

TinkerSheets: Using Paper Forms to Control and Visualize Tangible Simulations

Guillaume Zufferey, Patrick Jermann, Aurélien Lucchi and Pierre Dillenbourg
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
guillaume.zufferey, patrick.jermann, aurelien.lucchi and pierre.dillenbourg@epfl.ch

ABSTRACT

This paper describes TinkerSheets, a paper-based interface to tangible simulations. The proposed interface combines the advantages of form-based input and paper. Form-based input allows to set an arbitrary number of parameters. Using paper as a medium for the interface keeps the interaction modality consistently physical. TinkerSheets are also used as an output screen to display summarized information about the simulation. A user study conducted in an authentic context shows how the characteristics of the interface shape real world usage. We also describe how the affordances of this control and visualization interface support the co-design of interaction with end-users.

Author Keywords

Tangible User Interfaces, Paper-based Interfaces, Tabletop computing, Vocational Training

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

INTRODUCTION

Tangible User Interfaces [7, 10] have received a sustained interest in the HCI community for the last ten years. Many researchers have invested efforts in developing technologies and studying new ways of interaction made possible by this approach. Among the large scope of applications that can be efficiently supported by tangible interfaces, the ones with a strong spatial dimension particularly benefit from physical interactions. It comes with no surprise that early implementations of TUIs have taken place in fields like architecture, map exploration or urban planning [2, 8, 17].

The very close mapping between physical objects and their digital counterparts offered by TUIs allowed researchers to develop intuitive interfaces which rely on the implicit knowledge people have about real-world objects. Using Augmented Reality techniques, feedback can be given directly on top of the physical objects and create immersive experiences. An

ACM COPYRIGHT NOTICE. Copyright 2009 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept., ACM, Inc., fax +1 (212) 869-0481, or permissions@acm.org.

interesting opportunity offered by these systems is the ability to run simulations based on the spatial arrangement of physical objects. For example URP [18], an urban planning application, simulates the impact of the position of small-scale models of buildings on different parameters like shadows and pedestrians' behavior. Illuminating Clay [15] is another example where the topography of a small-scale landscape can be modified by the users and used to simulate phenomena like water flows and land erosion.

In these pioneering systems, the construction of the model to be simulated was the main focus of tangible interaction (users manipulated buildings and landscapes). The control of the simulation was restricted to a predefined set of parameters. These were set through dedicated tangibles (e.g. transparent wand in URP) or through predefined buttons operated via a traditional Graphical User Interface (e.g. Analysis Function Thumbnails controlled via a mouse in Illuminating Clay). The GUI approach offers the advantage to be able to control a larger number of parameters but comes to the expense of a second mode of interaction in complement to the physical interaction with the model. The dedicated tangible tools approach has the advantage to stay in the physical and spatial metaphor, but has a practical limitation with regard to the number of different parameters that can be controlled.

The output of these simulations typically consists of two types of information. Spatially dependent information (e.g. the shadow projected by a building in URP or the amount of erosion in Illuminating Clay) is projected directly on the tangibles or around them on top of the table. Summarized information is independent from a physical anchor and is represented by numerical values or graphical representations on a separate display.

In this contribution, we propose to use paper as a way to customize parameters and display the summarized output of a tabletop simulation. Paper is a generic tangible medium which can carry any kind of representation (from concrete drawings and photographs to abstract formulas). As a basis for a computer interface, paper allows to represent any graphical control (sliders, buttons, etc.). Using paper along with tangibles combines the best of two worlds, physical interaction with the model as well as the ability to represent an arbitrary number of parameters by well known GUI components.

TinkerSheets act as a paper-based interface to tabletop sim-

ulations. This control and visualization interface has been developed within the context of the TinkerTable (Figure 1), a tabletop learning environment for apprentices in logistics. The system's main functionality is the construction and simulation of a warehouse built with small-scale shelves. TinkerSheets solve two problems in this context: setting simulation parameters and visualizing summarized outputs in an integrated way, i.e. on the same physical space which is used to build the warehouse model.

This paper is organized as follows. After presenting related work and the context which motivated our work, we describe the TinkerSheets concept and its implementation. We then discuss the characteristics of this approach and present the results of a user study that we conducted with six groups of apprentices.

