StickySchedule: An Interactive Multi-user Application for Conference Scheduling on Large-scale Shared Displays

Vishal Doshi[1], Sneha Tuteja[1], Krishna Bharadwaj[1], Davide Tantillo[1], Thomas Marrinan[2], James Patton[1], G. Elisabeta Marai[1]

[1]University of Illinois at Chicago
Chicago, IL, USA
gmarai@uic.edu

[2]Argonne National Laboratory
Lemont, IL, USA
tmarrinan@anl.gov

ABSTRACT

Scheduling conferences is a common task in both research and industry, which requires relatively small groups to collaborate and negotiate in order to solve an often-large logistical problem with many nuances. For large conferences, the process can take days and it is traditionally a manual procedure performed using physical tools such as whiteboards and sticky notes. We present the design and implementation of StickySchedule, a multi-user application for use on interactive large-scale shared displays to better enable groups to organize large conference-scheduling data. To evaluate our tool, we present observations from novice users, and authentic use cases with expert feedback from organizers who are heavily involved in large conference scheduling. The main contributions of our work are documenting the collaborative and competitive aspects of conference scheduling, creating a tool that incorporates successful features and addresses identified issues with prior works, and verifying the usefulness of our tool by observing and discussing a variety of use cases, in both collocated and remote-distributed settings.

Author Keywords

Large-scale displays; collaboration; multi-user interaction; conference scheduling.

ACM Classification Keywords

H.5.3. Information interfaces and presentation (HCI): Group and Organization Interfaces; Computer-supported cooperative work.

INTRODUCTION

Scheduling podium sessions and supplementary events for large conferences with 10,000 attendees or more is a complex collaborative process that requires negotiation between multiple organizing chairs. While every conference has an overarching scope, talks are typically broken down into several subcategories, or themes, with each chair assigned to the theme most closely related to their area of expertise. Each chair, therefore, has a unique set of goals and constraints relating to speakers, schedule, venue, and audience. This complex collection of constraints and goals leads to a unique combination of collaborative and competitive actions. Given all their goals and constraints, organizers must assign all sessions to an initial date, time, and room. After this step, the organizers must resolve conflicts with the initial assignments and finally save the configuration for dissemination.

Such large-conference tasks preclude the use of ad hoc combinations of freely available, cloud-based and web-based collaboration tools. Because of this, current large conference scheduling techniques range from traveling to and gathering in a meeting room and using a whiteboard to manually schedule each event, to using specialized software where each organizer teleconferences into a scheduling session. Each of these methods provides affordances and has shortcomings in attempting to aid scheduling a large conference collaboratively.

At the same time, large-scale tiled display walls have become pervasive in today's society – they are currently found in public installations [14], financial institutions [6], and high-end visualization laboratories [11]. These digital spaces provide users with a large interactable surface, and foster multi-user collaboration [2]. Use of large-scale shared displays provides an opportunity to transform current workflows that benefit from large physical space, digital artifacts, and collocated or remote collaboration. Currently, there is no software system that leverages multi-user large-scale display technology for scheduling a large conference.

In this work, we present the design and implementation of StickySchedule, a multi-user application for use on interactive large-scale shared displays. StickySchedule was designed to better enable small groups to organize conference-scheduling data. We describe the challenges that had to be overcome in order to incorporate the main advantages of both the manual and software driven approaches for scheduling a conference, while also addressing their main drawbacks. Through observations

