Teaching with Tangibles: A Tool for Defining Dichotomous Sorting Activities

Audrey Mbogho
CUNY Graduate Center
365 5th Ave.
New York, NY 10016,USA
email: AMbogho@gc.cuny.edu

Lori L. Scarlatos
Brooklyn College, CUNY
2900 Bedford Ave.
Brooklyn, NY 11210,USA
email: lori@sci.brooklyn.cuny.edu

Magdalena Jaworska
Brooklyn College, CUNY
2900 Bedford Ave.
Brooklyn, NY 11210,USA
email: worska@yahoo.com

ABSTRACT
Learning activities with tangible user interfaces provide the benefits of active and peer mediated learning, while offering assistance from an autonomous guide on the side. Yet tangible user interfaces must typically be custom developed by computer scientists, and only rarely is the assistance of teachers sought.

We present a new tool that gives teachers the power to create their own educational applications with tangible user interfaces. Using actual scientific specimens, the teachers can define object attributes for the students to sort on, and also develop curriculum-appropriate hints. We describe our computer vision-based approach, which enables recognition of tags representing dichotomous keys defined by a teacher. Teachers use our back-end system to define the dichotomous keys and other parameters for the learning activities, while our front-end system uses those parameters to guide the students.

KEYWORDS: Tangible user interface, multi-modal input, multi-user interface, computer vision, multimedia feedback, educational application, dichotomous key.

INTRODUCTION
Scientists and naturalists around the world use dichotomous keys to identify and classify objects and organisms. Teachers use these concepts of classification to make students think in terms of attributes and values. As a result, students become more aware of the commonalities and differences among the entities that they are studying. When children work in groups with real physical objects (such as rocks, shells, leaves, etc.), they can reinforce their knowledge and understanding through active exploration. Yet for one teacher with up to 35 students in the classroom, providing adequate guidance to all of the groups can be challenging.

Tangible user interfaces provide a potential solution to this problem. With this paradigm, physical objects represent data in the computer, providing interface elements that groups of children can pick up, examine, and move around. Because the computer knows about the physical objects, it can give children the hints, suggestions, and encouragement that they need. The computer application is able to keep the children on task and record what they do, so that the teacher can go back and talk with the groups later on. The biggest problem with this approach is that computer scientists are typically needed to develop applications with tangible user interfaces. By themselves, computer scientists do not necessarily know what the children need to learn in the curriculum, or how the material should be presented. Even when teacher collaborators are employed, the resulting applications cannot be readily modified to meet the needs of a particular teacher or class of students.

We propose a system that allows teachers to create their own customized dichotomous sorting activities with tangible user interfaces. Teachers are given a back-end application that lets them define the subject matter, attributes that each object may possess, and possible values for those attributes (dichotomous keys). Teachers also specify the wording of questions to be given to the students (e.g. "Find all the aircraft that get <attribute> from <value>"). In addition, teachers can specify hints to be given when the children are having trouble. This produces a file which drives the front-end application that the children use. The front-end educational application asks the children to place objects with common attribute values onto a TICLE (tangible interface for collaborative learning environments) table. The application then checks what they have done, asking questions, providing encouragement, and giving hints as needed or requested.

 

Figure 1: Children have a tendency to handle objects in their environment, which makes TUIs natural for them.

Our system is unique in that it provides the following innovations and benefits.

- Teachers can define their own educational applications to fit their curriculum and their students' prior knowledge and abilities.
- We have developed a unique system for tagging items such that the computer can discern the values of multiple attributes for a given object.
- Students can be engaged with actual physical objects, such as leaves, shells, and rocks, while getting needed assistance from a computer system.

