H. Wandke, S. Kain & D. Struve (Hrsg.): Mensch & Computer 2009: Grenzenlos frei!?
München: Oldenbourg Verlag, 2009, S. 203-212

# Iterative design of tabletop GUIs using physics simulation

Philipp Roßberger[1], Kai von Luck[2]

adesso AG[1], HAW Hamburg[2]

**Abstract**

Ubiquitous computing and ambient intelligence point out the issue of usable soft- and hardware. Gadgets without manuals are crucial to the success of the vision of many computers per person. When we talk about seamless interaction, the gap between mental models provoked by the computer interface and the software below is a main indicator for ease-of-use. In this paper we discuss a desktop metaphor based on physics simulation as an antipole to symbolic iconic desktops nowadays. A physics-based user interface combined with gestures and touch technology promises a smaller gap between mental model and computer system for certain application areas. Furthermore we present a user-centered design process for rapid development of physics-based applications, which was used to create a prototype on basis of our tabletop application framework DynAmbient. Our approach enabled us to improve the usability of the application 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 standard set of graphical elements (e.g. tabs, scroll bars) and interaction techniques (e.g. double-clicking, Drag-and-Drop), which are known by most users.

While digital direct-touch tabletops have attracted a great deal of attention recently by HCI 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 (Morris 2006, Morris et al. 2006, Hilliges et al. 2007), which requires consideration of specific guidelines (Scott et al. 2003). Another central focus addresses 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

because users can view the display from different positions around the table. Furthermore observational studies (Kruger et al. 2003) 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 (Shoemake 1992, Liu et al. 2006), object decoration (Shen et al. 2004), situation-based (Magerkurth et al. 2003), environment-based (Ringel et al. 2004, Tandler et al. 2001) and person-based (Rekimoto & Saitoh 1999) approaches.

Amongst manual reorientation techniques a novel class of mechanisms (Mitchell 2003, Kruger et al. 2005, Agarawala & Balakrishnan 2006) 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. (1994) 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” (Rekimoto 2008) GUIs and interaction techniques that take advantage of our ability to anticipate behaviour of physical objects according to their characteristics, surroundings and manipulations is a promising way to improve the usability of tabletop applications significantly.

We introduce physics simulation as a strategy to interact with tabletop applications and to improve mental models of users regarding application behavior. We believ that this approach helps in creating tabletop functionality and behavior that can be grasped quickly by untrained users through leveraging their experience regarding real-world physical settings. On this basis we discuss concepts of mental models and how physics simulation can help to provoke appropriate models of software.

Further on we describe a tabletop application that provides physics-based interaction utilizing a framework called DynAmbient, which was developed at the HAW Hamburg (Roßberger 2008). DynAmbient allows the integration of virtual physics-based tabletop workspace configurations designed visually with 3D editing software. This functionality enabled us to rapidly improve the GUI of our application based on feedback we received from users over several iterations.

# 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. The physics-based interaction mechanism Drag (Mitchell 2003) computes the friction on objects for manipulating tabletop items, while RNT (Kruger et al. 2005) uses a more simplistic approach in form of a simulated force to integrate rotation and translation.

An application that uses physics more elaborately for working with objects within a virtual workspace has been proposed by Agarawala & Balakrishnan (2006). BumpTop, which is designed for pen-based touch interaction, utilizes a physics engine to create a dynamic work-

ing 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 behaviour 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. (2005) 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 (Mitchell 2003). There seem to be two reasons for this result: while conceptually similar, Drag employs a more accurate physics model than RNT, which makes it difficult for users to adequately predict Drag's behaviour while interacting with objects.

Furthermore Mitchell used a mouse as input device during evaluation, whereas Kruger et al. conducted their tests on a touch screen. This means that participants could apply their experience of performing traditional mode-based rotation via mouse input during Mitchell's evaluation tests, which yields for example from working with graphics applications. 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” (Mitchell 2003, 99ff.).

A qualitative user study of BumpTop conducted by Agarawala & Balakrishnan (2006) resulted in similar positive and encouraging feedback as 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 provides playful, fun and satisfying interaction.

