Metamouse: Multiple Mice for Legacy Applications

Kurtis Heimerl
UC Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720 USA
kheimerl@cs.berkeley.edu

Divya Ramachandran
UC Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720 USA
divya@cs.berkeley.edu

Joyojeet Pal
University of Washington
Mary Gates Hall, Suite 370
Seattle, WA 98195 USA
joyojeet@u.washington.edu

Eric Brewer
UC Berkeley
Soda Hall, 2599 Hearst Ave
Berkeley, CA 94720 USA
brewer@cs.berkeley.edu

Tapan Parikh
UC Berkeley
102 South Hall
Berkeley, CA 94720 USA
parikh@ischool.berkeley.edu

CHI 2009, April 4 – 9, 2009, Boston, MA, USA
ACM 978-1-60558-246-7/09/04.

Abstract
Single Display Groupware (SDG) solutions have been used to create software for disadvantaged children, particularly in the developing world. SDG allows for greater utilization of the limited infrastructure available to these kids. However, SDG faces challenges in working with legacy applications. Our technology, called *metamouse,* takes a step toward an integrated multi-user application by allowing users to collaborate within unmodified legacy educational software. We conducted a preliminary qualitative user study of our technology with educational software in schools around Bangalore, India. We found that Metamouse is easily learned, encourages collaborative discussion among students, and allows for the use of existing educational applications with no modification.

Authors Keywords
Education, Developing Regions, Single Display Groupware, Shared Computers

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies, Graphical user interfaces

Introduction
Several million children, especially in the developing world, never use a computer without a partner sharing the mouse and keyboard. Despite evidence that sharing

a computer is the norm due to resource constraints in several parts of the world [7] there is little work within the HCI community in examining ways of making computers easily shared, especially for use by children. Work with Single Display Groupware (SDG) [12] is particularly relevant to scenarios where screens and input need to be more equitably sharable. Recent work on the use of multiple mice for learning among primary school children has been a valuable step in this direction for two important reasons. First, researchers found mouse-sharing intuitive and usable [9]. Second, the use of multiple mice showed learning gains among children in computer aided learning scenarios [8].

Despite these encouraging early findings, two important barriers remain in deploying shared input technologies in real world settings. First, most legacy software would have to be significantly modified to effectively use multiple mice, and among other issues, many business challenges exist in doing this. The second barrier, potentially of great interest from the HCI perspective, is that of efficiently encouraging coordinated on-screen decision-making using the multiple mice. Most trials of Microsoft's Multimouse [7,8,9], while emphasizing the importance of collaboration, have been impeded by working within a "fastest-finger-first" race-clicking, or repetitive click modes that require all users to click on the same link for the function to continue. Our challenge was to work with legacy software, and yet provide an alternate way for allowing coordinated clicking to capture the proven learning gains of collaboration.

This paper introduces a framework for coordinated mouse use across multiple users by creating a *metacursor* at the average location of all on-screen mice in a shared scenario. Mouse clicks are disregarded unless the user cursors agree on the location of this metacursor. We present early results of field tests in real-world use scenarios among children who habitually share computers. Our results show that such a system is intuitive and usable, and increases collaboration by offsetting the problem of "gaming the system" through random clicks, a problem earlier recognized in multimouse use [6].

Related Work

There is work on SDG applications for collaborative education [10]. Microsoft's Multimouse [7,8,9] is a direct precursor to metamouse. Druin et al. [2] created a collaborative digital library interface and explored possible mechanisms for enforcing collaboration. We consider our research a continuation of these works, applying modifications of Druin's techniques to legacy educational applications.

We have found little work on utilizing SDG with legacy educational software. MIDDesktop [11] is one work, allowing for single-user Java swing applets to operate as multiple-mouse aware SDG applications without modification. Our work is more general, interfacing with any Windows XP application. There is research in this area dealing with integrating SDG technologies with the existing windowing systems [4]. We focus on schemes allowing for functional and intuitive use of legacy applications in SDG environments.

There is a large body of work relating to the mouse averaging technique we've used. Aggregate Pointer [5] uses a technique very close to ours, averaging the position of the various cursors into one metacursor. However, this is done with a more complicated

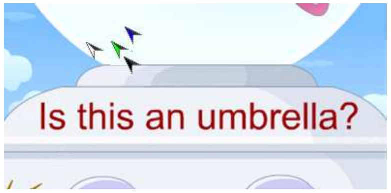

Figure 1. The metamouse in "green" mode when the user mice are close to each other. Clicks pass through the metacursor to the application.

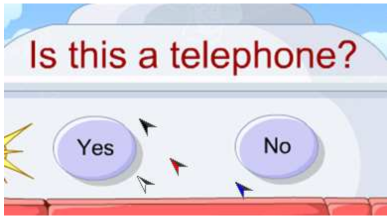

Figure 2. The metamouse in "red" mode when the user mice are dispersed. Clicks will be caught and disregarded.

clustering mechanism, allowing for multiple metacursors. Although interesting, it does not map to legacy software very well. Lauren Bricker's Colt system [1] has numerous techniques for cooperative interaction. The metacursor idiom is analyzed heavily in this work. Movement voting schemes are also analyzed. These techniques form the basis for the techniques we develop and use with legacy software.

