An Environment for Collaborative Iteration Planning

Lawrence Liu[*], Hakan Erdogmus[+], and Frank Maurer[*]
[*]*Computer Science Department, University of Calgary, Alberta*
{ypliu,maurer}@cpsc.ucalgary.ca
[+] *National Research Council Canada*
Hakan.Erdogmus@nrc.gc.ca

Abstract

Existing project planning software for agile development processes offers limited support for face-to-face, synchronous collaboration. In this paper, we describe an environment, AgilePlanner, that supports team collaboration during planning meetings. Our approach utilizes advanced technologies of pen computing and digital tabletop to create a collaborative work environment. It combines the benefits of paper index cards with those of traditional desktop planning solutions. AgilePlanner is intended as an integral resource in the planning process.

1. Introduction

Agile development processes encourage team collaboration by high-bandwidth communication and collocation. Desktop technologies do not support these tenets well as they "offer limited support for face-to-face, synchronous collaborations" [3].

Planning in agile software processes usually takes place in meetings attended by customers, developers, and other stakeholders. During iteration planning, agreements are reached upon about which features should be implemented based on the business value and effort estimates of the proposed features. Existing agile project management software is mainly used to store the decisions and output of the planning process and to monitor the progress of the project. One important aspect of planning is left out without proper support: face-to-face team collaboration. As a result, the agile planning workspace is mostly dominated by traditional media, typically, sticky notes, paper index cards, flipcharts, and whiteboards. However, traditional media do not offer convenient capabilities for searching and remote access that digital media can offer. As a result, planning information sometimes needs to be converted back and forth between traditional and digital media formats.

Support for agile iteration planning need not remain an after-the-fact solution. Tool developers could consider employing technologies that facilitate group collaboration during the planning meetings. *AgilePlanner* is a proof of concept environment that combines the benefits of asynchronous desktop solutions with synchronous collaboration support during group planning activities.

The environment specifically aims at supporting planning activities for Extreme Programming [24] and Scrum [18] because both methods have a strong emphasis on collocated collaboration.

AgilePlanner is an extension to an existing desktop project planning system, MASE [5]. It integrates pen-computing technology and digital tabletop technology to meet the needs of Extreme-Programming-like iteration planning activities. Pen computing allows users to produce planning artefacts the way they would in an ordinary low-technology planning meeting. The digital tabletop technology permits users to have shared access to these artefacts: users do not have to take turns to interact with the artifacts being manipulated. The artifacts are instantly stored in digital form and can be retrieved on demand for revision on a traditional desktop system for off-line activities or on a tabletop system during future planning meetings. As a result, while the planning activities are conducted as intended in a natural setting, persistency is achieved.

The paper is organized as follows. Section 2 describes the main activities of the iteration planning process that the environment intends to support. Section 3 presents the proposed solution from a conceptual perspective. Section 4 describes how to plan with the environment, and Section 5 examines the proposed solution with respect to standard guidelines for supporting collaborative work. Section 6 describes the architecture of the environment. Section 7 presents related work on similar and alternative environments. Section 8 discusses the approach proposed to evaluate the environment and other future plans. The paper is concluded in Section 9 with a summery and a discussion of the limitations of the environment.

2. The Process Supported

Both the Extreme Programming iteration planning activity *Planning Game* and the corresponding Scrum activity

Sprint Planning recommend a common project room to encourage instant feedback and on-going communication. Although differences exist between the planning activities of the two methods, they are not significant for our purposes. The two processes have complementary aspects, and are commonly combined. In the following, we describe a hybrid process that should be familiar to the practitioners of both Extreme Programming and Scrum. A blend of Extreme Programming and Scrum terminology is adopted to cover the relevant concepts. Iteration planning is usually conducted around a large vertical surface such as a whiteboard or a horizontal surface such as a tabletop. We assume a horizontal tabletop to represent this shared planning workspace. The process described next is an abstraction that we have adopted to concretely identify the needs for the collaboration environment. It is neither meant be complete or fully conformant to Scrum or Extreme Programming.

The planning process can be thought of as consisting of two main activities. During the first main activity, *user stories* (the Extreme Programming term for high-level user requirements) are created. The stories serve as reminders for “a promise to talk” between the customer and the developers. The content of the stories and the ensuing discussions during the planning meeting should be just detailed enough to estimate the implementation effort for the stories. Paper index cards are usually chosen to record the user stories because of their convenience. The limited space offered by the index cards force the story content to be concise. Hereafter, the term *story management* will refer to the group of activities that relate to the creation, modification, and revision of individual story cards.

After creation, stories are usually placed in a physical area representing the Product Backlog (a Scrum artefact that stores major items of remaining work) for the project or an upcoming iteration or release to be processed later, or laid out on the tabletop to be revised, organized, and processed immediately. Thus during the second main planning activity, team members together process the created stories: they reorganize stories, revise them, categorize them, split them, estimate the development effort, and order them. The term *iteration management* will refer to these activities that take place in the shared workspace, such as story prioritization, assignment of stories to an iteration, and removal of stories from the Product Backlog.

Different physical areas of the planning space often represent different project phases. For example, the pile of story cards that constitute the Product Backlog may be stored in one corner of the tabletop (the future). The central area may be used to organize the stories to be completed in the current iteration (the present). There may be other piles representing completed and deleted stories from previous iterations (the past).