PerDis '17, June 07-09, 2017, Lugano, Switzerland

ACM ISBN 978-1-4503-5045-7/17/06$15.00
http://dx.doi.org/10.1145/3078810.3078817

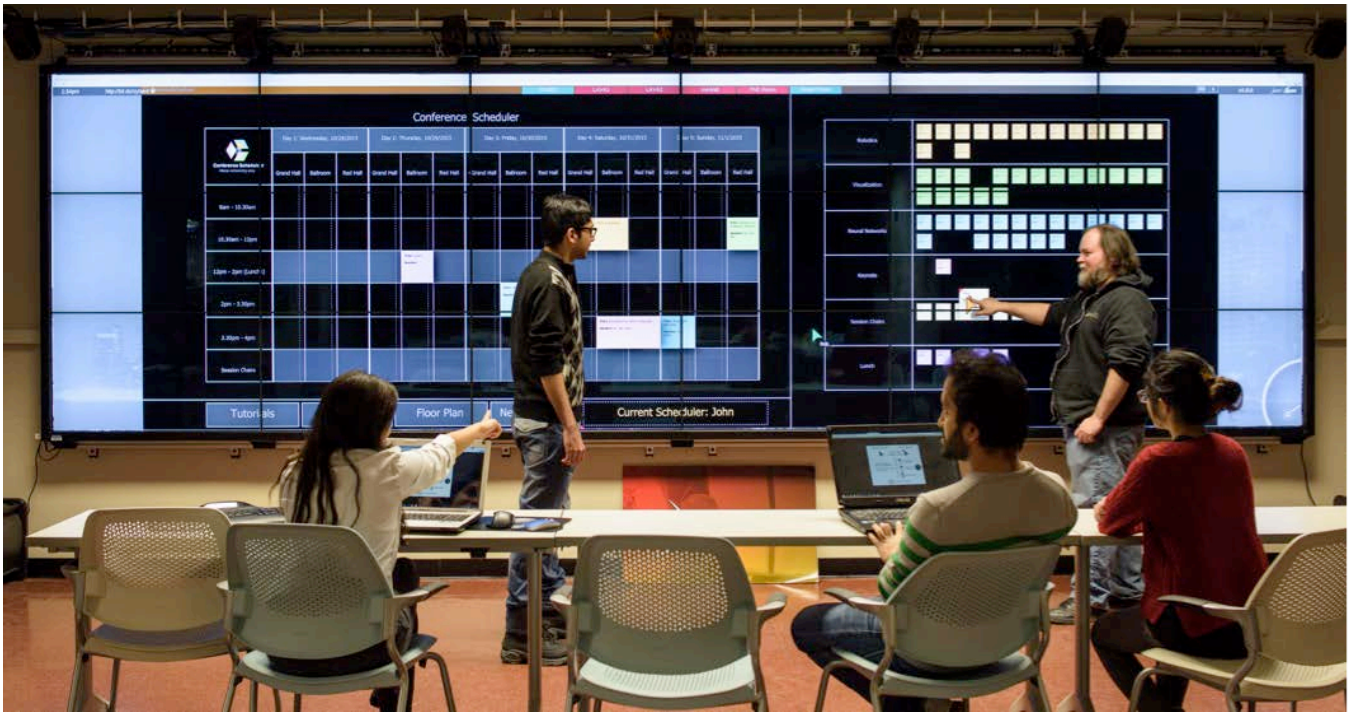

Figure 1. Simulated organization committee scheduling a conference using StickySchedule. Interaction is supported from both personal devices (e.g. a mouse from one's laptop) and touch directly on the large-scale shared display

from novice users and authentic use cases with expert feedback from experienced organizers, we show that StickySchedule (which supports direct touch interaction on the large-scale shared display, distant interaction with mouse pointers, and remote collaboration between multiple shared displays) more completely meets the requirements associated with large conference scheduling tasks. Figure 1 illustrates the use of StickySchedule by a collocated team of organizers.

RELATED WORK

Large-scale display systems have been shown to improve collaboration and significantly amplify the way users make sense of complex data [1,2,3]. An increase in both number of pixels and physical size enables users to simultaneously view overview and detail of rendered artifacts. Also, physical movement can be used to navigate through the data without the need of indirect interaction devices.

Shared use of large-scale displays has also shown great promise for enhancing collaborative actions. Stødle et al. [16] present work on extending laptop screens with Network Accessible Displays. They show that a large tiled display wall (up to 22 Mpixels) can be remotely connected to and utilized as an external monitor. To enable collaboration, multiple users can grab a subset of screens from the tiled display wall. Nutsi et al. [10] provides a set usability guidelines for developing multi-user applications for large-scale displays. One important finding is to ensure that the data being visualized is readable and understandable for bystanders as well as users currently interacting with the application.

