Proceedings of the Eighth IASTED International Conference
COMPUTERS AND ADVANCED TECHNOLOGY IN EDUCATION
August 29-31, 2005, Oranjestad, Aruba

COMPUTER-ASSISTED POP-UP DESIGN FOR CHILDREN: COMPUTATIONALLY-ENRICHED PAPER ENGINEERING

Susan L. Hendrix and Michael A. Eisenberg
Department of Computer Science and Institute of Cognitive Science
University of Colorado, 430 UCB, Boulder CO 80305 USA
Tel: 303-492-8091, Fax: 303-492-2844
Email: hendrixs@cs.colorado.edu, duck@cs.colorado.edu

ABSTRACT
Computationally-enriched crafts are activities that blend the advantages of computational media with the affective, social, and cognitive affordances of children's crafts. In this paper, we describe a design application, *Popup Workshop*, whose purpose is to introduce children to the craft (and engineering discipline) of pop-up design in paper. We describe the fundamental ideas behind computational crafts in general, present our application in its current implementation and offer a scenario for its use, explore the particular ways in which pop-up design serves as fertile ground for integrating computation and tangible design, and discuss our early pilot-tests with elementary-school children, as well as ongoing and related work.

KEY WORDS
Educational software, Computationally-enriched crafts, Design software, Pop-ups

1. Introduction: Computationally-Enriched Children's Crafts

Popular debates about the merits–or failures–of educational computing often focus on a putative divide between the abstract, "virtual" world of the computer (and perhaps the World Wide Web, as well) and the "real" world of physical materials and activities for children. In our view, such arguments are based on premises that are, if not false, at least not inevitable. We have argued, in contrast, for the design of *computationally-enriched craft activities* [1, 2, 3]: activities that combine the affordances of computational media with those of children's educational crafts. Such computationally-enriched craft activities can take a variety of forms, but in many cases they may be realized through design applications that permit children to create and experiment with possible constructions on the computer screen, and subsequently to build those constructions in physical materials.

In this paper, we present and describe such an application: a system for the design of pop-up forms in paper, geared toward use by K-12 students. Briefly, the application, *Popup Workshop*, permits students to create a paper template from which a pop-up form will be created, and to see that form simulated in a three-dimensional rendering. Once the design has been created to the child's satisfaction, the template may be printed and cut to produce the predicted moving structure.

Before going into further detail about this particular application, it is worth pausing at the outset to discuss in a bit more detail the motivating ideas behind this work. Popup Workshop is in fact one of several related systems that have been developed (or are under development) in our research group: there are similar design applications for paper polyhedral forms and sculptures, mathematical figures in string, wooden and plastic models of complex surfaces in three-dimensional space, mechanical automata, and so forth. Each of these domains has its own particular strengths and weaknesses as an educational craft, but there are general principles to support the idea of a détente between children's crafts and educational computing.

What do computer design applications lend to the practice of children's crafts? (Or, to put it another way: why should advocates of traditionally "low-tech" educational craft activities be lured by the use of computational tools?) We believe that there are several responses to this question. First, children can design far more complex, expressive, aesthetically appealing, and educationally potent artifacts with the use of computational design tools than they could otherwise. Second, the use of a computational tool generally allows students to exploit the advantages of many other computational tools. For example, a child's designs might be decorated in a system such as Adobe Illustrator; or annotated with a word processor; or placed on a website for others to inspect and use; or sent by email to a friend. Finally, and perhaps most interestingly, creating novel computational design tools encourages us, as educators, to examine craft activities in a new light: it allows us to think of *notations* themselves as artifacts for experimentation. Many craft activities have existing *ad hoc* notations (the diagrams in origami books or notations for knitting patterns come to mind); but these are rarely the objects of consideration,