3. Proposed Approach

The goal of AgilePlanner is to provide tool support for an iteration planning process that resembles the one described in Section 2.

We identified two metaphors that we believe are central to such a process: *pen-and-paper* for story management and *shared workspace* for iteration management.

AgilePlanner emulates story management with paper index cards and a pen through pen computing, while it emulates traditional, group interactions around a regular tabletop through a digital tabletop. The result of combining these two technologies is a fully digital collaborative planning environment.

Figure 1 depicts a high-level representation of our implementation of this environment using Tablet PCs, a digital tabletop system, and an existing traditional desktop planning tool MASE [5]. The tabletop system has either a touch sensitive display, a digitizing display that requires a stylus for input, or is equipped with overhead cameras that track users’ gestures over the display. Each participant has a Tablet PC. The Tablet PCs have digitizing screens that require a stylus for input from the user.

Participants interact with both the Tablet PCs and the tabletop display. They use the Tablet PCs to perform story management tasks. Via a wireless connection, changes made to story card are uploaded to the MASE system, which serves as the backend, and are reflected on the digital tabletop. The participants can then perform iteration management tasks on the digital tabletop to process and organize the story cards, as described in the previous section. Figure 2 demonstrates planning with an earlier version of AgilePlanner on a touch-sensitive digital tabletop display.

AgilePlanner mediates the interactions between the Tablet PCs, the digital tabletop and the MASE backend, and implements the necessary user interfaces on each platform. The users have the illusion of planning with traditional media, manipulating digital story cards the way they would paper index cards.

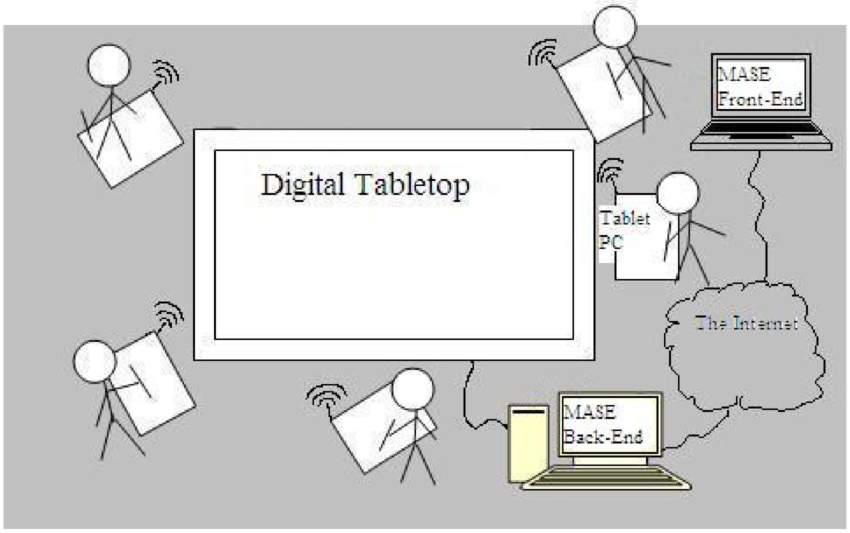


Figure 1: The AgilePlanner Environment.

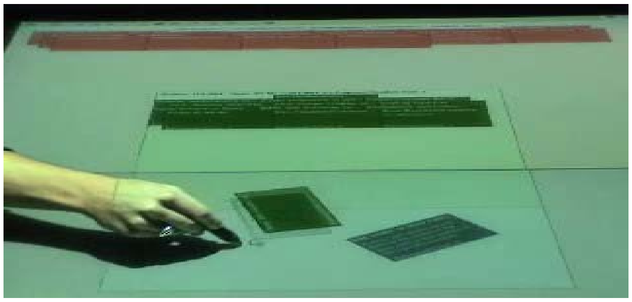


Figure 2: Iteration planning with a stylus on a horizontal tabletop display.

The planning results stored by MASE backend are available via a Web interface (the MASE front-end in Figure 1). This information can be shared and manipulated by distributed teams in a traditional, asynchronous manner using MASE's existing capabilities.

In the following subsections, we describe further the advantages and particulars of the different technologies involved.

3.1. Pen Computing

Pen computing technology enables the users to interact with a computing device through moving a stylus directly on the device's display. The display is either touch sensitive or has a digitizer that senses the location of the stylus on the display. Well known examples of pen-computing devices are PDAs and Tablet PCs.

The technology is designed to implement the pen-and-paper metaphor. Luff et al. [1] state that "despite computer systems being introduced to perform similar tasks, paper documents (pen and paper) are still being used in all settings", and although "screen-based text is often more legible than handwriting, it can be easier to write with pen and pencil on paper than type using a keyboard, particularly when interacting with colleagues or clients, or engaged in concurrent activities." Through handwriting recognition capability, pen computing technology today addresses both legibility and ease-of-use.

The Tablet PC is a relatively new pen-computing device. Users can interact with the device in a similar manner to writing on paper attached to a clipboard. Unlike laptop computers, it can be used standing up.

The handwriting recognizer is integrated into the operating system, and as such easily available to applications. Walker [9] states that the handwriting recognition performance has been satisfactory. Consequently, it was a good candidate for supporting story management in our environment.