Prior to our work, no software had been developed to leverage large-scale displays for conference scheduling. While small-scale conferences can typically be scheduled through a combination of online or offline tools (from emailing spreadsheets to web-based collaborative systems), the scale and complexity of large conference scheduling precludes the use of such tools in this context. For such large conferences, a number of specialized online tools allow organizers to upload conference data and to collaboratively create a calendar of events [4,5,12]. Each collaborator views and edits data from a personal computer, requiring teleconferences for synchronously discussing proposed schedules, or persistent notes for asynchronous scheduling. Wang et al. [19] developed Event Maps, which extends the basic online tools by also incorporating use by attendees during a conference. The main drawbacks of these tools are that 1) each organizer views data remotely from their own device, which limits face-to-face communication, and 2) personal computers do not possess sufficient screen resolution to depict the entire conference calendar in one view, which forces organizers to look at an undetailed overview or scroll through a subset of information in more detail. These limitations make collaboration among organizers difficult.

Zhang et al. [20] present Cobi as an alternative means for scheduling large conferences. The Cobi system relies on crowdsourcing preferences and constraints from the community to enable organizers to make informed decisions. While the Cobi system holds promising results, it does not fully address the issues present in scheduling large conferences. First, it requires active participation from a sufficient number of attendees, and second, it only provides the final decision makers with an initial outline. Therefore, it would still be required for the organization committees to meet and create a final schedule.

Because of the difficulties in using teleconference-based scheduling applications, many large-conference organizers begrudgingly use lower-tech solutions to better facilitate communication and cooperation. This involves traveling to a common location (sometimes overseas), gathering in a large room, and using physical materials – typically paper, whiteboards, sticky notes, and dry erase markers.

To facilitate a smooth transition from using a whiteboard to using a large display application, we investigated the study done by Tang et al. [18], who looked at group-based whiteboard use in order to determine design implications for large display applications. A whiteboard container was utilized as a location where information can be easily revisitable and readily updatable. For our application, we additionally support collaborative and competitive tasks.

Because we also aim to support group-to-group (mixed presence) collaboration with a high degree of awareness between groups [8,17], we developed StickySchedule using the SAGE2 application API [7,13]. SAGE2 is a windowed operating environment for large-scale displays that provides remote collaboration capabilities, and simultaneous multi-user interaction using mouse pointers and direct touch.

METHODS

Our design followed an Activity-Centered-Design paradigm [9], which is an extension of the classic Human-Centered-Design paradigm. The approach places particular emphasis on the user tasks and functional specifications. We implemented this paradigm through an iterative process where the research team met regularly with experienced conference organizers to confirm requirements, explore prototypes, refine the design, test the software, and verify that requirements were being satisfied.

Requirements Analysis

Requirement gathering started with a semi-structured interview of a biomedical conference Chair, followed by an observation session. Figure 2 shows a collaborative scheduling session from the Chair's experience the previous year – the session was performed using a regular whiteboard, sticky notes, and dry erase markers. The process had suffered from physical limitations (re-taping sticky notes that had fallen etc.), and had led the organizers to consider the idea of flying in to the meeting a large display, to be used in scheduling.

The interview established who the users of the system would be and how often the system would be used, a prioritized list of the main tasks performed by users, the data sources and flow of data, and non-functional requirements such as scalability to different screen sizes and support for local and remote collaboration.

Figure 2. Committee scheduling a conference using sticky notes and a whiteboard. The organizer in the blue shirt has been delegated to re-tape fallen sticky notes.

Interview Data and Task Analysis

In addition to the requirements list, the interviews elucidated several scheduling constraints. These range from a presenter who is giving multiple talks (cannot schedule talks at the same time) to ensuring attendees can easily listen to talks related to their interests throughout the day (related sessions should be presented in the same or a nearby room). The interviews further specified requirements related to the collaborative and competitive actions associated with the scheduling task: multi-user functionality; support for remote collaboration; visual scalability; creating, drag-and-dropping notes; enabling turn-taking for conflict resolution; marking conflicts; viewing supporting information to resolve conflicts; saving the schedule. We note the emphasis on turn-taking with multiple users in order to facilitate effective conflict resolution and collaboration.

