From Pilot to Practice: Creating Multiple-Input Multimedia Content for Real-World Deployment

Joyojeet Pal[1], Udai Pawar[2], Apurva Joshi[3], Mohit Jain[3], Sai Gopal Thota[3] ,Sai Teja P[3], Sukumar Anikar[4]

[1] TIER Research Group
University of California at Berkeley,
634 Soda Hall, Berkeley CA, USA 94720
joyojeet@berkeley.edu

[3]DA-IICT,
Gandhinagar, Gujarat-382007, India
{ apurva_joshi , miohit_jain ,thota_gopal, sai_teja}@daiict.ac.in

[2]Microsoft Research India Scientia,
196/36 2nd Main Sadashivanagar,
Bangalore, India 560080
Udai.Pawar@microsoft.com

[4]AzimPremji Foundation
#134,Doddakannelli,Next to Wipro Corporate Office,Sarjapur road,Bangalore-560035,India
anikar@azimpremjifoundation.org

ABSTRACT
In this paper we take further the experimental work on the use of multiple-input devices for developing regions and describe the process involved in creating a ready-to-deploy multimedia CD for English, as a second language, in vernacular-language-medium Indian schools. We briefly explore three areas here – first, we discuss the choice of learning English as a second language for our test application, and the pedagogical process used in designing the multimedia content. Second, we describe the various interaction designs for multiple-input modalities that we have employed, and discuss the motivations behind each, as well as the outcomes in preliminary trials. Finally, we lay out the practical challenges in both design and deployment of a real-world implementation of such a system.

Author Keywords
User Interfaces, Education, Developing Regions, Computer Aided Learning, Multiple Mice.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Following a renewed interest in interface design for low-income populations, the idea of using multiple-input to enable more equitable technology use in shared-use scenarios has gained currency among researchers in educational technology [1, 3, 11]. Experimental work so far has shown major gains in engagement with educational content [1, 15, 17], and gains in basic learning for children in cases where each child is assigned his or her own mouse when sharing a computer [3]. This calls for an examination of prevalent real-world multimedia educational content to understand how multiple-input technologies can be incorporated into use in existing classrooms. However, there are serious challenges in deploying experimental "Multimouse" [1,3] material to real-world scenarios. The incremental hardware cost of multiple mice is minimal, but software challenges are significant. Despite the almost universal sharing of screens by multiple children in low-income areas, most prevalent learning multimedia is designed for single users. So the shift to multiple input scenarios necessitates significant changes in the interaction designs and models. These changes involve enabling existing content to be usable for multiple input scenarios, as well as iterating designs and interactions for content specifically tailored for multiple users. Here, we describe some early results, and more importantly, the steps involved in creating deployable content for multiple-input scenarios

RELATED WORK
In addition to well-received scholarly works on design research for consumer-level services such as financial transactions [10] and engineering aspects of technology in such scenarios [12], a number of experimental deployments with a strong design focus have attempted to introduce technology-aided communication and operations to bottleneck scenarios. These include CAM-based mobile data capture for rural coffee cooperatives [6], text-free User Interface applications for illiterate and semi-literate users [5], learning English as a second language using mobile-phones [4] and in relation, a lot of work in the space of vernacular language development for developing regions.

Among such areas of interest, there has been design innovation for enabling children to better utilize computers in shared-use scenarios, of particular relevance in the resource-constrained developing world [1,2,3,11]. In this paper, we follow the thread of work on the use of multiple mice on a single computer in education settings, which has its roots in earlier work in Single Display Groupware [7,15,16,17,18], looking at programs that enable co-present users to collaborate via a shared computer with a single shared display with the simultaneous use of multiple input devices.

KEY CONTRIBUTIONS

In the past decade, there has been credible research focusing on the 'case for' technology research with the specific needs of the developing world [4, 13], and on the importance of innovative shared computing [14] for the developing world. These works have made clear that technologies initially developed with first world conditions in mind often do not adapt well to low-income situations, and often a lot of work is needed to bridge the gap between showing a prototype in a first-world lab, to actually deploying it in a developing country. The key contribution of this current paper work is to work on this logical next step of 'real-world deployment'. Here we have taken an idea within the ICTD field which had demonstrated benefits in experimental scenarios, and redesigned real-world content to examine practical applicability of such systems.

