

Papier-Mâché: Toolkit support for tangible interaction

Scott R. Klemmer

Group for User Interface Research

Computer Science Division

University of California

Berkeley, CA 94720-1776, USA

srk@cs.berkeley.edu

ABSTRACT

Studies of office workers, web designers, and oral historians have found that even in the digital age (and sometimes because of it), we are using paper more and more. The paper-saturated office is not a failing of digital technology; it is a validation of our expertise with the physical world. We use paper, and writing surfaces more generally, in their myriad forms: books, notepads, whiteboards, Post-it notes, and diagrams. We use these physical artifacts to read, take notes, design, edit, and plan. Here, we present Papier-Mâché, a toolkit for building tangible interfaces using computer vision, RFID tags, and bar-codes. The toolkit provides a high-level event model for working with these technologies. This abstraction layer also facilitates technology portability: for example an application can be prototyped with computer vision, and later deployed with RFID tags. We also present the user-centered design methods we employed to build this toolkit.

INTRODUCTION

Beginning with Wellner's Digital Desk [14], researchers have explored how to better integrate the physical and electronic worlds. Currently, researchers around the world are building Tangible User Interfaces (TUIs) [5].

The *Myth of the Paperless Office* [13] describes field research Sellen and Harper undertook over several years at a variety of companies. Their central thesis is that while paper is often viewed as inefficient and passé, in actuality it is a nuanced, efficient, highly effective technology. The authors are not asserting that "paper is better than digital" or vice versa, but that our naïve utopia of the paperless office is mistaken. Digital technologies certainly change paper practices, but rarely make paper irrelevant.

There are excellent reasons for researchers to embrace, not abandon, our interactions with everyday objects in the physical world. Paper and other everyday objects:

- Allow users to continue their familiar work practices, yielding safer interfaces [8].
- Are persistent when technology fails, and thereby more robust [9].
- Enable more lightweight interaction.
- Afford for fluid collocated collaboration.
- Are higher resolution, and easier to read than current electronic displays.

However, "tangible computing is of interest precisely because it is not purely physical" [2]. Researchers have

electronically augmented paper and other everyday objects to offer:

- An interactive history of an evolving physical artifact.
- Collaboration among physically distributed groups.
- Enhanced reading.
- Associative physical links to electronic resources.
- Physical handles for fluid editing of electronic media.
- Automated workflow actions.

While the research community has shown the substantial benefits of tangible interaction, these UIs are currently very difficult and time consuming to build, and the required technology expertise limits the development community. The difficulty of technology development and lack of appropriate interaction abstractions make designing different variations of an application and performing comparative evaluations unrealistic. In each of the 24 research systems we studied [6], at least one member of the project team was an expert in the sensing technology used.

Contrast this with GUIs, where developers are generally experts in the domain of the application, not in raster-graphics manipulation. The difficulties involved in building tangible interfaces today echo the experiences of the GUI community of twenty years ago. In 1990, Myers and Rosson found that 48% of code and 50% of development time was devoted to the user interface. One of the earliest GUI toolkits, MacApp, reduced Apple's development time by a factor of four or five [12]. We believe that similar reductions in development time, with corresponding increase in software reliability and technology portability, can be achieved by a toolkit supporting tangible interaction.

GUI tools have been so successful because, "tools help reduce the amount of code that programmers need to produce when creating a user interface, and they allow user interfaces to be created more quickly. This, in turn, enables more rapid prototyping and, therefore, more iterations of iterative design that is a crucial component of achieving high quality user interfaces" [11].

Papier-Mâché provides toolkit support for physical input, enabling developers to (1) quickly build TUIs and (2) change the underlying sensing technologies with minimal code changes. Papier-Mâché also enables further research by providing an open-source development platform.

Papier-Mâché supports vision, bar-code, and RFID tag input (see Figure 1). Supporting vision-based UIs is possibly the most important contribution of the toolkit, as vision is the

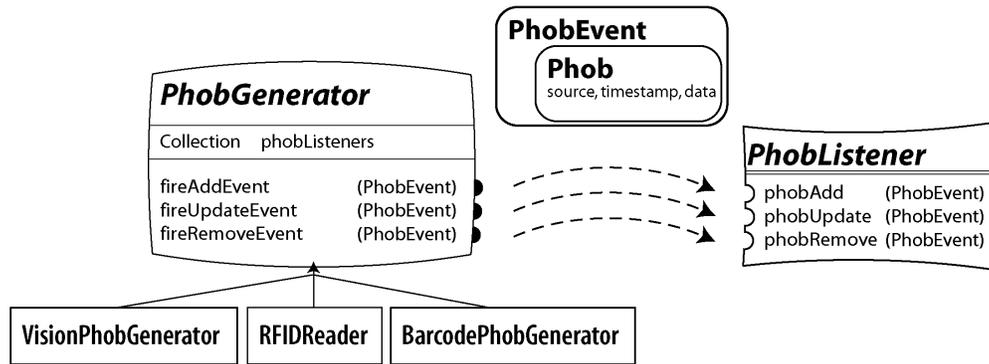


Figure 1. A *Phob* represents a physical object (e.g., RFID tag, bar-code, vision-recognized object). *PhobGenerators* create *Phobs*, notifying *PhobListeners* via *PhobEvents*. An *AssociationMap* maps *Phobs* to actions (e.g., play an audio clip, open a web page, add a graphical component to the screen).

