

Design Guidelines for Learner-Centered Handheld Tools

Kathleen Luchini, Chris Quintana, Elliot Soloway

University of Michigan

1101 Beal Avenue, Ann Arbor, MI 48109 USA

kluchini@umich.edu, quintana@umich.edu, soloway@umich.edu

Abstract

Handheld computers are mobile, flexible devices that can provide real-time, one-to-one support for students from within the context of their learning activities. This paper describes the design of three learner-centered handheld tools used as part of a nine-month classroom study involving thirty-three eighth grade students. A review of related work identifies some of the challenges of building educational software within the constraints of handheld screens, and two broad design guidelines are synthesized to help address these challenges. The first design guideline focuses on decomposing the learning activity to identify salient tasks and the type of supports (or scaffolds) students need to engage in these tasks, then building separate handheld workspaces to support each task. The second guideline focuses on methods for implementing scaffolds within these task-based workspaces while preserving the usability of the overall handheld software.

Categories & Subject Descriptors: H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. K.3.1. Computers and Education: Computer Uses in Education.

General Terms: Design.

Keywords: Handhelds, Learner-Centered Design

INTRODUCTION

Portable devices such as Palms, Pocket PCs, graphing calculators and scientific probeware have proven capable of supporting K-12 students during learning activities such as collaborative math projects [6], story writing [1], studies of population genetics [4] and hands-on science experiments [14]. These research efforts demonstrate that portable technologies can serve as powerful personal learning devices in K-12 classrooms, providing access to tools and information from within the context of learning activities.

While handheld tools have proven useful supports for students of all ages engaging in various learning activities, most of the existing research in this field focuses on the design of individual handheld tools that support specific tasks. This paper describes the **Mobile Learning Tools for Science (MaLTS)** project, which is an effort to understand and address the broader design challenges of building

handheld software to support learners engaged in a variety of science inquiry activities.

The MaLTS project includes three broad phases of work. First, we analyze related research in the design of desktop and handheld software to identify strategies for supporting learners during science inquiry activities and to develop guidelines for implementing these supports in software. Second, we develop three handheld learning tools to support the science inquiry activities of concept mapping, online research and scientific modeling. Finally, we study the use of these tools during a nine-month classroom trial and, based on the evaluation of this data, we offer additional strategies for understanding and addressing the challenges of building handheld tools for learners.

PHASE 1: ANALYSIS OF RELATED RESEARCH

The MaLTS project begins with a review of research from three areas: (1) methods for supporting students during science inquiry activities, (2) Learner-Centered guidelines for designing educational tools for desktop computers, and (3) User-Centered guidelines for designing usable interfaces for mobile devices. Synthesizing work from these three areas allows us to identify some of the challenges of building supportive educational software within the constraints of small handheld screens.

Scaffolding Strategies for Supporting Science Inquiry

Scaffolding is a term drawn from educational psychology to describe temporary supports that help students mindfully engage in unfamiliar new work [3]. Mindful engagement, in this context, means actively participating in and learning about the underlying concepts and processes involved with an activity, rather than simply completing a task by rote. Scaffolds can encourage mindful engagement by helping students identify important concepts and participate in the work processes of an activity that is initially unfamiliar. For instance, a math teacher might scaffold students during an algebra assignment by first modeling the process of how to solve a problem and then coaching students as they try to solve similar problems on their own. Scaffolds may also be integrated into educational software; for example, the software might prompt students to reflect on their work [5] or might make the steps of an unfamiliar research process visible to help students understand the activity [7].

Recent research [18] has analyzed existing desktop tools to identify what types of scaffolds students need in order to mindfully engage in science inquiry. This research identifies three categories of scaffolds that can assist students during science inquiry activities:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2004, April 24–29, 2004, Vienna, Austria.

Copyright 2004 ACM 1-58113-702-8/04/0004...\$5.00.

- **Process management** scaffolds help students plan and monitor their work and help them understand the rationale for engaging in the learning task.
- **Sense-Making** scaffolds help students analyze and understand the products of their inquiry.
- **Articulation** scaffolds help students review, reflect on and synthesize their work.

This research also offers specific guidelines for providing these supports, such as making concepts and terminology explicit in order to help students make sense of an activity.

