmer. Tilted tabletops: In between horizontal and vertical workspaces. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 49–56, October 2008.
- Miguel A. Nacenta, Dzmitry Aliakseyeu, Sriram Subramanian, and Carl Gutwin. A comparison of techniques for multi-display reaching. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 371–380, Portland, Oregon, USA, 2005. ACM.
- Donald Norman. *The Design of Everyday Things*. Basic Books, New York, February 1990.



- Dan R. Olsen, Jeffrey Clement, and Aaron Pace. Spilling: Expanding hand held interaction to touch table displays. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 163–170, 2007.
- Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 295–304, Florence, Italy, 2008. ACM.
- Peter Peltonen, Esko Kurvinen, Antti Salovaara, Giulio Jacucci, Tommi Ilmonen, John Evans, Antti Oulasvirta, and Petri Saarikko. It's mine, don't touch!: Interactions at a large multi-touch display in a city centre. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1285–1294, Florence, Italy, 2008. ACM.
- Ken Perlin and David Fox. Pad: An alternative approach to the computer interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 57–64, Anaheim, CA, 1993. ACM.
- Tommaso Piazza and Morten Fjeld. Ortholumen: Using light for direct tabletop input. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 193–196, 2007.
- David Pinelle, Mutasem Barjawi, Miguel Nacenta, and Regan Mandryk. An evaluation of coordination techniques for protecting objects and territories in tabletop groupware. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 2129–2138, Boston, MA, USA, 2009. ACM.
- Anne Marie Piper and James D. Hollan. Tabletop displays for small group study: Affordances of paper and digital materials. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, April 2009.
- Anne Marie Piper, Eileen O'Brien, Meredith Ringel Morris, and Terry Winograd. SIDES: a cooperative tabletop computer game for social skills development. In *CSCW*

- '06: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 1–10, Banff, Alberta, Canada, 2006. ACM.
- Jef Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2nd edition, April 2000.
- Jun Rekimoto and Masanori Saitoh. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, Pittsburgh, Pennsylvania, United States, 1999. ACM.
- Christian Remy, Malte Weiss, and Jan Borchers. A pattern language for interactive surfaces in collaborative workspaces. In *ITS '09: Extended Abstracts of ACM International Conference on Interactive Tabletops and Surfaces*, Banff, Alberta, Canada, November 2009.
- Kathy Ryall, Clifton Forlines, and Chia Shen. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 284–293, Chicago, Illinois, USA, 2004. ACM.
- Kathy Ryall, Meredith Ringel Morris, Katherine Everitt, Clifton Forlines, and Chia Shen. Experiences with and observations of direct-touch tabletops. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 89–96, Washington, DC, USA, 2006. IEEE Computer Society.
- Dan Saffer. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, 1st edition, November 2008.
- Toshiki Sato, Kentaro Fukuchi, and Hideki Koike. Implementation and evaluations of vision-based finger flicking gesture recognition for tabletops. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, 2008.
- Dominik Schmidt and Hans Gellersen. Show your hands: A vision-based approach to user identification for interactive surfaces. In *ITS '09: Extended Abstracts of ACM In-*

*ternational Conference on Interactive Tabletops and Surfaces*, Banff, Alberta, Canada, November 2009.

Stacey D. Scott, Karen D. Grant, and Regan L Mandryk. System guidelines for Co-Located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 159–178, Helsinki, Finland, 2003. Kluwer Academic Publishers.

Abigail J. Sellen and Richard H. R. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

Chia Shen, Frédéric D Vernier, Clifton Forlines, and Meredith Ringel. DiamondSpin: an extensible toolkit for Around-the-Table interaction. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 167–174, Vienna, Austria, 2004. ACM.

Ben Shneiderman. *Designing the User Interface*. Addison Wesley, Reading, MA, 3rd edition, 1997.

Lucia Terrenghi, David Kirk, Abigail Sellen, and Shahram Izadi. Affordances for manipulation of physical versus digital media on interactive surfaces. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1157–1166, San Jose, California, USA, 2007. ACM.