Summarizing the evaluation tests conducted with RNT, Drag and BumpTop, physics-based applications offer a number of advantages. However too accurate simulation of physics can affect users’ experience in a negative way as demonstrated by the evaluation of Drag. Therefore developers must carefully choose to which degree physics simulation is beneficial. Agarawala & Balakrishnan (2006) propose 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 manipulation should employ the speed and accuracy of computer programs. During transfer from reality to computer developers should abstract in order to create an improved version of the original. Like this it is possible to combine the advantages of physics-based interaction 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 appears 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 (Gentner & Stevens 1983, Rogers et al. 1992, Young 2008) 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 (Jacko & Sears 2003). Sasse (1997) 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 (Norman 1988, Norman & Draper 1986), which assumes that humans develop mental models of systems based on their assumptions.

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. How intuitively an application can be handled depends on how well the mental model of the developer and the user match.

Tognazzini (1992) recommends the use of analogies and metaphors to assist developers in creating successful mental models. Sasse (1997) defines an analogy as an explicit, referentially isomorphic mapping between objects in similar domains. A metaphor is a looser type of mapping which 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 transferred to the program.

People develop mental models regarding the behavior of physical objects under influence of external forces during their lifetime. 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 easy 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 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. The framework on which our prototype is built is consequently called DynAmbient (from dynamic ambient).

# 4 Design guidelines

Our physics-based prototype application allows users to browse and categorize photos and videos within a virtual working area. We defined the following set of interaction techniques applicable to photos and videos while using the software:

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

Furthermore we determined several non-functional key requirements: object manipulation should be easy to learn, lightweight and cause low cognitive load.

The final user interface of the application (cf. figure 1) realized with the DynAmbient framework resembles a billiard table seen from above: a rectangular horizontal plane with a hole on every long side surrounded by banks that keep objects from exiting the GUI unintentionally. 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. The holes are positioned in the middle of the long sides and thus equally well accessible for left and right handers. Incoming photos and videos fall from above into the three-dimensional GUI in front of the user and can be stacked (cf. upper left corner of figure 1), dragged and tossed around within the virtual workspace.


*Figure 1: Final GUI version including four sorting holes labeled "Copy Dest. (Destination) 1-4"*

While utilizing these mechanisms objects collide with each other and are eventually shoved away depending on the speed and momentum of pushing objects. 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 is a common approach for physics based interaction and is also used in BumpTop. 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 prototype application was clear in general at the beginning of development, the final gestalt of the GUI was created in a user-centered design process. The DynAmbient framework as the basis of a flexible system architecture allowed the realization of different physical models within short time periods.

# 5 System architecture and development workflow

The manual implementation of physics algorithms can be costly and error-prone. 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 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 as described in the next paragraph.

The physics engine Ageia PhysX was used to implement physics-based interaction, rigid body dynamics and collision detection due to a vital product feature: Ageia provides plug-ins that allow creation of dynamic objects using 3D modeling software like Autodesk 3ds Max. Created dynamic objects can be exported to a proprietary XML file format the PhysX engine is able to import and process. This allows developers to model objects like e.g. a cube within 3ds Max, configure its physical properties through the Ageia plug-in, export it to XML and re-import it into a dynamic scene that is computed by the Ageia PhysX engine. The described workflow enables developers to create dynamic objects without writing any code.

DynAmbient utilizes this mechanism to assemble GUIs dynamically: the framework loads a XML file during start up which defines the physical gestalt of the virtual working environment containing video 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 our tabletop application 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 tested using our tabletop application prototype. By following this approach we were able to improve the GUI steadily during each iteration.

The described functionality of DynAmbient allows to use 3D modeling software as toolbox for dynamic content creation. 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 behaviour and look of the GUI. The next section presents the test-driven development process of the virtual working environment provided by our application prototype.

# 6 User-centered design process

Three versions of the GUI were produced in total during the design process of our tabletop application. To evaluate the usability of the GUI various students of the UbiComp Lab and ourselves tested the tabletop application after each iteration. Tasks of the participants included moving and rotating photo and video items. Furthermore users were asked to throw several objects into the four sorting holes at the long sides of the working area. 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, mounted in front of the LCD, was used for touch detection. As the system is unable to relate multiple touches to individual persons, only one person was interacting with the table at a given time.