Design Guidelines for Learner-Centered Desktop Tools

Learner-Centered Design (LCD) is the second area of related research analyzed as part of the MaLTS project. LCD is an area of human-computer interaction research that focuses on developing tools that help people learn. While all computer users need to learn how to use specific software to accomplish a task, LCD recognizes that some computer users – the subset of “learners” – also need support for understanding the task itself [17]. Thus, the group of learners addressed by LCD are distinct from the audience of experts addressed by traditional approaches to user-centered design, which often focus on making software interfaces highly usable to speed completion of a task [15]. Instead, LCD defines learners as individuals who need additional support in order to mindfully engage in and learn about the content and work practices of an unfamiliar domain. While learner-centered software needs to be usable, it should not be *so* usable that the work is completed automatically, without allowing learners to engage in the underlying activity.

LCD-inspired software often incorporates scaffolds to assist learners in working mindfully within an unfamiliar domain. Several guidelines [17] have been developed for implementing scaffolds in desktop LCD tools, including:

- **Scaffold Visibility:** scaffolds should be highly visible within the interface since learners often miss or ignore supports that are not readily apparent
- **Scaffold Essentialness:** scaffolds should be essential for completing the task since learners may miss critical supports if they are mixed with non-essential scaffolds
- **Scaffold Coupling:** related scaffolds should be tightly coupled, meaning that supports that are in proximity in the interface should support similar tasks
- **Scaffold Usability:** scaffolds should be usable enough that learners are not unduly frustrated, but not so easy to use that learners are not mindfully engaged
- **Scaffold Representation:** both graphical and textual implementations can provide successful scaffolds, although graphical scaffolds may be misleading if the graphic is not obviously “clickable”

Design Guidelines for User-Centered Mobile Tools

In addition to exploring related research in scaffolding science inquiry activities and in creating learner-centered desktop tools, the MaLTS project also looks at the challenges of designing usable software interfaces within the

constraints of handheld computer screens. Research in User-Centered Design (UCD) is exploring a number of methods for creating highly usable interfaces for mobile tools like cell phones and handheld computers, including:

- **Streamlining** the interface to make the most essential information and tools highly accessible onscreen. This is often accomplished by reducing the functionality of the tool. For example, interfaces for “information appliances” [2] are highly specialized to support a single task, such as calendar scheduling, rather than providing a complex, multi-purpose tool.
- **Automating** the interface to reduce the need for users to enter text or commands. For instance, handheld web browsers often automatically remove graphics and summarize text to reduce the amount of time users have to spend looking for specific information [8, 9].

By streamlining and automating the interface, these UCD approaches work to create more usable mobile tools that can help experts complete familiar tasks more efficiently.

Challenges of Designing Scaffolded Handheld Software

Synthesizing research from these three areas offers several relevant lessons for the MaLTS project:

- In order to mindfully participate in learning activities, students need support for process management, sense-making and articulation.
- LCD research calls for scaffolds to be highly visible, accessible, tightly coupled, and essential to the task.
- UCD research suggests streamlining and automating mobile interfaces in order to create more usable handheld tools.

These lessons highlight a tension between the need to provide adequate functionality and support for learners and the need to keep mobile interfaces usable. For instance, UCD approaches call for automating functionality to improve the usability of handheld interfaces, yet LCD finds that students are unlikely to learn about work that is automatically completed for them. The remainder of this section focuses on two broad challenges highlighted by this review of related work: how to decompose learning activities to fit within the constraints of handheld screens, and how to incorporate scaffolds into handheld tools so that learners are supported but the interface is still usable.

Using Activity Decomposition to Structure Handheld Tools

One design challenge that must be addressed when building scaffolded handheld learning tools is determining what scaffolds, tools and information to incorporate given the constraints of handheld screens. Desktop LCD tools typically focus on making information and scaffolding highly visible, often using frames or windows to display multiple workspaces onscreen simultaneously. Yet within handheld screens it is difficult or impossible to show multiple workspaces at the same time while maintaining overall usability.

We know from UCD research with mobile tools that simply replicating desktop tools within a handheld screen does not result in usable software. Instead, we must analyze and decompose learning activities to identify the salient tasks and determine which should be included in handheld software. Such activity-based decompositions, which are commonly used in the design of learner-centered desktop software [17], preserve the conceptual integrity of the learning activity rather than breaking it up arbitrarily. This decomposition can serve as a framework for developing handheld software to support the learning activity by developing separate workspaces for each component task. Providing the scaffolds, information and tools required for each task within separate handheld workspace reduces the overall clutter of the interface and makes the individual workspaces easier to use.