RELATED WORK

An early work on spatial applications providing users with a TUI is Tangible Geospace, a prototype application for The metaDESK [17], an environment designed to explore the use of physical affordances. Users manipulate small-scale models of the MIT campus buildings to navigate on the map of the university. Some other tools allow users to get different views of the map (e.g. the active lens, which displays a three-dimensional view of the campus with its buildings in perspective). Other examples of pioneering work in this field are Urp and Illuminating Clay, described in the introduction of this paper. The objective of these systems was to explore the new interaction techniques that were made possible by a physical way of interacting with computers.

Envisionment and Discovery Collaboratory [1] is an environment that has been developed to study shared understanding and informed participation. One application of the EDC has been developed to support citizens in designing a transportation system for their neighborhood. Users interact with the system through two electronic whiteboards: a horizontal whiteboard, on which users manipulate physical objects that are recognized by the system, and a vertical whiteboard (reflection space) which presents information related to the task. Some specific tools, implemented as tactile icons on the horizontal whiteboard can be used to query information (e.g. walking distance tool). The vertical whiteboard allows participants to set some simulation parameters by filling web-based surveys. This environment was also designed to support end-users' customization through Visual AgenTalk [16], an end-user programming language.

Researchers in the HCI community have been interested for a long time in using the unique affordances of paper [13] to create user interfaces. Among the early works aiming at bridging the digital and the real world with paper, we can cite the DigitalDesk [19], a system that augments real paper on a standard office desk. Paper documents are recognized with a ceiling camera and users can interact with them by pointing with their fingers. This environment implemented several tools, like a calculator that could interpret handwritten equations, a translation system and a paper-based paint program. Johnson and colleagues [11] define the concept

of a paper user interface and use paper forms as an interface to control photocopier jobs. PaperButtons [14] were developed as a response to the challenge of adding new features to applications and allow users to interact with a system by touching buttons on a piece of paper. The approach has been developed in the context of the Palette system [12] that allows people to control electronic slideshows through a paper-based interface. PaperButtons were implemented with an electronic tagging technology that is embedded in the paper and has to be worn by the users on their fingers. VoodooSketch [3] mixes real ink and physical interfaces and allows users to create shortcuts to an application's functionalities using sketches and specific physical controls that are attached on an interactive surface.

We extend this previous work in the way we integrate both paper-based interface and a TUI into our environment, providing users with tangible objects to interact with the core model of a simulation and paper forms to control an arbitrary number of parameters and visualize data. We build on the similarity of these two interaction modalities to offer a consistent user interface based on physical manipulations.



Figure 1. The TinkerTable.

CONTEXT

TinkerTable

TinkerSheets have been developed in the context of the TinkerTable, a tabletop learning environment for apprentices in logistics (figure 1). The system is made of a 2m by 1.5m table covered with whiteboard material and a gallows carrying a camera, a projector and a mirror. The purpose of the mirror is to augment the projection area by increasing the distance between the projector and the tabletop while minimizing the height of the gallows. The camera is used to detect the position of objects using ARTag, a fiducial markers tracking library [5]. Apprentices use the system to build small-scale warehouses by placing wooden shelves, metallic pillars and a variety of cardboard rectangles (representing loading docks and administrative areas) on the table. Augmentations are projected directly on top of the objects and show additional information like security zones around

shelves or accessibility of shelves for storage (i.e. shelves that are too close to each other or to a wall are not accessible with a forklift). Another important aspect of the environment is that it allows to run simulations. The movements of forklifts and goods in and out of the warehouse are projected directly on top of the small scale model.

Realism was the main design criteria that was followed during the development of the system. A field study conducted at the beginning of the project showed that apprentices in logistics had difficulties to understand theoretical concepts presented in schools (apprentices in Switzerland go to school one day per week and work in a company the rest of the time). The concepts addressed in school are too often unrelated to the daily practice of apprentices. The TinkerTable aims at reducing the gap between theory and practice by providing apprentices with the opportunity to apply knowledge in authentic problem situations.

Towards TinkerSheets

The concept of TinkerSheets was not present in the early prototypes of the TinkerTable but emerged as the need for parameters control and data visualizations increased. We review here the different solutions that were implemented and that progressively led us to the design of the TinkerSheets.