Lucia Terrenghi, Aaron Quigley, and Alan Dix. A taxonomy for and analysis of multi-person-display ecosystems. *Personal Ubiquitous Computation*, 13(8):583–598, 2009.

Jenifer Tidwell. *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, Inc., November 2005.

Aaron Toney and Bruce H. Thomas. Considering reach in tangible and table top design. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 57–58. IEEE Computer Society, 2006.

Edward Tse, Jonathan Histon, Stacey D. Scott, and Saul Greenberg. Avoiding interference: How people use spatial separation and partitioning in SDG workspaces. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 252–261, Chicago, Illinois, USA, 2004. ACM.

Edward Tse, Chia Shen, Saul Greenberg, and Clifton Forlines. Enabling interaction with single user applications through speech and gestures on a multi-user tabletop. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 336–343, Venezia, Italy, 2006. ACM.

Douglas K. van Duyne, James A. Landay, and Jason I. Hong. *The Design of Sites: Patterns for Creating Winning Web Sites*. Prentice Hall PTR, 2nd edition, 2006.

Pawan Vora. *Web Application Design Patterns*. Morgan Kaufmann, March 2009.

James R. Wallace and Stacey D. Scott. Contextual design considerations for co-located, collaborative tables. In *TABLETOP '08. 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, pages 57–64, Amsterdam, Netherlands, 2008.

Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James Hollan, and Jan Borchers. SLAP widgets: Bridging the gap between virtual and physical controls on tabletops. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, April 2009.

Daniel Wigdor and Gerald Morrison. Designing user interfaces for multi-touch and surface-gesture devices. In *CHI '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 3193–3196, Atlanta, Georgia, USA, 2010. ACM.

Daniel Wigdor, Gerald Perm, Kathy Ryall, Alan Esenther, and Chia Shen. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 60–67, 2007.

Andrew D. Wilson and Raman Sarin. BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *GI '07: Proceedings of Graphics Interface 2007*, pages 119–125, Montreal, Canada, 2007. ACM.

- Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1083–1092, Boston, MA, USA, 2009. ACM.
- Mike Wu, Chia Shen, Kathy Ryall, Clifton Forlines, and Ravin Balakrishnan. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 185–192, Washington, DC, USA, 2006. IEEE Computer Society.
- Xianhang Zhang and Masahiro Takatsuka. Put that there NOW: group dynamics of tabletop interaction under time pressure. In *TABLETOP '07. Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 37–43, 2007.



# Index

Advantages of HCI design patterns .....	13–16
Alexander, Christopher .....	5
- keynote .....	9
ANOTO .....	76, 82
Anti-patterns .....	35, 38
Apple	
- iPad .....	72, 73
- iPhone .....	65, 66, 70, 90
Architectural design patterns .....	5
Augmented Reality .....	22, 29
BALANCED PARTICIPATION pattern .....	74
<i>BlueTable</i> .....	85
Borchers, Jan .....	10, 22
CHI 2000 design patterns workshop .....	10
Conflicting forces .....	<i>see</i> Forces
Connectivity of design patterns .....	14
CSCW .....	18
Definition of HCI design pattern .....	10
Design of Everyday Things, The .....	9, 13
Design pattern	
- context .....	11, 14
- diagram .....	12
- illustration .....	11
- name .....	10
- problem statement .....	11
- ranking .....	11
- rationale .....	12
- references .....	13
- solution .....	12, 14
DESKTOP ORIENTATION pattern .....	60
<i>DiamondSpin</i> .....	61
DiamondTouch	
- <i>DiamondTouch</i> .....	61, 62
Digital paper .....	<i>see</i> Paperwork
DODGE OBSTACLES pattern .....	92
<i>DTLens</i> toolkit .....	64

- 
- DYNAMIC KEYBOARD RELABELING pattern ..... 90
  - Educational use of HCI design patterns ..... 15, 35
  - Eight Golden Rules of interface design ..... 13
  - EMBEDDING ELECTRONIC DEVICES pattern ..... 84
  - ERGONOMIC HEIGHT pattern ..... 56
  - Ergonomics ..... *see* Tabletop patterns—ergonomics
  - Evaluation of tabletop patterns ..... 37–45
  - Examples in design patterns ..... 15, 32
  - EXTENDING REACHABILITY pattern ..... 86
  
