

Popup Workshop: Supporting and Observing Children's Pop-up Design

Ph.D. Dissertation Proposal

Susan L. Hendrix
Department of Computer Science
University of Colorado at Boulder

14 April 2004

1 Introduction: Supporting children's design

Children learn within a physical environment, play with physical objects, own physical objects, and enjoy making objects to trade, as gifts, and as treasures to be kept as they grow up. Their creations adorn our refrigerators and hold our paper clips. Some of these objects are made by following directions or patterns. But the most striking are often designed by children themselves within a setting that is presented to them, by handing them crayons and a blank piece of paper rather than a pre-printed picture to color, for instance.

By designing their own objects, children learn both general principles of design and details about the domain in which the design takes place. However, design is a complex issue, difficult to teach and difficult to analyse, and "Any attempt to describe something as complex as how children or adults design is fraught with difficulties. Not enough is known about an activity which is dependent on both the characteristics of the designer and the context within which designing takes place" [21, page 32]

Children could use some help in designing and making physical objects, as often tools are made for adults, there are few instructional materials, and those instructional materials may not be well-organized. Small fingers have trouble holding tools or drawing straight lines, and although simple objects may be easy to make in a given medium, more complex designs may take too long to get right. Children can benefit from computer software which helps them over some of these difficulties, but which does not take away the fun and surprise of making something one can hold in the hand and play with directly.

However, here we meet some unknowns in designing computer software to support children's design activities. We know too little about which tools might be appropriate to help children design physical objects. For example, we need to help them over the problem areas, but not take away too much of the value of the design activity. By taking away this effort, do we also take away the learning process? By transferring some of the work from physical object to screen, do we destroy too much of the value of making a physical object?

There are known difficulties with educational software in general. We need to be careful about the messages we send "...about teaching youngsters that exploring what's on a two-dimensional screen is more important than playing with real objects..."[27].

Additionally, much work needs to be done with respect to children's software interfaces. We know something of what works, especially in the area of web page design. Certainly children have been found to be quite forthcoming in letting us know when we ask, but there have also been found to be tricks in the asking.

In summary, we need more information about children's design and how best to facilitate it with software tools.

2 Approach: A design domain for exploration and study

Design needs to be learned by doing; in order to do design, there must be some product. Every form of design has some elements in common with others. One possible way to look at the whole area of children's design is to find some product that children will want to design, will enjoy designing, and will teach them something in the process. We can then build a tool to aid children in designing in this domain and study how the children use the tool, in order to try to find out which affordances are useful, which are not useful (or even harmful), and how children go about designing physical objects using computer tools.

Pop-up books and cards are not only mathematically interesting and challenging, but often amazing works of art. They can serve as these physical objects.



Figure 1: Two pop-up conceptions of small creatures: *Arachnophobia* [16] and *Butterflys* [26]

2.1 Pop-ups are interesting for children

Pop-ups are obviously enjoyed by children (and adults, I have noticed). Pop-up books sell well, in spite of their expense and fragility. They range from simple but well-designed books for young children [26] to amazing artistic creations [2]. However, few children try to make pop-ups on their own. There are several good books which give an introduction to paper engineering for children [10, 34], but it does not seem to be an activity that is common. Nor is this an activity common among adults. Robert Sabuda, one of the best-known paper engineers, has remarked "There may be 36 paper engineers on the planet. It is a small field. You don't see so many great pop-up books" [17].

Although few children make pop-ups, pop-ups can be very simple to design and produce. Figure 2 illustrates that pop-ups do not have to be complex to be amusing and creative. It is possible that the reason few children

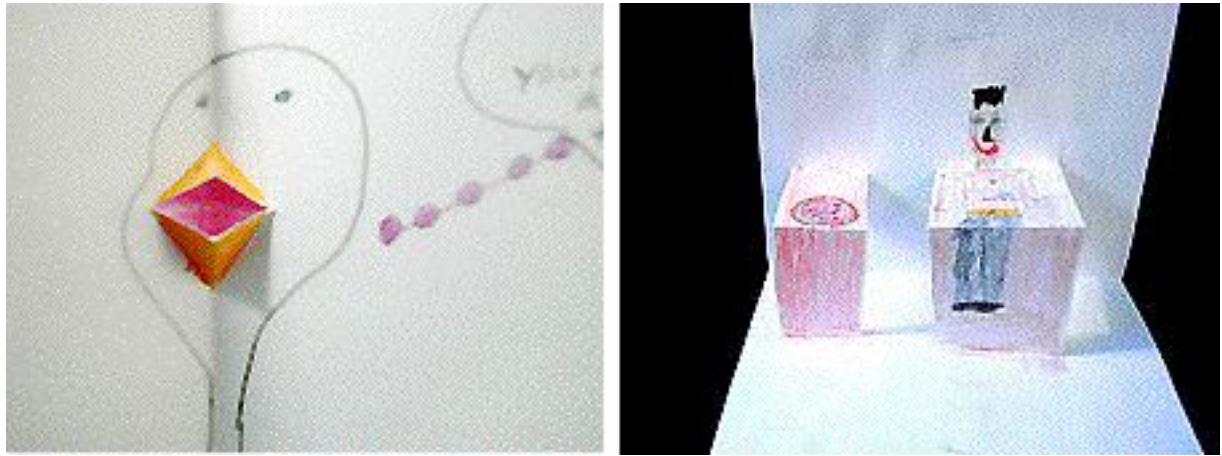


Figure 2: Two pop-ups made by children: *Yellow Beak* and *Pizza Please!* from [30]

make pop-ups is that they are never told that it is possible; pop-up books are seen as something that one buys, not makes.

2.2 Pop-ups are instructional

An activity that involves such an interesting object should be usable in a classroom context as well.

Pop-ups are useful in introducing mathematical concepts; Simmt[31] has used geometric single-sheet pop-ups in the classroom to introduce concepts such as growth patterns, limits, iteration and fractal dimensions. She describes this as a variable entry activity that “offers a rich space for mathematical exploration and discussion about the nature of mathematics” [31, page 108] Another variation on this approach, not for the classroom but for teens or adults, is Uribe’s book [33] which illustrates fractals, iteration and self-similarity using pop-up cards which may be cut out and constructed by the reader.

Pop-up books may also contribute to learning in art and writing. Many projects for the classroom using pop-ups may be found in Johnson[22]. In fact, one possible use of the software discussed in this proposal would be to allow an entire class to make a book, with each child designing a page. Because the pop-ups designed on a computer can be printed in quantity, each child could have a complete copy of the result.

It can be see from these examples that making pop-ups is instructive in several disciplines. In constructing a pop-up, art blends with mathematics. In constructing a book, writing blends with art.

2.3 Pop-ups offer a useful domain for studying design

A great deal of work has been done studying children and design both using software as the design space and designing physical objects directly. However, I have been unable to find any studies as yet of children using CAD systems (which this software is). I believe that pop-ups provide a good vehicle for this study for many reasons.

First, paper is a good material to work with. Paper is strong, common, and cheap. Children are familiar with its properties. They are also familiar with and fond of pop-ups. Therefore, they will be using a familiar material to make something they will enjoy.

Also, the tool itself can be used to examine how it is used. This is not a new idea. Embedding design history in design software has been done [28], and the redo feature (with or without a playback capability) is part of many software packages. But I have not yet found studies of children’s use of software using this method.

Third, the domain is varied, and yet constrained. There are a limited number of elements, simple combinations of folds, cuts, and attached pieces, that may be combined, and yet, as any glance at actual pop-up books indicates, there is a great deal of space for innovation.

2.4 There is a space for a tool to help design pop-ups

At present, pop-up books are designed by hand, by both adults and children. But there are good reasons for building a tool to help children in this process. The process of constructing pop-ups can be quite frustrating. Often mechanisms must be removed and remade, fiddled with and refined, in order to make them open and close smoothly and properly. Also, the constraints involved in many mechanisms are hard to learn. As we will see they involve parallel lines, equal angles, and other measurements that are difficult for children to make. Also some mechanisms must be quite smooth, straight, and regular or they will bend or seize.

Having a pattern to follow helps, but patterns produce standardization. If children can combine many mechanisms together in a variety of ways that will function correctly, the process of creating a complex pop-up can be eased.

It is important to note that this is design of physical objects. A pop-up design program would be a failure if the user did nothing but stare at a screen. These objects beg to be made, read, manipulated, and played with.

3 Thesis question

This proposal is guided by the following question:

Can a computer-aided design system using constraint satisfaction algorithms, which present pop-up books and cards as dynamic objects, be created that will enable children to design pop-ups and that will add to our knowledge of the process of design and the features of software which support children's design?

4 What I propose to do

Popup Workshop is an environment which I propose to develop and to use for observation of the design process used by children. The domain of Popup Workshop is the pop-up book or card. These range from the very simple, in which a single slit and three folds combine to make a bird's beak (as an example) to the complex and difficult constructions seen in children's and adult's pop-up books.