A major drawback in computer technology, and a chief concern of HCI research, is the complexity of the typical user interface. In the developing world where access to computers is limited, there are few opportunities to build familiarity with user interfaces sufficient for effective use of computers. On the other hand, handling of physical objects is quite natural, and children, especially, have a high affinity for doing so. Intuitively, then, tangible user interfaces can be expected to be easier to learn and to provide more transparent interaction with the application than traditional graphical user interfaces (GUI). In a classroom setting, time is usually brief, and it is beneficial to have a TUI-based system such as ours that gets the students up and running quickly.

RECENT WORK

Tangible Interfaces for Collaborative Learning Environments (TICLE) provides tools and techniques for creating learning activities enhanced with tangible user interfaces [8]. As part of this, we developed a TICLE table, which provides a surface for children to work on. A camera mounted below the Plexiglas tabletop "watches" as objects are placed on it and moved around. The educational application then uses this visual input to trigger verbal comments ("You're doing great!" versus "Would you like a hint?") and select appropriate hints when they are requested. Our first prototype had children solve Tangram puzzle problems. Observations of children using this system [10, 11] suggest that this interface keeps students on task and encourages discussion of what needs to be done and how to accomplish that. Students using our system were also far more likely to find a solution than the control groups. This is consistent with the view in educational research that combining approaches—behaviorist and social constructivist in this case—produces a better learning experience [8].

Others have also developed tangible user interfaces with educational applications in mind. The Illuminating Light project at the MIT Media Labs used similar vision-based techniques to tag and track objects in an optics simulator [12]. An earlier Triangles project uses physical triangular pieces to represent information that can be connected in a variety of ways [2]. Tangible interfaces have even been integrated into museum exhibits to educate the public, such as the Einstein exhibition at the American Museum of Natural History [1]. The wide variety of possibilities for tangible user interfaces in educational applications has led some to form conceptual models for tangibles and how they are used in learning [3].

Work is also being done to create tools that enable end-users to create applications with tangible user interfaces. At the MIT Media Labs, tangible interfaces are being integrated into interactive storytelling systems [4]. At the University of Maryland, where children are regularly involved in the design of software for children, a system has been developed that enables children to define interactive environments [6].

All of these applications demonstrate the exciting educational possibilities that come from new interface paradigms such as tangible user interfaces, perceptual interfaces, and ubiquitous computing. Yet none of these provide a way for teachers (without computer science training) to develop educational applications with tangible user interfaces for the science curriculum. Although educators may have been consulted in the process, they do not serve as more than just advisors or consultants. Our new system changes all that.

SYSTEM OVERVIEW

We have developed the following pieces for this system. First, we developed a back-end application for defining attributes (categories) and their possible values. This application produces an activity file which also specifies what hints are to be given and how questions are to be posed. Second, we developed an Xtra (a library of functions called by Lingo in a Macromedia Director application) that looks for – and interprets – bar codes within rectangular regions of a specified color. Third, we developed a front-end application that uses the activity file to generate challenges for the students, check their work, and provide hints if needed.

IDENTIFYING SPECIMENS

In dichotomous sorting activities, specimens have a variety of attributes or categories that help to define those specimens. For example, rocks might be characterized by their cleavage (crystal form), luster, or hardness. These form the dichotomous keys that can be used later in sorting. We needed a way of representing distinct values for several attributes simultaneously.

We decided to use a bi-directional barcode to represent the different values of an attribute. Each different attribute type is then identified by the barcode's background color (in the enclosing rectangle). We use highly saturated colors on a dark background to simplify the process of finding each barcode in the camera image.

A Color Code and a Method for Reading it

This color barcode reads the same forward and backward, for ease of processing. It represents 16 values but can be easily extended to encode as many categories as necessary. The method for reading it requires three distinct colors, one for the background, one for binary 0, and one for binary 1. There are two guard bars, one at either end. The guard bar marks the beginning of the code and also gives the color representing binary one. Following the leftmost guard bar, the next four bars are read as binary digits from left to right. The remaining half of the code is a mirror image of the first half, which allows it to be read in either direction. Figure 1 shows an example.

Figure 2. Red code for the number 9 (1001). The left half mirrors the right half of the code.

