uMeeting, an Efficient Co-located Meeting System on the Large-Scale Tabletop

Jie Liu and Yuanchun Shi

Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China
liujiejesse@gmail.com, shiyc@tsinghua.edu.cn

Abstract. In this paper, we present the uMeeting system, a co-located meeting system on the large-scale tabletop. People are used to sitting around a table to hold a meeting. It is natural and intuitive. The table has a central role to support team activities. Horizontal surfaces rather than vertical ones have inherent features to support the co-located meeting. Existing tabletops aren't large enough to support more than 4 people's meeting and the display area for each person is not large enough. Thus we develop uTable, a large-scale multi-touch tabletop. Based on uTableSDK we developed earlier, we develop uMeeting system that supports co-located meeting on the large tabletop uTable.

Keywords: co-located, collaboration, meeting, large-scale tabletop.

1 Introduction

Horizontal and vertical surfaces are two main kinds of interactive surfaces. In the past twenty years, much research has been done in vertical wall-like surfaces for co-located collaboration, like Liveboard[1] and CityWall[2]. If we simply apply these vertical surfaces into the meeting scenario, many problems may emerge. Firstly, the attendees have to stand in front of the surface at the meeting; so vertical surfaces are not suitable for long-time discussion. Secondly, attendees have to turn around to communicate with others by side for eye contact. It lacks face-to-face interaction, which largely reduces the efficiency of communication in the meeting. Last but not the least, the upper part of large-scale vertical surfaces usually is out of reach. So the actual available area for interaction is much smaller than the display area, which decreases the usability of the vertical surface greatly.

On the other hand, horizontal surfaces have inherent features to support the co-located meeting. People are used to sitting around a table to hold a meeting. It is natural and intuitive. The table has a central role to support team activities. Indeed, 55% of all documents and intermediary objects used during a working session are displayed on the meeting table[10]. The table remains the principal collaborative space. Horizontal surfaces use the table as a metaphor and are also known as tabletops. Tabletops are natural interaction substrates for people to meet around and collaborate on. It can easily solve the problems that vertical surfaces cannot solve as mentioned above.

J.A. Jacko (Ed.): Human-Computer Interaction, Part IV, HCII 2011, LNCS 6764, pp. 368–374, 2011.

Existing tabletops such as Microsoft Surface[3] and DiamondTouch[4] aren't large enough to support more than 4 people's meeting and the display area for each person is not large enough. Thus we develop uTable, a large-scale multi-touch tabletop. uTable has an interactive surface of 3.8m x 2m, 3840 x 2048 pixels. It is composed of ten unit tabletops. uTable supports simultaneous multi-touch and pen interaction, so It allows multimodal interaction and can support about 12 people's meeting.

Application systems on existing tabletops focus on general collaboration and don't feature the meeting scenario. Besides, these systems were designed for small group collaboration, and may not fit for large group meeting. We need to design a new meeting system based on uTable platform. Based on uTableSDK[7] we developed earlier, we develop uMeeting system that supports co-located meeting on the large tabletop uTable.

In this paper, we introduced related work about tabletops and co-located collaboration in section 2. We also give a brief introduction about uTable platform and uTableSDK we developed. In section 3, we explain the uMeeting system in detail and list several advantages this system holds. At last, we draw conclusions and show some future work.

2 Related Work

The study on interactive tabletops has a long history. It can be traced back to 1990s when Pierre Wellner introduced DigitalDesk[8]. It projects electronic images down onto the desk and paper documents. People can interact with it using pens or bare fingers. Fitzmaurice et al.[9] used physical artifacts called bricks to manipulate virtual objects on tabletops. This essentially new input device can be tightly coupled or attached to virtual objects for manipulation or for expressing action. In recent years, many tabletops have come into being. Some well-known tabletops are Surface, SMART Table and DiamondTouch, just to name a few. All the tabletops mentioned above share the same drawbacks that they are not so big enough as to afford many people simultaneously. Very large display and interactive area is in great need for large group collaboration.