most flexible, powerful, and unwieldy of these technologies. Tool support in this domain is minimal: while software libraries such as Java Advanced Imaging (JAI) and OpenCV [1] aid vision developers in image processing tasks, there are no tools that enable developers to work with vision-based UIs at the application/event level.

APPLICATION SPACE

To better understand the domain of tangible interfaces, we conducted a literature survey of existing systems, looking specifically for systems employing paper and other “everyday” objects (as opposed to mechatronic UIs). The twenty-four representative applications fall into four broad categories: *spatial*, *topological*, *associative*, and *forms*.

Spatial applications include augmented walls, whiteboards, and tables, used for collaboratively creating or interacting with information in a Cartesian plane. Collaborage is a *spatial* application: it is “a collaborative collage of physically represented information on a surface that is connected with electronic information, such as a physical In/Out board connected to a people-locator database” [10].

Topological applications employ physical objects as avatars (e.g., for airplanes, media files, and PowerPoint slides). Arranging these objects determines the behavior of the corresponding electronic system. Paper Flight Strips [8] is a *topological* application: the system augments the flight controllers current work practice of using paper flight strips by capturing and displaying information to the controllers.

With *associative* applications, physical objects serve as an index or “physical hyperlink” to digital media. Durrell Bishop’s Marble Answering Machine [5] is an *associative* application. An answering machine deposits a physical marble (with an embedded RFID tag) each time a message is left. To play a message, one picks up the marble and drops it into an indentation in the machine.

Forms applications, such as the Paper PDA [4], provide batch processing of paper interactions. The Paper PDA is a set of paper templates for a day-planner. Users work with the planner in a traditional manner, then scan or fax the pages to initiate electronically synchronize handwritten changes with the electronic data. Synchronization also executes actions such as sending of handwritten email.

These twenty-four applications share much functionality with each other, including:

- Physical input for arranging electronic content.
- Physical input for invoking actions (e.g., media access).
- Electronic capture of physical structures.
- Coordinating physical input and graphical output in a geo-referenced manner.
- An add, update, remove event structure.

This taxonomy omits haptic and mechatronic user interfaces, as these UIs are not the focus of our research.

MOTIVATING APPLICATIONS

As part of our user-centered design process, we conducted interview surveys with nine researchers who have built tangible interfaces. These researchers employed a variety of sensing techniques including vision, several RF technologies, capacitive field sensing, and bar-codes. We will report on the details of this study elsewhere. Here, we summarize the findings from 3 researchers that used vision.

Researcher #1 has a PhD in computer vision, and was the vision expert on an interdisciplinary research team. His team built a wall-scale, *spatial* TUI. Their driving user experience beliefs were:

- “People don’t want to learn or deal with formidable technology.”
- “They’re torn between their physical and electronic lives, and constantly trying work-arounds.”
- “Technology should make things more calm, not more daunting.”

They used vision because, “it gives you information at a distance without a lot of hassle, wires, and instrumentation all over the place. It puts all the smarts in one device and instrumentation is limited. It also is possible to retrofit existing spaces.” His main frustration with using vision was that “getting down and dirty with the pixels” was difficult and time-consuming.

Researcher #2 built a wall-scale, *spatial* TUI augmented with speech and gesture recognition. For rapid implementation, the system was originally implemented with a SMART Board. Later, this was replaced with vision

for two reasons: 1) SMART Boards are expensive and bulky, while cameras are inexpensive and small. 2) SMART Boards provide single-input of (x, y), while vision offers a much richer input space. This vision task is exactly the kind of task that Papier-Mâché can support.

Researcher #6 built a desktop *forms* UI incorporating image capture. His main frustration was that, “The real-time aspects of camera interfacing were probably the hardest.” This system was designed iteratively over a number of years. At each iteration, user feedback encouraged making the interaction techniques more lightweight and calmer. This echoed the experiences of the other two researchers, as well as our own group’s research.

All three researchers mentioned the difficulty of working with cameras. #2 avoided them initially. #1 plowed through anyway, lamenting “it’s not always worth it to live at the bleeding edge of technology. ... Make sure you have a very good reason if you choose to work on a problem whose solution requires pushing more than one envelope at most.”