The barcode reading algorithm, adapted from [7], is summarized in the following steps.

1. We find the bounding boxes of barcodes of color *x* (*x* is red in figure 2 above).

2. We scan the bounding box in at most four directions—top to bottom down the middle, left to right across the middle, top left corner to bottom right corner, and bottom left corner to top right corner. We stop with the direction that finds a valid tag (only the correct direction will find all 10 bars). Because the barcode is bi-directional, it is not necessary to scan in the other four directions that run opposite these. This significantly reduces computational overhead.

3. We read and record each bit encountered while scanning. Reading involves counting the color transitions that indicate that a bar (other than the guard bar) has been completely crossed. Thus in figure 1, starting from the second white bar, we note the transitions white-to-red (1), black-to-red (0), black-to-red (0), white-to-red (1).

4. We stop recording the bits read as soon as four are found, which means the middle of the tag has been reached. Scanning does continue to the end, however, in order to determine the validity of the tag. If 10 bars are found, the procedure reports the recorded bit string and exits. If a direction finds fewer than 10 bars, the recorded bits are ignored and the procedure begins again in another direction.

It is important to note that image quality is crucial for the accurate reading of tags. This issue is discussed in greater detail in [5], where it is shown that because the classroom environment can be controlled, it is particularly amenable to camera based applications.

DEFINING THE ACTIVITY

When we designed our back-end system for teachers, we wanted them to be able to define a wide variety of dichotomous keys and sorting activities for a wide range of student ages. At the same time, we wanted the system to be simple to use.

On startup, our application asks for a file name. The user can type a new file name and create a file or type an existing file name to open the file. When a new file is created, the user is directed to a second screen that asks for a subject name. In this screen, the teacher is also asked to enter the standard form for the prompts or questions to be posed to the students. For example, if the teacher is preparing a lesson on rocks and minerals, the question might be, “Which rocks have a <category> that is <value>?” When the user is done entering information on this screen (or if an existing file is opened), the application jumps to the third screen where categories and their values can be Added.

In the Add screen, the teacher defines different categories. Each category can have up to 16 values. A summary of what has already been entered (subject name, category number, category name and number of values) is displayed in the text field on the bottom of the screen. Figure 3 shows how to specify questions and related information in a rocks application. In the Edit screen, the teacher can modify entered data by typing the category number. After clicking the Display button, the specified category and its values get displayed in the text fields. Then the user can set the changes and save them in the database. On the Delete screen, entire attribute categories can be deleted by using the category number. The text field on the bottom of the screen reflects all the changes.

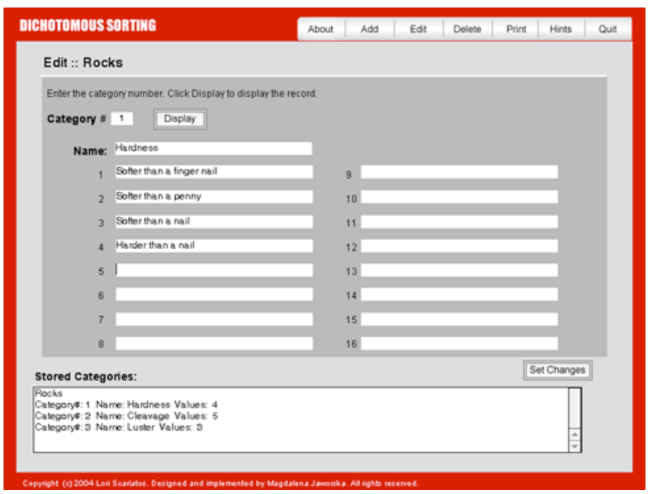

Figure 3: Back-end interface allows teachers to define categories (attributes), possible values for those attributes, hints related to the attributes, and questions regarding the attributes.