AgilePlanner takes advantages of the built-in handwriting recognition capability of the Tablet PC, and provides an interface that emulates writing on paper index cards. It also employs the Tablet PCs built-in wireless capability to communicate with the backend component and the digital tabletop. As a result, users can be mobile in the planning locale. Figure 3 illustrates story management using a Tablet PC.

Figure 3: Using AgilePlanner with aTablet PCs to create or modify stories.

3.2. Digital Tabletop Technology

Research suggests that desktop computers inhibit group collaboration. In many situations, people are forced to sit closer than it often is socially comfortable for them when sharing a traditional desktop workstation [2]. In addition, people have to resort to turn-taking because they cannot simultaneously interact with a single desktop workstation in a natural, synchronous manner [3]. Stewart et al. [6] state that the traditional desktop metaphor does not effectively support collocated, multi-user interactions because of its underlying one-user/one-computer design paradigm

The digital tabletop technology addresses these problems. First, users can move around the tabletop display freely, thus maintaining a comfortable distance between themselves and their collaborators. Moreover, most of the available systems support multiple input devices. As such, users can interact with the tabletop display concurrently, allowing truly synchronous collaboration. Since it takes little training for people to be at ease with digital tabletop systems, usability is not a major obstacle.

The current implementation of the Agile Planner environment uses a digital tabletop system that was custom-built by the *Interaction Lab* at the University of Calgary [4]. It involves a S*martBoard DViT* display, two high resolution projectors and two mirrors. The DViT display is touch sensitive, and it was originally built for a rear-projected interactive wall display. For tabletop collaboration, the board is placed horizontally to produce a surface area of approximately 1.5 m by 1 m. It has a display resolution of 2048 by 1024 pixels, not very high for a display of that size, but sufficient for our prototype implementation. The system uses a matrix of infrared LEDs along with light sensors installed in each corner to locate touch points on the surface. Figure 4 shows the setup of the system.

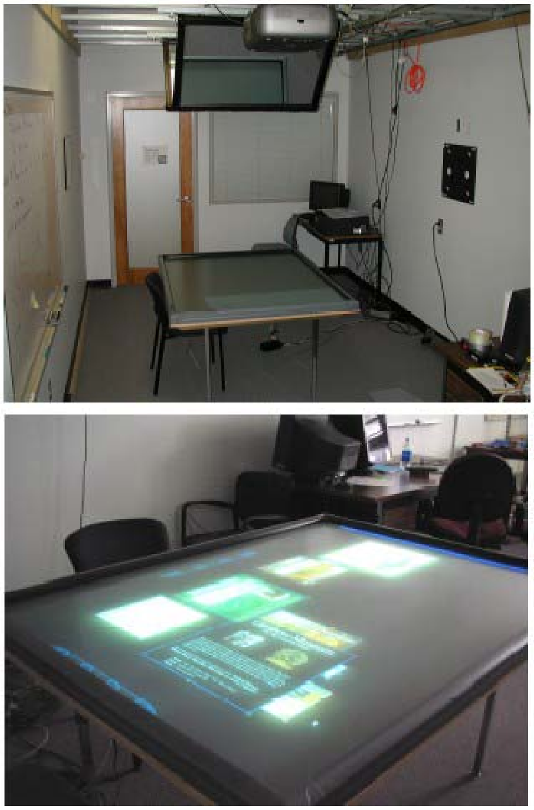


Figure 4: The digital tabletop at University of Calgary's Interaction Lab.

3.3. The MASE System

MASE is an project management system for agile software development. It has been developed at University of Calgary [5]. The system has a range of complementary functionality accessible via a Web interface, including:

- keeping track of the project team's composition and past performance;
- providing an integrated communication facility for the team members;
- providing Wiki–based services that support lightweight knowledge sharing; and
- managing project work products, such as stories, iterations, and releases.

The persistence layer of MASE is implemented using a relational database. MASE's data model is designed to support Extreme Programming's and Scrum's iteration planning practices.

AgilePlanner primarily uses MASE as a backend system to achieve persistence. Between iteration planning meetings, AgilePlanner users can access MASE in the usual way from a desktop station to perform off-line project management activities.

4. Iteration Planning with AgilePlanner

AgilePlanner has two user interfaces: the *Planning Area* and the *Story Editor*.

The Planning Area, shown in Figure 5, is normally accessed through the shared tabletop and supports iteration management tasks. It is meant for simultaneous use by multiple team members.

The Story Editor, shown in Figure 6, is normally accessed through the Tablet PC and supports story management tasks. It is meant to be used by individual team members when creating or modifying a story card.