Myers, Hudson, and Pausch [11] point to rapid prototyping as a central advantage of tool support. Vision is an excellent technology for rapid prototyping of interactive systems. It is a highly flexible, completely software configurable sensor. There are many applications where the final system may be built using custom hardware, but the prototypes are built with vision. An example application built with Papier-Mâché is the Physical Macros class project. Papier-Mâché’s high-level support for computer vision enabled these students to rapidly prototype their system in three weeks. Neither of the students had vision experience. Given the tight time schedule, this system would not have been possible otherwise.

HIGH LEVEL DESIGN

This dissertation introduces high-level events for TUIs. This is especially important for vision-based UIs, as building a vision system requires a huge amount of domain expertise. In contemporary vision-based UIs, the information that the vision system provides to the application tends to be ad-hoc and written in technology-centered terms. Papier-Mâché introduces an application-centric rather than recognizer-centric event mechanism. Additionally, an event layer abstraction enables application developers to write UIs that are less tightly coupled with the underlying technology. For example, bar-codes and RFID tags both offer an ID and a reader number. While in many cases these two technologies could be used interchangeably, without a layer of abstraction, this modularity can be difficult to achieve.

A well-specified event API enables a separation of concerns between algorithms development and interface development. Of concern to the application developer is, “When a user places a light-bulb on the work surface, display visual output based on the bulb’s location and orientation.” A savvy application developer is also likely interested in mediation techniques if an object was recognized with low confidence. These issues live above the event layer. The details of the recognition algorithms

are hidden by the event layer. The techniques for object detection can change completely, and the event API (and thereby, application code) does not need to be changed.

Papier-Mâché is open-source Java software written using the Java Media Framework (JMF) and JAI APIs. JMF supports any camera with a standard driver, from inexpensive webcams to high-quality 1394 cameras. The contribution of our research is not in the domain of vision algorithms. Our contribution is a novel set of APIs for building interactive systems and a toolkit that employs well-known algorithms that are effective for this task.

Motivating scenario: building The Designers’ Outpost

We introduce the software architecture with a scenario of how Papier-Mâché would help a developer build The Designers’ Outpost [3, 7], a tool that supports information design for the web. Web designers use pens, paper, walls, and tables for explaining, developing, and communicating ideas during the early phases of design. Outpost embraces and extends this paper-based practice through a large electronic wall with a tangible user interface.

Users have the same fundamental capabilities in Outpost as with paper and a whiteboard. One can create new pages by writing on Post-it notes, add them to the electronic wall and organize a site by physically moving Post-it notes around on the board. Paper in the physical world becomes an input device for the electronic world. A camera mounted inside the board captures the location of notes, detecting when notes are added, removed, or moved.

Outpost is a *spatial* TUI. To build Outpost, a developer would begin by instantiating a camera source for the internal camera. She would add an `EventFilter` that filtered for Post-it note sized objects as a listener to the camera’s event stream. She would then instantiate a `VisualAnalogueFactory` as a listener to the note filter, and perhaps add a `MotionlessTranslator` to filter out hand motions. She would have the factory create geo-referenced visual forms with a faint yellow shadow, acting as feedback that the notes had been recognized. With just this code, the developer has built a system that tracks Post-it notes placed on the screen and presents visual feedback. To support users tapping on notes, she could add standard Java mouse listeners to the visual forms the factory creates. To extend the system with a remote awareness shadow [3], she would add a second `EventFilter` to the camera. This would filter for person-sized objects. This filter would have a corresponding factory that created the outline shadow using the outline pixel data from the events source.

EARLY APPLICATIONS BUILT WITH PAPIER-MÂCHÉ

Two groups in the Spring 2003 offering of UC Berkeley’s graduate HCI class built projects using Papier-Mâché.

Physical Macros is a *topological* TUI. The students were interested in researching a physical interface to macro programming environments such as “actions” in Adobe Photoshop. The system provides paper function blocks that can be composed. As the user composes functions, the

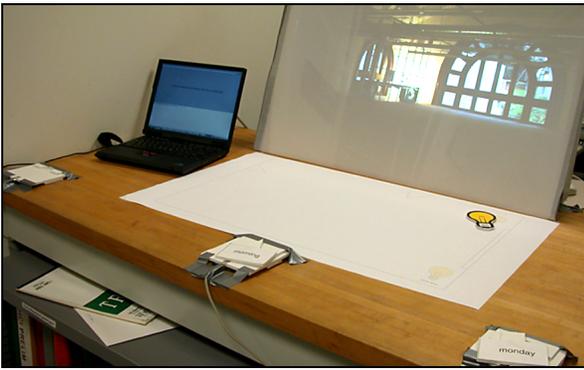


Figure 2. The SiteView home automation control system.

graphical display is updated accordingly. A set of functions can be saved in a save block for later reuse.