  - Forces ..... 11, 30, 39
  - Future work ..... 96–97
  
  - Gang of Four ..... 9, 23
  - Generativity of design patterns ..... 15
  - Gestures ..... 22, 28, 42, 63, 66–67, 71, 87, 96
  
  - HAND GESTURES pattern ..... 66
  - Haptic feedback ..... 28, 78, 90
  - HCI design pattern ..... 10
  - HIGH PRECISION INPUT pattern ..... 70
  
  - Inductive approach ..... 12, 37
  - Input devices ..... 29, 76–85
  - INPUT TANGIBLES pattern ..... 82
  - Interdisciplinary design ..... 10, 16
  - Invariant design solution ..... 30, 35
  - Iterative pattern development ..... 30, 33
  - ITS 2009 conference ..... 19, 32
  
  - LARGE COLLABORATION TABLE pattern ..... 54
  - Lingua franca* ..... 13
  
  - Macintosh Human Interface Guidelines ..... 13
  - Microsoft Surface* ..... 2
  
  - NARROW SUBSTRUCTURE pattern ..... 58
  
  - Obstacles ..... 28, 52, 72, 92–93
  - Occlusion ..... 28, 52, 58
  - ON-SCREEN KEYBOARD pattern ..... 80
  - Orientation ..... 29, 60–61, 85
  
  - Paperwork ..... 1, 29, 42, 72–73
  - Participatory design ..... 13
  - Pattern stubs ..... 30
  - Pen input ..... 29, 63, 76–77
  - PEN INPUT DEVICE pattern ..... 76
  - PHYSICAL KEYBOARD pattern ..... 78
  - PHYSICAL OBJECT STORAGE BIN pattern ..... 88



- 
- PLoP conference series ..... 9, 12, 25, 36, 96  
Precision ..... 28, 70–71, 76  
Private space ..... *see* Territoriality  
PRIVATE SPACE pattern ..... 68  
Problem-oriented approach ..... 28  
Prototyping ..... 41, 55, 57, 58
- Quality Without a Name ..... 6
- Reachability ..... 28, 50, 54, 86–87  
*Reactable* ..... 50, 82, 83  
REPLACE PHYSICAL PAPERWORK pattern ..... 72  
ROUND TABLE pattern ..... 50  
Rule of Three ..... 30
- Sensitizing example ..... *see* Design pattern illustration  
Seven Principles of Usability ..... 13  
SITTING WALL pattern ..... 7  
*SLAP* widgets ..... 82, 90, 91  
Software engineering design patterns ..... 6, 25  
Solutions in context ..... 14  
*SoundScape* renderer ..... 51  
Space issues ..... 50, 54–55, 64, 79, 80, 92
- Tabletop patterns ..... 50–93  
- ergonomics ..... 50–59  
- extending input ..... 76–85  
- interface ..... 60–65  
- special scenarios ..... 86–93  
- usability and specific collaboration aspects ..... 66–75  
- web site ..... 33  
Tangibles ..... 29, 44, 50, 79, 82–83, 85, 91, 93, 96  
Taxonomy  
- of 3D tabletops ..... 19, 20, 55  
- of gestures ..... 67  
- of multi-person-displays ..... 55  
- of tangibles ..... 82  
Territoriality ..... 28, 68–69  
Text input ..... 28, 52, 65, 70, 76–85, 89  
Tidwell, Jenifer ..... 10, 23  
TILTED TABLE pattern ..... 52  
Time in HCI design patterns ..... 9  
Turn taking ..... 29, 43, 74–75  
Typography as structure ..... 14
- USER IDENTIFICATION pattern ..... 62  
User interface design patterns ..... 10, 23
- Validation of design patterns ..... 15, 32  
View direction ..... 28, 60–61  
Vitruvius ..... 6

---

Writers' workshop .....	35, 96
ZOOMABLE INTERFACE pattern .....	64