The first version of the TinkerTable that was tested with apprentices only implemented basic functionalities. Teachers could choose among two scenarios (warehouse layout and movement optimization) that were subdivided in different phases that had to be followed in sequence. Selecting a scenario and moving from one phase to another was done by showing a specific tag to the camera. Each scenario thus consisted in several tags that were printed on independent sheets of paper and stapled together. This approach was not satisfying. For the teachers, showing a tag to a camera was not an intuitive way of interacting with the system and they would often get confused about what they were supposed to do to move to the next phase. The concept of a scenario was not appropriate either, as they preferred to organize their course in a more flexible way which did not necessarily follow the predefined phases.

In the next prototype, we implemented a screen-based hierarchical menu that was operated by a keyboard. This menu was displayed as an overlay over the projection of augmentations which consequently were partly hidden when the menu was open. Several options were provided, allowing teachers to choose among several visualizations, control basic parameters (e.g. speed) and run simulations (e.g. start, run, pause). Trial sessions were conducted with two teachers and showed that this approach was not appropriate. To use the menu, teachers were obliged to move to the computer, press a key to open it and then navigate to the option they wanted to set. This was disrupting the interaction with the class and teachers often got confused with the hierarchical organization of the menu, forgetting about the exact position of a given option. Moreover, teachers often forgot to close the menu after using it, thus leaving the projection on the tabletop partly occluded. Another unwanted effect of this computer-based

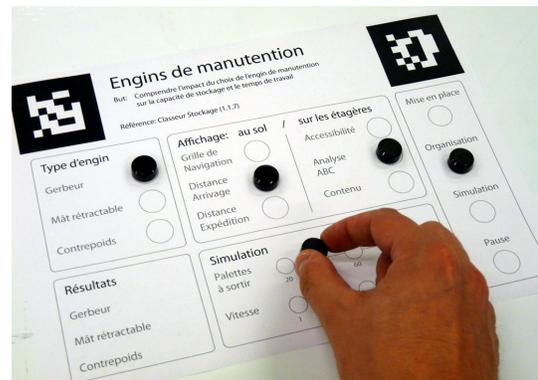


Figure 2. An example of a TinkerSheet with a user setting parameters using circular magnets.

menu is that it was implicitly reserved for teacher use. Apprentices did not dare to use the computer to change a parameter as it is perceived as a teacher tool that they are not allowed to use. The usability issues could have been corrected with a better design of the menu, but its negative impact on the interaction convinced us to develop another approach.

We tried to overcome these limitations with a third prototype that implemented a menu controlled through fingertips. This menu used the infrastructure of the TinkerTable to draw an augmented icon-based menu on a (blank) sheet of paper. Users could select an icon by pointing to it with their finger, which was detected by the system's camera. This augmented menu implemented the same set of functionalities as the keyboard-based menu. With this approach the system became mainly operated by the apprentices who felt more at ease with this interaction modality than their teachers. Our implementation suffered from several flaws due to the technical setting of the TinkerTable: using the image from the ceiling camera as unique input for controlling the menu did not allow the system to detect when a user was touching the table surface and resulted in too many false-positive detections. Teachers did not trust the system and were thus not willing to interact with it.

These successive design iterations allowed us to observe the main difficulties that our users were facing with different input modalities. As the needs for a way to control parameters and visualize simulation data were increasing, we progressively defined the main requirements of the interface that would fit the constraints of the hardware setting of the TinkerTable and allow teachers and apprentices to interact with these more abstract aspects of the environment. This eventually led us to the design of the TinkerSheets which are described in the next section.

TINKERSHEETS

Description

Figure 2 shows an example of a TinkerSheet, an interactive paper-based form that allows users to set different parameters and query information presented in textual or graphical form. Four basic primitives are currently implemented: buttons and sliders for input and text and graphical areas for

feedback. Users interact with a sheet in a variety of ways, using either physical tokens (e.g. magnets) or a pen by drawing a circle on an input area. This variety is possible because the system will consider any dark object or drawing which is circular and within a given size range as a possible input. The additional textual information printed on a TinkerSheet (e.g. labels and descriptions) is ignored by the system. Its purpose is to inform the users about the parameters or values associated with each interactive part of the sheet. Output areas are augmented with text or drawings whenever the sheet is put on the TinkerTable surface and updated in real-time.