In conceptualizing our design decisions, we kept in mind earlier research [2] that has shown that teacher and resource shortages in developing countries create 'babysitting' type scenarios in computer classes where access to human guidance is minimal or absent. Consequently, we have paid much attention to artificial intelligence factors in the design of children's interaction in creating several design options for real-world deployment. One such contribution here is our use of turn-taking as a machine-induced interaction assuming the absence of human intervention in our designs.

DESIGN APPROACH

There are three broad approaches for designing interfaces relevant to multi-mouse scenarios. **First**, simply "enabling" existing multimedia, designed for use by a single user, to be used by multiple users – without any changes to the content. A **second** approach is "redesigning" the interactive parts of the content without changes to the narrative flow or pedagogical structure. The **third** approach is to design multimedia content assuming a multi-user scenario from *first principles* (from 'scratch'). In this paper, we describe experiences with the first two cases, which in turn we believe help make a case for redesign from 'scratch'.

Enabling Existing Material for Multimouse

In our studies of educational content, we found that sizable fraction of software applications for children are graphics-intensive and designed using Macromedia Flash. Most such content also follows a typical narrative-interactive loop pattern – with some narrative content being shown to the user child, followed by a series of multiple-choice questions based on the narrative content. Such content usually has hyperlinks, and animations activated by clicks.

At the simplest level, "enabling" such material for multiple inputs would mean allowing each child to have a mouse and creating a "first click prevails" scenario for all screens. Thus the typical interaction mode is like a 'racing' scenario [3] – whichever child clicks first triggers the specific action on the screen that leads to the next step in the software, and so on. To examine ways of achieving this, we explored toolkits that have been created by researchers [17] in the past which enable multiple mice. We found that a major constraint in each of these was their being tied to particular platforms. Specifically, none of these worked with Flash.

We selected the MultiPoint SDK by Microsoft [9], which though based in .NET, could initiate Flash applications from within .NET without requiring any changes to the Flash runtime. Thus, it was technically possible to enable multiple mice on Flash-based multimedia content on a PC with .NET installed. Using the MultiPoint SDK in C#, we developed a tool which can host Flash content and enables multiple mice to interact with Flash. We created two C# applications, which run simultaneously, one hosts the Flash content and the other captures the multiple mouse clicks and informs the first application. The tool can support any number of mice, adding flexibility. This tool suffers from certain glitches – specifically some overhead in some settings, but it sets the stage for refined future iterations.

Using this tool and the MultiPoint APIs, the application instantly recognizes how many mice are plugged into a PC, and assigns a cursor to each, following which any of the students can cause the next action to take place on the screen by being the first to click. So even by using *existing* interactive modules, such as multiple choice questions, this creates a competitive environment (such as the MM-R "MultiMouse-Racing" discussed in [3]) which assigns a response to the first student to click.

Implementing MM-R using the tool requires no change to be made to the existing content, designed for single user scenarios. Past work has shown that this simple change can significantly increase children's engagement with such content, though for higher order learning outcomes, more attention needs to be paid to other factors such as collaboration. This leads us to the second design approach.

Redesigning Existing Multimedia

The bulk of our work was in "re-designing" existing educational multimedia content with the aim of minimizing changes to the pedagogical design of the content. This is a logical next step to simply "enabling" multiple mice, but it still allows one to re-use existing content. Such a re-design is not an easy case scenario since most content was originally designed with a single user in mind. More importantly the full breadth of features and interactions that a multiple-input system would offer might not be exploitable. We reviewed various interaction types typical to children's multimedia, and redesigned existing 'English as Second Language' (ESL) material using these.

