Middle School Students Using Tomorrow's Technology

Loki Jörgenson* Nathalie Sinclair†‡

Simon Fraser University, Burnaby, B.C. CANADA V5A 1S6

Abstract

Intending to create a constructionist environment for interactive learning, we are developing a JavaBeans-based tool building kit. It supports teachers' and students' use of Lego-like components to construct purposeful tools for teaching and learning mathematics. Due to the nature of the technology, it requires the user adopt an object-oriented, event-driven framework - a reasonable requirement for C++ and Java programmers but not typical for most middle school students.

Despite apparent technological handicaps and obstacles to learning, we were surprised at how readily the students assimilated the concepts and applied them creatively. This far and away exceeded our expectations and pointed at the tremendous potential borne by such constructionist technologies.

Keywords constructionism, mathematics education, JavaBeans(TM), collaboration, telelearning, applets, participatory design

Figure 1: The Island Pacific School on Bowen Island near Vancouver, British Columbia

1 Introduction

There has been growing support for component-oriented architectures for educational software. For example, diSessa [1] has described and advocated "Open Toolsets" such as Boxer; these are flexible and malleable collections of components which can be combined to create "microworlds". Microworlds offer students an opportunity to explore the interaction of elements they have constructed themselves. See also [7] for a description of component-oriented exploratory software for mathematics.

There are technical, social and pedagogical motivations for investigating the potential of component-oriented architectures. Their primary benefit is that sets of specific- or general-purpose tools can be built rapidly and relatively cheaply; they then can be easily modified, extended and combined to yield more tools. Moreover, tools built by other development groups can be integrated and customized to meet the needs of a wide community of learners [2, 5].

This study investigated the use of JavaBeans[1], a component-oriented architecture somewhat resembling a software version of Lego blocks. A set of JavaBeans specific to a certain activity or topic (such as mathematics) can be constructed and then subsequently interconnected as desired to form interactive applications. Such applications offer learners environments in which to visualize, transform and simulate mathematical concepts, processes which enable them to achieve deep conceptual understanding [6].

While there is confidence in the credibility of such an approach, it was not clear that it could be successfully applied by middle school students. Previous studies have shown that students are able to use object-oriented structures combined with LOGO [3] to build programs and models. However, in this case, potential barriers included

*Centre for Experimental & Constructive Mathematics

†Island Pacific School

‡PolyMath Development Gro up

[1] A programming standard advanced by Sun Microsystems

- the necessity of adopting an event-driven style of construction,
- the limited depth and breadth of the existing toolkit for building useful tools and resources,
- the toolkit's focus on a restricted domain of mathematics (geometrical transformations)

It was not clear that students would be able to adapt to the constraints of the implementation or learn concepts beyond the simple programming that would be introduced. The students were engaged with the technology as "design partners" in a process of guided collaboration; this cooperative research approach was based on the methods of participatory design [8,9].

2 Study Overview

The study involved 24 students at a middle school on Bowen Island, from grades 7 through 9. Author Sinclair was the mathematics and information technology teacher at the school and also a research associate at the Centre for Experimental and Constructive Mathematics at Simon Fraser University. Members of the PolyMath Development Group[2] developed and supported the use of the technology and assisted with some of the implementation. Members of the Assessment of Technology in Context Lab[3] observed and documented the classroom dynamics.

The first phase of the study introduced the students to applications that had been developed by author Sinclair using a specially designed mathematics JavaBeans toolkit. The students spent approximately 10 hours exploring and investigating different aspects of transformational geometry using these applications. Throughout, by means of oral and written communication, they were encouraged to reflect both on their own learning and on the technology. Although the students enjoyed the interactive and playful aspects of the applets, they observed that there were some weaknesses. These corresponded directly to the limitations of the toolkit in both size and scope.

In the second phase of the study, the students were introduced to a low-tech design tool called simCHET. It provides a facsimile environment that supports group learning and interaction, hands-on construction and heuristic design of tools; it is modelled after the functionality of a typical JavaBean construction tool, Java Studio (see Figure 2). Java Studio offered the level of functionality and user support that was viewed necessary for the students. However there were several technical problems that made it impractical to use. Consequently simCHET was employed to introduce the students to the principles of event-based, component-oriented construction and to develop their understanding of how such tools might be employed in their learning.