Not all the parameters can be set at any time, and a feedback is provided to indicate the values that are currently set. This feedback takes the form of a green cross on buttons which associated value is currently set and is done in a similar fashion for sliders. If a user tries to set a parameter which is not available (e.g. it is impossible to change the size of a warehouse during a simulation), a red cross is drawn on top of the selected value. The same feedback is given if two different values are set on a slider or if two buttons which values are mutually-exclusive are selected.

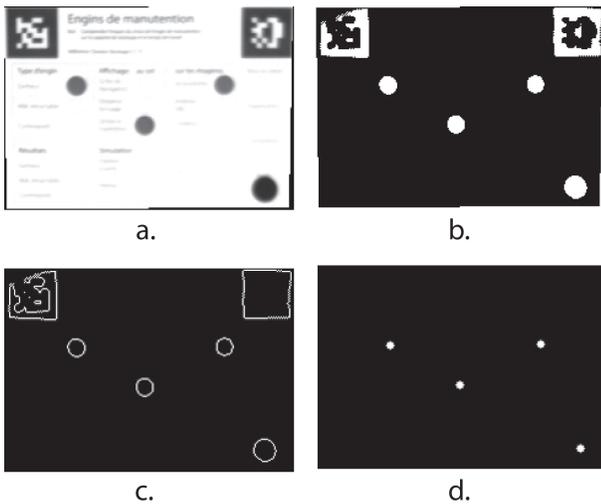


Figure 3. The four phases of the algorithm detecting input on a TinkerSheet. a) source image, b) after adaptive thresholding, c) after contours extraction, d) after selection of valid contours, four inputs are detected.

Implementation

An important objective of the design of the TinkerSheets was to build on the technology available on the TinkerTable, thus limiting the input to vision only. This was achieved through the use of ARTag for locating sheets on the table surface and computer vision techniques for input detection. The algorithm is the following (Figure 3). When a sheet is detected, an adaptive threshold [4] is applied on the image to segment dark objects and drawings from the sheet's background. The blobs of the resulting image are extracted and their shape is analyzed to discard the ones that do not lie in a given size range and are not circular. The remaining blobs are considered as valid inputs and their position is checked against each input area of the sheet. If the distance is below a certain threshold the input area becomes activated and its associated value is set.

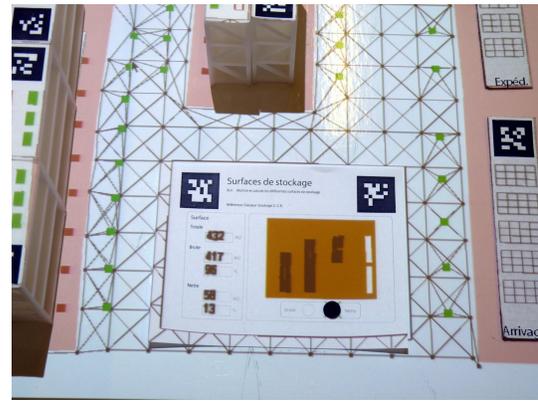


Figure 4. The surface sheet used during the training sessions. It displays a scaled and colored view of the warehouse which illustrates the concepts of effective and raw storage surfaces.

Architecture

TinkerSheets are developed as a set of paper sheets, each containing a subset of parameters and visualizations. In a complex field like logistics, the amount of sheets necessary for covering the whole set of customizable values and available information becomes too large for users to quickly find the one they need. To avoid this problem, we organized the sheets in three different layers: master sheets, companion sheets and raw sheets.

Master sheets control a subset of related parameters that correspond to a certain activity or, in the context of the TinkerTable, to a given logistics concept. Only one master sheet can be used at a given time. In addition to the parameters it allows to control, a master sheet sets a default value for all remaining parameters when it is first detected by the system. This ensures a consistent and predictable state of the simulation at the beginning of each activity. Each master sheet also proposes a standard palette of buttons which allows users to control the simulation state: a layout phase, where the warehouse is designed, and a simulation phase, where the design is evaluated.