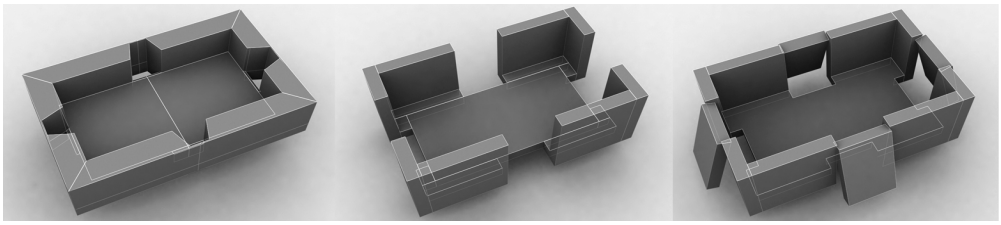

*Figure 2: Development stages of the three-dimensional workspace model used for physics simulation and GUI presentation*

*Version 1.* The first version of the physical model representing the three-dimensional workspace of our application is shown at left of figure 2. After importing the workspace model in XML-format using DynAmbient as described in section 5, the GUI of our application looked as displayed in figure 3. The central issue of this version was the integration of the sorting holes into the banks: the actual working area is reduced by doing so as the GUI must include the banks to make the holes visible. Furthermore users stated that the photo and video objects were too small. The wood texture on the model was also considered as distracting.

*Figure 3: First version of our application's GUI displayed on the 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 enlarged, removed from the banks and integrated into the actual working area. The photo and video items were scaled up as well. Additionally the workspace texture was replaced with a less distracting one that is also used in the third and final GUI version (cf. figure 1). Despite the improvements user tests revealed several shortcomings of the second version. One problem resulted from the segmentation of the workspace area that holds the photos and videos in multiple rectangular solids. Though there was no difference in elevation, photo and video objects tended to get stuck at solids' edges. This issue was probably related to rounding errors made by the physics engine. Another shortcoming of this application version was the lack of visual feedback when throwing photos or videos into sorting holes. Users noted that items exiting the workspace with high velocity, could not be seen falling, as items moved away horizontally for a certain distance.

*Version 3*. Based on the described shortcomings we redesigned our GUI model. The workspace of the third and final version of the GUI model (cf. figure 2, right) 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 allowing users to see them disappear. A visual sparkle effect was also added to emphasize exiting items.

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. The combination of physics simulation, visual GUI design and a flexible system architecture capable of loading physical GUI models provides a promising platform for rapid prototyping of tabletop applications.

Using a visual editor for virtual workspace modelling enables us to take advantage of real-world experiences regarding physical objects already at the design stage. Imagine users would request containers for collecting photo or video items within the virtual workspace. Using our approach we can add this feature simply by modelling a bowl and load it via DynAmbient into the application, where it would behave as users would expect from reality.

# 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 described how the transfer of real workspace and interaction analogies enables developers to create tabletop software that is intuitive as it 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 presented a tabletop application framework called DynAmbient, which is able to load its dynamic GUI from XML files. By applying our design approach we could improve the usability of our tabletop application prototype according to feedback received from users.

The next stage of this research will include intensive evaluation of our design approach and the DynAmbient framework in the context of the new "Living Place Hamburg" project at the HAW Hamburg, which will be fully functional in 2010. Besides the usability tests a comparison of our tabletop system with the Microsoft Surface table, arriving shortly in our lab, is in preparation.

**References**

Agarawala A. & Balakrishnan R. (2006). Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In: *CHI '06*. NY: ACM Press, pp. 1283–1292.

Gentner D. & Stevens A. L. (editors) (1983). *Mental models*. NJ: Lawrence Erlbaum Associates.

Hilliges O., Terrenghi L., Boring S., Kim D., Richter H. & Butz A (2007). Designing for collaborative creative problem solving. In: *C&C '07*. NY: ACM Press, pp. 137–146.