Informal interviews with two organizers of a large computer science conference further confirmed these requirements. The computer science organizers further provided access to their most recent conference program, hotel and room constraints, as well as a sampling of conflicts and conflict resolutions, which in the early scheduling stages of their conference had been painfully conducted and recorded via email. This sample mid-sized dataset contained 159 papers grouped into 34 podium sessions and spanning four themes. The conference took place over four days, had five timeslots per day, and had four rooms available for parallel tracks.

Interaction Analysis

Manual Approach. When using a manual approach, conference organizers have been physically gathered in one location and utilized the large space afforded by meeting rooms, thereby fostering small group collaboration. Their method follows a competitive / collaborative approach, which mixes turn taking with negotiation. Sticky notes have been used to represent each podium session or other event, with individual colors corresponding to the themes of the conference. A grid is drawn on the whiteboard to make a calendar marking the dates, times, and rooms available for the conference. Chairs then take turns placing a sticky note from their theme into a desired location in the calendar, followed by an explicit negotiation phase. The process has a number of drawbacks, from insufficient unique note colors to short-lived note adhesive, which causes some sessions to fall off the whiteboard. The process is further time consuming and susceptible to human error, e.g. recalling a particular session/timeslot constraint or transcribing data from the whiteboard to a digital format for dissemination.

Web-based Approach. An alternative approach is to use web-based conference scheduling software, e.g. papercept.net [12]. This approach allows organizers to teleconference instead of having to travel to a unified location, load data from prior years, and eliminate the transcription overhead. However, the teleconferencing makes difficult interaction, negotiation, and collaboration amongst the group, and the entire conference calendar does not fit on a standard desktop monitor.

Visual and Interaction Encoding

Prototype Data. To make testing by non-expert users possible, we used the sample data to generate a synthetic dataset that consisted of 57 podium sessions of four different themes over five days, with three rooms. The dataset included six timeslots each day, out of which the 12 p.m.-2 p.m. slot was reserved for lunch, while plenary sessions were reserved for after 4 p.m.

Environment. By selecting SAGE2 as a platform to develop our StickySchedule application, we were able to satisfy our two highest priority requirements: providing multi-user functionality and enabling the application to scale to ultra high resolutions. Additionally, using SAGE2 helped us

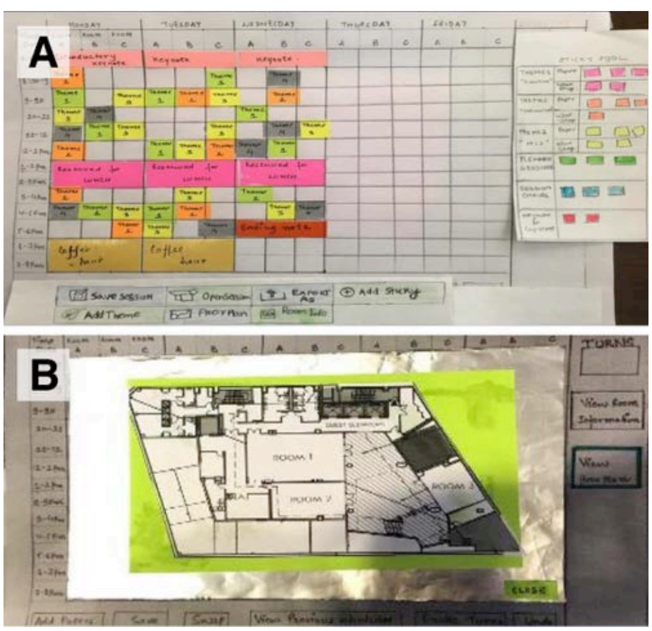

Figure 3. Sketch prototypes for StickySchedule. Panel A displays the main scheduling calendar interface. Panel B shows the conference floor plan as an example of the supplemental information provided to organizers.

satisfy our lowest priority requirement for free since it supports remote collaboration through synchronized shared applications between display walls.

Prototyping. A series of low-fidelity prototypes were sketched on paper and discussed, combined, and permutated with expert organizers in successive iterations. Figure 3 shows final prototypes approved by the expert organizers.

