Seamless interaction in interactive rooms - some preliminary remarks*

Philipp Rossberger
adesso AG
Rotherstr. 19
10245 Berlin
rossberger@adesso.de

Kai von Luck
HAW Hamburg
Berliner Tor 7
20099 Hamburg
luck@informatik.haw-hamburg.de

Abstract

Seamless interaction is a crucial aspect for the acceptance of tabletop applications. Recent research has shown, that applications using physics simulation provoke better metal models and are therefore easy to learn and understand as they allow users to potentially employ interaction and work strategies from reality. Based on these results we propose physics simulation as a strategy to improve user interaction regarding the GUI and functionality of tabletop applications. Furthermore we present an user-centered design process for rapid development of physics-based applications used for creating a tabletop application called DynAmbient. Our approach enabled us to improve the usability of our applications through several fast user participatory development iterations.

1. Introduction

Developing easy to understand and intuitive graphical user interfaces (GUIs) and interaction techniques for computer programs is a major challenge software developers face. Ideally the GUI should explain its functionality by itself without requiring the user to read a manual. Single-user applications for operating systems like MS Windows or Apple OS X typically use a set of well-known graphical elements (e.g. tabs, scroll bars) and interaction techniques (e.g. double-clicking, Drag-and-Drop), which are virtually know by every user.

While digital direct-touch tabletops have attracted a great deal of attention recently by HCI (Human-Computer-Interaction) researchers, there exists no comparable repertoire of established design principles for tabletop applications yet. A major challenge is the effective support of collaboration on tabletop displays [13, 12, 3, 10], which requires consideration of specific guidelines [21]. Another central focus concentrates on interaction mechanisms that are especially designed for the characteristics of tabletop systems.

Reorientation of digital objects for instance occurs on tabletops far more often than on desktop computers as users can view the display from different positions around the table. Furthermore observational studies [6] have shown, that orientation is critical for comprehension of information, coordination of actions and team communication.

There exist various methods for handling orientation on tabletops including use of specialized hardware [23, 2, 8], object decoration [22], situation-based [9], environment-based [18, 24] and person-based [17] approaches. Amongst manual reorientation techniques a novel class of mechanisms [11, 7, 1] leveraging people's skills in manipulating physical objects by using physics simulation seems especially promising.

These physics-based techniques comply with the seamlessness design concept of Ishii et al. [4] which considers continuity with existing work practices and everyday skills as essential. The concept of seamlessness design can not only be applied to object rotation but to the handling and GUI of tabletop applications in general. We believe that the creation of "organic" [16] GUIs and interaction techniques that take advantage of our ability to anticipate behavior of physical objects according to their characteristics, surroundings and manipulations is a way to improve the usability of tabletop applications significantly.

We introduce physics simulation as a strategy to improve mental models of users. This technics produces functionality and behavior of a tabletop application which can be grasped immediately by an untrained user analog to a certain real-world physical setting. Fürtheron we describe several applications that use physics simulation and discuss related evaluation results. On this basis we discuss concepts of mental models and how physics simulation can help provoke appropiate models of software. Summing up we present an user-centered design method that by describing the iterative development of a physics-based application in

*Presented at the World Usability Day 2008, Hamburg using a framework called DynAmbient recently developed at the HAW Hamburg[19].

2. Physics-based applications

In tabletop applications physics simulation has so far primarily been used for rotating and translating objects via a single contact-point for input. Drag [11] computes the friction on objects, while RNT [7] uses a more simplistic approach in form of a simulated force to integrate rotation and translation.

An application that applies physics more elaborately for working with objects within a virtual workspace has been proposed by Agarawala et al. [1]. BumpTop, which is designed for pen-based touch interaction, utilizes a physics engine to create a dynamic working environment where objects can be manipulated in a realistic manner. Objects in BumpTop can be dragged and tossed around according to their physical characteristics like mass or friction. Their behavior resembles that of lightweight objects on a real tabletop. By adding physics and thus more realism Drag, RNT and BumpTop allow users to potentially employ interaction and work strategies from reality.

Kruger et al. [7] evaluated RNT by comparing it to a traditional-moded (TM) rotation mechanism called "corner to rotate". The results of their usability study show, that RNT is faster, more efficient, and as accurate as TM. Furthermore test participants stated, that RNT was very easy to use and required less effort to complete tasks as object translation and rotation could be carried out in one movement as opposed to TM where these interaction techniques were separated.