4.1 Build an environment for creating pop-ups

The system description of the proposed tool is discussed in section 7 of this document. A prototype of Popup Workshop has been implemented and used by a 5th grade child.

The prototype uses a small subset of possible pop-up elements—straight cuts and folds directly on the base page, as opposed to elements cut from separate pieces of paper and attached to the base page.

There are three main goals of this part of the project.

- To create a computer-aided design system (for this pop-up domain) that is usable by children to create pop-ups

- An animated display, as a part of that system, of the pop-ups opening and closing, and using constraint systems to accomplish this.
- To build into the tool a recording facility, which will be useful in examining how the tool is used by children.

The first two items above have been partially explored in the prototype, as it has been used by one child to create a pop-up, and the animation is working using constraints, but with a fairly crude hill-climbing algorithm. Also, only a small subset of pop-up elements have been implemented. The third item has not yet been started.

4.2 Study children's design of pop-ups and the use of the tool

This research, however, involves more than the production of a tool for pop-up design. This proposal includes using this tool in case studies to observe how children will use it to design (in part by building into the environment the ability to record their actions), how the pop-ups they make change as they move from novice to seasoned user, and the resulting changes in their vocabulary in the process.

It is hoped that these case studies will add to our knowledge of how children learn design, in particular the design of physical objects, and which affordances of computer-aided design tools are useful to them and which are not.

5 The domain of pop-up cards and books

The making of pop-ups is a subset of the domain of paper engineering. Moveable books (the more general term) have been in existence since at least the 1300's, and began in three simple forms, gatefolds, flaps and wheels. Gatefolds consist of pages cut larger than the book containing them, and folded back to fit the book. This allows the size of the page to be extended. The Playboy centerfold is a good example of this form of moveable book. Flaps allow the reader to lift an attached piece of paper, exposing the illustration below it. Wheels are (usually) circular pieces attached so as to rotate; multiple wheels may be stacked. Both gatefolds and flaps were commonly used in anatomical books, and wheels in astronomical or astrological texts.

Moveable books were first made for children in the 19th century. The delay is not surprising, as books expressly for children are a modern innovation. Those early books are called harlequinades, turn-up books or metamorphoses. The reader moves flaps to change the picture. Other innovations followed such as slot books, which allow cut-out figures to be placed in slots in an illustration, panoramas, which carry the gatefold concept to extremes to allow the reader to fold out a long scene, and the peep show, in which a ribbon is pulled to raise additional layers of the scene [18].

However, the first true pop-up book (with pieces that stood up automatically when a page was opened 180 degrees) was not made until 1929 [14] and the term 'pop-up' was first used in 1932 [18].

Pop-up books today may combine not only pop-up elements, but gatefolds, tabs, wheels and lift-up flaps. I will use the term 'pop-up' to include all of these elements, however, my main focus is on true pop-up elements. I see no reason that the tool cannot be extended to cover flaps, pull tabs and similar constructions, but that is beyond the scope of this research project.

Pop-up books are still commonly designed and assembled by hand; the actual mechanisms are hand-made by experienced paper engineers. This process "can take several iterations, spanning many weeks per book"[23, page 21]. The pieces may be put into a graphics program to be sent to be die-cut in large quantities, but the final assembly is also done by hand, by skilled assemblers, often whole villages such as Ibarra in Ecuador [29].

An excellent illustrated guide to the process of design and fabrication of pop-up books may be found in Carter and Diaz[6].

This particular tool is envisioned as a tool for children to use in creating their own books. However, it is not out of the question that this or some similar tool could be useful to pop-up book designers, or adults interested in crafts.

Several books describe in detail the elements which make up the typical pop-up and methods for aspiring paper engineers. Carter and Diaz [6] is very useful as it contains actual pop-ups to illustrate many of the elements. Jackson [20] is very well-illustrated, with excellent pictures of many unusual pop-ups. A useful introduction can also be found in Hiner [19], as this has patterns to cut out and assemble, and contains a wide variety of mechanisms, including pull tabs and rotating arms. I am also extremely fond of a book by Birmingham [3] that is actually written for older children. It contains a large variety of mechanisms and many useful construction hints.

5.1 Pop-up basics

Let us put aside for the moment wheels, tabs and so on, and discuss pop-ups proper, in particular those types supported in the prototype tool. In this process, I will use terms from various sources, or define my own terms. Terms for pop-up elements are not well established. In most cases, I am selecting one particular term which is either often used, or which I think is particularly descriptive. I have included a glossary in Appendix B.

Pop-up elements, structures or forms can be divided into applied and single-sheet varieties. That is, there is a base page that one uses to make the pop-up. Applied elements are made from separate sheets and fastened onto the base page (or other elements). They are sometimes called 180 degree forms, since they commonly display best when the page is fully open.

Single-sheet elements are formed by cutting and folding from that base page or another element, without employing additional paper. They can be called 90 degree elements, since they look better when the page is only opened halfway.

Elements of both types may be combined in a single pop-up. In general, single-sheet forms need more advance planning, since they are formed by cuts on the base, and cannot be changed once produced. Applied forms, on the other hand, can be taken off and changed.

The prototype of this tool uses only single-sheet elements. Applied elements of at least one type will be added before the evaluation process. There are fewer single sheet elements; this seemed to be a reasonable simplification for the prototype.

There are three basic types of single-sheet elements. All three are placed on a fold which I will call the parent fold.

- Parallel double slit (figure 3)—In this form, which forms a quadrilateral, there are two cuts placed across the parent fold. The cuts do not have to be parallel to either the fold or each other. There are two folds which form the opposite sides of the four-sided figure. Both of these folds must be parallel to the parent fold. There is also a center fold on the element. The distance from one of the outer folds to the center fold must be the same as the distance from the other outer fold to the parent fold.
- Single slit (figure 4)—This element consists of a central cut across the parent fold. This does not have to be at any particular angle to the parent fold, and the distances of the ends of the cut from the parent fold may be different. There are two folds from the ends of the cut, which must meet on the parent fold. In this case, the angle between the parent fold and one of the outer folds must be the same as the angle between the center fold and the other outer fold.

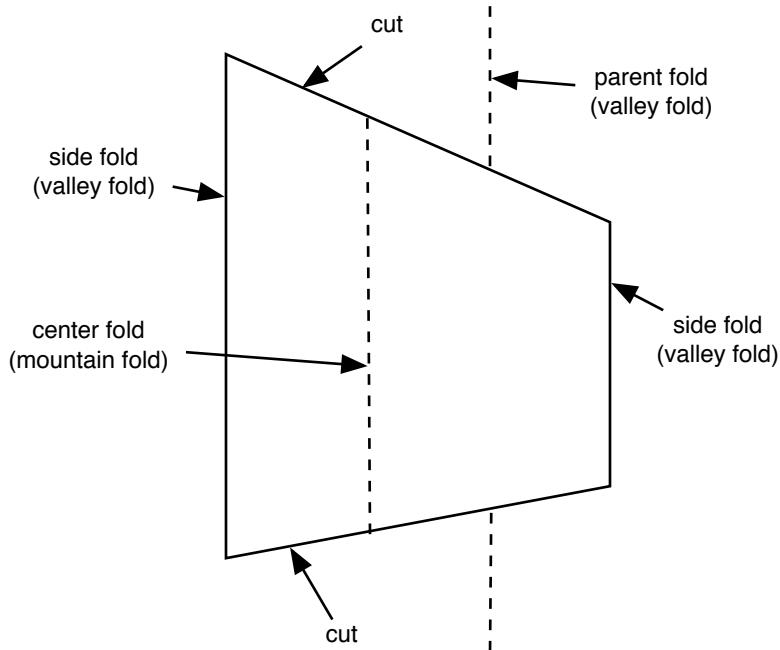


Figure 3: Parallel double slit

- Non-parallel double slit (figure 5)—This structure is a variation of the single slit element. Take a single slit and add a cut which takes off the tip. That cut may be at any angle to the parent fold. This turns the triangle of the single slit into a quadrilateral. All the constraints of the single slit must be preserved.

In all these cases, the cuts need not be straight, although for the purposes of the prototype, this limitation has been imposed. Folds in paper, on the other hand, are naturally straight. (It is also possible to obtain pop-up forms which rely on bending paper. This is beyond the scope of this research.)

A fold which pushes the paper toward the viewer is called a mountain fold. Its opposite is a valley fold. In each of the forms described above, it is possible to base the element on a mountain fold as opposed to a valley fold. The structure looks similar, and has the same constraints to open and close, but folds in a different direction. The side folds become mountain folds, and the center fold becomes a valley fold. I have chosen to call this variation an inverted structure. This gives us six possible single-sheet elements to work with, each of the three forms illustrated, and their three complimentary structures.

Applied elements of various types have other constraints for clean opening and closing, and each is different. This is an important point, and I shall return to it in section 6.1.