To have a low barrier to entry and a mild learning curve, the design followed the spreadsheet format already in use by conference organizers, optimized for large displays that would enable multi-user local or remote collaboration. We further emphasized a design that replicated the organizers' use of sticky notes for scheduling: we provided a "sticky pool" area for initially containing all colored sticky notes – each representing a single podium session or supplementary event, and separate area with a grid-based calendar for scheduling dates, times, and rooms. The sticky notes are each assigned a color according to their session theme. Below the calendar grid area, a widget area provides auxiliary information regarding the floor plan, room information, tutorials, and scheduler's turn.

Interaction Flow. The schedule grid is initialized and laid out according to a user-editable configuration file, and color-coded sticky notes are created and displayed in the pool area. The system supports two alternating types of activities, in accordance to the flow and interactions we gathered and observed during requirements engineering: a turn-taking phase, and a negotiation phase. The system can be set to either phase manually, and by default starts with the turn-taking phase, as specified by the user requirements analysis.

Turn-Taking Phase. During the initial assignment phase, Organizing Chairs take turns. The configuration files can specify information about the scheduling committee, for example the names of theme chairs and the theme they are in charge of. Using this information, we support the user-required turn-taking capability. A "Next Scheduler" button is pressed once a user is done with her scheduling turn. The name of the Chair who will be scheduling next is automatically displayed on the scheduler label. At any point users can access the room information, merge multiple rooms, access the tutorial, etc.

Low-level conflicts are highlighted and resolved during this stage. Constraints are automatically enforced – if a user tries to drop a sticky note to a location that will violate one of its constraints, the action does not complete and the sticky note jumps back to its prior location. A pop-up message opens, stating the reason for preventing the action. StickySchedule handles errors that violate constraints such as attempting to schedule a session in a slot that already has another session, room that cannot fit the estimated number of attendees, or reserved slot (e.g. lunch or a plenary talk).

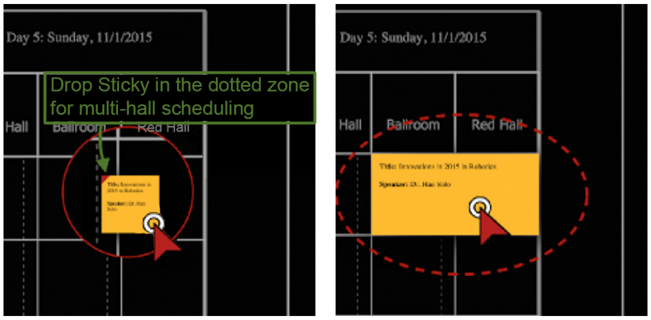

Figure 4. Multi-room scheduling. Dropping a sticky note in the dotted zone between two adjacent rooms will schedule the session for both rooms.

Negotiation Phase. Higher-level and subtler conflicts are tackled once all sticky notes have been allocated a preliminary time slot on the schedule. During negotiation phases, users can agree to trade slots through group and subgroup discussions, and enact these swaps through the large shared display.

In both phase types, adjacent halls can be merged into a single room for sessions that expect a high audience. To merge two halls, users can drag and drop a sticky note to the common boundary of two adjacent halls (Figure 4).

Implementation. We implemented StickySchedule as a single large Scalable Vector Graphic (SVG) canvas by using the open-source SAGE2 middleware, JavaScript and Snap.svg [15]. We implemented custom interactive widgets to closely mirror the experience of placing sticky notes on a whiteboard. Through SAGE2, our application can harness remote collaboration features in addition to local multi-user interaction. SAGE2 enables geographically distributed groups to simultaneously view and interact with the same application. This is particularly useful since overseas travel can be expensive, and organizing committees are typically an international ensemble of experts.

EVALUATION

To evaluate the system, we performed initially a pilot user study with novice users, followed by expert evaluation. We adopted an initial evaluation with novice users for two reasons. First, as indicated by the interviews with experts, new researchers join organizing committees each year, and so usage by novices was of particular interest. Second, collecting a reasonably large group of expert organizers solely for evaluation purposes was unfeasible, due to the experts' time and spatial constraints. Nevertheless, beyond the initial novice study, we observed and report here on the usage of StickySchedule by collocated experts, as well as during a remote collaboration setting.