We found no instances of our click-after-consensus scheme in any previous work, and we consider this scheme to be one of our primary contributions.

Design

Metamouse is built on CPNMouse [14], a research implementation of SDG for Microsoft Windows. This system allows for legacy software to be used in the SDG environment. It does this in the same way as Multimouse and CPNMouse, where each mouse action is viewed as an action from the singular system mouse.

These traditional SDG stacks provide one uniquely colored cursor for each user. Metamouse does this as well. These traditional stacks fail with applications that require exactly one user to operate correctly. To remedy this failure, we map all of the user cursors down to just one new cursor. We call this extra cursor the *metacursor*.

All input is passed to the legacy application through the metacursor. Each action on a user's own cursor translates to an action on the metacursor. There are two primary actions users make with their cursors, movement and clicking. We have mapped each of these to an action on the metacursor that is intuitive and encourages collaboration among users of the system.

Movement

We have to provide an intuitive scheme for mapping many cursors down to one metacursor. To do this mapping, we place the metacursor at the average of the user cursor locations. This means that with any action a user takes, the metacursor moves a small amount in the same direction. This is fundamentally the same technique used in previous works [1,5].

This system encourages cooperation between users. If the users want the metacursor on a particular part of the application, they work together to place all of the user cursors in that area.

Clicks

We detect when the users agree on the location of the metacursor and only allow clicks in that case. We define a threshold τ that determines whether the cursors are in agreement. If the distance between all pairs of mice is less than τ, we assume the users have agreed on the correct location, and allow their clicks to be accepted by the application. If they are not, the mouse clicks are disregarded.

Lastly, we need to give a visual indication to the user when they enter or exit these states. When the distances between pairs of mice are less than τ, the metacursor turns green, indicating that they are allowed to proceed. This is shown in Figure 1. When the user cursor distances are greater than τ, the metacursor is red, indicating to the users that the clicks will be dropped. This is demonstrated in Figure 2.

This design encourages the collaboration property. No actions can be taken in the application unless all users agree, as evidenced by all cursors converging. This eliminates the previously mentioned “fastest-finger-first" race-clicking, random click, and repetitive click modes that are detrimental to learning.

Evaluation

To evaluate this system, we performed a preliminary proof-of-concept study in Bangalore, India. The goal of the study is to observe in-depth how the students performed and interacted using the metamouse technology. The study was conducted by one researcher and one translator. Twenty fifth-grade children participated, seven boys and thirteen girls. We gathered field notes as well as video recordings of the children using the software.

The applications used by the students were a set of games produced by the Azim Premji Foundation's [3] educational initiatives. The games use simple point-and-click interfaces. The students were split into five groups of four, and each student was given a mouse. The researchers first explained and demonstrated the use of a mouse to the participants. In the first two groups, a teacher was present to give help on the questions. For the remaining, no teacher was present.

Collaboration

There were a number of interesting observations of the collaboration encouraged by the metamouse. First, the users interacted heavily with each other to reach consensus and make progress.

In the first group, a student with a firmer grasp of the material and correct answer directed other users to that answer. In the second group, a student actively helped, putting her hand over the hand of the student who was lagging to assist them in using the mouse as well as directing them to the correct answer. In groups three and five, the users all discussed the question among themselves, deciding as a group on the correct answer before all moving to the appropriate location. In group four, one child led, telling the others the answer, and then directing them to the correct answer. At times, the other children disagreed, leading to a discussion about the question before movement.

In all groups, the users worked together to solve the problem. This was either done with a leader dictating the answer to the other users, or by discussion among the users. The groups frequently alternated between these two schemes as the users agreed and disagreed with the leader. The users would also argue about who was not pulling their weight, putting the mouse on the wrong answer, and committing other actions that hindered progress.

Usability

All of the students were quickly able to make use of the technology. Even some nuanced aspects of metamouse were recognized: one student in group four remarked, “Why are you clicking? The cursors aren't all there yet!”

Some parts of the interface were not immediately understood. At times, although they have color differences, users confused the metacursor with their own cursors. Also, the users occasionally confused the coloration with application control. The green mouse was assumed to indicate the answer was correct, rather than that clicks were allowed.

The users were inexperienced with computers, which limited their initial progress. They quickly adjusted and were able to use the software. Their ability to use the technology efficiently also increased with time, ending with them much more proficient than when they began.

Future Work

Other SDG stacks multiplex not only the mice, but also keyboards and other peripherals [13]. An intuitive mapping from many to one for each of these will be necessary as well. For example, multiplexing the keyboards for legacy educational software may be as simple as requiring all keyboards to type the same key for input, or voting for the correct keystroke.