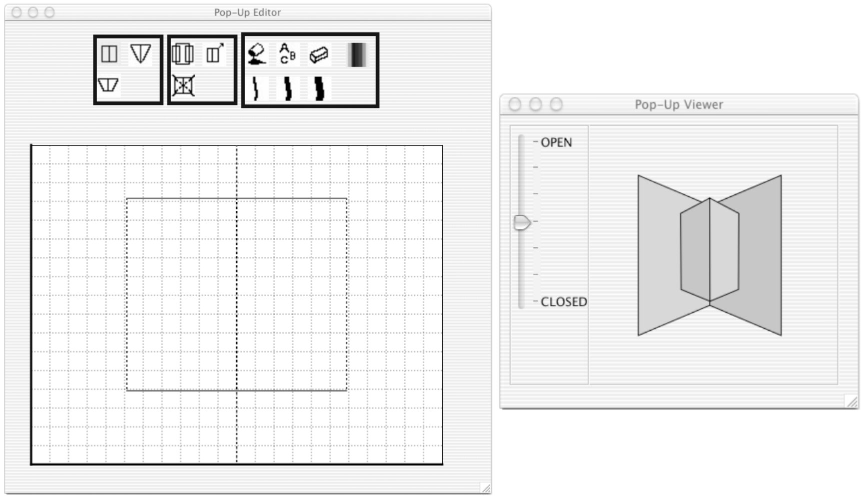

Figure 1. A screenshot of the Popup Workshop system: Placing a four-sided parallel element in the center of our pop-up construction.

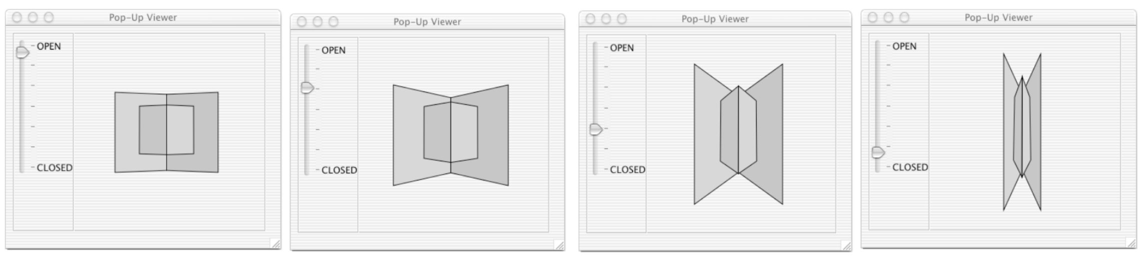

Figure 2 Using the slider in the Viewer window to open and close the 3D view of our construction.

exploration, and redesign in their own right. By representing craft designs in a design application, we find ourselves looking at the basic operations and techniques of the craft in a new light. (A longer argument along these lines can be found in [3].)

Turning to the converse question, we could ask: what do craft activities lend to the practice of educational computing? (Or, to put it another way: why should aficionados of “high-tech” educational innovations find themselves interested in the apparently unglamorous traditions of classroom crafts?) Again there are several elements to this argument. First, craft activities have a cognitive dimension–a dimension of tactile experience, working with materials–that is hard or awkward to reproduce in "pure software" applications. Materials such as paper, string, wire, wood, and plastic have essential characteristics (tensile strength, response to shear forces, “crumpling”, tearing, and so forth) that give tangible, viscerally understood form to fundamental ideas of engineering. From the intellectual standpoint, craft activities are (at their best) a rich source of mathematical or scientific ideas and images: paper polyhedra offer a tasteful introduction to solid geometry; string figures can illustrate surfaces in space; mechanical toys can reify the abstract physics of simple machines such as gears, cams, and levers. (Later in this paper, we will make a more extended argument along these lines on behalf of pop-up forms.) And beyond all this, there are *social* affordances of craft objects that seem, in some slightly mysterious way, hard to attain in the realm of pure software. Craft objects can be bestowed as gifts, kept as souvenirs, put on display, grouped into personal collections, cherished and re-examined for months or years; software artifacts rarely are the objects of this sort of affection or social utility. (For a useful discussion of the affective roles of physical objects in the lives of both children and adults, see [4].)