Another benefit of decomposing learning activities into their component tasks is that the most salient tasks can be identified and incorporated into the software, while secondary tasks may be automated or eliminated to improve the usability of the handheld tool. Of course, the salience of a particular task depends on the goals of the learning activity and on the nature of the software tool. For instance, a desktop tool might be designed to support many components of a scientific research process, whereas a handheld tool might only be designed to support one task, such as gathering data on-site from field experiments.

One drawback of decomposing activities into multiple sub-tasks supported in separate workspaces is that the overall learning activity is less visible. Since learners focus on only one task at a time while using the separate handheld workspaces, they may have difficulty recognizing the overarching work process and how the component tasks contribute to a larger goal. Thus, the activity decomposition must not only identify salient tasks but must also identify the links between those tasks. These connections should then be reflected in the design of the handheld tool.

These strategies are summarized by our first design guideline for building scaffolded handheld tools:

- **Design Guideline: Task-Based Workspaces.** When developing educational software for handheld computers with small screens, support different component tasks of the learning activity within individual workspaces that include the necessary scaffolds, information and tools. This task-based decomposition preserves the conceptual integrity of the learning activity while allowing the salient component tasks to be useably supported within handheld screens.

Increasing Interface Usability with Dual-Purpose Scaffolds

Another challenge that must be considered when building handheld tools for learners is how to include scaffolds without impairing the usability of the software. In desktop learning tools, scaffolding is typically placed in and around the activity workspace, and the overall work process is often depicted with large graphics or flowcharts that show the different steps in the activity. Screen space is at a premium in

handheld tools, however, and designers must find a balance between including enough scaffolds to support the learning activity while not including so many elements that the interface becomes unusable.

To address this challenge, our second design guideline focuses on creating dual-purpose interface elements that can provide both scaffolding and functionality:

- **Design Guideline: Dual-Purpose Elements.** When developing educational software for handheld computers with small screens, whenever possible design interface elements to serve a dual role by providing both functionality and scaffolding. Using such dual-purpose elements reduces the number of tools and scaffolds that must be included in an individual workspace and increases the usability of the handheld interface.

User-centered approaches to building interfaces for mobile tools often focus on streamlining software by reducing the number of onscreen elements [20]. Thus, employing our second design guideline can result in a more usable interface, where non-functional scaffolding is replaced by dual-purpose elements (e.g., buttons, menus) that provide both functionality and scaffolding. For example, instead of including a flowchart that describes the activity process but does not provide functionality, a navigation menu might be used to provide both the functionality for moving between workspaces and the process scaffolding that makes the overall work process visible. Creating dual-purpose elements reduces interface clutter and increases the amount of workspace available for actually engaging in the learning activity.

PHASE 2: DESIGN OF HANDHELD SOFTWARE

The second phase of the MaLTS project uses the two broad design guidelines synthesized from the previous review of related research to develop three scaffolded software tools. This software was designed for handheld Pocket PC computers equipped with wireless Internet access. The remainder of this section briefly describes each tool and the influence of the design guidelines on its development; more design information can be found in [11, 12].

Pocket PiCoMap

Pocket PiCoMap helps learners create and share concept maps, which are a type of graphical outline. In a concept map, individual ideas are drawn inside separate nodes and the relationships between ideas are shown as labeled arcs between the nodes [16]. Pocket PiCoMap provides separate workspaces for viewing and organizing the graphical map (Figure 1) and for a number of sub-tasks, such as creating and editing relationships (Figure 2).

Pocket PiCoMap also includes a number of dual-purpose elements that provide both functionality and scaffolding. For example, the text box shown in Figure 2 serves a dual role: it provides scaffolding by making the activity option of adding notes about a relationship visible to learners, and also provides the functionality to support this activity.