We need to develop a window manager that can switch among these schemes on a per-application basis. This has been done for just two schemes [4], but we hope to expand it to reach all legacy applications. This would allow the window manager to multiplex among the schemes and maximize the utility of computing infrastructure in the developing world.

We focused the first iteration of this work on the most common input interface for educational software, point-and click. There is a huge amount future work revolving around developing schemes for other input types. Some examples of these schemes are drag-and-drop, sliders, and moving targets. Each of the interfaces will require their own unique intuitive schemes. For instance, drag-and-drop may be supported by assigning one user to be the “dragger” or disallowing the dropping until the cursors agree on a location. Moving targets may use a modified point-and-click interface expanding the clicking range when the mice move together. This is a wide area for research, as there are many input schemes, and an appropriate mapping is not always obvious.

Even in our limited point-and-click scheme, other techniques may apply. We hope our system will allow for work in this area, as we expect that the correct answer may depend heavily on local culture, with no universal correct answer.

Lastly, we need to conduct a systematic study of the value of this technology. This will involve creating a list of popular educational software in the developing world. We then must prove that these applications with metamouse schemes provide more educational value than both naïve SDG techniques and traditional single-mouse solutions.

Discussion and Conclusion

In traditional educational software, learning among students is hampered if they do not have an input. We hoped to design a system that would mitigate this by allowing multiple users an intuitive interface to legacy software. This system would also need to encourage and create collaboration among the users. We believe we have accomplished each of these goals.

We found that each student participated, with the students actively working together to solve the problems. The students would assist others who were having difficulty in the interest of making progress themselves. We found no examples of undesirable mouse use modes, such as race-clicking. This behavior is exactly the result we had hoped for with this technology.

Our usability findings were encouraging, though not all positive. In general, the users found the interface intuitive and were able to make progress quickly despite their lack of technological familiarity. There are interface tweaks we need to make. The easiest is changing the cursor of the metamouse to be distinct from the user cursors. This has already been implemented, but was not completed before the end of the study. We believe that confusing the metamouse actions with application actions may be unavoidable, though we will explore possible ways to mitigate it.

We were able to use generic educational software successfully with multiple mice. The difficulties experienced with this are small and easy to solve. Finding educational software that included only point-and-click inputs was not difficult, as this is the dominant interface idiom for this software. We hope to continue our work and develop intuitive, collaboration-enforcing schemes for other interface patterns.

References

[1] L. J. Bricker, M. J. Baker, E. Fujioka, and S. L. Tanimoto. Colt: A system for developing software that supports synchronous collaborative activities. In EdMedia 99, pages 587–592, 1999.

[2] A. Druin, G. Revelle, B. B. Bederson, J. P. Hourcade, A. Farber, J. Lee, and D. Campbell. A collaborative digital library for children: a descriptive study of children's collaborative behaviors and dialogue. Technical report, Department of Computer Science, University of Aarhus Jordan B, Henderson A, 2003.

[3] A. P. Foundation. http://www.azimpremjifoundation.org.

[4] P. Hutterer and B. H. Thomas. Groupware support in the windowing system. In AUIC '07: Proceedings of the eight Australasian conference on User interface, pages 39–46, Darlinghurst, Australia, Australia, 2007.

[5] N. Osawa. Aggregate pointers to support large group collaboration using telepointers. In CHI '06: CHI '06 extended abstracts, ACM.

[6] O. Otto, A. Moed, J. Pal, M. Kam, U. Pawar, and K. Toyama. Reducing dominance behavior in multiple-mouse learning activities.

[7] J. Pal, U. S. Pawar, E. A. Brewer, and K. Toyama. The case for multi-user design for computer aided learning in developing regions. In WWW '06: pages 781–789, ACM.

[8] U. S. Pawar, J. Pal, R. Gupta, and K. Toyama. Multiple mice for retention tasks in disadvantaged schools. In CHI '07: ACM.

[9] U. S. Pawar, J. Pal, and K. Toyama. Multiple mice for computers in education in developing countries. In IEEE/ACM ICTD, 2006.

[10] S. D. Scott, R. L. M, K. M. Inkpen, and E. Lab. Understanding children's interactions in synchronous shared environments. In Proceedings of CSCL 2002.

[11] G. B. Shoemaker and K. M. Inkpen. Middesktop: An application framework for single display groupware investigations. Technical report, School of Computing Science, Simon Fraser University, Report No. TR 20001-01, April 2001.

[12] J. Stewart, B. B. Bederson, and A. Druin. Single display groupware: a model for co-present collaboration. In CHI '99: pages 286–293, ACM.

[13] E. Tse and S. Greenberg. Rapidly prototyping single display groupware through the sdgtoolkit. In Proceedings of the Fifth Australasian User Interface Conference, volume 28, 2004.

[14] M. Westergaard. Supporting multiple pointing devices in Microsoft Windows. Microsoft Summer Workshop for Faculty and PhDs, 2009