Companion sheets are meant to be used together with a master sheet and extend the controls available on the master sheet. They differ from the master sheet by the fact that they do not provide access to the high level simulation controls (start, pause) and provide controls which are thematically related. Companion sheets also embed visualizations that are meaningful in the context set by a master sheet. Ideally, teachers prepare activities by designing companion sheets which illustrate the ideas they want to address.

Raw sheets provide a high level of flexibility to the system as they embed the full list of parameters and visualizations available in the system. Contrary to the master and companion sheets, they are not related to a specific domain-related activity and simply correspond to the parameters that can be set in the underlying simulation. The expected usage scenario for raw sheets is that teachers use them in situations where they did not anticipate that a given parameter or visu-

alization could be useful.

This division in three layers allows us to create groups of sheets composed of a master sheet and several companion sheets. Each group corresponds to a given class in the apprenticeship of logisticians. Providing the teachers with the ability to control the full set of parameters through raw sheets potentially gives them unlimited possibilities for customization. However, our current experience suggests that it might be more efficient to use raw sheets for the co-development of new master and companion sheets. They allow developers and end-users to test different scenarios and design sheets in an interactive way.

USER STUDY

The TinkerSheets were tested with six groups of 2 or 3 apprentices in the second year of their apprenticeship (N=15) and two teachers during training sessions in a professional school. A smaller version of the TinkerTable was used during these sessions. It consists of a projector and camera mounted in a metal casing which is suspended above a regular classroom table by an aluminum gooseneck. Shelves, pillars and docks are scaled at 1:48. The functionality of the small version is identical to the large version presented above. This was the first time that the apprentices worked with the TinkerTable and the TinkerSheets. The teachers already knew the system and participated in the design of the TinkerSheets used during these sessions, but were using it for the first time for teaching.

The objective of the training sessions was to teach the partitioning of a warehouse's total surface (the surface of the building) into raw storage surface (the total surface minus staircases, administrative areas, service rooms) and effective storage surface (raw storage surface minus alleys, i.e. the surface occupied by shelves). A secondary objective was to illustrate the influence of the type of forklift on the effective storage surface available in a warehouse: a larger forklift needs wider alleys, and hence leads to a smaller effective storage surface. General instructions and three TinkerSheets were available to support these pedagogical objectives. The design of the sheets was discussed and agreed upon in close collaboration with the teachers during a meeting that preceded the training sessions.

The *master sheet* (Figure 2) allowed to switch among three simulation states (layout, organization and simulation), set the duration and speed of the simulation (expressed as the number of pallets which should be delivered and as a time acceleration factor), define the type of augmentation displayed on the shelves (accessibility of a storage space represented by green and red rectangles, content of a shelf represented by a 3 by 3 matrix of colored rectangles), define the type of augmentation displayed on the floor of the warehouse (simple monochrome navigation grid, distance from the loading or expedition dock as a green to red color gradation), and choose the type of forklift used during simulation (three types of increasing size which require different alley width). Two *companion sheets* were used together with this sheet: the *surface sheet* and the *process sheet*.

The *surface sheet* (Figure 4) was designed to illustrate the three types of surfaces numerically and graphically. The numerical output consisted of the total, raw and effective surface expressed as square meters and percentages. The graphical output illustrated the surfaces through the projection of a reduced and colored version of the warehouse. Users were able to switch between the visualization of raw and effective areas. It is worth noting that the graphical representation exactly matched the schema which is used in the corresponding course material.

The *process sheet* displayed numerical information about the content of the warehouse (number of shelves, number of available storage spaces and percentage occupied) as well as about the current state of the simulation (total number of pallets to move, number of pallets moved since the beginning of the simulation and average time to move a pallet).

The lesson consisted of a planning phase, an implementation phase, followed by series of testing phases (for two groups) and finally by an exploration phase (for four groups).