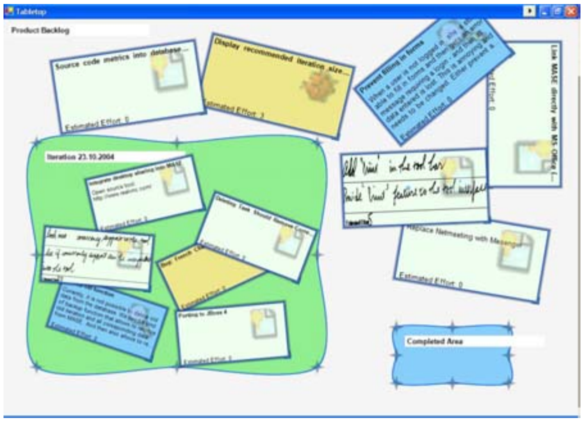
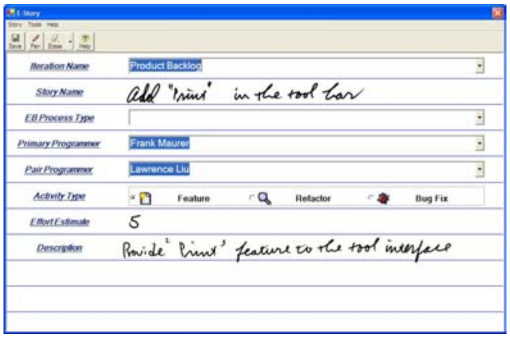

Tabletop
Product Backlog
Source code metrics into database...
Estimated Effort: 0
Display recommended iteration size...
Estimated Effort: 3
Prevent filling in forms
Estimated Effort: 0
Link MASE directly with MS Office (...
Iteration 23.10.2004
Add 'Print' in the tool bar
Provide 'Print' feature to the tool interface
Replace Netmeeting with Messenger
Estimated Effort: 0
Porting to JBoss 4
Completed Area

Figure 5: The Planning Area.



Iteration Name	Product Backlog
Story Name	Add 'Print' in the tool bar
EB Process Type	
Primary Programmer	Frank Maurer
Pair Programmer	Lawrence Liu
Activity Type	● Feature ○ Refactor ○ Bug Fix
Effort Estimate	5
Description	Provide 'Print' feature to the tool interface

Figure 6: The Story Editor.

In addition, two menu components –Dashboard and AdminMenu – are accessible both from the Tablet PCs and the shared tabletop display. The Dashboard allows team members to switch among projects and iterations within a project, and specify the workspace mode for the current planning task. The AdminMenu provides widgets for the creation of new projects, iterations, and story cards (typed or handwritten), as well as for the rearrangement of the stories.

Assume several story have already been created for the current iteration of an existing project in a previous planning session. The team indicates that they wish to continue planning the current iteration via the Dashboard. The tabletop will then display all the previously created stories in the same physical arrangement as they were left at the end of the last planning session. Capturing the last state of the tabletop assists the team to remember the past decisions made and actions taken. We now describe some typical interactions with the tool based on & few selected scenarios.

4.1. Modifying Existing Stories

Suppose the previous planning session ended with the Planning Area arrangement shown in Figure 5. Stories inside the green rectangular area belong to the current iteration, while the ones outside it are part of the Product Backlog. To modify a story, a user first displays the Planning Area on own Tablet PC. Then the user selects a story by double-tapping with the stylus on a story card inside the Planning Area. This action automatically launches the Story Editor.

The story Editor displays the selected story on the Tablet PC, as illustrated in Figure 6. Using the stylus again, the user may change the description to the story and save it. Figure 8 depicts a story on the Planning Area before and after such modification.

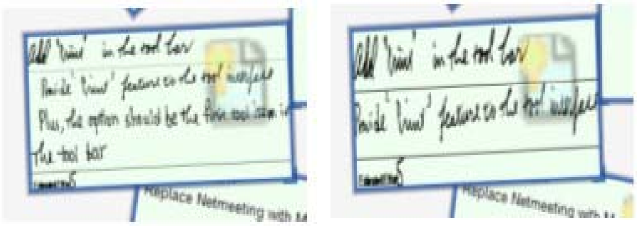

Add 'Print' in the tool bar
Provide 'Print' feature to the tool interface
Plus, the option should be the first tool item in the tool bar
Replace Netmeeting with M
Add 'Print' in the tool bar
Provide 'Print' feature to the tool interface
Replace Netmeeting with M

Figure 8: Image of handwritten story card on the Planning Area before (left) and after (right) modification.

4.2. Assigning Stories to an Iteration

Suppose the team now wants to move a story from the Product Backlog to the current iteration or from the current iteration to the Product Backlog. This is easily done on the tabletop display by a finger gesture: a team member simply points to a story and drags the story to the respective physical area of the display representing the Product Backlog or the current iteration.

When a story is indicated by pointing to it with a finger on the tabletop display, a small circle emerges representing the user's finger tip. The user drags this circle to move the story. If the story is not manipulated, the circle fades away and the story is deselected. The story can be rotated as it is being dragged and made to face a different direction, just like it is possible with an index card when it is being dragged by hand on a regular tabletop.

Users can select multiple stories at once by moving them closer to each other in a clump first and then specifying a large circular area. Stories that overlap the specified circular area can then be moved together. This action is illustrated in Figure 9. The blue circle overlaps three stories, which can be moved together by dragging the circle. When a collection of stories are being moved object rotation is disabled. Figure 10 shows how the Planning Area looks like before and after a collection of stories have been moved.

The size of the selected story cards can be reduced or enlarged together by dragging in or out a corner of the selection circle.