In the area of related work: there are numerous existing books on pop-up design. (Two particularly thorough and useful examples are from Carter [5] and Jackson [6].) There are also excellent examples of the use of pop-ups in the classroom [7, 8] and books specifically aimed at young students, such as [9] and [10]; and of course there are an endless number of books of pop-up creations for both children and adults. (One of our favorite recent examples is the refreshingly demented *Pop-Up Book of Phobias* [11].) Computer-aided design of pop-ups, and the mathematics of pop-up design, have been explored in Lee [12] and Glassner [13, 14]; though ours is the first program to our knowledge that is intended as a design application specifically for children’s work.

In the remainder of this paper, we focus this rather general discussion on the specific case of our Popup Workshop system. The following (second) section provides an overview of the software and a sample scenario. In the third section, we discuss the software from the standpoint of the paragraphs above, as an instance of computationally-enriched craft activity: that

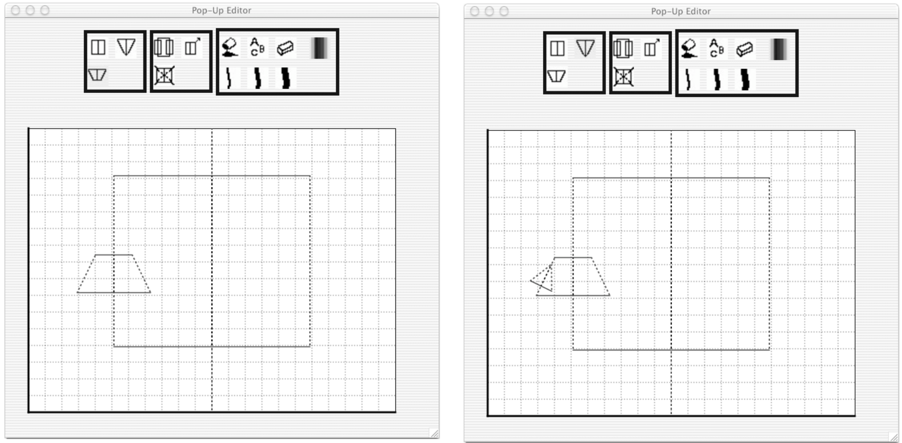

Figure3. Adding two new elements to the left side of our construction.

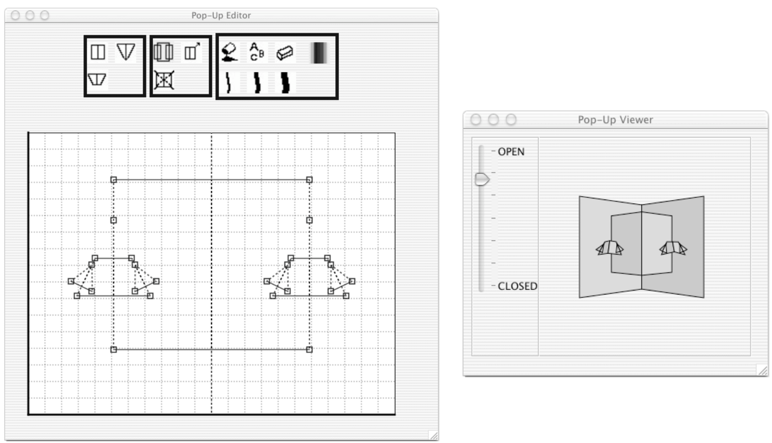

Figure 4. The Editor and Viewer windows, after duplicating the elements across the central fold.

is, we discuss the ways in which a computer design tool can expand the practice of pop-up design (and more generally, "paper engineering") and, in turn, how this particular craft domain can serve as a fertile source of design issues for educational computing. In the fourth section, we briefly describe our early efforts at pilot-testing the software with elementary-school children; and we conclude by outlining ongoing and potential future directions for work in this area.