Perhaps the simplest applied pop-up element is the v-fold. These elements are actually quite similar to single-sheet pop-ups, being extra pieces attached to each side of the page, with a central fold; they are often called tents, which they resemble. They are quite commonly used in commercial pop-up books. Other commonly used elements are moving arms, which are pieces attached to folds that rotate and move when the page is opened and can create quite dramatic effects, and parallel planes, which are pieces not placed over folds, but attached to those that are on folds with small braces to transfer movement. At the extreme end we find pop-up cubes, pyramids, and even cuboctahedrons [35].

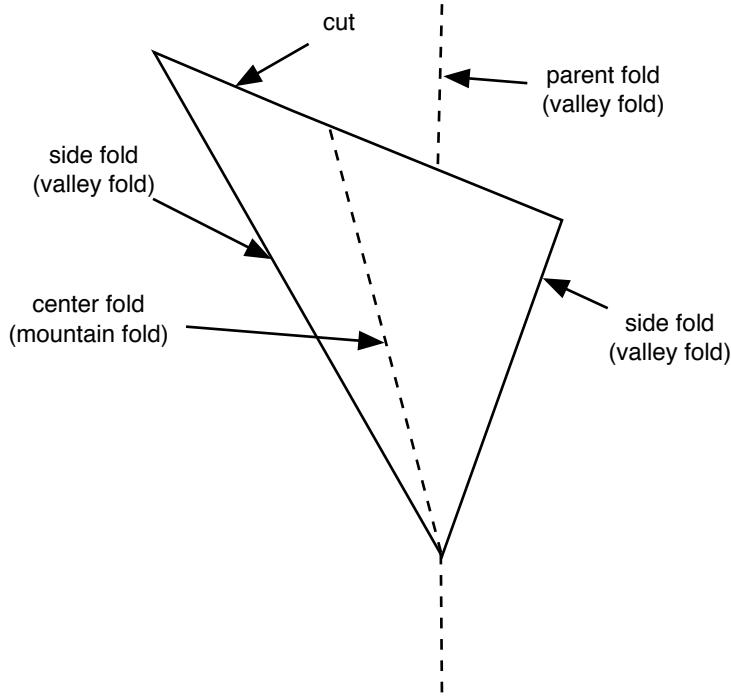


Figure 4: Single slit

Constraints play a role in pop-ups in two ways. A given element is constrained in the ways it may be constructed in order to lie flat within a closed page, and open smoothly when the page is opened. Besides constraining the pop-up elements so that they open and close properly, my work uses constraints in modeling and animating the opening of the page.

6 Related work

6.1 Pop-up Animation

This problem of pop-up animation has been approached previously in two lines of research with pop-ups.

Lee, et.al.[23] have made a mathematical treatment of v-folds. The authors developed the trigonometric equations to model these types of pop-ups and include drawings, produced with a professional 3D graphics package. It is indicated that they intend to produce a complete pop-up design system, but I have been unable to locate any publications after the 1996 paper.

A more varied set of elements are described by Glassner [14, 15, 12], who developed formulas for the animation of single-sheet elements as well as v-folds¹. In his publications, he states that he has a complete design system, but no details are given, other than that elements are added by drag-and-drop.

¹Glassner also has a patent[13] for a pop-up design system (and for the pop-ups as well!). There are no more details than in his other publications cited.

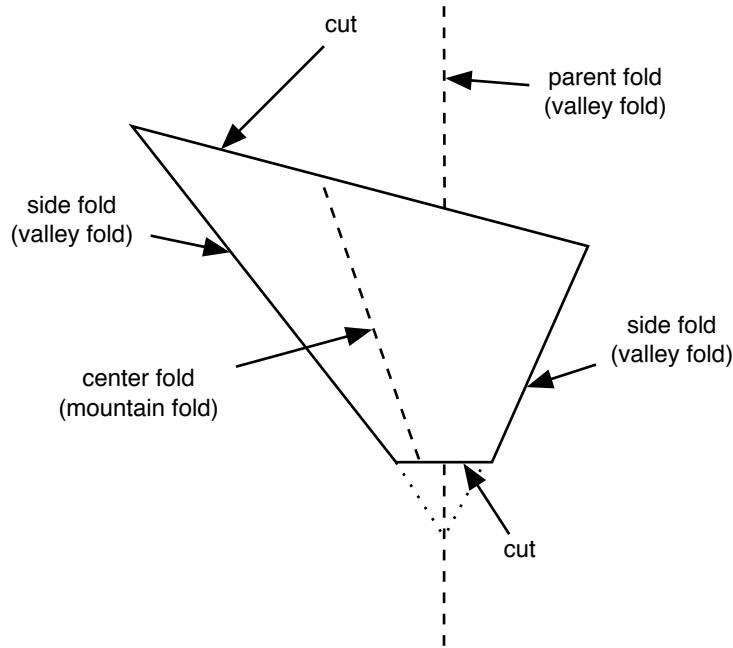


Figure 5: Non-parallel double slit

6.2 Related design tools

This work took its inspiration from other work of the Craft Technologies Group, most notably Hypergami[11], Machine Shop[4], and Hyperspider[9].

These three projects are all driven by the desire to design software to help children in the production of physical objects, although Hyperspider was not particularly designed for children, and I have continued in this tradition. Machine Shop is a computer aided design system for middle school children to design mechanical assemblies. Hypergami relates most closely to this work in that its domain is paper. However, Hypergami is a system to design paper sculpture using geometric solids, and in Hypergami, the design process is not done directly on the pattern.

6.3 3D Card Maker

3D Card Maker is a shareware program for designing pop-up cards, and is available on the web for download[25].

Figure 6 shows a sample of the pop-ups produced by 3D Card Maker and the main design window of the program. This does include an animation feature, as well as the ability to rotate the design in three dimensions and look at it from different angles. The pop-ups produced are combinations of the single-sheet parallel double slit only. But a look at what the program can make shows an impressive number of things that can be made with that element.

This particular style of pop-up, using parallel double slits to produce primarily architectural designs was pioneered by Masahiro Chatani[7, 8].

There are several ways in which the work proposed differs from 3D Card Maker. First, in 3D Card Maker the final card form is designed, and the program produces a pattern from that design. Although this is a compelling

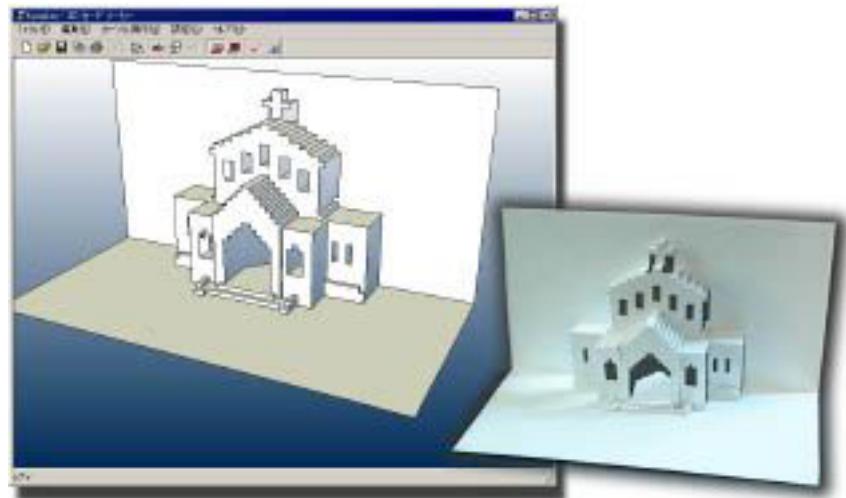


Figure 6: Sample Window and finished pop-up from 3D Card Maker

approach, especially given the limited subset of elements, there is no evidence to suggest that it is superior to designing the pattern directly. It seems to me that working on the pattern mimics the natural process of designing pop-ups more closely, and that the natural surprise of watching the effect of a change in the pattern on the opening of the resulting form is worth preserving. Second, the 3D Card Maker approach does not seem to scale well; it does not appear to be easily expandable to other pop-up forms. And third, it was not designed specifically for children, although children have used and liked it.

7 System description

Pop-ups are 3-dimensional objects constructed with a 2-dimensional material, paper. Moreover they are not static, but mutable and interactive; the reader opens them, pulls tabs, lifts flaps and changes the book over time as she manipulates it [1]. These aspects of pop-up books—2D material and 3d result, interactivity and temporality—should be preserved in making a design tool to create them.

In order to capture these aspects of space in designing with paper, and to introduce the elements of interactivity and time, the prototype version of Popup Workshop uses two windows. The Pop-up Editor window shows the flat starting page with the cut and fold marks, and the Pop-up Viewer displays the 3-dimensional result of those cuts and folds and the animation of that result produced by opening and closing the page, which is done by the user.