Unlike RNT, Drag turned out to be slower than TM object manipulation techniques when evaluated [11]. There seem to be two reasons for this result. While conceptually similar, Drag employs a more accurate physics model than RNT which made it difficult for users to adequately predict Drag's behavior while interacting with objects. Furthermore Mitchell used a mouse as input device while evaluation, whereas Kruger et al. conducted their tests with a touch screen. This means that participants could apply their training from performing traditional mode-based rotation via mouse input during Mitchell's evaluation tests, which yields from working with graphics applications or other standard software like MS PowerPoint®. This is an explanation for the performance advantages of traditional mode-based rotation in comparison to Drag since Mitchell also presumes "that direct input would enhance Drag"[11].

A qualitative user study of BumpTop conducted by Agarawala et al. [1] resulted in similar positive and encouraging feedback received for RNT. Users felt that interaction techniques like tossing were easy to discover and learn because the physics-based working environment of BumpTop allows leveraging of real-world experience. Participants also liked the software because the user interface allows playful, fun and satisfying interaction.

Summarizing the presented user evaluation tests conducted with RNT, Drag and BumpTop, physics-based application offer a number of advantages. However too accurate simulation of physics can effect users' experience in a negative way as demonstrated by Drag. Therefore developers must carefully choose to which degree physics simulation is beneficial. Agarawala et al. [1] propose for example a policy of "polite physics" where physics-simulation is restricted or turned off in certain situations. Direct copying of interaction techniques from reality for tasks like sorting or bulk object translation should employ the speed and accuracy of computer programs. During transfer from reality to computer developers should try to abstract in order to create an improved version of the original. Like this it is possible to combine the advantages of physics-based interaction techniques with the speed of computer supported work.

Generally physics-based interaction techniques are easy to learn and especially faster than traditional mode-based interaction mechanisms when used in combination with direct input devices like touch screens. Using physics simulation not only for interaction but also to provide dynamic workspaces where objects can be moved around reality-like seems to be the next logic step in developing intuitive tabletop user interfaces. How physics-based applications can help to achieve this aim by improving users' mental models of tabletop applications will be discussed in the next section.

3. Mental models of software applications

The concept of mental models has gained more attention in HCI during recent years. While interacting with computers and applications a user receives feedback from the system. This allows him or her to develop a mental representation (model) of how the system is functioning [5]. Sasse [20] states that a well-designed application and user interface will allow the user to develop an appropriate model of that system. This underlines the concept of Norman's design approach [14, 15], which assumes that humans develop mental models of systems based on their assumptions. In summary a mental model can be considered as a collection of assumptions a user has regarding the functionality of a system.

A central issue in GUI design results from the fact that the mental model of the developer differs from that of the user. This means that the application, which can be regarded as the manifestation of the developer's mental model, does not behave as the user would expect. The closeness between the mental of the developer and the user decides how intuitively an application can be handled.

Tognazzini [25] recommends the use of analogies and metaphors to assist developers in creating successful mental models. Sasse [20] defines an analogy as a explicit, referentially isomorphic mapping between objects in similar domains. A metaphor is a looser type of mapping that points out similarities between two domains or objects. Its primary function is the initiation of an active learning process.

According to Sasse's distinction, a physics-based application like BumpTop can be considered as an analogy since interaction techniques like tossing or grabbing and the physical characteristics of real-world objects were directly transfered to the program.

People develop mental models regarding the behavior of physical objects under influence of external forces during their lifetime. Because environmental conditions like gravity or air consistency are worldwide very similar one can assume, that certain mental models are shared by the majority of humankind. As a consequence developers as well as users probably possess a very similar mental model regarding the behavior of physical objects within a dynamic working environment provided by applications like BumpTop. By use of physics simulation, which allows the implementation of widespread mental models in form of real-world analogies, developers are able to create easily to grasp GUIs. The ability to close the gap between mental models of users and developers like this can be considered as key benefit of physics-based applications.

Due to the many advantages of physics simulation (see section 2) and the concept of mental models discussed in this section, we developed a physics-based tabletop application for touch input, that is based on the implementation of a real-world analogy and offers a dynamic working environment combined with realistic object handling. Our prototype is consequently called DynAmbient (from dynamic ambient). The next section describes the design of DynAmbient. First we present an application scenario from which we deduce a set of functional and non-functional requirements.