The existing content CD used, known as 'Friendly Animals' was an ESL CD being used in government primary schools for instructing seventh graders. Software mainly consisted of questions based on narrative content with three types of interactive modules - standard Multiple

Choice Questions (MCQ), Fill in the Blanks (FITB) with a blank to be filled from a list of clickable choices, and Ordering Questions in which the user has to re-arrange a jumbled set of words or phrases into the correct order, by clicking and moving the words or phrases, with the cursor.

The tool we developed was able to capture multi mice clicks, distinguish clicks by different mice, and inform Flash accordingly. For communication between Flash and C#, we used *fscommand,* a built-in function in Flash, and *setVariable,* a member function of the activeX wrapper class in C#. C# application also notified the Flash runtime about the number of mice connected to the computer.

The content was redesigned with no content additions or changes to narrative flow, to maintain a minimal interference with the curricular material, as well as to minimize the 'time-to-deployment'.

The interactive modules were redesigned building upon earlier research on what worked well [3, 7, 15–18] to attract and engage students such as color differentiations, animated cursors and personalized scoring to reinforce on-screen identity. We designed six types of multiple-input interactive modules, each was used following one of the narrative segments, by re-designing and replacing the existing interactive single-player modules. These were as follows.

Racing Model

As the name suggests, it's the 'fastest-finger-first' model. Any student can answer the questions, and the child who clicks the correct answer first, gets rewarded, in the form of stars (colored the same as their cursors) (Figure 1). Any content that tests basic concepts can use the racing model, as the questions can be answered quickly. This model is competitive in nature, and from [3] we expect that it will be engaging, however pedagogical efficacy is not guaranteed, and competition might not be the best way to go forward.

Figure 1. A multiple choice question with the Racing Model

Turn-Taking Model

Turn-Taking model has originated from the idea of 'one player at a time', as in traditional games like carrom, ludo, etc. In this model each student gets a chance, one after another. A question is targeted to a randomly selected student and only he/she is allowed to answer it. The 'instantaneous' interaction modality is similar in one sense to a one-pc-one-user scenario – i.e. at any given point there is only one active child, yet due to the pressure of 'getting on' with the game, it is not exactly a one-pc-one-user scenario. Also, past experience with multiple mice leads us to believe that even though a mouse might not be currently 'active', the fact that a student simply has a mouse in hand and a cursor onscreen makes him/her more involved.

Ideally, any content can be implemented using turn-taking model where questions are targeted to all the students sharing a single pc, in a round-robin fashion. This model can 'mechanically' enforce a condition that all students get equal opportunity to learn.

To implement the turn-taking modality in our test application, we need to emphasize which specific user's turn it is to answer. This is done by changing the font color of the question and answer text on-screen, according to the color of the active cursor. The active cursor is the one belonging to the student whose turn it is to answer. To prevent the other students from answering, their mice cursors are represented as a cross and disabled (though they are visible and can move, but cannot click on anything, Figure 2). Content developed using this model gives all the students a fair chance to participate as all of them get equal number of questions, which also makes this different from Inkpen's work [15] which also experimented with turn taking of sorts, but directed by the users themselves toggling a single on-screen cursor between two mice.

Figure 2. The game implemented using Turn-Taking Model

Directed MCQ Model

In the Directed MCQ model for the multi-mouse scenario, a question is followed with few option choices. Beside each option, there are colored boxes corresponding to each cursor (Figure 3). Each student has to choose one of the options by clicking over the colored box having the same color as his/her cursor. When a student clicks on an option, the option gets checked. When all of them have chosen their option, the correct answer is revealed, showing who all answered correctly and the game advances. As all the students participate at the same time, they remain engaged throughout. This model allows each student to exercise his/her choice independently.

Figure 3. The game implemented using Directed MCQ Model

Voting Model

The Voting model has originated from the concept of 'voting', where opinions of each individual are considered, and the final decision is based on the majority, or unanimity [3]. In this model, every student exercises his/her choice for answering a question and a suitable action takes place depending on the option selected by the majority. It is useful in situations where varied views are possible and decision is to be taken collectively, taking each students opinion into account.

In fact this extends to interaction settings beyond game-play. For instance, decisions affecting the global application operation like moving to the next game, playing the same game again, exiting the game, etc. can be implemented using voting model. In these cases, the game pauses and the voting screen appears (Figure 4).