Large display has many advantages. Ball et al.[11] reported an observation analysis of the use of a large tiled display and concluded that large displays can improve user performance for task switching or viewing large documents, increase ability to spatially position applications and shortcuts for quick access and recall, and can help work collaboratively. But large displays often have bezels, that are the covering around the edges of the display screens which distort the image and affect users' experience largely. When we build uTable platform, we take this into account and try to avoid the bezel effect. Thanks to the technology of Lanbo Corp.[5], the width of joint of unit tabletops is less than 2mm, so the display image and interactive input are seamless and interior bezels effect[6] can be ignored. Hence the quality of display on uTable is great as shown in Fig. 1 and visual and perceptual immersion on uTable is excellent.

Existing interactive surfaces with large display are almost vertical surfaces like screen walls. CityWall[2] is a big interactive wall installed in public circumstance.

CityWall's large physical size appeared to support making interactions visible to others both gestural and as effects on the display when this was wanted. The author also indicated that large display size and visibility also supports immediate availability of content to interact with. But vertical interactive surface isn't the first choice for collaboration for the reasons listed above. When people want to have a discussion, they tend to have a free talk, seated with eye contact. Horizontal interactive surface is preferred when acting as a meeting table. Meanwhile, some compromise and hybrid interactive surfaces are also introduced. Tilted Tabletops [12] combined vertical surface and horizontal surface. The vertical surface is considered to be public whereas horizontal display spaces let users have more privacy. Tilted tabletops provide the opportunity to collaborate side-by-side as well as face-to-face.

Much research on co-located collaboration with tabletops has been made. Scott et al.[13] analyzed digital tabletop systems and suggested several design guidelines for effective co-located collaboration around a tabletop display including supporting interpersonal interaction, supporting transitions between personal and group work, supporting simultaneous user actions and supporting fluid transitions between activities. When we design the uMeeting system, we also take into these guidelines into consideration. Other issues such as the orientation of each workspace and the ownership of workspaces should also be addressed. Scott et al.[14] conducted two observational studies of traditional tabletop collaboration and revealed that collaborators use three types of tabletop territories to help coordinate their interactions within the shared tabletop workspaces: personal, group and storage territories. Comprehensive reports about user experiences about co-located group collaboration also come out.

Fig. 1. uTable platform

We specify the meeting scenario, one of the common collaboration scenarios, to implement on uTable platform. uMeeting system can substitute for laptops, notebooks and printed materials. We classify the territories into three scopes, namely public, private and group scope. Presentation is a typical public scope application. Taking notes and anonymous vote are in private scope. Activities such as discussion, sharing digital material and making up a schedule are in group scope.

uTable is a very large tabletop we build. As shown in Fig. 1, uTable is tiled seamlessly. The surface is rear projected and multi-touched. To summarize, uTable has three main features: 1) very large interactive surface, support many users seated; 2)Tiled, seamless both in display and interactive input; 3) support simultaneous multi-touch and pen interaction based on diffused illumination.

To enable developers rapidly develop customized application on uTable, we also developed uTableSDK[7], a software development kit for uTable. uTableSDK provides many practical features, such as multi-users, multi-touch input handling, UI management policies on large tabletops, personal space management and communication approaches between each other. We use uTableSDK to develop uMeeting system with a lot of time and effort saved.

3 System Design and Implementation

To design an efficient meeting system on uTable, we first study what people do when attending a meeting and categorize most frequent activities by the scope of message transmission. There are three scopes, public, private and group scope. When attending a meeting, attendees can be seated at will and has their own workspace. The workspace can be customized. After a default workspace is presented, attendees can resize, rotate and choose preferred style of it. If an attendee moves around, the workspace should follow him where he is. The system should also support task migration, which is one same task can be transferred from one workspace to another workspace just like a pipeline. For presentation, the system must have the function of broadcasting. When someone is giving a presentation, all the workspaces should show the same content. For group discussion, several attendees share the same screen and can also have their own area.