After all the categories and their values have been entered, the teacher can use the Print screen to print out everything that has been entered (subject name, category number, category name and its values). This screen also contains customized labeling instructions for the teacher. To identify each category and its values, we decided to use barcodes with different color backgrounds. The red background represents category one, whereas yellow and blue are designated for categories two and three respectively. The barcode within each sticker represents the specific value. For instance, if category 2 is lift and value 1 is wings, the teacher will be told to use blue sticker #1 on the bottom of every object that gets lift from wings. A picture of the barcode appears next to instruction for clarity. The barcodes themselves can be printed out separately. The teacher is expected to use this guide to label all of the objects that will be used in the dichotomous sorting activity.

A Hint screen provides a place where teachers can specify the background material and hints that should be provided by the application to the children. Teachers are expected to create their own supplementary material in a format – HTML, XML, SMIL, Shockwave, etc. – that can be displayed by a web browser. By using web formats, we allow the teachers to use familiar tools to create their hints. Teachers can even have their lesson link to web pages on the Internet. Teachers can enter one file or URL for the background material, and an unlimited number of hint files or URLs for each attribute category that they have specified. If the children ask for help during their activity, the front-end application will show them these hints – for the appropriate category – in the order that they were listed.

Finally, an About screen contains a brief description of the program and a few examples of Dichotomous sorting. Once the teacher is finished with setting the activity file, he/she may exit the program by clicking the Done button. All the data gets saved automatically in the text file which will be used in the front-end application.

IMPLEMENTING THE ACTIVITY

Our front-end application uses the activity file – produced by the previous application – to drive the educational activities for the children. The teacher simply needs to copy the appropriate activity file to a file named Dsorting.ini in the directory where this application resides, and then start the application.

A group of children will be sent to the TICLE table by their teacher. A screen behind the table will ask the students to touch the screen to begin. This generates a time-stamped file that will record the children’s activities. A set of objects to be sorted is arranged around the perimeter of the table, out of the camera’s view.

Children then have the option of reviewing the background material, or starting the activity. If no background material is provided by the teacher, the application will simply jump to the first part of the activity: sorting on attributes.

Sorting on Attributes

Using the questions posed by the teacher, the application asks the children to find all objects that have a particular value for a particular category (attribute). These instructions are vocalized by a simulated voice, while text printed on the screen reinforces them. The actual values and categories are selected randomly. The students are expected to find all of the objects that meet the requested criteria, place them in the center of the TICLE table, and then touch the Done button on the screen when they are done. This gives the children ample opportunity to examine the objects, discuss the objects among themselves, and even look at hints related to the selected category.

The application then checks the barcodes on the table. If all of the objects on the table have the correct value for the given attribute, the application congratulates the students and goes on to the next random question. If a wrong barcode is detected, the children are shown a hint to reinforce their understanding of the category that they are currently considering. They are then asked the same question again.

Occasionally the system will ask, “Are you sure?” when the children touch the screen. This will happen whether they are right or wrong. This will teach the children to be confident in their answers and abilities.

Exploratory Sorting

After the children have amply demonstrated that they understand the attribute categories and their values, the application will ask them to make their own groupings. This allows the children to test their own understanding of the subject matter, and think creatively.

When the children have their selected grouping on the TICLE table, they touch the screen. The application then tries to “guess” what those objects have in common. If the application cannot determine this (perhaps because the children have thought of another attribute), the computer will tell them so. In either case, the children’s groupings are recorded in the time-stamped file (which has been recording their activities) so that the teacher can discuss these later. He or she might also have the children record their groupings on paper, and write about what those objects have in common.

Figure 4: Barcodes on the rocks indicate their crystal form, luster, and hardness.

Sample Subjects and Their Categories

We have collected and labeled a number of objects to be used in testing this front-end application. Figure 4 shows a collection of rocks and minerals that we have tagged. For this subject, our categories are cleavage or crystal form (which can be basal, cubic, rhombohedral, prismatic, octahedral, or dodecahedral), luster (metallic, shiny, glassy or earthy), and hardness (softer than a fingernail, softer than a copper penny, softer than glass, and harder than glass). By using the actual rocks, students can pick up, study, and even scratch the minerals to determine what the values of their attributes are (figure 5).