2. Popup Workshop: An Overview of the Software, and a Scenario

Figure 1 shows a screen view of Popup Workshop in its current implementation. There are two main windows in the system: an Editor window in which pop-up forms may be designed, and a Viewer in which the current forms may be observed in a simple 3D rendering. The various buttons at the top of the Editor window are grouped according to function. The first group of buttons allow the addition of various forms of pop-up elements, the second group of buttons allow change, duplicate and remove operations, and the final group allow decoration of the elements.

Most of the Editor window (the bottom region) is employed to show the "flat" version of the pop-up under construction. The dark vertical centerline on this display indicates a *valley fold* (a fold away from the viewer), and this is indicated as well by the rendering shown in the Viewer window (the two side edges of the page are closer to the viewer than the center).

2.1 Constructing a Working Pop-up Form: a Scenario

To begin constructing a pop-up in the system, the user must first choose one of the pop-up elements in the left group of buttons. For instance, by choosing a four-sided parallel element, and placing that element over the center valley fold, the user creates the starting structure shown in Figure 1. The new element is shown as two parallel valley folds (on the left and right portion of the page, respectively) and a *mountain fold* (i.e., a fold toward the viewer) in the center of the page. The new state of the construction-under-development is updated immediately in the Viewer window, and what we now see is a folded

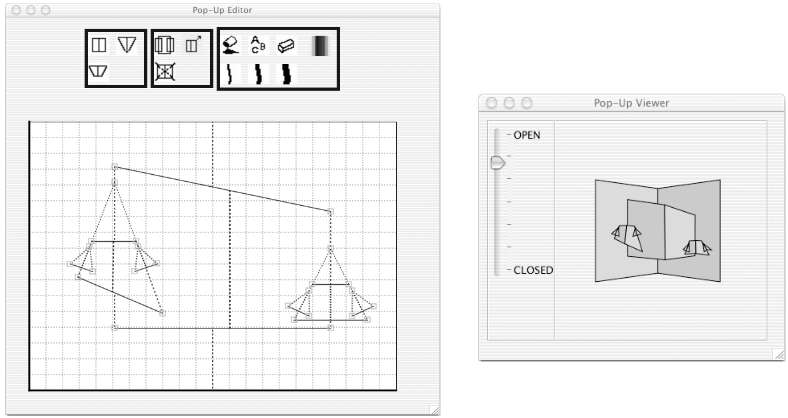

Figure 5. The Editor and Viewer Windows, after shifting the positions of a couple of points to create an asymmetric pop-up template.

page with a central region folding out toward us. By adjusting the slider at the left of the Viewer window, the user can observe the current pop-up form in various stages of opening and closing (Figure 2).

Now that we have placed one pop-up element on the page, there are several available valley folds on which new elements can be placed (the two left and right folds of our newly-placed element, and the upper and bottom portions of the original center valley fold). In Figure 3, we show one particular way to proceed: first, by placing a four-sided fold (with non-parallel edges) on the left side, and then adding to this fold still another one with only three sides. Finally, we use the "duplicate" button several times–first on the small triangular element (to duplicate it across its nearest fold on the non-parallel four-sided fold), then on the four-sided fold itself (to duplicate this structure across the center fold), and finally on the triangular element again; the result is the relatively complex structure shown (along with its 3D rendering in the Viewer window) in Figure 4. Finally, we can use the "change" operation to alter the elements that we have already made. For instance, in Figure 5 we see the result of making alterations to the positions of two individual points in the previous figure's construction, producing an asymmetric diagonal pop-up whose 3D view can again be seen (and manipulated into various stages of "openness") in the Viewer window. Note that altering points still preserves necessary geometric constraints in the overall construction: for example, although the upper right point of our original four-sided fold has been moved, the two major (vertical) folds on either side of the center remain parallel.