When the students wrote their system, Papier-Mâché had no visual analogue facilities. Looking through their code, we found that geo-referenced event handling and graphical presentation was a substantial portion of the code. Reflecting on this, we realized that many of our inspiring applications, including Outpost, also require this feature. For this reason, we introduced the visual analogue classes.

SiteView is a *spatial* UI presenting physical interaction techniques for end-user control of home automation systems. On a floor plan of a room, users create rules by manipulating physical icons representing conditions and actions. The system provides feedback about how rules will affect the environment by projecting photographs onto a vertical display. It employs a ceiling mounted camera and three RFID sensors. The camera tracks the objects whose locations the application needs. The RFID reader is used when only the presence or absence of the physical icon is necessary.

SiteView uses `EventFilters` to find the location and orientation of the thermostat and the light bulbs on the floor-plan. The thermostat is distinguished by size, the bulbs are distinguished by size and color. In general, the system worked well, but human hands were occasionally picked up. This inspired our addition of a `MotionlessTranslator`. With this in place, human hands do not seem to interfere with recognition. SiteView is roughly 3000 lines of code (including comments); of this only about 30 lines access Papier-Mâché code; we consider this a tremendous success. (Outpost, built on top of OpenCV, required several thousand lines of code to achieve a comparable amount of vision functionality.)

CONCLUSIONS

We have presented Papier-Mâché, a toolkit supporting tangible interaction. Our event abstractions shield developers from having to get “down and dirty” working with technologies such as cameras and RFID tag readers. Vision applications can be built with real camera input, or they can be prototyped with a Wizard of Oz image slide

show. The two class projects built using our system show how vision-based applications can be built by developers without vision experience. We are currently building an event layer that has some independence from the underlying technology, enabling developers to experiment with this technology. For example, a developer may be unsure whether vision or RFID is a more appropriate technology for their application. Our goal is to make switching easy. Papier-Mâché is open-source Java software, available for download at: <http://guir.berkeley.edu/papier-mache>.

REFERENCES

- 1 Bradski, G. Open Source Computer Vision Library, 2001. <http://www.intel.com/research/mrl/research/opencv>
- 2 Dourish, P., *Where the action is: the foundations of embodied interaction*. Cambridge, Mass.: MIT Press. 233 pp, 2001.
- 3 Everitt, K.M., S.R. Klemmer, *et al.*, Two Worlds Apart: Bridging the Gap Between Physical and Virtual Media for Distributed Design Collaboration. CHI, Human Factors in Computing Systems, *CHI Letters*, 2003. **5**(1): p. 553–60.
- 4 Heiner, J.M., S.E. Hudson, and K. Tanaka, Linking and messaging from real paper in the paper PDA. UIST: User Interface Software and Technology, *CHI Letters*, 1999. **1**(2): p. 179–86.
- 5 Ishii, H. and B. Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. Proc. *Human Factors in Computing Systems: CHI '97*. pp. 234–241, 1997.
- 6 Klemmer, S. Papier-Mâché: Toolkit support for tangible interaction, 2002. <http://guir.berkeley.edu/projects/papier-mache/quals.pdf>
- 7 Klemmer, S.R., M.W. Newman, *et al.*, The Designers’ Outpost: A Tangible Interface for Collaborative Web Site Design. UIST: User Interface Software and Technology, *CHI Letters*, 2001. **3**(2): p. 1–10.
- 8 Mackay, W.E., Is Paper Safer? The Role of Paper Flight Strips in Air Traffic Control. *ACM/Transactions on Computer-Human Interaction*, 1999. **6**(4): p. 311–340.
- 9 McGee, D.R., P.R. Cohen, *et al.*, Comparing paper and tangible, multimodal tools. *CHI Letters*, 2002. **4**(1): p. 407–414.
- 10 Moran, T.P., E. Saund, *et al.*, Design and Technology for Collaborative: Collaborative Collages of Information on Physical Walls. UIST: User Interface Software and Technology, *CHI Letters*, 1999. **1**(2): p. 197–206.
- 11 Myers, B., S.E. Hudson, and R. Pausch, Past, Present, and Future of User Interface Software Tools. *ACM Transactions on Computer-Human Interaction*, 2000. **7**(1): p. 3–28.
- 12 Myers, B. and M.B. Rosson. Survey on User Interface Programming. Proc. *Human Factors in Computing Systems: CHI '92*. Monterey, CA: ACM Press. pp. 195–202, May 3–7, 1992.
- 13 Sellen, A.J. and R. Harper, *The myth of the paperless office*. Cambridge, Mass.: MIT Press. 231 pp, 2002.
- 14 Wellner, P., Interacting with Paper on the DigitalDesk, *Communications of the ACM*, vol. 36(7): pp. 87–96, 1993.