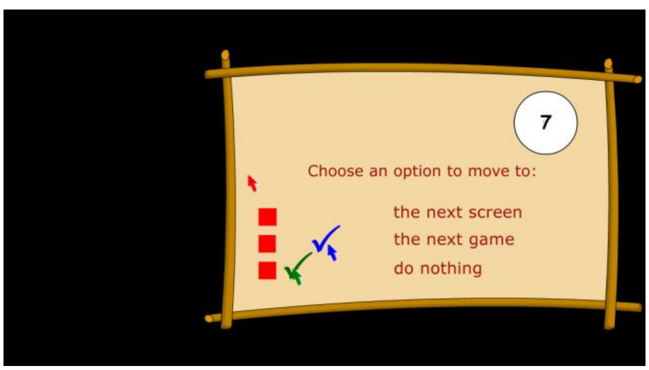

Figure 4.Implementation of the Voting Model for decisions

The game moves forward only when each student has chosen his/her option. If a student doesn't select any option the game doesn't proceed. To overcome this problem, a timer was introduced (Figure 4). The timer is set to a specified limit and it starts as soon as the voting starts. The users are allowed to vote within that time span. If any of the students do not vote within the given time, the game resumes from the same point.

Unity Model

As the name suggests, in this model all the students need to collaborate among themselves to answer the question – collaboration is enforced. This model encourages sharing of knowledge among the students and also develops the spirit of team work. The content where the answer requires proper selection and arrangement of options can be comfortably implemented using this model.

Figure 5. Showing the game implemented using Unity Model

This model was implemented for a game which involves rearranging jumbled words to make meaningful sentences (Figure 5). The game was redesigned for the multiple-mice scenario such that each of the five jumbled words is randomly colored to one of the five cursor colors. A student can only click on the word corresponding to his/her cursor color. This brings that word to a sentence queue.

To reach the next level all the students need to collaborate amongst themselves and click on the words in a particular sequence to place these words in a sentence queue, so as to form a meaningful sentence.

Split-Screen Model

The split-screen model has also been trialed in other research and found to be highly effective in increasing collaboration without losing engagement and competitiveness within a group [8]. In our implementation, the game screen was split into two halves so that two students (or two groups) can play simultaneously in their respective halves. In the game developed using this model, two random teams were formed by dividing the students (Figure 6). Each team was allotted one part of the screen, and teams can only answer the questions appearing in their part. The team which answers first gets rewarded. The game moves forward only after questions in both the halves were answered correctly.

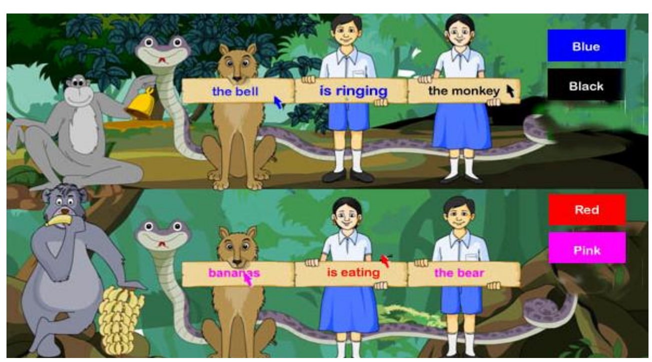

Figure 6. The game implemented using the Split-Screen Model

This encourages team effort amongst the students, leading to higher level of interaction, along with incorporating competitive incentives. Hence, it satisfies the three-fold objective of engagement, collaboration and learning.

OBSERVATIONS

Our usage examination here is a preliminary qualitative participant observation which helps the iterative design process and gives us insight into the right questions to ask. These observations set the stage for controlled experimentation.

At the time of publication, we have observed a total of 30 children using the material over 6 groups. Each of the tests took place in a real-world setting, at a low-income corporation school which was participating in the state computer-aided-learning program. The observations were all done during school hours inside the computer lab with 5 children per computer at the time of the test. No specific instructions related to multi-mouse were given to the students, to observe how quickly they get acquainted with the new scenario.