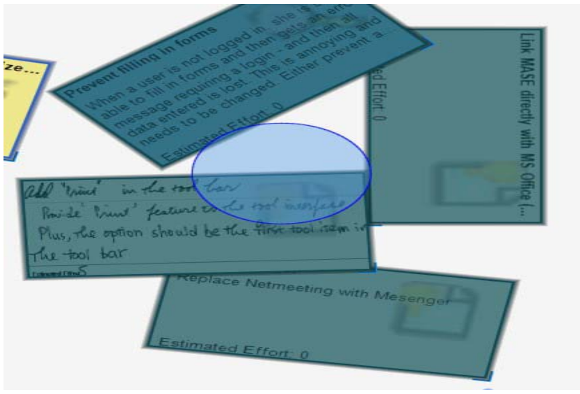

Prevent filling in forms
When a user is not logged in, she is able to fill in forms and then gets an error message requiring a login - and then all data entered is lost. This is annoying and needs to be changed. Either prevent a
Estimated Effort: 0
Link MASE directly with MS Office (...
ed Effort: 0
Add 'Print' in the tool bar
Provide 'Print' feature to the tool interface
Plus, the option should be the first tool item in the tool bar
Replace Netmeeting with Mesenger
Estimated Effort: 0

Figure 9: Selecting multiple stories.

4.3. Public and Private Workspace Modes

AgilePlanner has two so-called *workspace modes*. In *public* mode, the information on the tabletop and the Tablet PCs are constantly synchronized. In contrast, when a Tablet PC user switches into *private* mode, any changes performed on the story cards and the Planning Area are only saved on the Tablet PC and will not be reflected in the shared Planning Area on the tabletop display. The private mode is reserved for off-line changes.

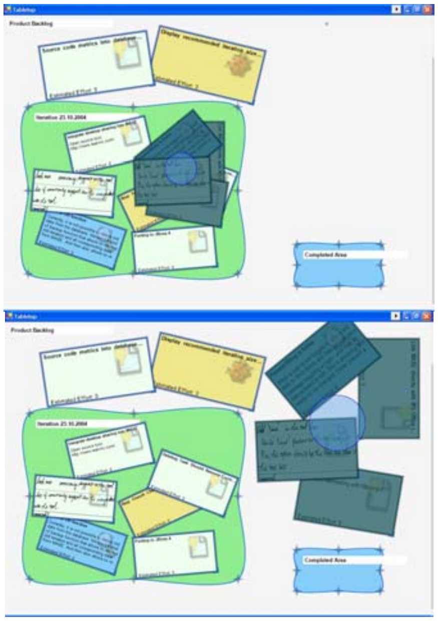


Figure 10: The Planning Area before (top) and after (bottom) moving a collection of stories.

For example, the Tablet PC user and the tabletop may start out with the shared image of the Planning Area shown on the top panel of Figure 11. The Tablet PC user may then switch into private mode, move a few story cards from the Product Backlog to the current iteration area, and create a new handwritten story card, ending up in a local configuration of the Planning Area shown in the bottom panel of Figure 11. While the user stays in private mode, the image of the tabletop will not change and the user will retain the option of scrapping the changes. When the user wants to release the changes to the whole team, she can switch back into public mode, an action that resynchronizes the shared tabletop data and the local Tablet PC data.

If the state of the shared Planning Area has changed in the meantime, a conflict may arise or a merge may be required. Although the current implementation of AgilePlanner does not yet support handling of conflicts and merges during resynchronization, this capability is being planned for future releases of the environment.

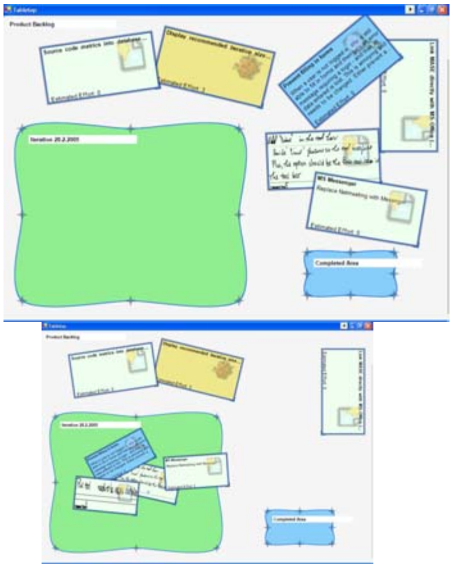


Figure 11: The local and shared images of the Planning Area in private mode: displayed on the tabletop (top panel, visible to the whole team) and on the Tablet PC of a single user (bottom panel, visible only to the Tablet PC user).

5. Adherence to Environment Design Guidelines

In this section, we scrutinize the AgilePlanner environment against the design guidelines suggested by Scott [7] for tabletop environments that are intended to support collaborative work.

