Bridging the Gap: Fluidly Connecting Paper Notecards with Digital Representations for Story/Task-Based Planning

Tom Hurlbutt
Stanford University HCI Group
Computer Science Department
Stanford, CA 94305-9035 USA
hurlbutt@cs.stanford.edu

Scott R. Klemmer
Stanford University HCI Group
Computer Science Department
Stanford, CA 94305-9035 USA
srk@cs.stanford.edu

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.
ACM 1-59593-298-4/06/0004.

Abstract
Programmers use both paper and digital artifacts to aid in the process of software planning. This paper presents a prototype of a system that uses digital pen technology to integrate paper notecards and digital task plan representations, allowing programmers to utilize the affordances provided by both techniques. Through an ethnography of programmers who practice planning using both physical and digital artifacts, we discovered common actions performed by the programmers included card creation, card augmentation, card combining, and scheduling of card for completion. We designed interaction techniques to facilitate these actions and conducted a usability study (n=10) to evaluate the techniques. Through the study, we discovered that the initial prototype provided both positive and negative experiences for the user, providing insightful design implications for the future.

Keywords
Story/task-based planning, paper user interfaces, tangible user interfaces, Extreme Programming, digital pen input.

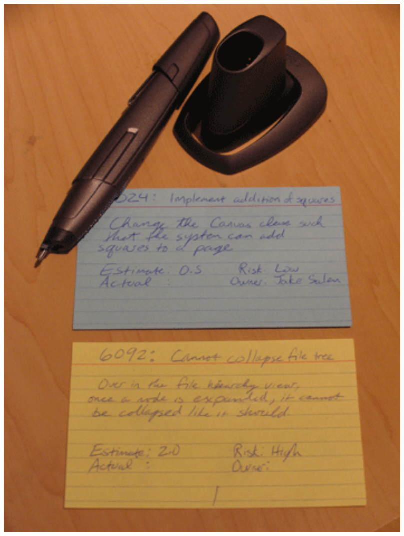

Figure 1. Using pen-based input techniques, paper notecards can be easily digitized, allowing users to utilize the affordances provided by both physical and digital representations of the information.

ACM Classification Keywords

H.5.1: Multimedia Information Systems—*artificial, augmented, and virtual realities*. H.5.2: User Interfaces— *input devices and strategies; interaction styles; prototyping*.

Introduction

There are multiple ways in which programmers can approach the planning phase of software development. One such approach is story-based or task-based planning, where the programmers create different user scenarios about how the program will be used by the target users, and then create tasks to realize the story. Programmers who practice story-based planning most commonly use two different interfaces through which they can record stories and tasks: paper notecards and project management software. Each approach provides its own affordances and benefits. Notecards are light, flexible, constantly visible, portable, disposable, and inexpensive. Project management software allows for multiple views of information, calculations on data, and information sharing among various programmers. While each method provides powerful, distinct benefits to programmers who practice story-based planning, no system exists that connects the two together. Programmers who wish to digitize their physical artifacts must do so by hand in form-based interfaces, creating a significant bottleneck. With a system that would make the transition from physical to digital fluid, we believe that these programmers will be able to work more effectively and efficiently than ever before.

Ethnography

To gain insights into how these notecards are used by the programmers, we observed a local Bay Area company that practices story/task based programming. The ethnography was in conjunction with another study examining Extreme Programming (XP) [1]; the observations occurred over a period of twelve weeks with each visit lasting 1.5 to 3 hours. The company observed uses notecards in their planning process. The cards come in four colors: green, blue, yellow, and red, each with its own meaning. Green cards are *story* cards on which the programmers create different scenarios of how they believe users will interact with the program. Blue cards are *task* cards (generated from the green *story* cards) on which the programmers take the stories that are created and break them down into specific parts that need to be implemented in order to realize the story card. Yellow cards are *bug* cards, and they are generated independently when the programmers discover errors in the program during testing. Finally, red cards are *customer emergency* cards that are generated whenever the company receives a complaint from one of their customers about any serious problem with the software. Each card has an identification number, a title, a brief description, and additional metadata written on it (see Figure 2). These cards are placed on a card board that is visible to the entire team of programmers.

In addition to notecards, the company also uses a project management software system in order to keep track of all of the cards. The identification number on the card accesses the associated information in the database. In addition to the data contained on the card, the project management system includes other information, usually additional notes from the programmers about the card that they want to share with the team. Once the card is created, the programmers then manually enter that information again into the database. Any update to the card by a

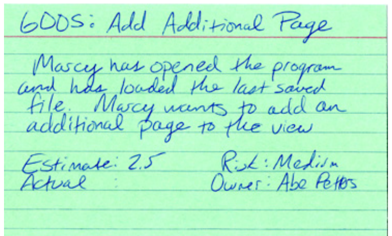

Figure 2. Cards used in story/task-based planning contain information such as the task description, title, and current owner, in addition to other metadata. The color of the card indicates which type of card it is to the programmers.

program also warrants a change to the information in the database of cards.