The Popup Workshop windows in their startup configuration are shown in figure 7. The editor window (the larger window on the left) shows the base page for the pop-up. This is the window in which the user makes changes to the pop-up. The viewer window (the smaller window on the right) shows the opened pop-up as it will appear in 3D. The user is able to open and close the pop-up and see its motion via a slider, but is not able to otherwise change the design from this window. A summary of a sample interaction with the prototype tool is provided in Appendix A.

The prototype software is written in Java under Mac OSX. Java2D was used, but not Java3D, as this was not available at the time the prototype was being written. No effort, beyond the choice of Java, has as yet been made to make the application portable to other platforms; this should be fairly straightforward, although testing will

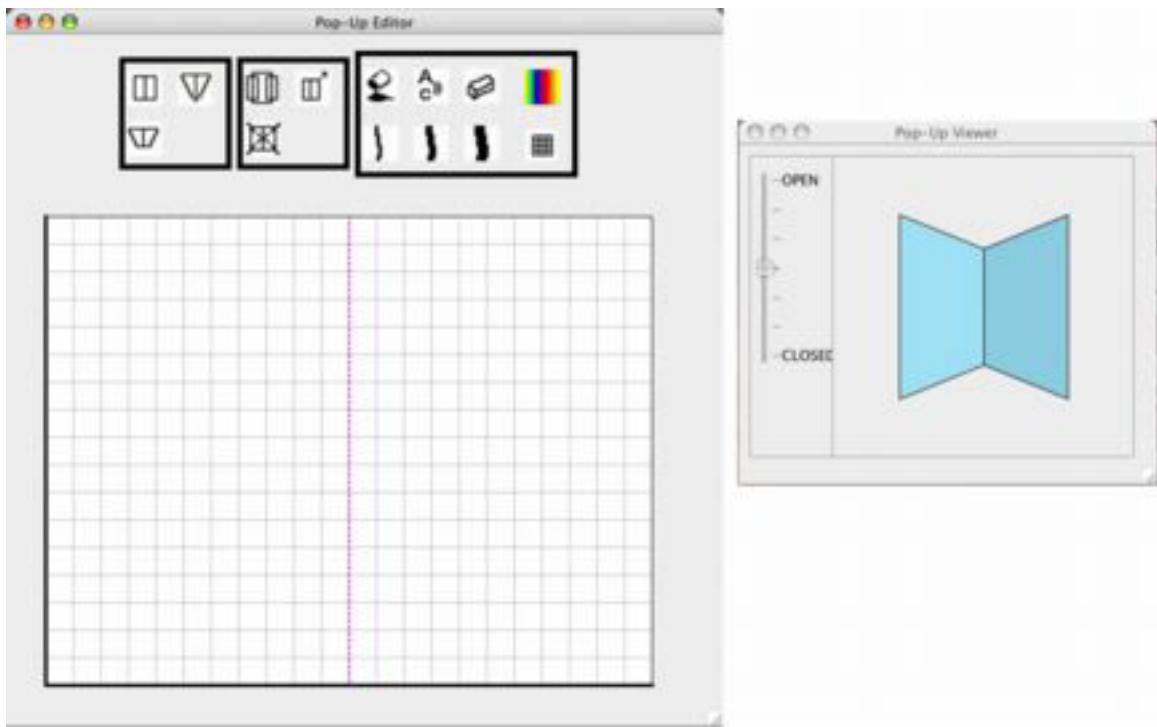


Figure 7: Popup Workshop starting windows. the Popup Editor is on the left, and the Popup Viewer is on the right.

need to be done on multiple platforms, and is planned. It is also planned to make this software open source, and to make certain that it is available on the web.

Let us look at the major components of the software as they exist in the prototype, and as they are envisioned in the final dissertation version of the tool. These major divisions are the Pop-up Editor, Pop-up Viewer, and the instrumentation needed to record the use of the tool.

7.1 Pop-up Editor

The Pop-up Editor window is the pattern for the page. It also serves as the place where the user may add, delete and change the pop-up elements and add decorations or text.

The software could have been designed so that the user works on the final view of the product (in the viewer). Two similar applications take this approach—3D Card Maker [25] and HyperGami [11]. It was decided that working on the flat page would more closely mimic the hand production of a pop-up, and might also preserve some of the surprise that pop-ups produce.

Currently, in order to produce a physical pop-up, the user has to use a screencapture tool, print the resulting graphic, cut on the cut lines and fold on the fold lines. Figure 8 shows three pop-ups made with the prototype.

7.1.1 What does the prototype editor look like?

At the top of the editor is a control panel containing three sets of buttons which are used to choose the current operation. The buttons are separated into three groups: adds, changes, and decoration options. There are three

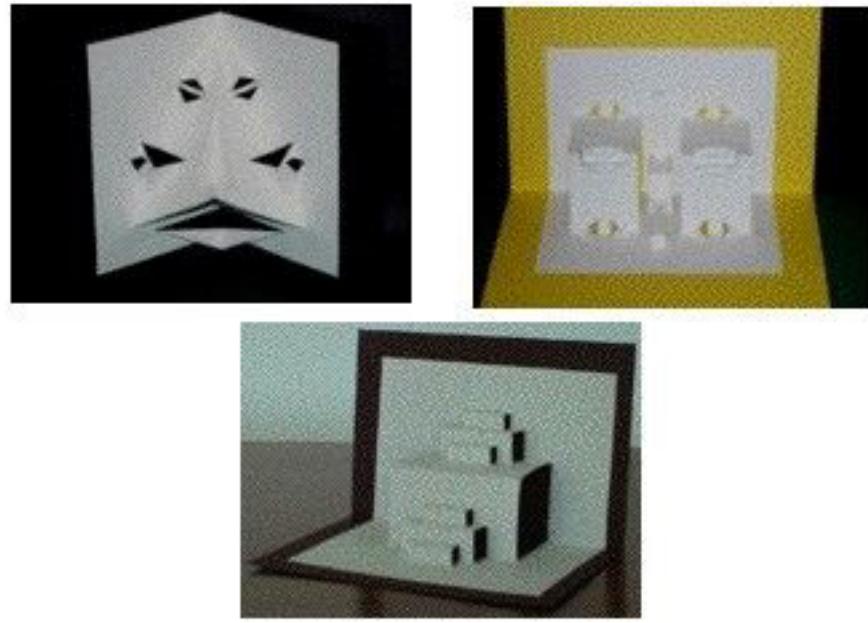


Figure 8: Three pop-ups produced with PopupWorkshop

add buttons at present, representing the three types of elements which are currently supported. The second group of buttons invoke change, delete and replicate. In the third group are decorating options such as fill and various sizes of pens for drawing or painting. Also in this group is a button to turn the grid lines off and on (to hide them for printing), and a button to bring up a color chooser for the pens and fill. The groups are surrounded by borders which takes on the current fill color.

There is a second area in the editor which shows the current page. I have chosen to use solid red lines for slits, and dashed lines for folds (colored magenta for valley folds and blue for mountain folds). Dashed lines for folds and solid lines for cuts are frequently used in many types of patterns, and might be a metaphor already familiar to the user. For operations on the shapes of the structures, or for removing or mirroring structures, small squares appear on applicable corners, and the user clicks or grabs and moves them. This preserves consistency in the operations.

The series of figures from figure 9 to figure 12 shows the creation of a simple pop-up. This shows a series of adds, a replicate operation to make the pop-up symmetric over the page, and a change to introduce asymmetry into the structures.

One should note that change allows only changes which result in a foldable structure, and that change is a recursive operation. In other words, if a change is made which alters folds in a structure on which other structures sit, those structures must be moved as well. This can be seen in operation in figure 12 where the user has moved the corners of the large center element, and caused the single slit elements on its side folds to move as well, before changing the smaller elements.

Not shown in this series is the deleting of a structure. Once again, this is a recursive operation. Not only the base structure, but all those that sit on its folds must be removed.

Inverted structures are not shown, but are included in the prototype. If a structure is added on a mountain fold, it will be inverted, that is, it will fold in and not out.