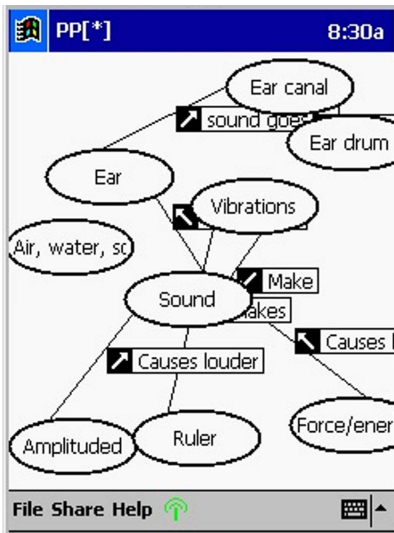


Figure 1: Pocket PiCoMap Map Interface

Figure 2: Pocket PiCoMap Relationship Interface

Figure 3: ArtemisExpress Driving Question Interface

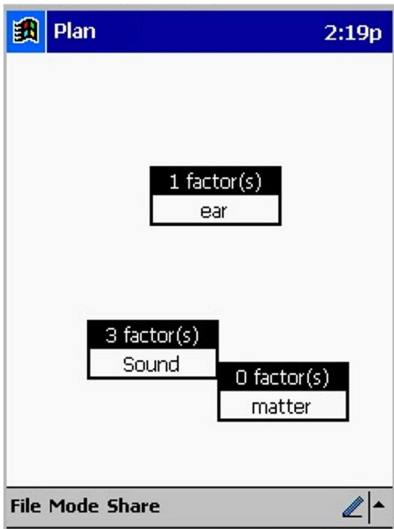


Figure 4: Pocket Model-It Plan Interface

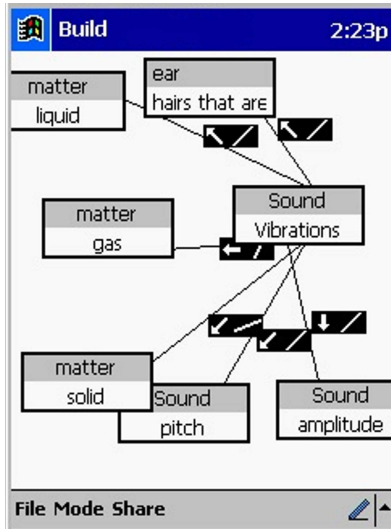


Figure 5: Pocket Model-It Build Interface

Figure 6: Pocket Model-It Test Interface

ArtemisExpress

ArtemisExpress supports research activities using the online Artemis digital library [21], which catalogs websites suitable for students. Using ArtemisExpress, learners can search for relevant websites and create Driving Question folders (Figure 3) to collect their research and add their own notes. ArtemisExpress is divided into four major workspaces: Search, Driving Question, Share and Tools. The navigation bar that appears at the top of each of these workspaces is one example of a dual-purpose element in ArtemisExpress; it provides process scaffolding by making the components of the overall research activity visible and provides functionality for moving between workspaces.

Pocket Model-It

Pocket Model-It helps students build and test models of dynamic systems, which gives them the opportunity to see the impact of changing factors within models [13]. Pocket

Model-It supports using three main workspaces: Plan, Build and Test. In Plan (Figure 4), students add objects to their models; for instance, a model about hearing might include the object “sound”. In Build (Figure 5), students add factors (also called variables) to describe their objects; for example, they might add a “pitch” factor to the “sound” object. Students also create relationships between factors in the Build workspace. Finally, in Test (Figure 6), students simulate the dynamic behavior of their model, changing factor values to observe the consequent impact on the rest of the system. Pocket Model-It includes a number of dual-purpose elements, such as the simulation tools shown at the top of Figure 6. These tools allow learners to control the simulation by adding meters and changing variables, while also providing activity option scaffolding to help learners understand what tasks and activities are available in the current workspace.

PHASE 3: EVALUATION OF CLASSROOM STUDY

The three MaLTS software tools were tested as part of a nine-month classroom study involving thirty-three students in two eighth grade science classes at a private, urban school in Michigan. The students each received handheld Pocket PC computers that could be used in all of their classes, could be taken home, and could access the Internet via the school's wireless 802.11b network.

The primary data analyzed for this study was collected from eight focus students (two girls and two boys from each class) who each used the MaLTS software for eight 30-minute activities. The data consisted of video and audiotape of the handheld computer screen, supplemented by logfiles and the final products of students' activities. The evaluation of this data focused on determining the "effects with" [19] the MaLTS software, which means looking at what students do with these handheld tools and how they use them to engage in the learning activities.