First-time Usage

We asked twenty computer science graduate students to perform two simplified scheduling tasks using both a light version of our system and a real whiteboard with sticky notes, in random order. After a brief demonstration, we conducted 5 study runs in groups of 4 people each.

In both settings, each group member was handed a constraint sheet (nine to ten papers with "must be after paper X", room, day, or slot-hours constraints). Several further conflicts were introduced in the sheets and in the calendar grid (e.g., pre-booked slots for lunch breaks). Values were assigned both to sessions and timeslots (the higher the number the more valuable the session or coveted the timeslot). These values allowed us to simulate organizer competition, since each Chair desired to schedule valuable sessions of their theme in the coveted timeslots.

We compare the two approaches in terms of the overall score of the final schedule (sum of session value * timeslot value) and by the task success and time-to-completion. In both approaches and all runs, the groups successfully solved the task assigned. Analysis of the results further showed no statistically significant difference of the overall score between the two approaches: 141.4 ± 1.3 (whiteboard) and 142.4 ± 4.8 (StickySchedule). The average time-to-completion was also similar: 26 min 55 sec and 27min 26 sec, respectively. These findings show that the StickySchedule approach can successfully replace the physical whiteboard approach, while circumventing shortcomings of the manual approach, and offering expanded functionality.

Collocated Expert Usage

In a second stage, we conducted a qualitative analysis with two expert organizers. The organizers had both expressed interest in using large displays for conference scheduling. Both organizers were given an opportunity to use the application.

The experts provided enthusiastic feedback with the scheduling application ("It looks wonderful". "You have met the requirements". "Thank you to all for this opportunity". "How soon can I use this?"). The feedback further expressed satisfaction with the real estate provided by the large displays, and the displays' ability to show lots of information, without loss of context or need for scrolling. They commented that the display successfully captured the relevant data, and that the system had intuitive natural interactions that were superior to existing software approaches and made for a fluid transition from a manual whiteboard-based approach.

Particular comments further noted how important the body visual cues were to the scheduling process, in particular during the interactive negotiation phase. The ability to accommodate multiple users in the same space, and the ability to take turns, similar to the round-robin approach used for conflict resolution in physical rooms, was much appreciated.

The experts particularly appreciated additional capabilities characteristic to the application, such as automatic reports of violated constraints, visual scalability to screens of varied dimensions, flexibility, and the fluid transition from a manual whiteboard-based approach.

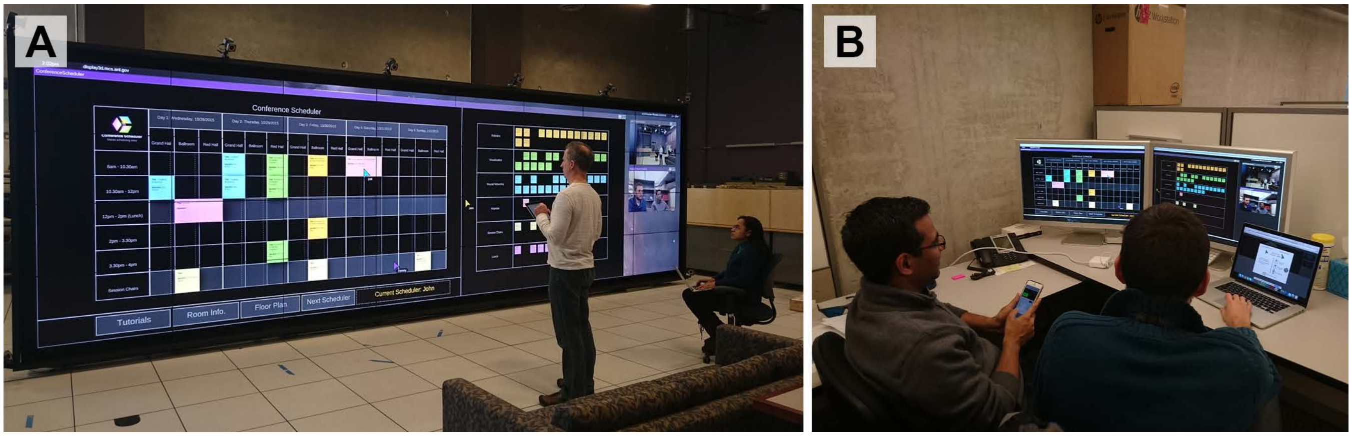

Figure 5. Mixed presence collaboration between a partially distributed team. StickySchedule is being used to collaboratively schedule a conference across distance, with each group having a unique display setup. All interactions are completely synchronized and a videoconference enables verbal communication.