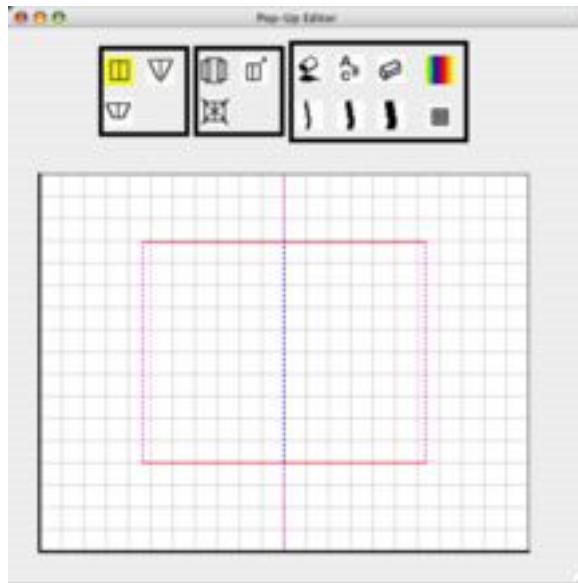


Figure 9: Adding a center parallel double slit structure

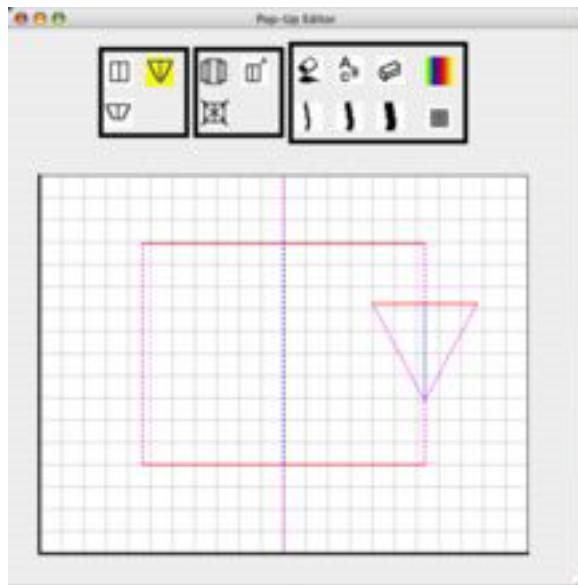


Figure 10: Adding a single slit structure to the page

7.1.2 How does the prototype editor work?

Since this code is written in Java, it takes advantage of the object-oriented nature of the language to organize the parts of the pop-up in classes. The main classes used to capture pop-ups include:

- Point3D—A point in 3D-space.
- LineFeature—A line segment. They come in several subclasses: MountainFold, Cut, ValleyFold. This class

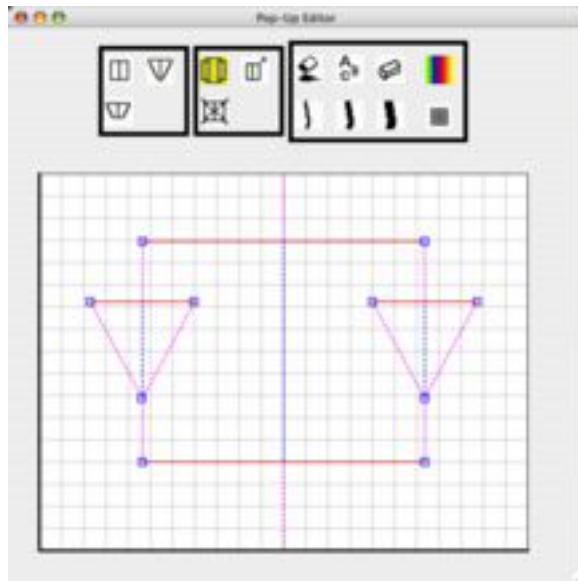


Figure 11: Using the replication feature to mirror the structures

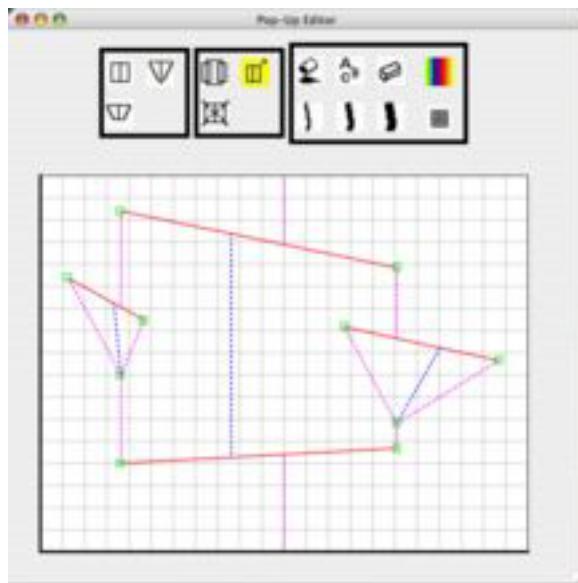


Figure 12: Using change to create asymmetric structures

also has operations that act on lines.

- Plane—A flat surface defined by the points at its corners. Each structure is composed of multiple planes. The planes are displayed in the viewer window, rather than the editor window, but they are important in defining the structure itself.
- Structure—A combination of folds and cuts that acts as a unit. It also contains information on which planes define it, and which planes it sits on. Additional types of structures will be created to add new element

types to the program. The Structure is the most important class, since each type of structure 'knows' such information as how to open itself, and what constraints exist to make it a valid structure (sides parallel, etc.)

The important classes comprising the editor interface are:

- PopUp—There is one PopUp for a page. (At present, the user may only operate on one page at a time.) This stores the structures, planes, and other global features of the page (such as the current opening angle in the viewer). This class contains methods which act on the whole page.
- PopUpEdito2d—This is a single editor window displayed by this class, which is composed of an EditorToolArea (for the buttons) and an EditorDisplayArea (for the 2D editing area). In the future, with the addition of multiple pages of parts to be added to the main page, there will be additional editor windows for these pages.

7.1.3 Full dissertation version of the editor

There are several changes or additions to existing features needed in the dissertation version of the Popup Editor.

Although I have included various decorating options for the user, such as fill, draw, and text, I am reluctant to push decoration options too far. The reader is directed to Johnson's book on children's pop-up activities [22] for some fine examples of pop-ups hand-made and decorated by children. Particularly for young children, the value of hand-decorating and writing the text would seem to be large. In the help to be added for the finished version of the program, there should be decorating ideas (crayon, paint, collage and so on) to encourage this activity.

One button on the editor window is currently inoperable. The text button should be supported to allow text blocks to be added to the page. (Support for font changes can be added with this.)

Error strategies are an interesting question in this interface. The user is prevented from adding or changing a single structure that will not open. However, she might add a structure which crosses several others (which will not work) or structures which interfere with each other on opening. It seems that there might be some value in allowing these errors in order to encourage learning about pop-up mechanisms. In addition, they would be very difficult to check. This policy might have to be changed, and this should be one concern in early user tests.

In addition to changes in existing features, there are several new features. In order to prove the general case of the use of constraints, and to make the tool more useful to young paper engineers, I will expand the repertoire of possible pop-up designs, and show that applied structures (at least one) can be added. The most likely additional element would be the V-fold (sometimes called the tent). V-folds are commonly used, adaptable to the production of many designs, and one of the most generally used and usable elements in pop-up books. Rotating arms and parallel planes (see [6] and [3] for views of these elements) are other possible additions.

There is currently no help menu. I anticipate adding some general help, and also a text box in the editor window that indicates briefly how to perform the operation currently selected.

One problem which must be solved in adding applied structures is the representation of extra pieces (and how they are related between pages), which might be facilitated by adding and labeling tabs for attachment.

Another problem is placement on the extra sheet. This ensures minimal use of extra paper by placing the pieces economically on the sheet. A possible algorithm is given in [24]; pieces may be combined on a single sheet at printing time. However, it may be reasonable to use one sheet per added structure and ignore paper waste.

In order for this prototype to be a usable program, the user needs to be able to produce usable output easily. Currently, pop-ups are printed by taking screen-shots and printing them; in the final software, a print feature will be provided. Besides printing, one should be able to save the result as a .jpg, .gif, or in another graphic format. At least one of these formats will be supported. This will allow a pop-up to be imported into other software to be further decorated.

Finally, it is necessary to be able to save current work and continue at a later time. Save and restore will be supported in XML format, which is human-readable and easily debugged. Undo and redo are also planned.

7.2 Pop-up Viewer

The window which supports the 3-dimensional representation and the simulated motion of the pop-up is the Pop-up Viewer. This displays a view of the pop-up currently in the editor in a 3D form, which can be opened and closed by the user.

This window uses a geometric constraint system to model the opening and closing of the pop-up. This approach has not been used before in animating pop-ups. Glassnerag:interactive1 dismisses constraint systems for this application. "...they have three big drawbacks for this application: they are typically large and difficult to write and debug, they are notoriously sensitive to numerical instability, and they can get stuck while searching for a solution and end up with no solution at all." [14, page 82]

These are all valid criticisms of constraint systems in general. However, in this domain, I believe that they make sense. The hill-climbing system was not large, and not particularly difficult to debug, although when other algorithms are experimented with they may turn out to be so. There are methods available to combat instability. And to answer the last point, the problems we are dealing with are actually overly-constrained, and the space of solutions is small. Part of this work is to investigate alternatives to the hill-climbing (which has problems) and to establish whether a constraint system can work well in this domain.

It must be noted at this point—getting back to the large domain of applied elements—that constraint systems offer a more adaptable method of expanding the program to other elements. If we have some standard way to express the constraints of a given structure, new variants may be added more easily and reliably. This eliminates the need to find mathematical expressions for each point of a complicated form.