4. Application scenario and design guidelines

A central research focus of the UbiComp project at the HAW Hamburg are computer applications and hardware for emergency situations. Based on this application scenario we designed a tabletop application, that could be used in a control room to browse and categorize incoming photos and videos sent by rescue workers during a live rescue mission. We defined the following set of interaction techniques which are applicable to photos and videos while using the software:

- Translate, rotate and resize
- Translate and rotate simultaneously
- Categorize

According to the emergency scenario we determined several non-functional key requirements: object manipulation should be easy to learn, lightweight and cause low cognitive load since users operate in a stress situation.

Regarding the design of our GUI we followed the motto of computer pioneer Theodor Holm Nelson, who postulated that "a user interface should be so simple that a beginner in an emergency can understand it within ten seconds." The final user interface of this rescue control room table realized with the DynAmbient framework (see figure 1) resembles a billiard table seen from above: a rectangular horizontal plane with a hole on every long side surrounded by banks. Photos and videos can be moved on top of the plane within the embankment. Categorization of photos and videos is carried out by throwing objects into the holes whereas each hole represents a certain category (i.e information relevant for hospitals, policestations, firefighters . . .). The holes are positioned in the middle of the long sides and thus equally well accessible for left and right handers.

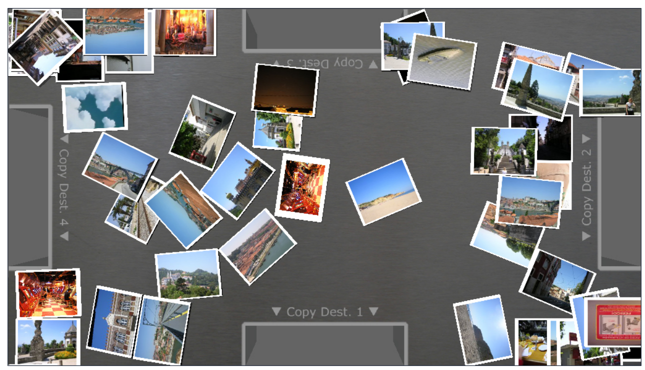

Figure 1: *Final rescue control room table GUI with four sorting holes labeled "Copy Dest. 1-4"*

Incoming photos and videos fall from above into the three-dimensional GUI in front of the user. Photo and video objects can be stacked (see upper left corner of figure 1), dragged and tossed around within the virtual workspace. While utilizing these mechanisms objects collide with each other and are eventually shoved away depending on the speed and momentum of the pushing object.

A photo or video object can be grabbed by "touching" it, i.e. the user puts down a finger or pen onto the touch screen over the object. The object is then attached by an invisible dampened spring to the cursor position and can be dragged around as long as the contact exists. This a common approach for physics based interaction, which is also used in BumpTop [1]. Reality-like grabbed objects behave according to the touch position: while performing the same movement a contact point at the edge of an object will result in a stronger rotation than one close to the object's center.

While the functionality and appearance of the rescue control center table was clear in general at the begin of development, the final gestalt of the user interface was created in a user-centered design process. The DynAmbient framework as the basis of a flexible system architecture allows the realization of different physical models within short time periods. This possibility was required for a development approach described in the next section.

5. System architecture

The manual implementation of physics algorithms can be costly and prone to error. Instead we recommend the integration of existing real-time physics engines used for computer game dynamics or scientific simulation, which simulate rigid body dynamics with sufficient accuracy. Physics engines allow the definition of three-dimensional objects along with their physical properties like mass or friction. They can furthermore simulate the effects of collisions and external forces depending on the characteristics of the affected objects.

Creating and configuring complex dynamic objects for physics engines through programming languages if often cumbersome, as the visual verification of every change usually requires a rebuild and restart of the program. Lengthy, complex and hard to understand passages of code may be another result of coded object definitions. To overcome these problems we propose a visual approach for modeling and testing dynamic scenes and objects for tabletop systems.