Figure 6 shows a collection of shells that we have tagged for a lesson on mollusks. For this subject, our categories are shell type (univalve or bivalve), shape (round, oblong, triangular, fan, pointed), and texture (smooth, prickly, ridged, knobbed). By using the actual shells, students can actually feel the textures and examine the hinges or openings in the shells.

Figure 5: Students sorting actual rock specimens can test hardness by scratching the rocks with a nail.

Figure 6: Barcodes on the shells indicate their shell type, shape, and texture.

FUTURE WORK

Our software will be made available to graduate students (most of whom are practicing teachers furthering their education) in the School of Education. Graduate students in the Advanced Practice in Science course will be encouraged to use this tool to develop science lessons that they will then bring to their elementary school classrooms to try out. These students/teachers experiences and observations will then be able to tell us how easy the application is to use, and make suggestions for improvements. They will also be able to provide feedback on how effective this application is for teaching students about attributes and sorting.

Finally, we intend to make this application freely available to teachers through our web site. Included at this site will be instructions for constructing a TICLE table from readily available materials.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Eleanor Miele for initially suggesting that we apply tangible user interfaces to dichotomous sorting activities. We are grateful to her for providing a forum where we could test these ideas out. We would also like to thank Denise Dixon for being our first teacher/user, and providing the content for our sample application. This work is being supported by the National Science Foundation under Grant No. 9984385 and 0203333, and by a grant from PSC-CUNY.

REFERENCES

1. Abinader, G., Stein, J. and Lenore, M. Moving clocks and bending space: a learning/interactive museum environment (LIME), *Educators program from the 30th annual conference on Computer graphics and interactive techniques*, (2003) 1 – 1.

2. Gorbet, M.G., Orth, M. and Ishii, H. Triangles: tangible interface for manipulation and exploration of digital information topography, *Proceedings of the SIGCHI conference on Human factors in computing systems* (1998), 49 – 56.

3. Marshall, P., Price, S. and Rogers, Y. Conceptualising tangibles to support learning, *Proceeding of the 2003 conference on Interaction design and children*, 101 – 109.

4. Mazalek, A., Davenport, D. and Ishii, H. Tangible viewpoints: a physical approach to multimedia stories, *Proceedings of the tenth ACM international conference on Multimedia* (2002), 153 - 160.

5. Mbogho, A.J.W. and Scarlatos, L.L. Towards Reliable Computer Vision-Based Tangible User Interfaces. *Proceedings of IASTED-HCI 2005* (to appear).

6. Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W. and D'Amour, A. Physical programming: designing tools for children to create physical interactive environments, *Proceedings of the SIGCHI Conference* (2002), 299 – 306.

7. Parker, J.R. *Practical Computer Vision Using C*, John Wiley and Sons, 1994.

8. Quintana, C., Krajcik, J., Soloway, E. and Norris, C. A Framework for Understanding the Development of Educational Software. *The Human-Computer Interaction Handbook* (Jacko, A.J. and Sears, A., Eds), Lawrence Erlbaum Assoc. Mahwah, NJ (2003), 824 – 834.

9. Scarlatos, L.L. TICLE: Using Multimedia Multimodal Guidance to Enhance Learning. *Information Sciences 140* (2002), 85-103.

10. Scarlatos, L.L. and Landy, S.S. Experiments in Using Tangible Interfaces to Enhance Collaborative Learning Experiences, *CHI 2001 Extended Abstracts*, 257-258.

11. Scarlatos, L.L. An Application of Tangible Interfaces in Collaborative Learning Environments, *SIGGRAPH 2002 Conference Abstracts and Applications*, 125-126.

12. Underkoffler, J, Ishii, H. Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface, *Proceedings of CHI '98*, 542-549.