7.2.1 What does the prototype viewer look like?

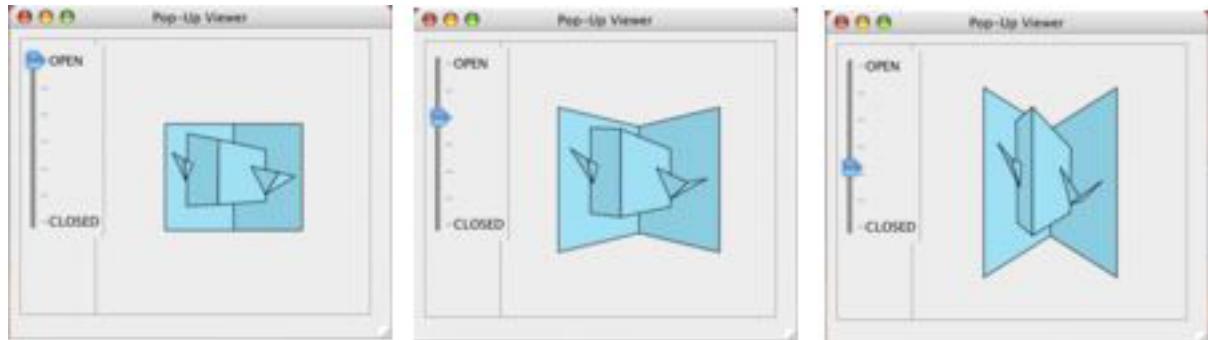


Figure 13: Using the slider to open and close the design

Figure 13 shows the manipulation of the design produced in figure 12. The planes produced by the cuts

and folds are displayed in a shaded format in which the planes on one side of an element are darker than those on the opposite side.

7.2.2 How does the prototype viewer work?

The most important class for the purposes of the viewer is the Plane. As noted for the editor, each Structure object stores the Plane objects that are a part of it and that it sits on. These Plane objects contain the original flat points of the corners, and the points in 3D space after opening.

As noted above, the opening and closing for each structure is done by the structure class. The structure knows which planes are below it, and can request information on how they have opened or closed (the changes in the points). The structure can then recalculate the position of points that lie on those planes. Using the constraints that fold and cut lengths do not change, the structure then gets the new positions of its points that do not lie on the planes below. At the present time, this is being done by calling a general function for hill-climbing with values for the current structure.

It is important to note that this is an overly-constrained system. There are two possible solutions at any point: the solution with the structure 'popped out', and the solution in which the structure's planes stay with the planes below. There is a mechanism, therefore, to forbid the 'non-popped out' solution if the angle between the pages is not 180°.

7.2.3 Full dissertation version of the viewer

The full version of this tool will have a better constraint system. At present, a hill-climbing algorithm is being employed, with the start point being the last position of the point in question. This works fairly well, but it is somewhat slow when opening pop-ups with many elements, and occasionally produces unexpected results, particularly when the structure is new and the pop-up has not been opened and closed.

The goal is to improve this by exploring other algorithms. Part of this work is in searching for possible algorithms to use, and part is experimentation to see what works best for this particular problem.

In the process, I would like to produce some general constraint language to describe a given structure. This would make it easier to add new types of elements. If the structure could be expressed as a set of constraints, those constraints could be used both in the editor (to constrain changes in the structure to those which open and close smoothly) and in the viewer (to model the opening and closing.)

7.3 Recording tool

In order to study the actions children use to design with Popup Workshop, it would be advantageous to record the user's actions. Such a tool can be added in association with the redo feature. It will keep track of each button-press, structure change, the motion of the slider, and the starting and ending file.

This tool must be configurable. That is, the researcher should be able to turn it off (without otherwise affecting the software) so that the information is not stored when the tool is not being used. Once again the output will be in XML, in order to keep it easily readable by humans or custom analysis tools which may need to be created.

7.4 Future system changes

This section deals with those changes that may be desirable at some future time, but that are not planned for this research project. Some of these changes may be made, however, if user testing indicates that they will solve

blocking problems for the users.

- The ability to add straight or curved cuts. All the current structure types would still be openable with non-straight cuts.
- User definable grid sizes (the options currently are grid and no-grid) and the ability to snap-to-grid.
- paper-size and orientation choices.
- Allow the 3-dimensional view to be rotated using the mouse. In order to do this, the planes drawn must indicate where they are cut away to create new structures. Currently this is not being done, and will be a large change. A wire-frame might be easier, but does not represent the problem domain as well (paper is not transparent), and might be confusing to the user.
- One-sided printing is planned, but there are situations in which one might like two-sided printing. Certain pop-up elements are viewable on both sides, and it would be helpful to have cut and fold lines on the back. This is complicated by current printer technology which makes accurate registration of two-sided prints difficult. It might be desirable to be able to design and print mirrored copies, to paste together in the final card or book.
- Sending output to a laser cutter or knife printer. These devices could cut, or even print and cut in one operation. Some pop-ups need to be carefully cut in order to work. Knives are not recommended for children's class work. (Children using this tool for this research will use scissors for now.)
- The recording tool for tracking the user's actions might at some point be extended to replay a sequence of actions.

7.5 Why this system is different

This system shares some elements with those described in Lee [23] and Glassner [12], and with 3D Card Maker [25]. However, it differs in that:

- Constraints are used in the simulation of the opening and closing of the pop-up. This allows new types of elements to be more easily modelled.
- The software is designed with children in mind.
- The system will be expandable to a larger range of pop-up elements. The interface is designed with expansion in mind, the viewer uses constraints to open and close the pop-up, and the object-oriented nature of Java makes it easy to add new structures.
- Pop-ups are designed through manipulations of the 'material' of the page. That is, changes are made on the 2D surfaces rather than on the 3D representation.
- This system is designed to be portable to multiple systems, and will be free, open source, and readily available.
- A recording tool is included to study use.

I believe that these items make this project unique and valuable.

8 User testing and evaluation

8.1 What will I evaluate?

There are three major areas of evaluation:

- Design process as revealed by actions of the users
- Complexity of designs produced
- Vocabulary

8.1.1 Study of the design process

Novice users and expert users of this system may show differences in the way they approach the design task. Early users of the tool will probably throw away many attempts. I would expect to see more experimentation with the tool at the start of the process of learning. On the other hand, I would expect less experimentation from expert users, who know what they want. Some things to look for:

- number of redos
- number of deletes of structures

I would expect these to decline over time.

By using the recording tool, it may be possible to find other patterns in the design methods used. It should also be possible to find out which features of the software are used most often, and which not at all.

8.1.2 Study of complexity of designs produced

There are several metrics which might be used to judge the value of this application. One might be the time needed to create a pop-up. But pure efficiency and speed of production are not among the goals we have in making design tools for children. Another approach would be to investigate aesthetic appeal, or the enjoyment in creating the pop-up. However, these are hard, if not impossible, to measure.

A more reasonable measurement might be the complexity of the completed pop-ups. It is hoped that the complexity of the pop-ups made with the tool would increase for more experienced users.

The following is a non-exhaustive list of the complexity items that will be measured:

- number of structures
- size of the smallest structure (smaller is harder)
- number of different kinds of structures
- number of asymmetric structures
- the number of ‘layers’—structures over others

I would expect a rough increase in each of these numbers (except the size, which should get smaller) as the users progress from novice to expert.

8.1.3 Study of vocabulary changes

Children play with and look at pop-ups, but are seldom asked to describe them. By talking to children about some pop-ups before and after either using the tool or making pop-ups by hand, I hope to discover how they learn to describe what they see. I would expect to see changes in vocabulary in particular (names for pop-up elements, for example.)

8.2 How will I evaluate?

I will require a small group of test subjects for a case study, as I will be working with them closely and gathering a great deal of data for each child. I plan to recruit 6-8 children in late grade school and middle school (4th-7th grade). I hope to have a group of mixed genders and ages.

Subjects will be working by themselves, with the researcher providing answers to questions asked. Books on pop-up construction will be on-hand as well.

8.2.1 Evaluation before the experiments

In order to obtain a baseline for the language use portion of the data, I will spend some time (approximately 1/2 hour) with each subject exposing them to pop-up books of various sorts and talking about how they are made. This discussion will be video recorded. I will also ask them to duplicate a simple single-sheet pop-up.

8.2.2 Evaluation during the experiments

Ongoing evaluation of the subjects will consist of periods of making pop-ups. During these work periods:

- The sessions will be videotaped to obtain data about the interactions between the children, the tool and the researcher, particularly their vocabulary and work styles.
- The recording tool in the environment will be activated in order to record the sequence of actions that the children use to produce pop-ups.
- The subjects will be able to keep any pop-ups they make. Copies of the files and photos of the pop-ups will be saved for later evaluation.

In all cases, the researcher will answer questions, and help the students look at the books or the software to find out how to do what they want to do.

8.2.3 Final evaluation

At the conclusion of the experiment:

- Each child and I will have a conversation about example pop-ups (similar to the conversation at the beginning of the tests) which will be recorded to compare with the initial conversation. Once again, the subjects will be asked to duplicate a simple single-sheet pop-up, which will be slightly more complicated than the pop-up used for the pre-test.
- The data from the recording tool will be examined to see what can be learned about the design process. Such information as false starts, number of changes, etc. will be evaluated.