Ageia PhysX was used to implement physics-based interaction, rigid body dynamics and collision detection due to a vital product feature: Ageia provides plugins that allow the creation of dynamic objects by using 3D modeling packages like Autodesk®3ds Max®. The plugins are also able to export created dynamic objects to a proprietary XML file format, which can be imported and processed by the PhysX engine. This allows developers to model for example a cube within 3ds Max, configure its physical properties through the Ageia plugin, export it to XML and re-import it into an dynamic scene that is computed by the Ageia PhysX engine. The described workflow makes it possible to create dynamic objects without writing any code.

DynAmbient utilizes this mechanism to assemble its GUI dynamically: the application loads a XML file during startup that defines the physical gestalt of the virtual working environment that contains the videos and photo objects. The shape of the virtual working environment and hence the GUI can be changed by replacing the XML definition file. This concept enabled us to develop the GUI of DynAmbient test-driven in a user-centered design process. Modifications to the working environment were accomplished by using a 3d modeling package. The modified model was then exported and could be directly tested within DynAmbient. By following this approach we were able to improve the GUI steadily during each iteration.

The described system architecture of DynAmbient makes it possible to use a 3D modeling package as a toolbox for creating dynamic content. In summary our approach significantly shortens and simplifies the creation of tabletop applications that use physics simulation and enables also people who can not program to modify the behavior and look of the GUI. The next section presents the test-driven development process of DynAmbient's virtual working environment.

6. User-centered design process

6.1. Test method

Three versions of the GUI were produced in total during the design process of the rescue control room table using the DynAmbient framework. To evaluate the usability of the GUI various students of the Ubicomp Lab and ourselves tested the rescue control room table after each iteration. Tasks of the participants included translating and rotating photo and video objects. Furthermore users should throw several objects into the four sorting holes at the long sides of the working area. There was no time limit for the tests but users could experiment with the application as long as they wished. We asked participants subsequently to propose improvements regarding the GUI design.

Application tests were conducted on a 42 inch LCD with a resolution of 1920 x 1080 pixels powered by a Quad-Core Apple Mac Pro running Windows XP. An infrared touch screen from IR Touch, that was mounted in front of the LCD, was used for touch detection. As the system is not able to relate multiple touches to individual persons, only one person was interacting with the table at a given time.

6.2. Design iterations

The GUI of the rescue control room table resembles a billiard table, as described in section 4. This basic concept was the starting point for the gestalt of the application's working environment. As photo and video objects can be tossed around freely in this rescue control room table, it was necessary to surround the visible working area with banks. This keeps objects from exiting the GUI unintentionally.

Version 1. The first version of the physical model which represents the three-dimensional working area in the DynAmbient framework is shown at the top of figure 2. After importing the model as XML file into DynAmbient as described in section 5, the GUI of application looked like shown in figure 3. The central issue of this version is the integration of the sorting holes into the banks: the actual

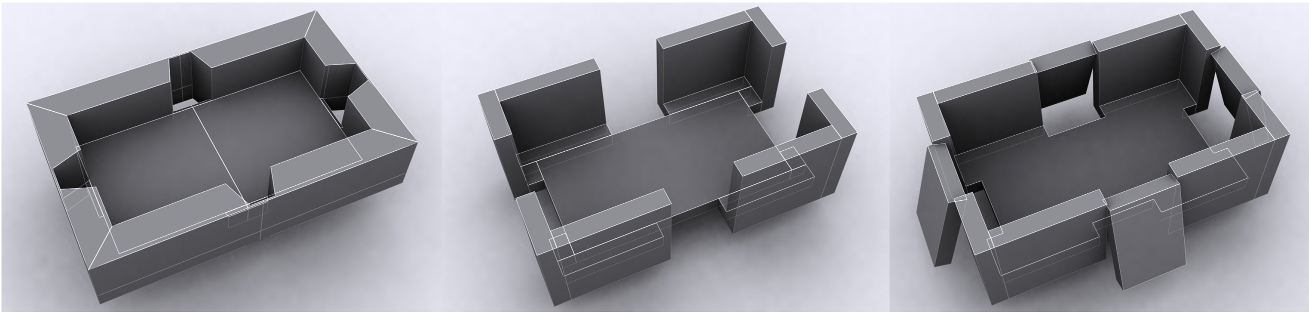

Figure 2: *Development stages of the three-dimensional workspace model used for physics simulation and GUI presentation*

working area is reduced by doing so as the GUI must show the banks to make the holes visible. Furthermore users stated that the photo and video objects were to small. The wood texture on the model was also considered as distracting.