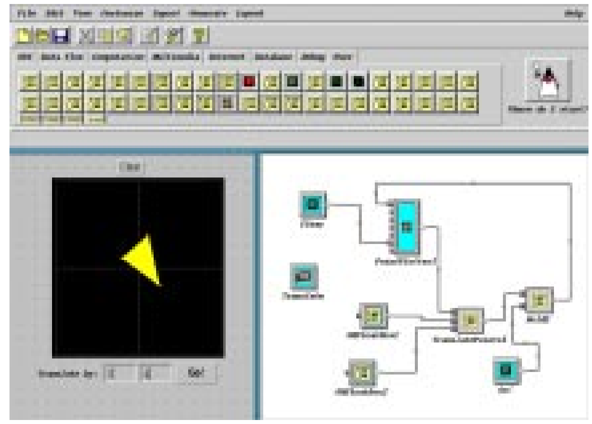

Figure 2: The JavaStudio construction interface: Note the applet being built on the left with its construction schematic on the right.

3 simCHET Implementation

With guidance from authors Sinclair and Jörgenson, the students identified the components required to construct a simple applet, defined their functionalities and dynamics, and inter-connected them to create a prototype applet. This session lasted two hours during which extensive elaboration and discussion was conducted in order to clarify the notions of components and events.

In a subsequent one-hour session, the students attempted a design of a slightly more complicated applet which they had all previously used. This was intended to reinforce their experience on the previous day without invoking all the support and materials needed for a full run. The students were able to construct the applets without difficulty in a much shorter time period.

The final session engaged students to employ simCHET in designing their own applets using the JavaBeans toolkit. They were encouraged to develop their own ideas around the general theme of geometrical transformations. The researchers stressed the fact that their creativity would be particularly welcomed, as would their suggestions for new JavaBean

[2] Responsible for network technology development within the CECM; directed by Dr. Stephen Braham

[3] Headed by Dr. Ellen Balka, School of Communications

Figure 3: A full simCHET-designed applet

components. The students began by brainstorming ideas both individually and in groups. They were very excited by their ideas and very comfortable working within the framework of the existing technology. As they saw the need for a new JavaBean, they would inquire whether it was necessary and possible to make.

The students then planned the layout of their applets and made a list of JavaBeans that would be needed in order to make the applet work. After having discussed their designs with their teacher and members of the PDG team, the students began the construction of their applets. Figure 3 shows an example of a completed design.

4 Observations

Initially it was anticipated that students would have difficulty adapting to the JavaBeans framework. It was expected that they would be able to reproduce the recipes for the applets they were shown and to comment on their experience with the technology. It was not clear how well or how quickly they would grasp the nature of a JavaBean or be able to adopt the construction methods of Java Studio.

In the initial simCHET session where JavaBeans were introduced, it was quite clear that most of them had no problem with the concepts. In the following session they verified this by accurately reproducing the design for the target applet using simCHET. In their final session, they were asked to be creative and original in their design using their acquired knowledge.

As the students worked on developing their own applets it became clear that the JavaBeans available to them were insufficient. Students repeatedly requested Timer JavaBeans, Collision JavaBeans, and Drawing JavaBeans. Although they were not constrained in their design ideas, it was evident that in order to support the imagination of the students, a more sophisticated set of JavaBeans would have to be provided.

Remarkably though, the students demonstrated the depth of their understanding by explicitly suggesting new directions for research. They anticipated some of the JavaBeans still under development such as the Collaboration JavaBeans (these would allow students to construct tools which could be simultaneously shared across the network) and, by the requirements of their designs, defined the priorities for subsequent research.

Prior to the final session, we were very concerned that the students would feel restricted by the limitations of the toolkit. It was expected that they would only be able to reproduce one of the applets that they had already seen. However, the students were able to create applets which, though grounded in the concept of transformation, exhibited much more complex ideas.