1. ***Natural interpersonal interaction.*** Gutwin and Greenberg [17] suggest that people facilitate workspace awareness by low-level mechanics that gather and convey information about the actions performed, the time at which they are performed, and the person performing them [6]. The technology must not be invasive, for interference causes "breakdown on collaboration" [7]. AgilePlanner emulates a natural, manual planning environment. As such, it avoids interfering with such low-level mechanics.
2. ***Transitions between activities.*** The overhead of switching between different activities, such as writing, drawing and manipulating artefacts, should be minimized [15]. Such minimization permits fluid transition between activities, and allows users to focus on communication [6]. Examples from AgilePlanner of adhering to this guideline include the following. Moving and rotating story cards on the tabletop display happens in a single action. Writing and drawing on a Tablet PC are indistinguishable to the user when using digital ink. Stories are manipulated in the same manner on both the Tablet PCs and the digital tabletop; hence transitions are not required when switching between activities performed on the two types of hardware.
3. ***Transitions between personal and group work.*** It has been observed that people frequently transit between individual and group work by maintaining distinct areas for each type of work. [19, 16]. Workspace distinction is provided in Agile Planner in both the physical and the virtual sense. The intended uses of the Tablet PC and the tabletop platforms are different: the former is for individual activities and the later for collaborative activities. In addition the environment supports two kinds of virtual workspace, private and public, for off-line individual work and for on-line group work, respectively. Thus, individual work can be in a physically and virtually separate area from the shared workspace.
4. ***Transitions between tabletop collaboration and external work.*** Scott [7] states that "co-located group interaction is only one part of daily collaborative activities." Group members must also be able to incorporate work generated externally to the tabletop system into the current tabletop activities. AgilePlanner allows externally or off-line prepared artefacts to be incorporated into group activities through the MASE front-end and the private workspace mode.
5. ***Shared access to digital objects.*** Interacting with shared artefacts can help maintain the group focus and facilitate awareness because body positioning and eye gaze of the group members attending to the same object can be easily interpreted by other group members [7]. Using AgilePlanner, story cards are shared on the digital tabletop. Gestures that manipulate the story cards on the tabletop are instantly visible to all participating team members; which facilitates group communication.
6. ***Flexible user arrangements.*** Physical properties of the table, such as its size and shape, can influence user arrangements. Also, people typically prefer to be positioned at least at an "arm's length" from each other [2]. Prolonged period of invasion of the personal space would induce social awkwardness [7], possibly affecting the planning outcomes. The physical configuration of our prototype can accommodate small teams of up to four people, but larger tabletop displays can accommodate more. The mobility provided by the wireless Tablet PCs allows users to flexibly and comfortably arrange themselves around the tabletop.

7. ***Simultaneous user interactions.*** Agile Planner's current implementation does not yet support simultaneous user interactions, but we are investigating ways of customizing the available tabletop system to handle multiple concurrent users. The main difficulty here is identifying the users without requiring invasive methods such as wearing patterned bracelet tags (as some camera-based systems do) or requiring uniquely identifiable pointing devices.
8. ***Use of physical objects.*** We currently have no plans to support the use of physical objects. A system that employs physical objects is described in Section 7.

6. Architecture and Implementation

Figure 12 illustrates the high-level architecture of the AgilePlanner environment.

Microsoft's Windows XP Tablet PC Edition is currently the only operating system that supports pen computing and ink-smart applications on Tablet PCs. Therefore the client component of AgilePlanner that implements the front-end functionality currently runs only under Windows XP. The tabletop system runs the same client software, but without support for digital ink.

The operating system provides the necessary functionality to translate ink input from Tablet PC users into a form that can be transmitted and stored. The translated data from the tabletop and the Tablet PCs are exchanged with MASE via Web Services using the SOAP protocol. MASE is responsible for all back-end functionality, including storing the planning data in a relational database.

MASE handles multiple users by treating them as clients that belong to the same broadcast group. If a client releases changes to MASE, then MASE notifies all clients in the same broadcast group. The AgilePlanner client is then responsible for updating the Planning Area on the Tablet PCs and on the tabletop display. MASE treats the Tablet PC clients and the tabletop client in the same way. Outside the environment, the stored planning data is accessible to regular desktop clients though MASE's webs interface for off-line project management activities.

A drawback of the traditional tabletop applications has been the treatment of object relocation and rotation as separate operations. Research however has demonstrated their inseparability on a physical table [12]. Jacob et al. [13] suggested that an integrated manipulation of objects would ensure the fluidity of tabletop collaboration. AgilePlanner implements this suggestion by using Kruger's "Rotate'N Translate" algorithm [11] to allow users to simultaneously drag and rotate story cards around the tabletop display in a natural way.

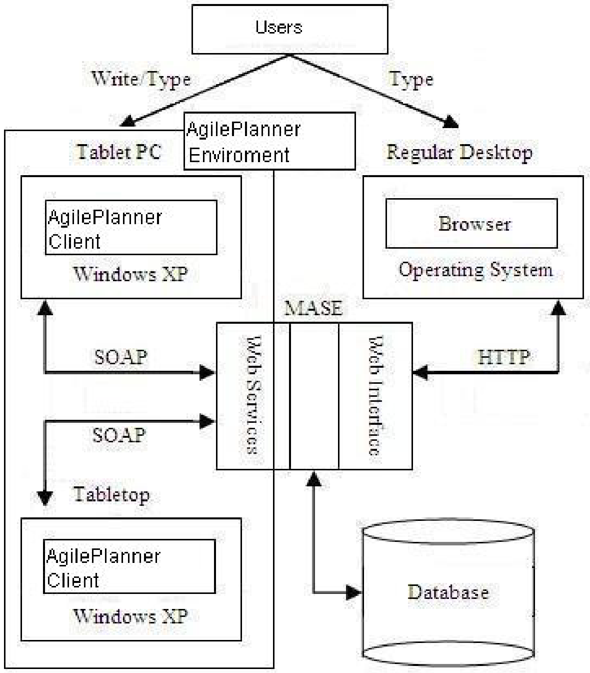


Figure 12: The architecture of AgilePlanner.

7. Related Work

Project wikis often provide flexible, informal support for project planning and management tasks in agile software development. When more structured support is required, specialized software packages are used.