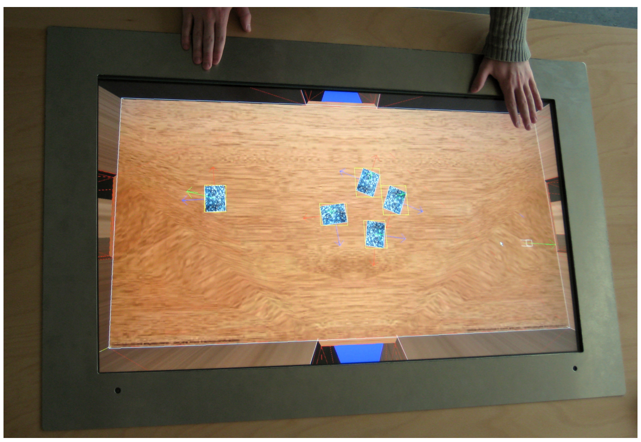

Figure 3: *First version of the rescue control room table GUI displayed on a touch screen we used for evaluation*

Version 2. Considering the proposed improvements the GUI was redesigned as shown in the middle of figure 2. The sorting holes were removed from the banks and integrated into the actual working area. As the photo and video objects were enlarged, the holes were resized as well. Additionally the workspace texture was replaced with a less distracting one, that is also used in the third and final GUI version (see figure 1). Despite the improvements user tests revealed several shortcomings of the second version. One problem results from the fact, that the plane that holds the workspace objects consists of multiple rectangular solids. Although there was no difference in elevation, photo and video objects tended to hang at touching points of the solids. This issue is probably related to rounding errors made by the physics engine. Another point of criticism was the lack of visual feedback users received when throwing photos or videos into sorting holes. If an object exits the working area through a hole with high velocity it will not start to fall until it has moved away horizontally for a certain distance.

Version 3. On basis of the described shortcomings of the second version the model of the GUI was improved. The plane in third and final version of the GUI model (see figure 2 bottom) consists of one piece. To improve the visual feedback for objects exiting the working area tilted banks were added behind the holes. This causes objects to rebound and fall down straight which allows users to see them while they disappear. Additionally a visual effect was added to emphasize the exit of objects as demonstrated in figure 4.

Figure 4: *Screenshot sequence that illustrates the visual effect which is triggered when objects leave the visible working area*

6.3. Discussion

All described modifications regarding the physical model of the user interface were conducted with a 3D modeling package. Using this visual design approach modifications to the GUI could be carried out and tested within in minutes (cf. figure 5). The combination of physics simulation, visual GUI design and a flexible system architecture that is capable of loading physical GUI models dynamically provides a promising platform for rapid prototyping of tabletop applications. The use of an editor which allowed graphical modeling of DynAmbient's working environment enabled us to take advantage of real-world experiences regarding manipulation of physical objects already at the de-

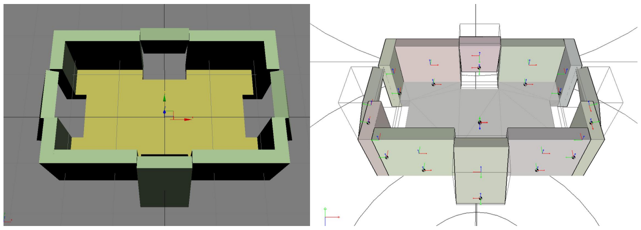

Figure 5: Design of a scene model with help of a graphics editor and its integration in the DynAbient framework

sign stage. Imaging for instance a container for collecting objects in the working area. This requirement can be directly implement into the application for instance by modeling a bowl that behaves like bowl in reality and is thus seamlessly integrated into working environment.

7. Conclusions and future work

The first contribution of this paper is the application of physics simulation to improve mental models of tabletop applications. We have presented how the transfer of real workspace and interaction analogies enables developers to create tabletop software which is intuitive and allows users to take advantage of their evolved dexterity with physical objects.

The second contribution is the introduction of a visual and test-driven design process for physics-based tabletop applications that aids interface designers in developing and modifying user interfaces rapidly. Therefore we present a tabletop application framework called DynAmbient, which is able to load its dynamic GUI from XML files. By applying our design approach to DynAmbient we could significantly improve the usability of the application according to feedback we received from test users.