One group of students designed a labyrinth game applet in which Theseus chases the Minotaur through a 3-dimensional maze by means of selected transformations. Another group created a falling objects game which drew on the popular Tetris game but required explicit descriptions of transformations; it was designed to be played by multiple computers across the network. Another group created a wallpaper making studio in which users could create designs by using tesselations. The students were able to adopt the concepts and design applets which had personal meaning for them and which they could share with the class. This is exemplary of the kind of constructionist learning environment desribed by Papert [4].

5 Conclusions

The evidence from this study at Bowen Island School indicates that middle school students are easily capable of using JavaBeans- based technologies to program tools in support of their own learning activities. While they were not exposed to the preferred construction environment, JavaStudio, they showed little difficulty in adopting and applying the low-tech facsimile tool simCHET in a creative manner.

In particular, we found that the framework for constructing tools using event-driven components was easily within their grasp; this remains to be confirmed through the use of the actual JavaBean toolkit. Their understanding was reflected in their accurate assessment of the shortcomings of the technology in its current state (insufficient range and depth in the toolkit). Further, we found that the "premature adoption" of the technology was validated by their ability to apply it creatively in design while remaining faithful to its constraints.

The experimental approach to cooperative design was surprisingly effective[4]. The students were able to priorize the development of subsequent JavaBeans and even to propose JavaBeans not yet considered by the developers. They demonstrated an ability and an interest in working with researchers.

Acknowledgements

This project has been made possible by the work of the entire PolyMath Development Group: Stephen Braham, Trevor Bradley, Carlton Chan, Jen Chang, Paul Irvine, and Terrance Yu as well as authors Jörgenson and Sinclair. Also essential was the work of the ATiC lab, director Ellen Balka and ethnographer Michael Jones.

We would also like to thank BC Tel Advanced Communications for donation of an ISDN line to Bowen Island, Zentra Computers for donations of PC hardware, and Innovative Computing Solutions and Redesign for their networking support.

This work has been in part supported by research and equipment grants from the TeleLearning - National Centre of Excellence (TL-NCE) and the Pacific Institute for the Mathematical Sciences. We would also like to thank the Centre for Experimental & Constructive Mathematics, the Assessment of Technology in Context lab and the Island Pacific School for their support and participation in this project.

References

[1] diSessa, A. A., (1997). Open toolsets: New ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.) *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education, Vol. 1.* Lahti, Finland, 47-62.

[2] Kaput, J and J. Roschelle, Educational software architecture and systemic impact: The promise of component software. Journal of Educational Computing Research, 14(3):217-228, 1996

[3] Noss, R., Lulu Healy and Celia Hoyles. The Construction of Mathematical Meaning: Connecting the Visual with the Symbolic. Educational Studies in Mathematics (in press)

[4] The process of "guided collaboration" was dynamically developed as part of the project - it was based on well established theories and techniques drawn from participatory design

[4] Papert, S., (1991) Situated Constructionism, in S. Papert and I. Harel (Eds.) *Constructionism* , Ablex Publishing Corporation.

[5] Roschelle, J., Spoher, J.: 1997. Banking on Educational Software: A Wired Economy Unfolds *TECHNOS Quarterly*, Vol. 6, No. 4. www.technos.net/journal/volume6/4roschel.htm

[6] Roschelle, J. and Kaput, J. (1996). SimCalc MathWorlds for the Mathematics of Change *Communications of the ACM*, 39 (8), 97-99.

[7] Kynigos C., Koutlis M. and Hadzilacos Th. 1998: "Mathematics with Component Oriented Exploratory Software", to appear in the International Journal of Computers in Mathematics Education, Kluwer Academic.

[8] Druin, A., Bederson, B., Boltman, A., Miura, A., Knotts-Callahan, D., and Platt, M. (1998) Children as Our Technology Design Partners *The design of Children's Technology: How we design and why?* Druin, A. (ed) Morgan Kaufmann.

[9] Carroll, J. M. (ed) (1995) Scenario-based Design, John Wiley and Sons