1. Planning phase (45 minutes): Apprentices were instructed to draw a blueprint of a warehouse on paper. They were given the scale of the drawing (1:48), as well as the size of the shelves, docks, and administrative area. They first had to decide where to place pillars, loading docks and an administrative area. The drawing of these elements defined the architectural constraints. The next planning step consisted of drawing shelves in a way to maximize the effective storage surface.
2. Implementation phase (5 minutes): Apprentices put their blueprint under the projector. They then precisely placed the tangible docks and shelves on the locations they had drawn. The length and width of the administrative area were purposely constrained by the teachers to match the space necessary to place the master sheet which allows to control the simulation.
3. Testing phase (5 minutes): Once all shelves were placed on the blueprint, apprentices used the master sheet to start a simulation of the warehouse (Figure 5). This was accomplished by moving a token on the master sheet from *layout* to *organization* to *simulation*. As soon as the token was recognized, the simulated forklifts started moving in the warehouse. Apprentices inquired about the quality of their layout by placing the surface and process sheets above or next to the master sheet. They then wrote down the result of their initial design and moved on to test the next apprentice's design (two groups of 5 apprentices compared their designs like this and did not move on to the exploration phase).
4. Exploration phase (30 minutes): Four smaller groups of 2 to 3 apprentices implemented alternative warehouse designs by tinkering with the spatial arrangement of shelves. Each alternative design was tested by starting a new simulation and checking results with the surface and process TinkerSheets. When they thought to have reached their best possible result, they used a pencil to fix the solution on the blueprint by outlining the edges of the shelves.



Figure 5. Apprentices at work. Companion sheets are placed next to the simulation.

RESULTS

We present the results of the user study we conducted through a list of characteristics of the TinkerSheets that we observed during the trial session and its preparation with teachers. We also discuss how these characteristics shape a real use situation.

Intuitive The paper form metaphor underlying the development of the TinkerSheets appeared to ease the use of the simulation. Teachers were able to explain to their apprentices how to use the system without the help of the researchers. After a short demonstration, apprentices started to build a warehouse and handled TinkerSheets without difficulty. Because of the small scale of the sheets (A6 paper) some users used the tip of a pencil to move tokens. Another strategy to avoid moving the sheets while manipulating tokens consisted in pinning down the sheets with one or two fingers. Some apprentices carefully avoided occlusions by holding the sheets down between the tags. Others however, did not pay much attention to the tags and tried to interact with a sheet partially occluded by another. We need to implement a more obvious feedback which signals whether a sheet is recognized by the system or not. The current implementation draws a rectangle around active sheets and removes other augmentations (e.g. navigation grid) from the sheet's surface. A possible solution consists of using a more obvious color code to signal recognition by the system (e.g. a thick green frame drawn around active sheets).

Visible Because TinkerSheets can be spatially arranged on the side of the interactive surface, the interaction space of the software is available for visual inspection. Teachers placed the master and companion sheets next to the projector at the outset of the lesson. While progressing through the exercise and following the general instructions, apprentices were querying the TinkerSheets for the one which would allow them to answer particular questions. They chose the *surface* sheet to measure surfaces and the *process* sheet to measure warehouse efficiency. In comparison with a menu-based system which only dis-



Figure 6. Teachers and one researcher redesigning the master and companion sheets.

plays one level of information at the time, the layout of TinkerSheets on the table makes all possible interactions available for inspection. Retrieval is simply a matter of picking up the sheet and placing it under the light. A valuable side effect is that the simulation interface can be explained and understood without a computer or augmentations.

Annotable TinkerSheets can be cheaply reproduced by any standard copier or printer. Therefore, TinkerSheets could be considered as single use artifacts and could be used to take notes. For instance, we left some blank room on the master sheet to allow users to write down the results of the simulation for later discussion. Indeed, one apprentice suggested that it would be easier to directly write down results on the sheets, but the teacher instructed him to write them down on a separate piece of paper. As we provided teachers with a limited set of printed sheets to be used during the sessions, they did not allow apprentices to annotate their sheets. We could also speculate that TinkerSheets acquire a higher value than traditional paper because they allow to control a complex application, and are perceived as somewhat "magic" by the users. We plan to give an electronic version of the sheets to the teachers who will then be able to print enough copies of sheets beforehand and will thus not be afraid of not having enough sheets.

