End-User Programming to Support Classroom Activities on Small Devices

Craig Prince
University of Washington
cmprince@cs.washington.edu

Abstract

We believe it is unreasonable to assume that all students will own a laptop. One potential solution is to depend on the students to bring whatever computing devices (cell phones, portable gaming devices, etc.) that they already own to class. This leaves a unique challenge for teachers both in creating activities that work across a wide set of devices with different input modalities and in interpreting the responses to these activities. We seek to explore an end-user programming solution for teachers to abstractly define an activity for use over a diverse set of devices – allowing for easy distribution and aggregation of the resulting responses.

1. Introduction

We believe that the future of technology in the classroom is one where every student will use some sort of computing device – exactly how this will occur is still uncertain. Most current efforts focus on provide a laptop for every child or at least a classroom set of laptops. However, we believe that this is not a sustainable long-term solution to having computers in the classroom. By focusing on laptops we are ignoring a large spectrum of low cost computing devices on the market today that people already own. These range from low-cost cell phones, to gaming devices, to ultra-mobile PCs. As of 2007, 86.1% of U.S. undergraduate students owned a cell phone (56.3% owned a gaming device) [1]. Such devices are well suited to the classroom because they 1) are light-weight, 2) have a long battery life, and 3) are relatively low cost.

Previously we have developed Classroom Presenter (CP) a Tablet PC-based application that allows teachers to give students activities electronically during class. The students can then complete the activities and submit their responses to the teacher in real time. The teacher can then view these responses, integrating them into the lecture. This has been shown to be effective [2]; however, deployment has been an issue because most students do not own a Tablet PC. An obvious solution is to port CP to other devices.

Porting CP to personal hand-held devices holds many potential benefits for learning. Schools that cannot afford a classroom set of computers can still have students gain from the benefits of using computers to augment classroom lectures. Also, integrating computing devices into the learning process provides motivation to both teachers and students to make class more interactive [2] and to learn to use the technology effectively as a tool for education.

2. The Problem

If every student potentially has a different device with different input modalities and different feedback capabilities, several interesting challenges arise. Consider, for example, an activity asking the students to draw a graph with a clique. On a Tablet PC, this can be done with digital ink; however, on a cell phone it might be more appropriate to have the student simply draw vertices and edges by placing predefined icons. Deciding the “best” interaction to complete an exercise on a given device requires specialized knowledge, and we do not want to burden the teacher with having to decide how the exercise should be done on every potential device that may be in his/her classroom.

Another challenge is how to interpret the responses from the various devices. Not only might the responses be using different digital media (e.g. digital ink vs. shapes) the solutions themselves might be different visually but correct semantically. In our example above, one student might create a 3-clique, while another creates a 4-clique. These graphs have different structures, so graph isomorphism is not enough.

3. Proposed Solution

Our proposed solution to these problems is to allow teachers to specify activities using a declarative language in a way that allows CP to automatically 1) target the activity to the capabilities of each student device and 2) aggregate the results from each device into a form that is useful to the teacher.

We propose three steps to specify an activity. The first step is to author the activity content. This is currently done in CP by creating a PowerPoint slide – which we will continue to use. We will augment this by allowing the teacher to specify additional properties for each object on the slide (moveable, resizable, etc.) or for the slide as a whole (e.g. can new objects be created? which objects?).

The second step is to specify the semantically important parts of the exercise. For example: does the exercise involve creating a new diagram, modifying an existing diagram, or highlighting a part of a diagram?; what scale is important for the response – a single point, an area, or an exact boundary?; what number of responses is required?; etc. The specifications from the first two steps should allow our system to decide on the type of interactions needed to complete the activity and select the most appropriate UI for each device.

The third step of the specification is code for grouping and classifying answers. We plan to use a visual language similar to HyperFlow [3]; however, the focus will be on clustering and classifying student responses. The teacher will use the visual language to create a dataflow diagram where the input is the student response and the output is one or more labels. Each node in the diagram can include either logical primitives and/or functions. This code will have access to all the content and specifications from the previous two steps. In [4] they found that having built-in functions were very useful for teachers; we will provide built-in functions for: 1) machine learning, 2) sketch understanding, and 3) graph operations.

In order to encourage teachers to use the system we will implement two tools to allow teachers to progressively improve their expertise. The basic CP system will require no special skills beyond the ability to make PowerPoint slides. Once a teacher is comfortable with the system he/she will be able to use a specialized authoring tool/wizard to assist in the creation of activity specifications. This wizard will have some but not all of the power of the language itself. As teachers become more and more acquainted with the system they can begin to work directly with the language. This will be done via a “specification editor”. This editor can be used to examine and change the specifications generated by the wizard or to directly author a specification from scratch. This tool is vital because it allows teachers to share activities between each other – modifying them to suit the needs of a specific class. Sharing was found to be important to teachers in [4]. Teachers also can learn from what others have done and become better at using the system themselves.

4. Current and Remaining Work

So far our work has been focused on building the CP application and developing pedagogy for it. As part of this work, we have collected a large number of examples of activities that teachers have used CP for on the Tablet PC. These are useful as a baseline for the types of activities we would like to support over all hand-held devices. In [5] we compared CP on the Tablet PC with a paper version of CP utilizing Anoto Pen technology. This pilot study allowed us to explore some of the limitations of not having the feedback associated with the Tablet PC screen. Currently we have begun development on a version of CP for the Nintendo DS.

To complete this work the first step is for CP to be ported to a variety of devices. Then the work will focus on implementing the tools and language specified above. As we complete each part of the outlined solution we will conduct both lab studies and in-class deployments to validate each component.

We will pay particular interest to the tool/wizard for helping teachers create good specifications. One challenge here will be in making it clear to users how CP will use the specification generated by the tool. Another challenge will be in balancing the expressiveness of the language against its usability by teachers with little programming experience.

5. References

[1] The ECAR Study of Undergraduate Students and Information Technology, 2007.

[2] Anderson R.J., Anderson R.E., Davis K.M., Linnell N., Prince C., Razmov V. “Supporting Active Learning and Example Based Instruction with Classroom Technology”. SIGCSE 2007.

[3] Dotan D., Pinter, R.Y. “HyperFlow: an Integrated Visual Query and Dataflow Language for End-User Information Analysis”. VL/HCC 2006.

[4]Wiedenbeck, S. “Facilitators and Inhibitors of End-User Development by Teachers in a School Environment”. VL/HCC 2006.

[5] Liao C., Guimbretiere F., Anderson R., Linnell N., Prince C., Razmov V. “PaperCP: Exploring the Integration of Physical and Digital Affordances for Active Learning”. INTERACT 2007.