

Tidebreak: Taking Room-based Ubiquitous Computing Technology Out of the Research Lab and into the Real World

Brad Johanson
CTO and Co-Founder, Tidebreak, Inc.
bradj@tidebreak.com

Introduction

Tidebreak was founded in early-2004 by two alumni of the Stanford Interactive Workspaces project [4], to produce commercial versions of the iROS system [6], and extend the technology in new directions based on the needs of real-world users. During the last year and a half, we have investigated the needs of businesses and academic institutions for room-based collaboration technology. Based on what we have learned, we have evolved and extended the research quality code into a commercial product, TeamSpot, which addresses the needs of these real-world users. Along the way we have had to address how to package discrete capabilities into a useful product, how to introduce users to the new capabilities provided, and how to effectively introduce new paradigm shifting features while still supporting existing work practices. This paper gives an overview of some of our experiences in the context of the TeamSpot product and details some of our general observations.

TeamSpot: A Solution for Walk-up Interaction Spaces

TeamSpot packages iROS components and several extensions to allow one or more people to walk up and use their laptops with public machines on large displays in technology enhanced public or communal spaces. Other members of the Interactive Workspaces project investigated the initial concept in a custom trial deployment for student use in one of Stanford's libraries which was called TeamSpace [7]. In a TeamSpot, after a simple connection process, users have access to several interaction components. Using the PointRight system, each users can move their cursor off the top of their screens and onto the large public displays to begin controlling them with their own keyboards and mice. To move to additional adjacent large displays, the cursor need only be moved off the edge of one screen onto the next. This simple and quick mechanism for shifting control allows users to quickly take turns pointing out and making changes to shared work being done on the public displays without interrupting the flow of collaboration to pass around a keyboard, switch cables back and forth, or issue orders to the single person controlling input to a publicly displayed screen.

Another important user need addressed in TeamSpot is moving information between the various displays. This is accomplished in a TeamSpot primarily using CrossWarp, which allows users to transfer files or URLs to any other machine in the space using drag-and-drop to a small widget always present on each machine running the software. Along with PointRight, this mechanism allows users to move personal files to the public screens to work on them along with the group, move work from the public screens back to personal laptops for polishing before more group work, or take home a copy of a result produced on a public screen.

A final useful tool which is included in TeamSpot is Action Archive, a system which captures a log of what files and URLs are transferred during the meeting, certain significant interaction events, and any 'comments' that users manually add to the Archive. During a session, users may view the Action Archive in their web browser where it appears as a table displaying what actions have happened over time. Capture begins automatically when a set of new users join an empty space, and as users leave they are given the option of saving a copy of Action Archive's log of their session. The automatic nature of the Action Archive was inspired by research on the Workspace Navigator project [2] which showed that it often does not become clear which information is important until sometime after the meeting ends. The automatic capture is predicated on the assumption that information important enough to share is worth capturing.

Key Issues

As we work to move concepts from the research code into commercial products, we have started to address a number of issues that would be relevant for other next-generation meetings spaces.

Issue #1: Minimizing Software Setup and Connection Complexity and Time. Users coming to a meeting or collaboration session are focused on the task at hand and typically only have an hour or two total to spend, so few will bother to even try new technology if setup takes more than a few minutes. As an example, configuration of the original prototype research software was so complex, that some members of the Interactive Workspaces research group would not bother installing the software despite known benefits to collaboration once it was installed. After several iterations, the TeamSpot install process has been simplified so that first time installation on a personal machine takes about five

minutes, joining a new space for the first time once the client is installed takes about a minute, and rejoining a previously used space takes a few seconds.

Issue #2: Providing powerful mechanisms for collaboration while reducing complexity to minimize distraction from the actual collaboration process. Any extra cognitive load placed on users by the software can distract from the thread of discussion being pursued by participants, where focus on individual information fragments lasts for only six seconds on average [1]. An example of how TeamSpot is designed to account for this is the control transfer mechanism in PointRight which allows a user to take control of a public screen in less than a second. The CrossWarp mechanism was similarly designed to be straightforward, with an omnipresent drop target that can be used to transfer content.