Evaluation Criteria

In order to better understand the impact of small screens on the design and use of the MaLTS tools, two aspects of students' use of this software are examined. First, we evaluate whether the design and implementation of each scaffold results in a useful and usable support. This evaluation is necessary since students cannot benefit from scaffolds if they are unusable within handheld screens. To conduct this evaluation, we adapted the following evaluation criteria from LCD research on evaluating the design of scaffolds in desktop software [17]:

- **Accessibility:** can students find the scaffold within the handheld tool?
- **Use:** do students make use of an accessible scaffold?
- **Efficiency:** how much difficulty do students have using the scaffold?
- **Accuracy:** does the scaffold help students complete the task in an accurate and appropriate manner?
- **Progression:** is this scaffold used differently over time?
- **Reflection:** do students reflect on or review their work when using this scaffold?

In addition to evaluating the success of individual scaffolds, we also looked at the overall use of each MaLTS tool, since even the best scaffolds are not helpful if the software as a whole is overly difficult to use. The following criteria synthesized from LCD and UCD research were used to evaluate each tool:

- **Contextual Awareness:** how do small screens impact students' awareness of the overall context of the learning activity?
- **Artifact Organization:** how do students organize their work within the constraints of handheld screens?
- **Navigation:** how do students navigate within the current workspace and between different tasks and workspaces?

- **Information Transfer:** how do students move information between and within handheld workspaces?

RESULTS OF THE CLASSROOM STUDY

Evaluating the data from the classroom study highlights two key factors that impact students' ability to use the MaLTS tools to mindfully engage in science inquiry activities: (1) the design of the handheld software, and (2) characteristics of the learning activity, such as the complexity of the task or learners' familiarity with it.

We found that creating task-based workspaces generally resulted in usable handheld software. However, segmenting the activity into multiple workspaces did make it harder for some students to understand the overarching work process, particularly for activities that were complex or unfamiliar. We also found that creating dual-purpose interface elements, which provide both functionality and scaffolding, generally resulted in usable handheld tools. However, learners sometimes need more scaffolding than what can be integrated into functional interface elements. For instance, the MaLTS study found that learners often require more process scaffolding than what was provided by the dual-purpose elements, especially when the learning activity was complex or unfamiliar to students. The remainder of this section describes the results of the classroom study in more detail and offers refinements to our design guidelines for building scaffolded handheld software.

Evaluation of "Task-Based Workspaces"

The evaluation of the classroom study data indicates that although some students had difficulty understanding or using individual interface elements (as discussed later in the evaluation of dual-purpose elements), the task-based workspaces were generally very usable. The study does highlight several issues that should be considered when designing task-based workspaces, such as:

- Ensuring that all of the tools and information required for a task are accessible from within the appropriate workspace, even though this may require duplicating the same tool or information in multiple workspaces.
- Providing sufficient navigational support for moving between the workspaces of closely related tasks.
- Supporting the organization and visualization of work, particularly for large or graphical tasks.
- Using caution when selecting which tasks to automate or eliminate to ensure that learners do not develop incorrect mental models of the learning activity.

The remainder of this section describes each of these issues in more detail.

Ensuring Access to Tools and Information

One of the challenges of decomposing learning activities into multiple tasks that are supported within separate handheld workspaces is that individual tools and data may be relevant to more than one task. For example, in order to save a website to the correct Driving Question (DQ) folder in ArtemisExpress, students need to know which DQ is

currently set to “active” status. This information is visible within the DQ workspace but not from the Search workspace, which is where students review and save websites. To address this problem, we tested two versions of ArtemisExpress during the MaLTS classroom study – one version that displays the DQ workspace when students log on, and a second version that starts students at the Search workspace. Interestingly, making the DQ workspace visible initially did not improve students’ ability to recall the currently active DQ folder when saving websites from the Search workspace. Five of the eight focus students in the study saved sites to incorrect DQ folders, making an equal number of errors when using either version of ArtemisExpress. Obviously, making the active DQ folder visible at the outset of the activity is not sufficient, and future versions of this tool will explore methods for making the active DQ visible from within the Search workspace.

Pocket Model-It provides another example of the need to make common functionality and information accessible from within all relevant workspaces. In this tool, factors can be added to existing objects by either drawing a new factor in a blank area of the Build workspace or by opening an object in Plan mode and clicking on an “add factor” button. While this duplicate functionality was provided to make the relevant task accessible from both workspaces, our design inadvertently made the task more difficult in one third of the models, where students used the visible “add factor” button in Plan mode exclusively. Although drawing new factors in Build mode is more straightforward, there is no visible indication of this activity option and some students assumed that the “add factor” button in Plan was the only way to complete this task. Thus, while it is important to include all relevant tools and information in a workspace, designers must also ensure that these activity options are readily accessible.