Figure 7. Preliminary field tests with school-children

Unchanged Existing Content Enabled with Multiple Mice

We tested first, the basic implementation of simply adding multiple mice capability to existing single user CDs. This was a particularly important test because in terms of a real-world usage scenario, this offers the cheapest and most immediately deployable option. Trials showed that students were more engaged due to the competitive aspect, but this in turn made the control of the interaction somewhat ungainly since any child could move on to a 'next' screen. An encouraging result is that a share of the clicking was distributed among all the users, which indicated a fairly wide involvement. However, during the MCQ sessions, the clicking itself was based on a speed-based competitive strategy rather than one oriented to thinking through the options. In this strategy, children hoped to score in the game through lucky clicks, validating [3], so there was no real need for actually building content knowledge. On the whole, our observations suggest that while there is encouraging increase in engagement, it may take some hand-holding and getting used to individual mice for children to start effectively using them.

Racing Model in Multimouse Enabled Content

The students played the game with enthusiasm as they get rewarded for answering correctly. The games implemented using racing model proceeded quite fast, as students rushed to answer the questions, without discussing among themselves. A non-collaborative environment developed in such situations. Moreover in a particular case, a lagging child eventually lost interest in the game and sat idle.

Turn-Taking Model in Multimouse Enabled Content

Children easily understood this model as it was similar to the one-pc-one-child scenario which they were already aware of. Since, each question was targeted to a randomly-selected student, all the students remained attentive waiting for their turn. In cases where a student got stuck with a question, others helped him/her answer the question, encouraging discussion among students. Children remained idle after their turns.

Directed MCQ Model in Multimouse Enabled Content

As all the students have to choose an option for each of the questions, everyone remained involved throughout the game. In few cases, the lagging child was observed following the leading child.

Voting Model in Multimouse Enabled Content

As each individual's decision is taken into consideration, students felt responsible and participated actively. It was seen that the leading student was compelling others to choose an option of his/her choice. The lagging child was forced by others to be quick.

Unity Model in Multimouse Enabled Content

In this model, maximum amount of discussion was observed. It was observed in a case that the leading girl not only formed the complete sentence, but even recited it, so that the answer was known to all. Albeit the sample size is low, girls seemed to be more cooperative as compared to boys. Since the students have to discuss in order to form the correct answer, this model consumed a lot of time, but we postulate that the discussion would lead to better learning.

Split Screen Model in Multimouse Enabled Content

Children learned to play in a team and co-operated with their teammates to win the game, as the answers could be framed only after discussion within the team. Since the game doesn't move forward until the questions on both halves of the screen are answered correctly, the first team to finish remained idle till the second team completed.

General Observations for Multimouse Enabled Content

Apart from these specific observations, in general when children were forced to wait due to the slow pace of others, they got frustrated. This was more common in boys as compared to girls. In few cases, the leading student forcefully tried to answer for others by taking over their mice. Moreover, few students indulged in random clicking.

These observations open up possible avenues for future developments (Table 1) for multiple-mice interactions.

TABLE 1. Interaction models and related characteristics

Model	Application	Risks	Future Work
Racing Model	To increase engagement, competition	Gaming system through rapid clicks	Negative marking to discourage random clicking
Turn-Taking Model	Creating equitable access	Decreased engagement of non-active children, resentment towards slow movers	Artificial intelligence: push questions for children performing lesser than others
Directed MCQ Model	Simultaneous and individual participation	Disinterest among faster-finishing children	Rewards based on the timing of clicking
Voting Model	Encouraging consensus	Free riding, contrived agreement, no decision-making even if one doesn't answer	Decision-making based on received votes
Split Screen Model	Balancing collaboration with engagement	Complicated interface	Using scores from previous rounds to form a balanced team
Unity Model	Fostering group responsibility	Slower interaction	Setting time limits for decision-making

CONCLUSION

We started this work under the assumption that shared computing is a likely direction for the future given the cost of technology in the developing world. This work is meant to serve as a reference for researchers looking at simultaneous shared computer use by offering ways in which such interactions can be designed as well as discussing the pros and cons of these designs.