At this juncture, the construction may be decorated with the various tools provided in the set of buttons at the top right of the Editor window. For instance, by selecting the "color-change" button, the user can bring up a standard color-picker window; select a color; and fill a particular contiguous region of the pop-up (i.e., a region delimited by folds) with the desired color. Figure 6 shows several complex (undecorated) pop-ups that have been created using the system.

3. Pop-Up Design as a Computationally-Enriched Craft Activity

Although Popup Workshop is still at a relatively early stage of implementation (ongoing work on the system will be described in the following section), it serves as a particularly interesting instance of a computationally-enriched craft activity for children. In this section, we re-examine the arguments of the introductory paragraphs for the particular domain of pop-up design.

Employing a computational design tool in pop-up design has a number of distinct advantages over (e.g.) paper and pencil design, or trial-and-error experiments with paper. First, by using a tool such as Popup Workshop, it is relatively easy to experiment, quickly and informally, with a wide variety of design options, trying one type of fold after another; moreover, the design tool offers invaluable assistance in exploring and predicting the patterns of movement that a particular construction will undergo when realized. The upshot of this is that the "complexity barrier" for construction is markedly raised, particularly for children and beginners: that is, one may construct far more complex designs (cf. Figure 7) than would generally be expected in a classroom setting.

A second advantage of a computational design tool for pop-up design is that it permits the creative exploitation of a huge, and continually expanding, landscape of ancillary software tools and systems. For example, even though the decorative tools of our own system are relatively simple, it is easy to extend them via other programs: it is a relatively easy matter to import a design into paint programs, annotate (with text), send (by email), or post (to a website) a pop-up design created in Popup Workshop.

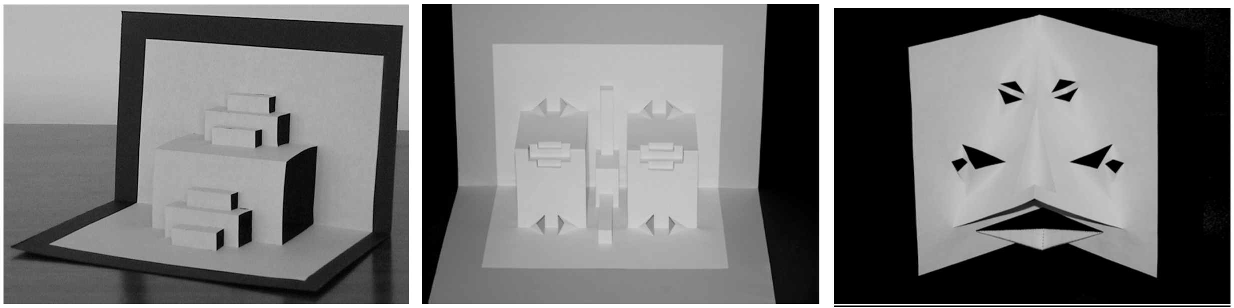

Figure 6. Several pop-up constructions designed with Popup Workshop.

Finally, the use of a computational design tool permits us to view the craft itself in a fresh aspect, exploring new operations for design and new notations for representing design. In our own system, this exploration is still at an early stage; but the use of the "duplicate" operation, for instance, suggests a way of replicating fold patterns that (to our knowledge) is not encountered explicitly in existing pop-up design books. Duplicating pop-elements across folds is an easy operation for a computer program–indeed, it is almost irresistibly suggested in a medium such as our system; but it is a much more tedious (and hence unexplored) technique in paper-and-pencil design. Moreover, the "change" operation in our system is likewise facilitated by the presence of the computer: changing a particular element is facilitated by the system's ability to maintain necessary folding constraints among all the affected elements of the construction.

The previous paragraphs described several important ways in which the use of a computational tool could expand children's practice of the craft of pop-up design. In turn, there are several avenues along which a craft domain such as this one can enrich the practice of educational computing.