Co-design The nature of TinkerSheets, mixing augmented and printed forms of information can be used as a bridge between the physical and the digital worlds. During the lesson preparation meeting which took place at the cafeteria, teachers numbered and annotated the TinkerSheets to structure the lesson and imagine how and when they would use each of the sheets (Figure 6). The annotation of control sheets allowed researchers and teachers to co-design the interface and the usage scenario. The TinkerSheets play a role similar to card board interface mockups currently used in software design. The difference between the paper mockup and the final paper-based interface is however much smaller than between a paper mockup and a traditional graphical user interface. TinkerSheets serve

as boundary objects [9] for teachers, researchers and programmers.



Figure 7. Carefully stacking a Companion sheet on top of the Master sheet.

Stackable Because the settings of a TinkerSheet are persistent, they can be stacked on top of each other. Occlusions don't erase settings. The master sheet stayed on the table for the whole duration of the simulation. Depending on the layout of the warehouse, apprentices either stacked companion sheets on top of the master sheet (if the alleys were too narrow, see Figure 7) or placed them side by side. Because the tokens used on the master sheet were made of lightweight rubber material, stacking sometimes disturbed the arrangement of tokens. As a workaround, apprentices held the companion sheets 2 cm above the master sheet to avoid displacing the tokens. On the large-scale TinkerTable this problem does not appear as we use small circular magnets as tokens and that sheets are placed on a metallic tablet. The problem could also be avoided by replacing tokens with pen annotations. Companion sheets were used either to monitor the state of the simulation or to briefly lookup the value of a parameter. Monitoring requires that the sheet is placed on the table. Lookup is short lived and can be implemented by flashing the sheet under the lamp.

Stacking saves space on the limited interactive surface, but limits the visibility of the TinkerSheets, giving access to the top sheet only. Users thus have to make a trade-off between storage area and parallel access to information. It is worth noting how apprentices tried to circumvent this limitation by keeping sheets in the off-line space and bringing them in for short checks only. Some groups put *companion sheets* on top of shelves during simulations to have a continuous feedback. While flashing sheets to get feedback is satisfactory during a layout phase, where the focus is on the physical arrangement of shelves in the storage space, access to parallel sources of information given by TinkerSheets is more important during simulation phases, where the focus is on the evaluation of dynamic data which is best understood as absolute values or chart representations. To support this need, we are now

working on the possibility to save physical layouts. Users can then reload them at a smaller scale (after removing the physical objects), saving space for placing several TinkerSheets on the interactive surface.

Embeddable and Integrated The teaching material used in the professional schools is divided into a theoretical section and accompanying practice sheets. TinkerSheets are designed to be embedded into the material as supplementary practice sheets. The integration of TinkerSheets in the "official" binder offers a strong argument when convincing new teachers of the relevance of the system for teaching. From the point of view of instructional design, TinkerSheets allow for the design of custom interfaces that combine an arbitrary set of controls onto a master sheet. This facilitates the integration of a new technology into an existing curriculum. From the many design meetings conducted with teachers, we have found out that it is not sufficient to propose a general purpose interface to the simulation ("see how you can control everything") and expect that teachers will combine the possibilities on the fly. The integration into teaching has to be carefully planned and designed. TinkerSheets embody the translation from pedagogical objectives into simulation functionalities.

CONCLUSION

We have described the implementation and characteristics of a paper-based interface as a complementary input modality for a tangible tabletop simulation. Tangible simulations involve two related activities, building a mockup model and controlling a simulation. The control and configuration of the simulation has been implemented in previous systems either through ad hoc Graphical User Interfaces or by dedicated and specialized tangible controls. The TinkerSheets that we presented in this contribution allow users to control the simulation through physical manipulation (moving tokens or checking form fields with a pencil) without restrictions concerning the number of parameters that can be controlled. TinkerSheets are metaphors [6] of paper or electronic forms which are used in everyday life (e.g. administration, choosing the features of computer bought online, answering a questionnaire). As a result, the interaction modality is consistently tangible across the main task (building a warehouse) and the secondary tasks (configuring the simulation and visualizing output).

The user study that we have conducted illustrated the usability of the system, but above all its usefulness in a real context of use. Companion sheets allowed apprentices to efficiently inquire about the properties of the warehouse they built and simulated. Master sheets were central resources in the discussions between system designers and teachers. They acted as boundary objects which facilitated the translation of pedagogical objectives into simulation controls. The possibility to physically and pedagogically embed TinkerSheets into the training material used by the logistics training community will be essential for the acceptance of the simulation in future schools which did not participate in the initial developments.