To put all these features into the uMeeting system, we need a well-structured and scalable framework. The framework is shown in Fig. 2 below. The architecture of uMeeting system is much like MVC (model, view and controller) style. For every attendee, there is an independent view for him. Attendees can get all information from their own view and interact with the view. The view gets the input instruction and parses it, then sends it to the controller. The controller takes charge of the communication between the user and the system. The controller keeps track of users' information and the tasks that they are running and issues requirement to the model. The model holds all the functions the system has and manages the all the tasks. When a requirement comes, the model processes it and returns a notification.

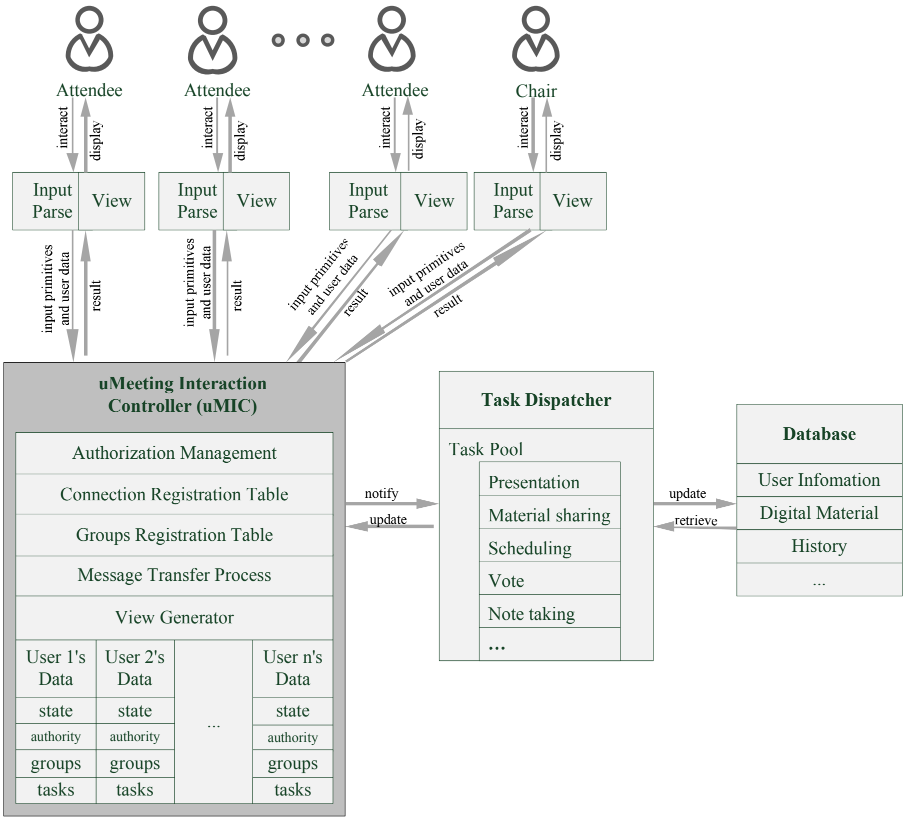

Fig. 2. uMeeting architecture

We emulate the process of the meeting and the roles of each attendee plays. The chair in the meeting has the administration privilege, and can conduct the progress of the meeting. The system creates a stack for every attendee who logs into the system after authentication. In each attendee's stack, there are state and authority information, group information that which this attendee belongs to, and tasks that this attendee is running. Based on these stacks, uMeeting Interaction Controller (uMIC) keeps track of all activities running on the system. Attendees interact with the system by multi-touch, which is a direct and intuitive interaction modal. Input Parse module processes these interaction signals and send out input primitives which can be interpreted as instructions by uMIC. uMIC updates some registration tables and notifies Task Dispatcher to implement these instructions. Then Task Dispatcher selects a task from the task pool and allocates resources to the task to processes the instructions. After completion, results return to uMIC. Then uMIC updates registration tables and states, mixes the result with some visual effect and transfers these data to the View module. At last, attendees see the result.