Issue #3: Providing adequate information security without impeding the collaboration process. Unlike in the research environment where much of the software was conceived and all users were trusted, TeamSpots are designed to be deployed in communal areas like University libraries and corporate conference rooms. In such spaces it is important that control sharing only be allowed for local users, and that information shared in the space be available only to parties who are physically present, or have been invited to participate remotely. To keep network information private, one of the first upgrades we made to the underlying network coordination infrastructure, the Event Heap [3], was to add support for SSL encrypted connections to prevent packet sniffing.

Issue #4: Providing a privacy model that differentiates between physically local and remote participants. Initial connection to a TeamSpot is controlled by forcing the user to enter an authentication code that is displayed on a local public screen. This prevents a remote user from connecting unless local participants help them do so by reading him the code. At the end of meetings, users are automatically disconnected if they suspend their machines and leave the space. Access to the Action Archive is also controlled by providing machine specific URLs to access the Archive which are only valid as long as that machine is connected to the local TeamSpot Host machine. We are still in the process of understanding security and privacy needs. Issues we have yet to address include providing different levels of access privileges for users to large displays and other public resources, and allowing sub-groups to have sidebar conversations without risk of others in the local space being able to track what they are doing.

Conclusion and Future Directions

TeamSpot will be deployed at several academic institutions, including Stanford, MIT, and Emory, during the 2005-2006 academic year. Tidebreak is working with these institutions to use the underlying instrumentation capabilities of our infrastructure to gather anonymous usage data at each of the installations [5]. Through this

and explicit studies run with student groups, we hope to better understand how interactive spaces can enhance collaboration among students for group projects. Results from these studies will then be applied to future software development.

For corporate environments, a dominant need is supporting group-to-group collaboration between two or more sets of collocated participants. We are investigating the needs and requirements in this area as we work with pilot customers and design next-generation products to address their needs. Some important issues in this area are: allowing all participants to view synchronized material on large public screens while still allowing private workspace on personal machines, and providing mechanisms for interactions like sidebar conversations.

Biography

Brad Johanson is the lead architect for the computing infrastructure system that lies at the heart of Tidebreak's products. His Ph.D. research at Stanford focused on computing infrastructures to support interactive spaces, with emphasis on how backend middleware software systems need to be designed to support user interface mechanisms [3]. He was also the lead designer on the PointRight mouse/keyboard redirection system and has participated in the design of many of the other HCI mechanisms included in Tidebreak's products. Previously he has held technical positions at Silicon Graphics, S3 and Intel Corporation.

Acknowledgements

Tidebreak would not exist without the support and encouragement of too many people to mention by name. Special thanks go to Andrew Milne, Tidebreak's co-founder, for his helpful suggestions and criticisms of this paper.

References

1. Baya, V., *Information Handling Behavior of Designers During Conceptual Design: Three Experiments*. Ph.D. Dissertation, Mechanical Engineering. 1996, Stanford, CA, USA: Stanford University.
2. Ionescu, A., M. Stone, and T. Winograd, *WorkspaceNavigator: Capture, Recall and Reuse using Spatial Cues in an Interactive Workspace*. Technical Report, TR2002-04. 2002, Stanford, CA, USA: Stanford University. 8. <http://www.stanford.edu/~arna/persist/wkspcNav-286.pdf> Verified: 11/2002).
3. Johanson, B., *Application Coordination Infrastructure for Ubiquitous Computing Rooms*. Ph.D. Dissertation, Electrical Engineering. 2003, Stanford, CA, USA: Stanford University. 231.
4. Johanson, B., A. Fox, and T. Winograd, *The Interactive Workspaces project: experiences with ubiquitous computing rooms*. IEEE Pervasive Computing, 2002. 1(2): p. 67-74.
5. Milne, A., *An Information-Theoretic Approach to the Study of Ubiquitous Computing Workspaces Supporting Geographically Distributed Engineering Design Teams as Group-Users*. Ph.D. Dissertation, Mechanical Engineering. 2005, Stanford, CA, USA: Stanford University.
6. Ponnekanti, S., et al. *Portability, Extensibility and Robustness in iROS. In 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*. 2003. Dallas-Fort Worth, Texas, USA: IEEE: p. 11-19.
7. Shih, C.C., et al. *Teamspace: A Simple, Low-Cost and Self-Sufficient Workspace for Small-Group Collaborative Computing*. in *Submitted to the Conference on Computer-Supported Cooperative Work*. 2004. Chicago, IL, USA: p.