Supporting Navigation

Decomposing learning activities to fit within the constraints of handheld screens makes the overall tool more usable, yet also introduces a local usability challenge – how to support students in navigating between different tools and activities. In the MaLTS tools, navigational support is generally provided through dual-purpose interface elements, such as the navigation bar displayed at the top of each ArtemisExpress workspace. The results of the classroom study indicate that the navigation bar made it easy for students to switch workspace in ArtemisExpress while also providing process scaffolding that made the components of the online research activity visible.

In contrast, another navigational task in ArtemisExpress – moving from websites to note taking areas inside the DQ folders – was not supported within the software because the handheld web browser provides a “back” button. During the classroom study, however, students were quickly annoyed at the need to click “back” repeatedly, and three students even resorted to alternative note-taking methods such as creating notes in a separate word processor and later copying them

into the DQ folder. Results such as these from the MaLTS study suggest that, although dual-purpose elements can provide navigational support, designers should ensure that common or particularly onerous navigational tasks are supported within the interface even if they cannot be combined with scaffolding to create a dual-purpose element. While this may reduce local usability and screen space, supporting common navigation tasks can reduce learners’ frustration and increase the overall usability of the tool.

Supporting Organization and Visualization

The constraints of handheld screens can make it difficult for learners to organize and visualize their work, particularly when artifacts are scattered across multiple workspaces or when viewing the workspace requires extensive scrolling. The MaLTS software allows students to scroll vertically and horizontally within the handheld workspace in order to view and modify their work. This technique worked well in ArtemisExpress, where students were able to scroll through websites and organize their bookmarks and notes with little difficulty. Students’ use of Pocket PiCoMap had more mixed results, however. While half of the students had no difficulty scrolling and creating complex, well-organized maps, the other half did little or no scrolling and their concept maps were poorly structured and often contained overlapping concepts and relationships. Ongoing research in alternative visualization techniques, such as zooming [10] or peephole displays [22], may offer solutions for helping students visualize large or complex work within small handheld screens.

Using Caution when Automating or Eliminating Tasks

Automating less-salient portions of the learning activity has two benefits: students can focus on more central aspects of the learning task, and designers can use automation to simplify the handheld interface. Yet if an interface is too automated, students may complete the task by rote rather than mindfully engaging and learning about the task. For example, in Pocket Model-It students use a pull-down menu to create relationships between factors. This menu includes a default value that inadvertently made the task too easy for many students, who simply accepted the default instead of actively selecting an appropriate description for each relationship.

Another cautionary lesson highlighted by the MaLTS study is that students may not realize that specific components of a learning activity are being automated, which causes problems when students use tools that expect students to complete these tasks on their own. For instance, saving was automated in both Pocket PiCoMap and Pocket Model-It. Although students were also instructed to save their work using a menu option, in practice students quickly learned that their work was saved automatically and they often neglected to save it manually. This habit proved problematic when students used ArtemisExpress, where saving was not automated, and several students inadvertently lost work when they forgot to save.

Evaluation of “Dual-Purpose Elements”

The dual-purpose elements in the MaLTS software were generally successful at providing both functionality and scaffolding. However, the classroom study does raise two issues that should be considered when designing dual-purpose interface elements for scaffolded handheld tools:

- Ensure that both the functionality and the scaffolding are visible and accessible in dual-purpose elements.
- Provide additional scaffolds within the workspace if the dual-purpose elements are insufficient, particularly for activities that students find complex or unfamiliar.

Ensure Functionality & Scaffolding are Visible & Accessible

One difficulty encountered during the MaLTS study was that students sometimes failed to recognize or to understand the dual roles of some interface elements. This was particularly true when the implementation combined graphical and textual features, such as in the factor editor in Pocket Model-It (Figure 7). The factor editor provides functionality to let students customize factors in their models and provides activity option scaffolding to make the customization choice visible to students.