Because of the high flexibility of TinkerSheets (stackable,

copiable, annotable), the final design will come from the users who will eventually adapt the interface to their needs and possibilities. The underlying software architecture allows us to envisage the development of a TinkerSheet generator as a next development step. The generator would enable users to create custom sheets by combining their choice of controls and visualizations. We think that providing users with an intuitive way to control a potentially large set of parameters and visualize the associated data will foster the development of applications based on tangible interactions that can be used in real-world situations.

ACKNOWLEDGEMENTS

This project is part of Dual-T, a leading house on technologies for vocational training, funded by the Swiss Department of Economic Affairs. We would like to thank the teachers, André Ryser, Jacques Kurzo, Hans Erni and Boris Seiler for their strong support and willingness to innovate.

REFERENCES

1. E. Arias, H. Eden, G. Fischer, A. Gorman, and E. Scharff. Transcending the individual human mind—creating shared understanding through collaborative design. *ACM Trans. Comput.-Hum. Interact.*, 7(1):84–113, 2000.
2. E. Arias, H. Eden, and G. Fisher. Enhancing communication, facilitating shared understanding, and creating better artifacts by integrating physical and computational media for design. In *DIS '97: Proceedings of the conference on Designing interactive systems*, pages 1–12, New York, NY, USA, 1997. ACM Press.
3. F. Block, M. Haller, H. Gellersen, C. Gutwin, and M. Billinghamurst. Voodoosketch: extending interactive surfaces with adaptable interface palettes. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 55–58, New York, NY, USA, 2008. ACM.
4. D. Bradley and G. Roth. Adaptive thresholding using integral image. *Journal of Graphic Tools*, 12(2):13–21, 2007.
5. M. Fiala. Artag revision 1. a fiducial marker system using digital techniques. Technical report, National Research Council of Canada, November 2004.
6. K. P. Fishkin. A taxonomy for and analysis of tangible interfaces. *Personal Ubiquitous Comput.*, 8(5):347–358, 2004.
7. G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of CHI 1995*, pages 442–449. ACM Press/Addison-Wesley Publishing Co., 1995.
8. M. Fjeld, F. Voorhorst, M. Bichsel, K. Lauche, M. Rauterberg, and H. Krueger. Exploring brick-based navigation and composition in an augmented reality. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 102–116, London, UK, 1999. Springer-Verlag.
9. K. Henderson. Flexible Sketches and Inflexible Data Bases: Visual Communication, Conscriptioin Devices, and Boundary Objects in Design Engineering. *Science Technology Human Values*, 16(4):448–473, 1991.
10. H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *CHI '97*, pages 234–241, 1997.
11. W. Johnson, H. Jellinek, J. Leigh Klotz, R. Rao, and S. K. Card. Bridging the paper and electronic worlds: the paper user interface. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 507–512, New York, NY, USA, 1993. ACM.
12. L. Nelson, S. Ichimura, E. R. Pedersen, and L. Adams. Palette: a paper interface for giving presentations. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 354–361, New York, NY, USA, 1999. ACM.
13. K. O'Hara and A. Sellen. A comparison of reading paper and on-line documents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 335–342, New York, NY, USA, 1997. ACM.
14. E. R. Pedersen, T. Sokoler, and L. Nelson. Paperbuttons: expanding a tangible user interface. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems*, pages 216–223, New York, NY, USA, 2000. ACM.
15. B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 355–362, New York, NY, USA, 2002. ACM.
16. A. Repenning and T. Sumner. Agentsheets: A medium for creating domain-oriented visual languages. *Computer*, 28(3):17–25, 1995.
17. B. Ullmer and H. Ishii. The metadesk: models and prototypes for tangible user interfaces. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223–232, New York, NY, USA, 1997. ACM Press.
18. J. Underkoffler and H. Ishii. Urp: a luminous-tangible workbench for urban planning and design. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 386–393, New York, NY, USA, 1999. ACM Press.
19. P. Wellner. Interacting with paper on the digitaldesk. *Commun. ACM*, 36(7):87–96, 1993.