Remote Collaboration Usage

Remote collaboration was tested between two groups using SAGE2's perfect mirror mode (32:9 aspect ratio). One group used a large-scale display with a resolution of 15360x4320, while the other group used a dual desktop monitor setup with a resolution of 5120x1440. A videoconference was set up between the sites and video from both sides was shared via SAGE2. The two groups (each with two users) used a synthetic conference dataset to simulate scheduling a conference. Fluid and effective interaction, negotiation, and collaboration were observed amongst all four users. Pointer icons were present for each user providing awareness indicators for remote collaborators, and the videoconference enabled verbal communication (Figure 5).

DISCUSSION AND CONCLUSION

The case study evaluation with conference organizers shows that large displays and StickySchedule provide a similar look and feel to the traditional whiteboard approach for conference scheduling. This similarity fosters group collaboration. Additionally, StickySchedule removes many constraints present in that traditional workflow. Digital sticky notes cannot fall off the wall, there is no transcription time since original data can be automatically pre-processed for use in the application and post-processed for dissemination of the final schedule, and constraints are automatically reported to the users. Additionally, supplementary data such as room layout are all present in a unified location.

Our initial observations of first-time users indicate that StickySchedule provides an intuitive interface, and enables organizers to accomplish tasks efficiently. In addition to local groups successfully scheduling conference events, we observed remote groups using StickySchedule. The ability to utilize the same groupware for remote scheduling in addition to collocated scheduling provides unique opportunities for organizing committees. The seamless interactions we observed in the two distributed groups shows good potential to reduce travel while still maintaining a high level of coordination and ease of negotiation.

In conclusion, we have introduced StickySchedule, a novel scheduling application that enables conference scheduling on interactive large-scale shared displays. We documented the requirements engineered through interviews with and observation of experienced conference organizers, with particular emphasis on collaborative / competitive interactions. Our human-centered-design addresses these interaction requirements by enabling both turn-taking and simultaneous input collaboration. The design implements a friendly sticky note and calendar grid design, as well as drag-and-drop interactions, both of which are consistent with traditional scheduling applications.

Beyond traditional physical approaches, StickySchedule provides multi-user support for local and remote collaboration. Unlike traditional systems, StickySchedule can also indicate and enforce scheduling constraints and can provide information to support the scheduling process, such as floor plans and room details. The application is designed for large displays, but also scales down elegantly to regular monitor sizes.

While conference scheduling is a relatively specialized task, the types of interactions users engage in are generalizable to event organization as well as other scheduling and logistics based task. Considering the enthusiastic reception and feedback from multiple conference organizers, we believe that the lessons learned from StickySchedule will be valuable in research and industry settings that require small groups to collaborate and negotiate in order to solve nuanced logistical problems.

ACKNOWLEDGEMENTS

We would like to thank the students who participated in our user study, as well as the conference organization experts for their valuable feedback. This work was supported in part by the National Science Foundation (NSF), awards NSF CAREER IIS-1541277, ACI-1339772, OCI-0943559 and CNS-0959053.