Desktop planning software, such as VersionOne [21], ReleasePlanner [19], and Scrumworks [22], supports planning based on a one-user/one-computer paradigm. Unfortunately, as stated earlier, this paradigm breaks down the flow and dynamics of collaboration. As a result, desktop planning software is often only useful for data management and persistence and asynchronous collaboration, but not for synchronous collaboration. Large horizontal displays are sometimes used during group activities to achieve synchronous collaboration, however still only one user can interact with the planning software at any given time. This scheme may prove adequate in situations when the meetings are well structured and facilitated, but falls short when the meetings are less structured and the interactions are highly fluid, random, and open-ended.

Several tabletop systems that support different types of collaborative work exist. ConnecTables [20] is one example designed to support interaction-intensive collaboration with pen-enabled, small tabletop devices. The devices are mobile (mounted on wheels), networked and context-aware. They are dynamically coupled to form a shared workspace when in close proximity of each other. However, the system requires specific identical hardware and supports only two-way collaboration. Earlier collaboration systems that integrate various technologies involving large displays, SmartBoards, pen-based interfaces, and desktop environments are described in [25, 26, 27, 28].

An example that uses physical objects to assist collaboration is the BUID-IT system. It allows users around a tabletop to collaboratively create a model [23]. Cameras and state-of-the-art computer vision technology is used to track physical objects being rearranged on a horizontal surface. The physical objects represent virtual objects of the model being built. The model is projected onto a wall adjacent to the table. The idea of using physical objects to represent virtual objects is interesting. In our context the idea would imply using physical index cards to represent virtual index cards. The contents of physical index cards could in theory be automatically recognized and stored. However the implementation of such a scheme would be much more heavy-weight than that of the current Agile Planner environment.

8. Future Work

8.1. Evaluation of the Environment

AgilePlanner is still under development. As such, we have not yet evaluated it with real users. We are in the process of designing a case study to qualitatively evaluate the usability and usefulness of the environment.

The case study will be conducted during the team project portion of a senior undergraduate course at the University of Calgary. The students taking this course are required to complete a project consisting of multiple iterations, following a hybrid agile process based on Extreme Programming and Scrum.

The teams will use paper index cards in the first iteration. Then first MASE only (representing traditional, desktop-based planning support) and later AgilePlanner (representing collaborative, tabletop-based support) will be introduced in subsequent iterations. Team members participating in the study will be asked to fill out questionnaires before and after each iteration planning session about their expectations and experiences with the tools introduced. The feedback from questionnaires will be used to compare the different levels of support provided by the tools with manual, pen-and-paper-based planning.

At the same time, we hope to obtain input from the case study participants that will help us improve the environment.

8.2. Improvements to the Environment

The future development of AgilePlanner will focus on supporting concurrent interactions with the tabletop display, improving existing functionality and interfaces, and supporting the underlying metaphors in more natural ways.

Allowing concurrent interactions with the tabletop display is a top priority since concurrency is necessary for seamless and fluid collaboration. This will be a challenge with the existing capabilities of the SmartBoard DViT system currently employed in environment. We are actively investigating alternative solutions.

Regarding the enchantments to existing features, releasing private changes is currently an all-or-none proposition. We would like to provide finer control so that the users can selectively indicate the changes to release and discard the rest. Another point regarding the release of private changes is the ability to merge changes released by multiple users (e.g. though serialization) and detecting and handling collisions when they arise. These capabilities are yet not supported, but since they concern mostly off-line activities rather than real-time collaboration, they do not have high priority.

Regarding metaphors, one improvement being contemplated is to implement story selection from and addition to the shared Planning Area, as well as the release of modified stories into it, through a drag-and-drop gesture with the stylus, moving it between the Tablet PC and the tabletop display. This gesture would emulate the physical transfer of a paper story card between a clipboard (represented by the Tablet PC) and a table (represented by the tabletop display). A similar metaphor for story card deletion would be to use a throwing gesture over the edge of the Planning Area on the tabletop.

9. Conclusions

This paper described a prototypical environment to support iteration planning in agile software development It demonstrated how such an environment can be constructed by integrating state-of-the-art technologies to facilitate team collaboration during planning meetings.

Existing planning tools are predominantly based on the traditional desktop metaphor. However, this metaphor does not effectively support collocated, multi-user interactions due to its underlying one-user/one-computer design paradigm. As a result, these tools are not good fits for the practices that rely on high-bandwidth collaboration.

Our solution integrates pen computing on mobile Tablet PCs and the digital tabletop technology with an existing desktop-based iteration planning system. The result is the AgilePlanner environment. The idea of combining a mobile pen computing technology with a tabletop system is novel. To our knowledge, none of the existing collaboration environments takes advantage of both technologies simultaneously.

The immediate applicability of the solution is however limited. The pen computing devices we use, Tablet PCs, have only became available during the last couple years and ad still considered a novelty. The cost of supplying each user with one would for now be substantial. Digital tabletop systems are even less common outside research environments. Most are one-of-a-kind systems that are custom built from multiple components such as smartboards, underlit projection displays, touch-sensitive displays, and overhead cameras. Their cost is for now prohibitive and their setup and maintenance are difficult. However, as these technologies become more mature and widespread, we expect the feasibility of deploying AgilePlanner-like environments in medium to large organizations to increase.