Ishii H., Kobayashi M. & Arita K (1994). Iterative design of seamless collaboration media. *Comm. ACM*, 37(8), 83–97.

Jacko J. A. & Sears A. (editors) (2003). *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. NJ: Lawrence Erlbaum Associates.

Johnson-Laird P. N. (1983). *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness*. Cambridge: Cambridge University Press.

Kruger R., Carpendale S., Scott S. D. & Greenberg S. (2003). How people use orientation on tables: comprehension, coordination and communication. In: *GROUP '03*. NY: ACM Press, pp. 369–378

Kruger R., Carpendale S., Scott S. D. & Tang A. (2005). Fluid integration of rotation and translation. In: *CHI '05*. NY: ACM Press, pp. 601–610.

Liu J., Pinelle D., Sallam S., Subramanian S. & Gutwin C. (2006). Tnt: improved rotation and translation on digital tables. In: *GI '06*. Toronto: Canadian Information Processing Society, pp. 25–32.

Magerkurth C., Stenzel R. & Prante T. (2003). Stars – a ubiquitous computing platform for computer augmented tabletop games. In: *UBICOMP '03*. Berlin: Springer, pp. 267–268.

Mitchell G. D. (2003). *Orientation on tabletop displays*. M.sc. thesis. Burnaby: S. Fraser University.

Morris M. R. (2006). Supporting effective interaction with tabletop groupware. In: *TABLETOP '06*. Washington, USA: IEEE Computer Society, pp. 55–56.

Morris M. R., Paepcke A. & Winograd T. (2006). Teamsearch: Comparing techniques for co-present collaborative search of digital media. In: *TABLETOP '06*. Washington, DC, USA: IEEE Computer Society, pages 97–104.

Norman D. A. (1988). *The Psychology of Everyday Things*. NY: Basic Books.

Norman D. A. & Draper S. W. (1986). *User Centered System Design: New Perspectives on Human-Computer Interaction*. NJ: Lawrence Erlbaum Associates.

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

Rekimoto J. & Saitoh M. (1999). Augmented surfaces: a spatially continuous work space for hybrid computing environments. In: *CHI '99*. NY: ACM Press, pp. 378–385.

Ringel M., Ryall K., Shen C., Forlines C. & Vernier F. (2004). Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In: *CHI '04*. NY: ACM Press, pages 1441–1444.

Rogers Y., Bibby P. & Rutherford A. (editors) (1992). *Models in the Mind: Theory, Perspective and Application*. London: Academic Press.

Roßberger P. (2008). *Physikbasierte Interaktion in kollaborativen computergestützten Umgebungen*. M.sc. thesis. Hamburg: HAW Hamburg, CS Department.

Sasse M. A. (1997). *Eliciting and Describing Users' Models of Computer Systems*. PhD thesis. Birmingham: University of Birmingham.

Scott S., Grant K. & Mandryk R. (2003). System guidelines for co-located collaborative work on a tabletop display. In: *ECSCW '03*. Norwell, MA, USA: Kluwer Academic Publishers.

Shen C., Vernier F. D., Forlines C. & Ringel M. (2004). Diamondspin: an extensible toolkit for around-the-table interaction. In: *CHI '04*. NY. ACM Press, pp. 167–174.

Shoemake K. (1992). Arcball: a user interface for specifying threedimensional orientation using a mouse. In: *Proc. of the conf. on Graphics interface '92*. San Francisco: Morgan Kaufmann Publishers, pp. 151–156.

Tandler P., Prante T., Müller-Tomfelde C., Streitz N. & Steinmetz R. (2001). Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. In: *UIST '01*. NY: ACM Press, pp. 11–20.

Tognazzini B. (1992). *TOG on Interface*. Boston: Addison-Wesley Longman Publishing.

Young I. (2008). *Mental Models: Aligning design strategy with human behaviour*. NY: Rosenfeld Media.

## Contact

Philipp Roßberger, adesso AG, Rotherstr. 19, 10245 Berlin, E-Mail: rossberger@adesso.de