The next stage of this project will be to explore the features physics simulation offer to model working environments and interaction techniques. Mimicking forces like magnetic attraction could be used to group objects while sticky surfaces would allow users to stick objects to vertical surfaces. Actual physics engines allow for instance the use of fluid and particle dynamics which can be used for novel and interesting effects. These features can be considered as tools that support the creativity to develop tabletop applications that are fun, easy to use and can be extended with less effort.

Actual work covers the combination of infrared-based multitouch technology with DynAmbient to recieve seamless interactive tables as shown in figure 6.

Another research topic actual worked on at the HAW Hamburg covers the distribution of applications like DynAmbient over several systems in a collaborative workspace setting. This would allow the creation of a physics-based virtual working environment that could be viewed from different perspectives by using power walls and tabletops. According to the orientation of displays physics simulation would facilitate realistic behavior of shared digital workspace objects. Thus users could pass objects between input devices for instance by throwing or flicking.

Figure 6: The multitouch tabletop of the HAW Hamburg

References

[1] A. Agarawala and R. Balakrishnan. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1283–1292, New York, NY, USA, 2006. ACM Press.

[2] M. Chen, S. J. Mountford, and A. Sellen. A study in interactive 3-d rotation using 2-d control devices. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 121–129, New York, NY, USA, 1988. ACM Press.

[3] O. Hilliges, L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz. Designing for collaborative creative problem solving. In *C&C '07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 137–146, New York, NY, USA, 2007. ACM.

[4] H. Ishii, M. Kobayashi, and K. Arita. Iterative design of seamless collaboration media. *Commun. ACM*, 37(8):83–97, 1994.

[5] J. A. Jacko and A. Sears, editors. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 2003.

[6] R. Kruger, S. Carpendale, S. D. Scott, and S. Greenberg. How people use orientation on tables: comprehension, coordination and communication. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 369–378, New York, NY, USA, 2003. ACM Press.

[7] R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 601–610, New York, NY, USA, 2005. ACM Press.

[8] J. Liu, D. Pinelle, S. Sallam, S. Subramanian, and C. Gutwin. Tnt: improved rotation and translation on digital tables. In *GI '06: Proceedings of the 2006 conference on Graphics interface*, pages 25–32, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[9] C. Magerkurth, R. Stenzel, and T. Prante. Stars – a ubiquitous computing platform for computer augmented tabletop games, 2003.

[10] M. Mandviwalla and L. Olfman. What do groups need? a proposed set of generic groupware requirements. *ACM Trans. Comput.-Hum. Interact.*, 1(3):245–268, 1994.

[11] G. D. Mitchell. Orientation on tabletop displays. M.sc. thesis, Simon Fraser University, Burnaby, British Columbia, Canada, 2003.

[12] M. R. Morris. Supporting effective interaction with tabletop groupware. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 55–56, Washington, DC, USA, 2006. IEEE Computer Society.

[13] M. R. Morris, A. Paepcke, and T. Winograd. Teamsearch: Comparing techniques for co-present collaborative search of digital media. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 97–104, Washington, DC, USA, 2006. IEEE Computer Society.

[14] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.

[15] D. A. Norman and S. W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1986.

[16] J. Rekimoto. Organic interaction technologies: from stone to skin. *Commun. ACM*, 51(6):38–44, 2008.

[17] J. Rekimoto and M. 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, New York, NY, USA, 1999. ACM Press.

[18] M. Ringel, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1441–1444, New York, NY, USA, 2004. ACM Press.

[19] P. Rossberger. Physikbasierte Interaktion in kollaborativen computergesttzten Umgebungen. Master's thesis, HAW Hamburg, Computer Science Department, 2008.

[20] M. A. Sasse. *Eliciting and Describing Users' Models of Computer Systems*. PhD thesis, University of Birmingham, 1997.

[21] S. Scott, K. Grant, and R. Mandryk. System guidelines for co-located collaborative work on a tabletop display, 2003.

[22] C. Shen, F. D. Vernier, C. Forlines, and M. 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, New York, NY, USA, 2004. ACM Press.

[23] K. Shoemake. Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the conference on Graphics interface '92*, pages 151–156, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[24] P. Tandler, T. Prante, C. Mller-Tomfelde, N. Streitz, and R. Steinmetz. Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 11–20, New York, NY, USA, 2001. ACM Press.

[25] B. Tognazzini. *TOG on Interface*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.