- The pop-ups produced will be analysed for complexity. Possible data include: asymmetries, repeated structures, number of structures, etc.
- The pop-ups made by hand at the beginning and the end of the sessions will be compared and analysed.

8.2.4 Future research items

There are some types of evaluation that would be desirable at some point in the use of this tool. While beyond the scope of this research project they should be noted:

- A controlled study with larger numbers of children would be desirable, but would require several researchers to handle the larger numbers of children.
- A comparison between children who design pop-ups by hand and those who use the tool would also be very valuable. Once again, with only one researcher, and the larger numbers of children required to add another test group of subjects, this is not possible in this particular study.
- Classroom studies would be useful in developing curricula to use the tool in an educational environment. One possible use, for instance, would be to allow a class to produce a book, with each child able to have her own copy, since the patterns are computer-printed.
- Studies of automation of cutting, by means of a laser-cutter, knife-cutter, or printer/cutter would be interesting as a way to obtain more data about how much physical interaction with the materials is desirable, and how much these output devices might facilitate the design process.

9 The contributions of this research

When completed, the research described in this proposal will have made the following contributions:

- It will have created a tool which may be used by children to create pop-up cards and books.
- It will have shown that constraint systems can be used in modelling pop-up action by creating a 3D image of the pop-up at each stage of opening and closing.
- It will have produced data about the complexity of pop-up designs produced by children, and the change in that complexity over time.
- It will have produced data about the design processes in paper engineering used by children.
- It will have produced data about vocabulary and the change in vocabulary used by children in describing pop-up books.
- It will have produced data to determine to what extent automated data collection and processing can illuminate our understanding of the use of computerized design tools.

10 Exposures

The possible problems which I may encounter are detailed in this section.

10.1 The system

The system that currently exists is a prototype, and there is much work still to be done. Unforeseen difficulties could arise in the following:

- Additional pop-up elements need to be added. These will be applied elements, which involve additional pieces which are added to the page. Such pieces require additional windows, and some method for linking elements on different windows together, which will probably involve labeled tabs for glueing pieces on the page. In addition, the constraints must be established for new elements.
- Printing, exporting as a .jpeg or .gif, and save and restore need to be added. These should be straightforward additions, but will take time. In particular, the XML format for save and restore will need to be established.
- The recording component of the software must be designed and coded. This is new functionality.
- The constraint system must be generalized and improved. The current hill-climbing algorithm is somewhat slow, especially with pop-ups containing many elements, and it sometimes becomes confused. Although a hill-climbing algorithm provides moderately good animation, it is unknown how good or general the system can be. In addition, it is desirable to have a more general constraint description, which would allow element types to be added more easily.

10.2 The users

Since I will be working with children, there may be possible problems.

- It may be difficult to find enough subjects, or subjects may drop out for one reason or another.
- The subjects may not be motivated enough, or the software may be too daunting for them to use it over the time span necessary to gather the data.
- On the other hand, the software may be too simple to hold the interest of the subjects.
- Part of the work depends on the willingness of the subjects to talk about their experiences.
- Whether the subject can construct the items they design remains to be seen. Construction of pop-ups can be difficult, often requiring accurate cutting. The researcher may have to do some cutting.

10.3 Additional exposures

- Data analysis may take some time, especially since I will be analyzing a lot of data from the program recordings and video tapes. I anticipate making tools to do some of this, but it may delay the finish of the project.
- The funding under which I have been working runs out at the end of the spring semester of 2005. If the work is not complete at that point, the search for new funding and/or time involved in a new position may impact this work.

11 What needs to be done

The following list is my best approximation of the work to be completed:

11.1 The system

There are items still to be added to the software:

- Add additional pop-up elements to the system.
- Add recording tool to the system.
- Add save, export, restore and print to the system.
- Add help.
- Add the ability to write text blocks on the page. The button exists on the screen, but is non-functional.
- Experiment with constraint systems to find the best way of handling the opening of the pop-up.

In addition, time must be allowed for testing and adjusting the interface (some walkthroughs and testing with children.)

In the time line, I've combined the software work into one item, for the purposes of coding and testing.

11.2 User testing

During the user testing phase, which will overlap development of the system, I will need to do the following.

- Obtain Human Research Committee approval. (I have applied for approval for user interface testing, and need to apply for the case study.)
- Create the protocol for working with the subjects.
- Obtain the subjects.
- Test with the children.

11.3 Data analysis and writing

- Go over the tapes of tests for vocabulary changes.
- Analyse the data from the recording tool. This may involve tool-writing as well, as the data files may be more conveniently examined by using a tool to extract and/or count items of interest.
- Analyse the pop-ups created for complexity.
- Write the dissertation.
- Defend the dissertation.

12 Timeline

12.1 Spring 2004

- Propose
- Coding and testing of the system.

12.2 Summer 2004

- Get HRC approval.
- Coding and testing of the system.
- Informal tests with some children to make sure the interface is usable.
- Test protocols complete.
- Find subjects.

12.3 Fall 2004

- User testing with subjects.
- System modifications as needed.

12.4 Spring 2005

- User testing with subjects.
- Begin analysis of data.
- Begin writing dissertation.

12.5 Summer 2005

- Finish analysis of data if needed.
- Write dissertation.

12.6 Fall 2005

- Write dissertation
- Defend dissertation

13 Acknowledgements

The authors thanks Mike Eisenberg for his advice and patience. Glenn Blauvelt provided great help and encouragement. In addition, his proposal [5] provided a model for this document. Also contributing invaluable assistance and comments were Tamara Sumner, Eric Scharff, Kirsten Butcher, and Alena Sanusi.

The work described in this proposal has been supported by National Science Foundation grant REC0125363.

Appendix A A sample interaction with Popup Workshop

This is an actual interaction which I observed.

Heather is a 5th-grade girl who is introduced to the pop-up environment at school. She is shown several geometric pop-ups, and sits down at the computer to try something for herself.

At first, Heather draws a few random shapes. The person with her points out how to change shapes and remove them. She does that few times.

Heather draws a box (parallel double slit), and then says “Can I make a face?” When told “Why not,” she starts trying to figure out how a face might be made. It is frustrating at first that she can’t put eyes in the center of the box sides (structures can only be added to folds.) So she goes on to the nose instead.

She tries a non-parallel double slit for the nose, doesn’t like it, and removes it. She asks, “Can I draw a triangle with the point up?” (She has only drawn single slits with the point down to this point.) She tries that for the nose, and finds that it can be done. She re-adjusts the nose, sometimes by removing it and putting in a new one, sometimes by using the change feature.

She tries a mouth, using another single slit, but doesn’t think that it looks enough like a mouth. She then adds a non-parallel double slit, saying that it looked more like a mouth than a nose. She removes it and draws in a new one, oriented the other way, so it looks like a smile and not a frown.

Now back to the eyes. Heather moves the mouse around a bit and lands on the idea of putting the eyes at the side of the head, where there is a fold. She tries a single slit, but decides finally on a parallel double slit (for square eyes.) She adjusts the one eye she has created to change the size, then asks, “How can I put one in the other side that matches?”

The person helping her shows her the mirror feature that will place the other eye on the other side of the face.

Heather uses the fill button to paint the face, mouth and nose. She draws in a pupil for the eye, but doesn’t like it. She decides to erase it by filling in the plane with white but gets black instead. “That’s not bad,” she says, and proceeds to match the black square on the other eye.

After the face is printed on the color printer, she cuts the cut marks. The helper shows her how to score the folds to make them easier to fold. After folding, Heather has a pop-up card.



Figure 14: Heather’s pop-up of a face

Appendix B Glossary of pop-up terms

Many of these definitions have been taken from [32], which has an excellent glossary. Some of these terms are my own.

90-degree elements—Single-sheet elements.

180-degree elements—Applied elements.

Applied element—Applied pop-up elements are made from separate pieces of paper and attached to the base page.

Base page—The material forming the page of the book itself.

Flaps—One or more illustrated papers glued at one point. Lifting a flap exposes the illustration directly beneath it.

Gatefolds—Edge of a page folded back on itself which when unfolded extends the size of the page providing more text or extending an illustration.

Harlequinades, turn-up books or metamorphoses—Named after the popular English figure, illustrations folded up or down over themselves changing the original illustration and moving the story along.

Inverted structure—A single-sheet element having folds in the opposite direction to the standard form.

Mountain fold—Mountain folds point toward the reader.

Moveable books—Any books which have wheels, tabs, pop-ups or any other mechanism which changes the illustrations or the form of the book.

Non-parallel double slit—Pop-up element formed by two cuts and two folds not parallel to the parent fold (but meeting on it if extended) and having a quadrilateral shape.

Panoramas—An extended gatefold in which the entire book folds into a long scene.

Parallel double slit—Pop-up element formed by two cuts and two folds which are not parallel to the parent fold, but if extended would meet on the parent fold and having a quadrilateral shape.