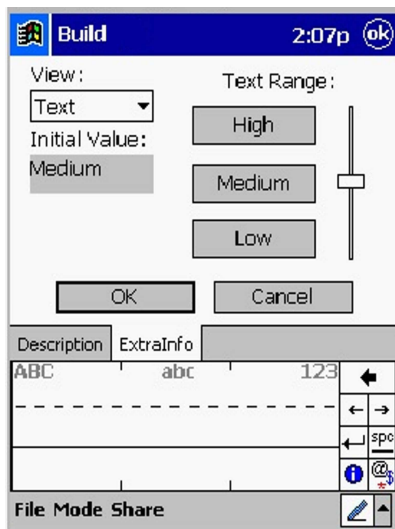


Figure 7: Pocket Model-It Factor Editor Workspace

During the classroom study, students customized factors in one-third of the models, usually by moving the slider to set the initial value of the factor. However, only one student discovered that the labels on the “High”, “Medium” and “Low” range buttons could also be customized by clicking on the button and changing the text. LCD research with desktop scaffolds has demonstrated that graphical implementations can be misleading to students if the scaffolds are not obviously changeable elements of the workspace [17], and the graphical implementation of the factor editor in Pocket Model-It proved problematic for students in the MaLTS study. To solve this usability problem, scaffolds could be added to make the re-labeling process visible, or another implementation, such as text boxes, could be used.

Additional Scaffolding for Complex or Unfamiliar Work

Integrating functionality and scaffolding in a single element can improve the usability of the interface and increase the amount of screen space available for engaging in the task. However, the results of the MaLTS study suggest that designers must be cautious not to exclude scaffolding if it cannot be integrated into a dual-purpose element. For example, the MaLTS study indicates that students’ need for visible process scaffolds increases as the complexity and unfamiliarity of the learning task increases. In Pocket PiCoMap, which does not include any visible process scaffolds, students were able to build substantive concept maps with little difficulty. Given that concept mapping was a straightforward and familiar activity for the students in the MaLTS study, our design decision to eliminate visible process scaffolds in favor of increased workspace resulted in a useful and usable handheld tool.

In contrast, students had difficulty understanding the overall process of scientific modeling when using Pocket Model-It, which did not incorporate sufficient, or sufficiently visible, process scaffolds. While the name of the current workspace was always visible onscreen, the full Plan-Build-Test modeling process was only visible within a menu option used to switch between workspaces. All of the students were eventually able to use Pocket Model-It, but many had trouble understanding the modeling process and found this handheld tool difficult and frustrating to use. Clearly, students required additional process scaffolds in order to understand and engage in this unfamiliar and complex learning activity.

CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK

By reviewing related work from educational psychology, LCD and UCD, the MaLTS project highlights some of the design challenges that must be considered when building scaffolded educational software within the constraints of handheld computer screens. The design guidelines described here – creating task-based workspaces and using dual-purpose elements – suggest decomposing learning activities into component tasks that can be supported within individual handheld workspaces. Within these workspaces, dual-purpose interface elements can be used to provide scaffolding and functionality, which allows necessary supports to be included in the software while preserving the usability of the handheld interface. The classroom study evaluates three handheld science inquiry tools developed according to these design guidelines, and the results offer several additional considerations for designing scaffolded handheld tools for learners.

The MaLTS study focused on the “effects with” handheld tools, or how students use these tools when engaging in learning activities. One avenue of future research are additional classroom studies to analyze the “effects of” [19] the MaLTS software, which means evaluating whether using scaffolded handheld tools changes students’ understanding of the underlying learning activity. A second area for further research is exploring alternative methods for addressing some of the design challenges and trade-offs identified

through the MaLTS study. For instance, recent research on the use of peephole displays [22] and zooming interfaces [10] might be adapted to allow students to view handheld workspaces across multiple scales, providing a “big picture” view of the workspace to help compensate for some of the loss of context and workspace imposed by the constraints of handheld tools.

ACKNOWLEDGMENTS

Special thanks to Adam Wiczorek, Kyle Reese, Nayan Nandihalli, Chris Farah, Chris Gleason, Ann Novak and the students and staff at Greenhills School. This work is supported in part by Intel Research, Microsoft Research and the National Science Foundation under grant number NSF ITR 0085946 and a Graduate Research Fellowship. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