There are many advantages in the design of uMeeting system. First, attendees can be seated flexibly without restriction. Attendee simply creates a personal stack where he or she is seated. By using task pool and user's data stack in uMIC, one kind of task needs only one process to handle it. At the same time, each attendee who is running this task can get a different view of the result. Moreover, task migration in collaboration is simple since there are no consistency problems. We use Groups Registration Tables to separate public, private and group scope. Private scope can be treated as a group of the attendee himself. Message Transfer Process module transfer message among each group and View Generator generates views for each attendee. Besides, if a group of attendees want to share a big view, View Generator can join the views and generate one view for this group.

Fig. 3. One scenario when using uMeeting system

4 Conclusions and Future Work

In this paper, we have presented the uMeeting system, a co-located meeting system on the large-scale tabletop. First, we compare the horizontal interactive surfaces and the vertical ones and draw the conclusion that horizontal surfaces are preferred for the meeting scenario. Then we list some activities that attendees do when attending meetings. We classify the workspace into three scope, that is public, private and group scope. For each scope, we list some specific activities. We implemented the uMeeting system based on uTableSDK and gave a detailed explanation about the architecture and advantages of the uMeeting system.

Future work includes conducting a comprehensive user study to find to how much degree this meeting system helps to facilitate the progress of a specific meeting. Which category of activities in the meeting helps attendees a lot and which helps little. With the changing of devices used in meeting from notebooks and laptop computers to large-scale tabletop, the impact on the attendees' behavior should also be studied.

Acknowledgements. This work is supported by National High-Tech Research and Development Plan of China under Grant No.2009AA01Z336.

References

1. Elrod, S., Bruce, R., et al.: Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration. In: Proc. of CHI 1992, pp. 599–607 (1992)
2. Peltonen, P., Kurvinen, K., Salovaara, A., et al.: It's Mine, Don't Touch!: Interactions at a Large Multi-Touch Display in a City Centre. In: Proc. of CHI 2008, pp. 1285–1294 (2008)
3. Microsoft Surface, http://www.microsoft.com/surface/en/us/default.aspx
4. Dietz, P., Leigh, D.: DiamondTouch: A Multi-User Touch Technology. In: Proc. of UIST 2001, pp. 219–226 (2001)
5. Lanbo Corp, http://www.bjled.com/
6. Bi, X., Bae, S.H., Balakrishnan, R.: Effects of Interior Bezels of Tiled-Monitor Large Displays on Visual Search, Tunnel Steering, and Target Selection. In: Proc. of CHI 2010, pp. 65–74 (2010)
7. uTableSDK, http://utablesdk.codeplex.com
8. Wellner, P.: Interacting with paper on the DigitalDesk. Commun. ACM 36(7), 87–96 (1993)
9. Fitzmaurice, G.W., Ishii, H., Buxton, W.: Bricks: Laying the Foundations for Graspable User Interfaces. In: Proc. of CHI 1995, pp. 442–449 (1995)
10. Laborie, F., Jacquemond, D.: Cooperation in Highly Distributed Design Processes: Observation of Design Sessions Dynamics. In: Proc. of CDVE 2005, pp. 42–51 (2005)
11. Ball, R., North, C.: An analysis of user behavior on high-resolution tiled displays. In: IFIP International Conference on Human-Computer Interaction, pp. 350–364 (2005)
12. Tomfelde, C.M., Wessels, A., Schremmer, C.: Tilted Tabletops:In Between Horizontal and Vertical Workspaces. In: Proc. of TABLETOP 2008, pp. 49–56 (2008)
13. Scott, S.D., Grant, K.D., Mandryk, R.L.: System Guidelines for Co-located, Collaborative Work on a Tabletop Display. In: Proc. of ECCSCW 2003, pp. 159–178 (2003)
14. Scott, S.D., Sheelagh, M., Carpendale, T., Inkpen, K.M.: Territoriality in collaborative tabletop workspaces. In: Proc. of CSCW 2004, pp. 294–303 (2004)