REFERENCES

1. Christopher Andrews, Alex Endert, and Chris North. 2010. Space to Think: Large High-Resolution Displays for Sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 55-64. http://dx.doi.org/10.1145/1753326.1753336.
2. Robert Ball and Chris North. 2005. Analysis of User Behavior on High-resolution Tiled Displays. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT'05), 350-363. http://dx.doi.org/10.1007/11555261_30.
3. Xiaojun Bi and Ravin Balakrishnan. 2009. Comparing Usage of a Large High-Resolution Display to Single or Dual Desktop Displays for Daily Work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1005-1014. http://dx.doi.org/10.1145/1518701.1518855.
4. BusyConf. Retrieved September 15, 2016 from BusyConf: http://busyconf.com/.
5. ConfTool. Retrieved September 15, 2016 from ConfTool: http://www.conftool.net/en/index.html.
6. Valentin Heun, Anette von Kapri, and Pattie Maes. 2012. Perifoveal Display: Combining Foveal and Peripheral Vision in One Visualization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (UbiComp '12), 1150-1155. http://dx.doi.org/10.1145/2370216.2370460.
7. Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Krishna Bharadwaj, Victor Mateevitsi, Luc Renambot, Lance Long, Andrew Johnson, and Jason Leigh. 2014. SAGE2: A New Approach for Data Intensive Collaboration Using Scalable Resolution Shared Displays. In *Proceedings of the IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing* (CollaborateCom), 177-186. http://dx.doi.org/10.4108/icst.collaboratecom.2014.257337.
8. Gregor McEwan, Markus Rittenbruch, and Tim Mansfield. 2007. Understanding Awareness in Mixed Presence Collaboration. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces* (OZCHI '07), 171-174. http://dx.doi.org/10.1145/1324892.1324924.
9. Donald A. Norman. 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books, Inc., New York, NY, USA.
10. Andrea Nutsi and Michael Koch. 2015. Multi-User Usability Guidelines for Interactive Wall Display Applications. In *Proceedings of the 4th International Symposium on Pervasive Displays* (PerDis '15), 233-234. http://dx.doi.org/10.1145/2757710.2776798.
11. Charilaos Papadopoulos, Kaloian Petkov, Arie E. Kaufman, and Klaus Mueller. 2015. The Reality Deck-an Immersive Gigapixel Display. *IEEE Computer Graphics and Applications*, vol. 35, no. 1, 33-45. https://doi.org/10.1109/MCG.2014.80.
12. Papercept. Retrieved September 15, 2016 from Papercept: https://controls.papercept.net/.
13. Luc Renambot, Thomas Marrinan, Jillian Aurisano, Arthur Nishimoto, Victor Mateevitsi, Krishna Bharadwaj, Lance Long, Andrew Johnson, Maxine Brown, and Jason Leigh. 2016. SAGE2: A Collaboration Portal for Scalable Resolution Displays. *Future Generation Computer Systems*, vol 54, 296-305. http://dx.doi.org/10.1016/j.future.2015.05.014.
14. Markus Rittenbruch. 2013. CubIT: Large-scale Multi-user Presentation and Collaboration. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces* (ITS '13), 441-444. http://dx.doi.org/10.1145/2512349.2514923.
15. Snap.svg. Retrieved September 15, 2016 from Snap.svg: http://snapsvg.io/.
16. Daniel Stødle, John Markus Bjørndalen, and Otto J. Anshus. 2007. The 22 Megapixel Laptop. In *Proceedings of the 2007 Workshop on Emerging Displays Technologies: Images and Beyond: the Future of Displays and Interaction* (EDT '07), Article 8. http://dx.doi.org/10.1145/1278240.1278248.
17. Anthony Tang, Michael Boyle, and Saul Greenberg. 2004. Display and Presence Disparity in Mixed Presence Groupware. In *Proceedings of the Fifth Conference on Australasian User Interface* (AUIC '04), vol. 28, 73-82.
18. Anthony Tang, Joel Lanir, Saul Greenberg, and Sidney Fels. 2009. Supporting Transitions in Work: Informing Large Display Application Design by Understanding Whiteboard Use. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work* (GROUP '09), 149-158. http://dx.doi.org/10.1145/1531674.1531697.
19. Jingtao Wang, Danny Soroker, and Chandra Narayanaswami. 2010. Event Maps: a Collaborative Calendaring System for Navigating Large-scale Events. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '10), 3691-3696. http://dx.doi.org/10.1145/1753846.1754040.
20. Haoqi Zhang, Paul André, Lydia B. Chilton, Juho Kim, Steven P. Dow, Robert C. Miller, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2013. Cobi: communitysourcing large-scale conference scheduling. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13), 3011-3014. https://doi.org/10.1145/2468356.2479597.