The environment is still a pure research prototype, and yet to undergo a formal evaluation with real users. It is intended as a proof of concept. Through qualitative evaluation, we expect to soon obtain some preliminary results on usability and usefulness. We hope that the results will fuel further research on similar environments in particular, and on leveraging technology to improve collaborative software development in general.

References

[1] Luff, P., Heath, C., and Greatbatch, D. Tasks-in-interaction: paper and screen based documentation in collaborative activity. In Proc. of the ACM conference on Computer-Supported Cooperative Work, December 1992.

[2] Hall, E. *Distances in Man: The Hidden Dimension.* Double Day, Garden City, NY, 1966.

[3] Scott, S., Mandryk, R. and Inkpen, K. Understanding children's collaborative interactions in shared environments. *Journal of Computer-Aided Learning*, 19(2), 2002. pp. 220 - 228

[4] Habelsk, S., Carpendale, S., Scott, S., and Sonnet, H. *Realization of Territory-Based Interaction Techniques for Supporting Tabletop Collaboration.* Internship Thesis, 2004

[5] Maurer, F. Supporting Distributed Extreme Programming. In Proc. XP Agile Universe, Extreme Programming and Agile Methods, Chicago, IL, 2002. LNCS Vol. 2418, Springer. pp. 13-23.

[6] Stewart, J., Bederson, B., and Druin, A. Single Display Groupware: A Model for Copresent Collaboration. In Proc. of the ACM Conference on Human Factors in Computing Systems (CHI), 1999. pp. 286 - 293

[7] Scott, S., Grant, K., and Mandryk, R. System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In Proc. of ECSCW'03, European Conference Computer-Supported Cooperative Work, Helsinki, Finland, September 14-18, 2003

[8] Meyer, A. Pen Computing – A Technology Overview and a Vision. *ACM SIGCHI Bulletin*, 27(3), July 1995. pp. 46 – 90

[9] Walker, G. The Tablet PC: A detailed look at Microsoft's proposed Tablet PC, *Pen Computing Magazine*, July 2001.

[10] Bier, E., Freeman, S. MMM: A User Interface Architecture for Shared Editors on a Single Screen. In Proc. of ACM Symposium on User Interface Software Technology, 1991. pp.79 - 86.

[11] Kruger, R., Carpendale, S., Scott, S., and Tang, A. Fluid Integration of Rotation and Translation. In Proc. of ACM Conference on Human Factors in Computer Systems, Portland, Oregon, USA, April 2-7, 2005.

[12] Wang, Y., MacKenzie, C., Summers, V., and Booth, K. The Structure of object transportation and orientation in human-computer interaction. In Proc. of ACM Press on Human Factors in Computing Systems (CHI), 1998. pp. 312 - 319

[13] Jacob, R., Sibert, L., MaFarlane, D., and Perston, M. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction,* 1(1), 1994. pp. 3 - 26

[14] Gutwin, C., Greenberg, S. The Mechanisms of Collaborations: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. In Proc. of the IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2000.

[15] Bly, S. A Use of Drawing Surfaces in Different Collaborative Settings. In Proc. of the ACM Conference on Computer-Supported Cooperative Work (CSCW), 1988. pp. 250 - 256.

[16] Tang, J. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34, 1991. pp. 185 - 209

[17] Mandviwalla, M., Olfman, L. What Do Groups Need? A Proposed Set of Generic Groupware Requirements. *ACM Transactions on Computer-Human Interaction*, 1(3), 1994. pp. 245 - 268

[18] Schwaber, K., Beedle, M. *Agile Software Development with Scrum.* Prentice Hall, 2002

[19] D. Greer, G. Ruhe . Software Release Planning: An Evolutionary and Iterative Approach. *Information and Software Technology*, 46, 2004. pp 243-253

[20] Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., and Steinmetz, R. ConnecTables: dynamic coupling of displays for the flexible creation of shared workspaces. In Proc. 14. Annual ACM Symposium on User Interface Software and Technology (UIST), 2001. pp. 11-20

[21] http://www.versionone.net

[22] http://www.Scrumworks.com

[23] Fjeld, M., Bichsel, M., Voorhorst, F., Lauche, K., Krueger, H., and Rauterberg, M. Designing graspable groupware for co-located planning and configuration tasks. *EACE Quarterly*, 3(2), 1999. pp. 16 – 21

[24] Beck, K. *Extreme Programming Explained: Embrace the Change*. Addison Wesley, 1999.

[25] Moran, T., van Melle, W. Tailorable Domain Objects as Meeting Tools for an Electronic Whiteboard. In Proc. ACM Conference on Computer Supported Cooperative Work (CSCW), 1998.

[26] Stewart, J., Bederson, B. B., Druin, A. Single Display Groupware: A Model for Co-Present Collaboration. In Proc. Human Factors in Computing Systems (CHI), 1999.

[27] Streitz, N. et al. i-LAND: An interactive landscape for creativity and innovation. In Proc. ACM Conference on Human Factors in Computing Systems (CHI), 1999.

[28] Streitz, N., Geißler, J., Haake, J., and Hol, J. DOLPHIN: Integrated meeting support across LiveBoards, local and desktop environments. In Proc. ACM Conference on Computer Supported Cooperative Work (CSCW), 1994.