REFERENCES

- Ananny, M. Supporting Children's Collaborative Authoring: Practicing Written Literacy While Composing Oral Texts. in *Proceedings of CSCL 2002*, Lawrence Erlbaum, Hillsdale, NJ, 2002, 595-596.
- Bergman, E. (ed.), *Information Appliances and Beyond*. Morgan Kaufmann, New York, 2000.
- Bransford, J.L., Brown, A.L. and Cocking, R.R. (eds.). *How People Learn: Brain, Mind, Experience, and School (Expanded Edition)*. National Academy Press, Washington, D.C., 2000.
- Danesh, A., Inkpen, K., Lau, F., Shu, K. and Booth, K. Geney: Designing a Collaborative Activity for the Palm Handheld Computer. in *Proceedings of CHI 2001*, ACM, New York, NY, 2001, 388-395.
- Davis, E.A. and Linn, M.C. Scaffolding Students' Knowledge Integration: Prompts for Reflection in KIE. *International Journal of Science Education*, 22 (8). 819-837.
- Davis, S.M. Research to Industry: Four Years of Observations in Classrooms Using a Network of Handheld Devices. in *Proceedings of the IEEE Workshop on Wireless and Mobile Technologies in Education*, IEEE Computer Society, Piscataway, NJ, 2002, 31-38.
- Golan, R., Kyza, E., Reiser, B.J. and Edelson, D., Scaffolding the Task of Analyzing Animal Behavior with the Animal Landlord Software. Presented at the Annual Meeting of the American Educational Research Association (AERA), (New Orleans, 2002).
- Gonzalez-Castano, F.J., Anido-Rifon, L. and Costa-Montenegro, E. A New Transcoding Technique for PDA Browsers, Based on Content Hierarchy. in Paterno, F. ed. in *Proceedings of the Mobile HCI 2002*, Springer-Verlag, Berlin, 2002, 69-80.
- Jones, M., Buchanan, G. and Thimbleby, H. Sorting Out Searching on Small Screen Devices. in Paterno, F. ed. in *Proceedings of Mobile HCI 2002*, Springer-Verlag, Berlin, 2002, 81-94.
- Jul, S. "This is a lot easier!": Constrained Movement Speeds Navigation. in *Extended Abstracts of CHI 2003*, ACM Press, New York, 2003, 776-777.
- Luchini, K., Quintana, C., Krajcik, J. and Soloway, E. Designing Learner-Centered Scaffolds Tools for Handheld Computers. in *Proceedings of ICLS 2002*, Seattle, WA, 2002, 268-275.
- Luchini, K., Quintana, C. and Soloway, E. Pocket PiCoMap: A Case Study in Designing and Assessing a Handheld Concept Mapping Tool for Learners. in *Proceedings of CHI 2003*, ACM, Ft. Lauderdale, FL, 2003, 321-328.
- Metcalf, S.J., Krajcik, J. and Soloway, E. Model-It: A Design Retrospective. in Jacobson, M.J. and Kozma, R.B. eds. *Innovations in Science and Mathematics Education: Advanced Design for Technologies of Learning*, Lawrence Erlbaum Associates, Mahwah, New Jersey, 2000, 77-115.
- Moher, T., Ding, X., Wiley, J., Conmy, D., Hussain, S., Singh, P. and Srinivasan, V. Combining Handhelds with a Whole-Class Display to Support the Learning of Scientific Control. in *Extended Abstracts of CHI 2003*, ACM, 2003, 882-883.
- Norman, D.A. and Draper, S.W. (eds.). *User Centered System Design*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.
- Novak, J. and Gowin, D.B. *Learning how to learn*. Cambridge University Press, Cambridge, 1984.
- Quintana, C., Krajcik, J. and Soloway, E. Issues and Approaches for Developing Learner-Centered Technology. *Advances in Computers*, 57. 271-321.
- Quintana, C., Reiser, B., Davis, E., Krajcik, J., Golan, R., Kyza, E., Edelson, D. and Soloway, E. Evolving a Scaffolding Design Framework for Designing Educational Software. in *Proceedings of ICLS 2002*, Seattle, WA, 2002, 359-366.
- Salomon, G., Perkins, D.N. and Globerson, T. Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher*, 20. 2-9.
- Shoemaker, P.B. Designing Interfaces for Handheld Computers. in *Proceedings of CHI 1999*, ACM, New York, 1999, 126-127.
- Wallace, R., Soloway, E., Krajcik, J., Bos, N., Hoffman, J., Hunter, H.E., Kiskis, D., Klann, E., Peters, G., Richardson, D. and Ronen, O. ARTEMIS: Learner-Centered Design of an Information Seeking Environment for K-12 Education. in *Proceedings of CHI 1998*, ACM, Los Angeles, 1998, 195-202.
- Yee, K.-P. Peephole displays: pen interaction on spatially aware handheld computers. in *Proceedings of CHI 2003*, ACM Press, New York, 2003, 1-8.