Parallel planes—Flat pieces which are not placed on a fold, but which are attached to elements which are, causing them to lift when the page is opened.

Parent fold, base fold—Pop-ups work by being placed over a fold, which provides the motion to open the mechanism. The fold on which the pop-up is placed is the parent fold. This may be the central fold of the page, or a fold on another pop-up.

Peep show—A series of illustrations die-cut to the shape of the illustration, spaced one behind the other, and supported by side panels. The front cover has one or more die-cut openings to allow viewing of the inside illustrations. The overlapping of the various illustrations creates a sense of depth, as in looking into a tunnel.

Pop-up—An illustration which when activated either by the opening of a page or lifting a flap rises above the level of the page. A three-dimensional illustration.

Rotating arm—A piece which is placed on a fold in order to lift and rotate as the page is opened. Rotating arms can create dramatic animation effects.

Single-sheet element—Single-sheet elements are part of the base page, made by cutting and folding the base

page.

Single slit—Pop-up element formed by one cut and two folds and having a triangular shape.

Slot books—Books with slots in the picture which allow cut-out figures to be placed into the illustration

Structure, form, or element—The basic mechanism of a pop-up. They come in many varieties.

V-fold—Also known as a tent, this pop-up element is an applied element having a tent-like shape.

Valley fold—Valley folds point away from the reader.

Wheels—An illustrated disc of paper secured to the base (stationary) page by a paper disc or grommet between two adjoining pages. Stacked wheels are called volvelles.

Appendix C Reading list for the proposed work

This is a list of possible sources for my dissertation which I have not yet read. I list them here as an indication of the papers and books which I will read next. This list is not definitive or complete. In particular, there is a large segment of literature on children's design which must be explored. There are some items on this list concerned with this topic, but this is only a start.

Nathalie Bonnardel and Tamara Sumner. Supporting Evaluation in Design.*Acta Psychologica* 91 (1996), 221-244.

Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, 2003.

Allison Druin, Ed.. *The Design of Children's Technology*. Morgan Kaufmann Publishers, San Francisco, 1999.

Susan Dunn and Rob Larson. *Design Technology: Children's Engineering*. RoutledgeFalmer, 1990.

L. Hanna, Kirsten Risdan, Mary Czerwinski and Kirstin J. Alexander. *The Role Of Usability Research In Designing Children's Computer Products*. The Design of Children's Technology. (1999), 3-26.

Glenn A. Kramer. *Solving Geometric Constraint Systems*, MIT Press, Cambridge MA, 1992.

Kim Marriott and Peter J. Stuckey. *Programming with Constraints: An Introduction*. MIT Press, Cambridge MA, 1998.

Jean Piaget, Bärbel Inhelder and Alina Szeminska, translated by E. A. Lunzer. *The Child's Conception of Geometry*. Harper and Row, New York and Evanston, 1960.

Thomas C. Reeves. *Enhancing the Worth of Instructional Technology Research through 'Design Experiments' and Other Developmental Research Strategies*. Paper presented April 27, 2000, International Perspectives on Instructional Technology Research for the 21st Century, (it.coe.uga.edu/treeves/AERA2000Reeves.pdf).

Les Tickle. *Understanding Design and Technology in Primary Schools: Cases from Teachers' Research*. Routledge. 1996.

References

- [1] AVELLA, N. *Paper Engineering: 3D design techniques for a 2D material*. Roto-Vision, Mies, Switzerland, 2003.
- [2] BAUM, L. F. *The Wonderful Wizard of Oz*. Little Simon, New York, NY, 2000. paper engineering by Robert Sabuda.
- [3] BIRMINGHAM, D. *Pop Up!: A Manual of Paper Mechanisms*. Tarquin, Norfolk, UK, 2000.
- [4] BLAUVELT, G., EISENBERG, M., AND WRENSCH, T. Creating Mechanical Toys: Steps Toward a CAD Tool for Educational Automata. Presented at WCCE2001, The World Conference on Computers in Education, Copenhagen, Denmark, July 2001.
- [5] BLAUVELT, G. R. A Design Environment for Automata: Supporting Children's Construction of Mechanical Reasoning and Spatial Cognition. Ph.D. Dissertation Proposal, Department of Computer Science, University of Colorado, 1999.
- [6] CARTER, D. A., AND DIAZ, J. *The Elements of Pop-Up: A Pop-Up Book for Aspiring Paper Engineers*. Little Simon, New York, NY, 1999.
- [7] CHATANI, M. *Pop-up Greeting Cards*. Ondori/Japan Publications, Tokyo, Japan, 1986.
- [8] CHATANI, M., AND NAKAZAWA, K. *Pop-Up Geometric Origami*. Ondori/Japan Publications, Tokyo, Japan, 1984.
- [9] CHEN, T. Y. Hyperspider: integrating computation with the design and construction of educational crafts. Master's thesis, Department of Computer Science, University of Colorado at Boulder, 1999.
- [10] DIEHN, G. *Making Books that Fly, Fold, Wrap, Hide, Pop Up, Twist, and Turn*. Lark Books, New York, NY, 1998.
- [11] EISENBERG, A. N. *An Educational Program for Paper Sculpture: A Case Study in the Design of Software to Enhance Children's Spatial Cognition*. PhD thesis, Department of Computer Science, University of Colorado at Boulder, 1999.
- [12] GLASSNER, A. Interactive Pop-up Card Design. Tech. Rep. TR 98-03, Microsoft, jan 1998.
- [13] GLASSNER, A. Methods for producing pop-up cards, and cards produced thereby, oct 2001. Patent Number 6,311,142.
- [14] GLASSNER, A. Interactive Pop-up Card Design, Part 1. *IEEE Computer Graphics and Applications* 22, 1 (jan-feb 2002).
- [15] GLASSNER, A. Interactive Pop-up Card Design, Part 2. *IEEE Computer Graphics and Applications* 22, 2 (mar-apr 2002), 74–85.
- [16] GREENBERG, G. *The Pop-up Book of Phobias*. William Morrow and Company, Inc., New York, NY, 1999. paper engineering by Matthew Reinhart.
- [17] HEDGES, C. In Him, Storyteller Meets Architect. New York Times, December 9, 2003.

- [18] HEINO, A. Movemania. http://www.euronet.nl/users/aheino/public_html/.
- [19] HINER, M. *Paper Engineering for Pop-Up Books and Cards*. Tarquin, Norfolk, UK, 1999.
- [20] JACKSON, P., AND FORRESTER, P. *The Pop-Up Book : Step-By-Step Instructions for Creating over 100 Original Paper Projects*. Henry Holt and Co., New York, NY, 1996.
- [21] JOHNSEY, R. Observing the way primary children design and make in the classroom: an analysis of the behaviors exhibited. In *IDATER 93, Loughborough University of Technology*.
- [22] JOHNSON, P. *Pop-Up Paper Engineering: Cross-Curricular Activities in Design Engineering Technology, English and Art*. RoutledgeFalmer, London?, 1992.
- [23] LEE, Y., TOR, S., AND SOO, E. Mathematical Modelling and Simulation of Pop-up Books. *Computers and Graphics* 20, 1 (1996).
- [24] MILENKOVIC, V., DANIELS, K., AND LI, Z. Placement and compaction of nonconvex polygons for clothing manufactures. In *Proc. 4th Canad. Conf. Comput. Geom.* (1992), pp. 236–243.
- [25] MITANI, J. 3d card maker. http://www.page.sannet.ne.jp/jun_m/card3d/index-eng.html.
- [26] NIELSON, C., AND STEER, D. *Snappy Little Bugs*. Silver Dolphin Books, San Diego, CA, 1999. paper engineering by Richard Hawke.
- [27] OPPENHEIMER, T. The Computer Delusion. *The Atlantic Monthly* 280, 1 (July 1997), 45–62.
- [28] REEVES, B. N. *Supporting collaborative design by embedding communication and history in design artifacts*. PhD thesis, Department of Computer Science, University of Colorado at Boulder, 1993.
- [29] RUBIN, E. G. My trip to mecca. <http://www.popuplady.com/MECCA.html>.
- [30] SABUDA, R. Robert Sabuda Home Page: Children's Book Creator. <http://robertsabuda.com/>.
- [31] SIMMT, E., AND DAVIS, B. Fractal Cards: A Space for Exploration in Geometry and Discrete Mathematics. *Mathematics Teacher* 91 (feb 1998), 102–108.
- [32] THE POP-UP LADY. The pop-up lady: History of pop-ups. <http://www.popuplady.com/History>
- [33] URIBE, D. *Fractal Cuts*. Tarquin, Norfolk, UK, 1999.
- [34] VALENTA, B. *Pop-O-Mania : How to Create Your Own Pop-Ups*. Hamlyn, London, UK, 1997.
- [35] WALSER, H. The Pop-up Cuboctahedron. *The College Mathematics Journal* 22, 2 (mar 2000).