Once an iteration is complete, the project manager takes the completed cards and enters that information (e.g., hours to complete, estimated hours, level of risk, etc.) into an Excel spreadsheet. With the information in the spreadsheet, the project manager can perform custom calculations, such as tracking the velocity of the team and average number of points per iteration, to judge the performance of the team.

In addition, the program manager was experimenting with a new project management software system called *Rally*. When asked about *Rally*, the program manager responded that Rally provided some features that he believed that would be extremely helpful for his team, such as the ability to compile cards for a particular iteration and have the system check that the tasks do not exceed a given limit of time allotment and risk. However, the greatest drawback he cited was the web form based interface necessary for entering all in all of the information into the system. He felt it was clunky and hard to use.

Design

Taking the observations from the ethnography, we designed interaction techniques used to create, perform actions on, and navigate through the cards. Based on the ethnography, we discovered that actions the programmers took with the cards included 1) creating cards and adding them to the system, 2) augmenting a particular card with additional information, 3) "collapsing" cards, or taking the tasks associated with multiple cards and joining them on one card, and 4) assigning a card to be completed during a particular iteration, a particular timeframe during the coding cycle. The system uses a Nokia SU-1B digital pen (based on the Anoto technology) to capture the ink and translate it into a digital format.

In designing the interaction techniques, we focused on using the pen as the sole means of inputting information and commands; this way, we can better evaluate the situations in which pen-based input is beneficial. Starting with the first scenario, the user needs only to create the card as he/she would normally; the system would be able to recognize and parse the input fields without any additional action by the user. For users to be able to augment a card with additional information, they need to access the digital version of the card. Following in our research interest to make all input pen-based, users simply tap on the physical card with the pen in order to bring up its digital representation on the screen. From there, the user can view and modify the additional information only available on the digital version of the card. Collapsing physical cards would be accomplished by stapling the cards together; however, this would have no bearing on the digital representation of the cards. In order to collapse cards digitally, the user stacks the cards and then draws a mark down all of the cards, terminating the mark by looping up and around from left to right. Finally, the user can schedule a card for a particular iteration by marking the week number on the reverse side of the card inside of a diamond.

Evaluation

We performed an informal evaluation of this prototype with ten participants. The subjects were unfamiliar with story/task-based programming, though they were briefed about the cards and their purpose. While all

Figure 3. In the study, subjects were presented with cards that were previously entered into the system. The subjects used these cards and cards that they created themselves in order to complete the different tasks.

pen, all of the information used was preloaded into the system. Since we were only concerned with the interaction techniques, and not the technology, the system was irrelevant to the study. Subjects were presented with cards that were already created and loaded into the system. Subjects were told about the gestures used to invoke particular commands, and that gestures were terminated by docking the pen in its cradle. The users were then asked to complete a set of four tasks, including *adding* a card to the system, *collapsing* cards together, *accessing* the electronic version of a card and augmenting it, and *assigning* tasks to a particular project iteration. Users were observed while interacting with the system; afterwards, they were provided with a questionnaire asking about their experience.

Results and Findings

The results from the user testing provide both positive and negative evaluations about various features of the interaction techniques. Overall, the subjects reported that their initial experience ranged from somewhat difficult to extremely difficult, but they also reported that they began to get the hang of using the system by the end of the study. Users were excited about the possibility of being able to input information into the computer through their handwriting, but found the prototype initially frustrating, for reasons discussed below:

Card Creation

Subjects found it quite easy to create a new card and add it to the system. Some subjects did not realize that synchronization only occurred when the pen was placed in the cradle (though this was explained to the subjects beforehand.) Once the information was uploaded, the subjects were presented with a digital photograph of their card. They reacted favorably to the picture of the card and the underlying information, believing that the pen actually uploaded the information. The subjects had mixed opinions regarding the usefulness of having the handwritten version of the card in a digital form. On one hand, the subjects felt it was good because it provided them visual feedback that their actions were successful and it allowed them to verify that the information had been inputted correctly. On the other hand, some subjects found it unnecessary since the actual card was in front of them, and they could have used that to verify that the information was correct.

Navigation

Navigation proved to be the most difficult task for the users to complete. Users first tried to use the computer to attempt to search and/or navigate towards the correct card; however, the system was designed to only allow for the pen to facilitate navigation, not the computer. When they found out that they could not complete the task this way, most of them took some time to think about their next move before they recognized that they could use the pen to select the correct card. Once the eight of the ten subjects grabbed the correct card from the stack and tapped on it using the digital pen, they waited a substantial amount of time for the screen to change to that card, again forgetting to place the pen in the cradle for synchronization. While the subjects understood that the pen needed to be returned to the cradle, they felt that the action was too cumbersome and wanted immediate feedback regarding the success of their actions. The issue of immediate feedback represented a shortcoming of the prototype that needs to be addressed in the next design iteration.