Although it may have something of a reputation for lightheartedness (even frivolity), the creation of pop-ups in paper is in fact a task of engineering–indeed, it is arguably one of the most creative (and certainly the most inexpensive) engineering exercise that elementary-school-age students are likely to encounter. Moreover, even though this craft lends itself to simple constructions, it is also extensible to the point of extreme adult-level professional mastery. (A glance at the prizewinning entries in pop-up design in the annual Dimensional Illustration competitions–see, e.g., [15]–will provide breathtaking confirmation of this.) It is not accidental that many professional pop-up designers refer to themselves as "paper engineers" [5], since the essence of the craft is to create dynamic, structurally robust three-dimensional forms in the medium of paper.

Even the elementary techniques supported by Popup Workshop provide a real-world introduction to notions such as recursive patterns in design (see, for instance, the first and third examples in Figure 5) maintaining essential structural constraints (such as parallel folds) within design elements, and predicting kinetic behavior of structures. Crucially, and consistent with our earlier discussion of the value of craft activities, these engineering and design ideas are materially embodied in the tangible medium of paper. It is odd to note (though intuitively apparent) that there is little interest in a pop-up design that is purely realized in computer graphics; the value of the artifact only truly comes alive when the object is held, opened, and closed in one's hands. A "virtual pop-up" design tool would have little appeal, and most likely would lack many of the physical affordances and constraints of paper—its surprising tensile strength (but simultaneous susceptibility to tearing), its longevity, and so forth.

There is also a social utility for crafts such as pop-ups that, again, seems to escape artifacts in the "purely virtual" worlds of computer software. Pop-ups may be retained, displayed, and examined over a period of months or even years; they can be used as purely intellectual illustrations of types of motions and mechanical linkages, as elements of mathematical or scientific illustrations, or (most often) purely as objects of delight. Not infrequently they may be employed as gifts or tokens of affection (again, in stark contrast to most "pure software", or screen-based, artifacts). In other words, these craft objects can be woven into children's (and adults') lives and day-to-day activities in a manner that makes of engineering something playful and human.

Finally, it is perhaps worth mentioning an observation from the software designers' standpoint (i.e., our own), rather than the user's: namely, that implementing a computer system for pop-up design serves as a rich source of interesting, and often challenging, algorithmic and interface-design research issues. The problem of representing a flexible pop-up structure in three dimensions (as in the Viewer window) is not at all trivial, for instance. Indeed, our own algorithm for that purpose employs a greedy search strategy to move three-dimensional points into their correct positions when the Viewer-window slider is shifted; and that algorithm is both somewhat slow (the slider must generally be moved fairly gently and incrementally), and occasionally subject to error (so that a patently wrong three-dimensional rendering suddenly appears on the screen). Improving and refining this algorithm is one of the many tasks to be undertaken in our continuing work on the program, other aspects of which are discussed in the final section of this paper.

4. Conclusion: Ongoing and Future Work

Although Popup Workshop is operational in its current state, it is still at a relatively early stage of continuing development. There are numerous avenues, along which we plan to extend, refine, and improve the current version of the program. Some of these planned improvements are a matter of adding functionality: We are currently adding to the (still-minimal) basic set of elements already present in the program. These new elements are produced by adding additional pieces of paper to the original page, and result in the creation of additional editor windows. (Numerous ideas along these lines may be found in design books such as [6].) Improvements to the Viewer window, the choice of various sheet sizes, and additional decorative options are planned. Freely available and downloadable over the World Wide Web [16], the system has been implemented in the Java language and is available for both the Mac and Windows.

We have recently completed our first pilot-tests of the system with fifth-grade children in a local elementary school. Although this work with children is early and informal, it has been encouraging; that is, the program seems to be both understandable and usable for students of that age. Moreover, it appears that the hand-eye coordination required of children to produce working pop-ups from a printed template is not beyond their capabilities. Figure 7 shows a pop-up card of a smiling face designed, decorated, and constructed by a fifth-grade girl.