We find in our tests that children easily adapt from one type of interaction model to another with limited or no explanation. For multiple mice to be used effectively in currently prevalent learning scenarios, no single design, rather a combination, is likely to be used. One finding consistent through many of the trials was expanding the scope of artificial intelligence in realizing the true benefits of multi-mouse.

With the increase of interest in multiple mice both within the industry and in policy circles, it is possible that in the near future, real world deployments of such technology are highly likely. Interaction designers are likely to play a critical role in the development of shared screen technology going forward. This study and others like it highlight some of the key issues for researchers to iteratively discuss.

REFERENCES

1. Pawar, U.S., Pal, J. and Toyama, K. Multiple Mice for Computers in Education in Developing Countries. In *Proc. of IEEE/ACM ICTD'06,* 2006, pp. 64-71.
2. Pal, J., Pawar, U.S., Brewer, E.A. and Toyama, K. The Case for Multi-User Design for Computer Aided Learning in Developing Regions. In *Proc. of WWW 2006*, pp.781-789.
3. Pawar, U.S., Pal, J., Gupta, R. and Toyama, K. Multiple Mice for Retention Tasks in Disadvantaged Schools. In *Proc. CHI 2007*, ACM Press (2007), pp. 1581-1590.
4. Kam, M., Ramachandran, D. and Canny, J. MILLEE: Mobile and Immersive Learning for Literacy in Emerging Economies. *CHI 2007 Workshop*, 2007.
5. Medhi, I. and Toyama, K. User-Centered Design and International Development. *CHI 2007 Workshop,* 2007.
6. Schwartzman, Y. and Parikh, T.S. Establishing Relationships for Designing Rural Information Systems. *CHI 2007 Workshop*, 2007.
7. Stewart, J., Bederson, B.B. and Druin, A. Single Display Groupware: A Model for Co-present Collaboration. *Proc.' CHI 99*, ACM Press (1999), 286-293.
8. Field/mice project. http://groups.ischool.berkeley.edu/fieldmice/
9. The Microsoft Windows MultiPoint Software Development Kit (SDK). http://www.microsoft.com/downloads/details.aspx?FamilyID=A137998B-E8D6-4FFF-B805-2798D2C6E41D&displaylang=en
10. Parikh, T., Ghosh, K. and Chavan, A. Design studies for a financial management system for micro-credit groups in rural India. In *Proc. of the Conference on Universal usability*, 2003.

11. Patra, R. et al. Usage Models of Classroom Computing in Developing Regions. *Proc of IEEE/ACM ICTD2007*, December 2007.
12. Guo, S., Falaki, M.H., Oliver, E.A., Ur Rahman, S., Seth, A., Zaharia, M.A. and Keshav, S. Very Low-Cost Internet Access Using KioskNet. *ACM SIGCOMM Computer Communication Review*, October 2007, 95-100.
13. Jhunjhunwala, A., Ramamurthi, B. and Gonsalves,T.A. The Role of Technology in Telecom Expansion in India. *Communications Magazine, IEEE*, 36 (11), pp. 88-94, 1998.
14. Dillenbourg , P. and Traum, D. Does a shared screen make a shared solution?. In *Proc. of the 1999 conference on Computer support for collaborative learning*, p.14-es 1999.
15. Inkpen K., Booth K.S., Gribble S.D., and Klawe M. Give and Take: Children Collaborating on One Computer. Short papers in CHI' 95, ACM Press (1995), 258-259.
16. Bricker, L.J., Tanimoto, S.L., Rothenberg, A.I., Hutama, D.C., and Wong, T.H. Multiplayer Activities that Develop Mathematical Coordination. Proc. CSCL (1995), 32-39.
17. Shoemaker G.B.D. Single Display Groupware research in the year 2000. TR2001-1 (2001), Simon Fraser University.
18. Benford, S. et al. Designing storytelling technologies to encouraging collaboration between young children. Proc. CHI '00, ACM Press (2000), 556-563.