Actions on Cards
Most subjects were successful in performing both a collapse action and an assignment to an iteration with a set of cards. However, some subjects experienced some difficulty performing the actions. One of the subjects hesitated to make the appropriate mark on the cards because the card looked "neat" and he did not want to "deface" it by doing so. Another subject complained that she did not like writing the action on top of the card because it would make the card harder to read after making commands on the card over time. Some users also reported difficulty drawing the shape of a diamond to complete an assignment of a card to an iteration, asking that a different shape should be used.

Design Implications
Based on the usability study, our findings have led us to take the following considerations when revising our prototype:

Using the Entire Card
Currently, the programmers only use the front of the card for information, leaving the back completely blank. For the target users, the back of the cards are never seen both globally and locally, so there is not an assigned meaning to that area. Because the subjects indicated that they did not like issuing commands on the top of the card, it would be beneficial for all of the actions on cards to be marked on the back of the card, allowing the front of the card to remain unmarred from frequent commands.

Immediate Feedback
One of the largest problems experienced by the subjects was the lack of immediate feedback about interacting with the cards. Many of the subjects expected that their actions would be recognized by the computer immediately, and were thus confused when the computer did not react. While some actions should be executed immediately (such as searching for a card,) other actions do not necessarily need to be executed immediately (e.g. creating cards or assigning a stack of cards to an iteration,) since they occur in batches. As the system becomes more robust, we will need to examine which actions should occur wirelessly and which ones occur through docking.

Related Work
In their discussion of paper user interfaces, Johnson et. al. lay out three requirements for paper-based user interfaces [3]. First, processing with these systems must be quick and reliable. Second, the system must handle all of the scanning of the information on behalf of the user. Third, all of the information that can be represented physically must be able to be represented electronically. Digital pen technology is at the point where all three of these requirements can be met.

The prototype is similar to the Paper PDA [2], a system that augments paper materials with computational abilities. Both systems allow users to operate with the paper representations as normal, later synchronizing with the system to produce the papers' digital representations.

Both *Collaborage* [7] and *Outpost* [4] demonstrate how a collection of physical objects on a surface (usually a wall) can be sensed by computers and translated into a digital format. Here, ceiling-mounted cameras capture the information and translate it to an electronic form, handling the transformation on behalf of the user. This system is similar to our system in that the main form of input into the computer system is pen and paper, though both systems' computer vision requires a large, specialized environment in order to operate optimally.

In addition to paper based interfaces, Liu, Erdogmus, and Maurer have developed a system using touch-sensitive tables and tablet PCs to completely digitize the story/task based programming process [5]. Programmers can use their tablet PCs to create cards, and then they can place those cards in the touch-sensitive table so that they can be "passed" around and shared with others. The system completely eliminates the need for paper notecards. However, by digitizing the process, the programmers lose the affordance provided by paper notecards [6]. In addition, the system described requires an expensive overhead that might not be currently feasible for software companies.

Conclusion

In this paper, we have presented a work-in-progress that integrates the physical with the digital using notecards within the context of story/task-based programming. After conducting an ethnography into the practices of the target user, we designed a system to test interaction techniques using digital pens to facilitate more efficient ways of adding, navigating through, and augmenting physical notecards. The results from our investigation have provided important insights, and we intend to incorporate these findings in our next prototype as we attempt to redesign it.

Acknowledgements

We would like to thank Brandon Burr, Jan Chong, Özgür Eris, Larry Leifer, Bob Plummer, George Toye, and Ron Yeh for all their support and guidance.

References

[1] Chong, J. Plummer, R. Leifer, L., Klemmer, S.R., Eris, O., Toye, G. Pair Programming: When and Why it Works. In Proceedings of Psychology of Programming Interest Group 2005 Workshop. Brighton, UK, June 2005.

[2] Heiner, J., Hudson, S., Tanaka, K., "Linking and Messaging from Real Paper in the Paper PDA", CHI Letters (Proceedings of the ACM Symposium on User Interface Software and Technology), pp. 179-186.

[3] Johnson, W., Jellinek, H., Klotz Jr., L., Raol, R. Card, S., Bridging the Paper and Electronic Worlds: The Paper User Interface. Proceedings of the SIGCHI conference on Human factors in computing systems, April 1993, pp. 507-512.

[4] Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., Landay, J. A. The Designers' Outpost: A Tangible Interface for Collaborative Web Design. Proceedings of UIST 2001, The ACM Symposium on User Interface Software and Technology, CHI Letters, 3(2) ,pp. 1–10.

[5] Liu, L., Erdogmus H., and Maurer, F. An Environment for Collaborative Iteration Planning. Agile 2005.

[6] Mackenzie, A., Monk, S. From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice. *Computer Supported Cooperative Work*. Volume 13, Issue 1. Kluwer Academic Publishers (2004), 91-117.

[7] Moran, T.P., Saund, E., Van Melle, W., Gujar, A.U., Fishkin, K.P., Harrison, B.L. Design and technology for Collaborage: collaborative collages of information on physical walls. Proceedings of the 12th annual ACM symposium on User interface software and technology. November 1999, pp. 1-10.