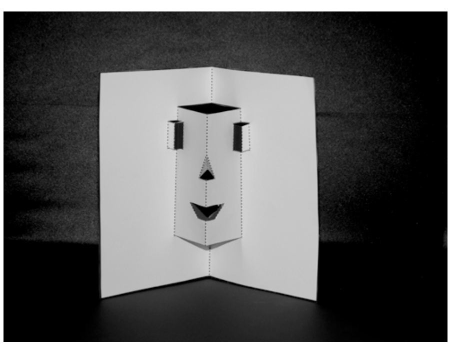

Figure 7. A pop-up of a face, designed and constructed by a fifth-grade student.

In the (slightly) longer term, we hope to augment our computational work with more straightforwardly "curricular" elements, addressing and emphasizing the "paper engineering" side of the craft, and encouraging children to think about both the structure and dynamic behavior of their creations. In both the shorter and longer term, we envision no lack of work in extending or assessing our system, and in exploring the educational role of this elegant, child-friendly branch of engineering.

5. Acknowledgements

The work described in this paper has been supported in part by National Science Foundation grant REC0125363, and by a gift from Mitsubishi Electric Research Laboratories (MERL) in Cambridge, Massachusetts.

References:

[1] M. Eisenberg, et al. As We May Print: New Directions in Output Devices and Computational Crafts for Children, *Proceedings of Interaction Design and Children 2003*, Preston, UK, 2003, 31-39.

[2] M. Eisenberg, Output Devices, Computation, and the Future of Mathematical Crafts, *International Journal of Computers for Mathematical Learning*, 7(1), 2002, 1-44

[3] M. Eisenberg, A. Rubin, and T. Chen, Computation and Educational Handicrafts: a Strategy for Integrative Design, *Proceedings of the International Conference on the Learning Sciences, 1998*, Atlanta, 1998, 84-90.

[4] M. Csikszentmihalyi, and E. Rochberg-Halton, (1981). *The Meaning of Things (*Cambridge, UK: Cambridge University Press, 1981).

[5] D. Carter and J. Diaz, *The Elements of Pop-Up : A Pop-Up Book for Aspiring Paper Engineers* (New York: Little Simon, 1999).

[6] P. Jackson and P. Forrester, *The Pop-Up Book: Step-By-Step Instructions for Creating over 100 Original Paper Project*s (New York: Henry Holt and Co, 1996).

[7] E. Simmt. and B. Davis. Fractal Cards: A Space for Exploration in Geometry and Discrete Mathematics. *Mathematics Teacher*. 91, 1998, 102-108.

[8] P. Johnson, *Pop-Up Paper Engineering: Cross-Curricular Activities in Design Engineering Technology, English and Art* (London: RoutledgeFalmer, 1992).

[9] D. Birmingham, *Pop Up! : A Manual of Paper Mechanisms* (Norfolk, UK: Tarquin Publications, 2000).

[10] G. Diehn, *Making Books that Fly, Fold, Wrap, Hide, Pop Up, Twist, and Turn* (New York: Lark Books, 1998).

[11] G. Greenberg, *The Pop-Up Book of Phobias* (New York: Harper Entertainment, 1999).

[12] Y.T. Lee, S.B. Tor, and E.L. Soo, Mathematical Modelling and Simulation of Pop-up Books, *Computers and Graphics*, 20(1), 1996, 21-31.

[13] A. Glassner, Interactive Pop-up Card Design, Part 1. *IEEE Computer Graphics and Applications*, 22(1) 2002, 79-86.

[14] A. Glassner, Interactive Pop-up Card Design, Part 2. *IEEE Computer Graphics and Applications*, 22(2), 2002, 74-85.

[15] Dimensional Illustrators, Inc., *3-Dimensional Illustration Award Annual 5* (Rockport, MA: Rockport Publishers, 1995).

[16] S. Hendrix, Popup Workshop, Version 1.1, l3d.cs.colorado.edu/~ctg/projects/popups/index.html.