Chapter 4

Pop-ups and Computation

Paper engineering was introduced in Chapter 3 as a representative craft with history, practice, and value as an activity for children. During the discussion of pop-ups as a craft, mention was often made of their modularity, in that pop-ups are composed of simpler elements.[1] This modularity influences the way that the craft is practiced and learned. For instance, in developing the knowledge necessary to be a paper engineer, it is important to learn which elements are available, the motions they produce, and ways in which they can be combined.

Since modularity is so important to the craft of paper engineering, this chapter examines the pop-up elements that form its building blocks. Two of the ways that computation can aid the learning of a craft that were presented in Chapter 2 are particularly applicable here. First, the computer can simplify the design process. In the case of the pop-up elements, not only are a limited number of elements used in pop-ups, but each element requires certain geometric constraints be followed in its construction. These geometric constraints are amenable to computational enforcement. Second, the learner can benefit from seeing both the notation or pattern for a design and the finished construction, in this case the 2-dimensional paper to be cut, folded and glued alongside the 3-dimensional pop-up in motion. The computer can present these simultaneously.

In order to develop the high-level requirements for a system that aids users in learning pop-up design, it is appropriate to first examine the elements that form the foundation of pop-

[1] In that discussion, the term *device* is used to indicate everything from a single moving part, such as a flap, to an entire page of moving paper mechanisms. The word *element* is used for the more basic structures that make up pop-ups specifically.

ups in more detail. In addition, previous research into software for pop-up design will provide some guidance.

This chapter begins with a description of movable book devices and terminology, and more specifically, pop-up element types and construction. A literature review of research in the area of pop-up animation, mathematics and modeling follows. Finally, a brief summary of the principles and philosophy for pop-up design software for children is presented, along with a discussion of how these follow from the literature, from the discussions in previous chapters, and from the pop-up elements themselves.

4.1 The Composition of Movable and Pop-up Books

This section introduces a taxonomy of pop-up elements and how they are related to other movable book devices, the terminology needed to discuss them, the construction, motion, and use of those elements, and how they can be combined. The most useful sources for this information come from instructional books about paper engineering. A list of these books can be found in Appendix B. In particular, the current section relies largely on four references. Carter and Diaz, *The Elements of Pop-up* [14] is a large collection of pop-up elements in actual, functioning form. Jackson and Forrester's *The Pop-up Book* [56] is an excellent source for variations and combinations of elements and includes a design section and striking photos. The best description of most elements, and certainly the most complete and useful explanation of constraints in pop-up elements is found in Birmingham's *Pop-up! A Manual* [6], although it is primarily aimed at older children and teens. Finally, it is useful to be able to consult a pop-up book made for the use of young children, Valenta's *Pop-o-Mania* [123]. This book uses a highly restricted set of elements, and yet demonstrates the large variety of pop-ups that can be constructed from them.

A few words should be added here about terminology. The terminology in pop-up making is not standardized, and the case of elements illustrates this perfectly. Various books use different terms for what here will be called an element: device, form, mechanism, structure, element, construction and technique. In the same way, the names for individual elements, portions of elements and construction techniques also vary greatly. The names for elements used here were chosen for their simplicity ("step" rather than "180° parallel double slit" for instance) or their commonness ("v-fold" is used commonly, although not universally, for one element). In part, this is to allow the software to use child-friendly names, but consistency in naming also simplifies the discussion of element types.

4.1.1 Movable Devices

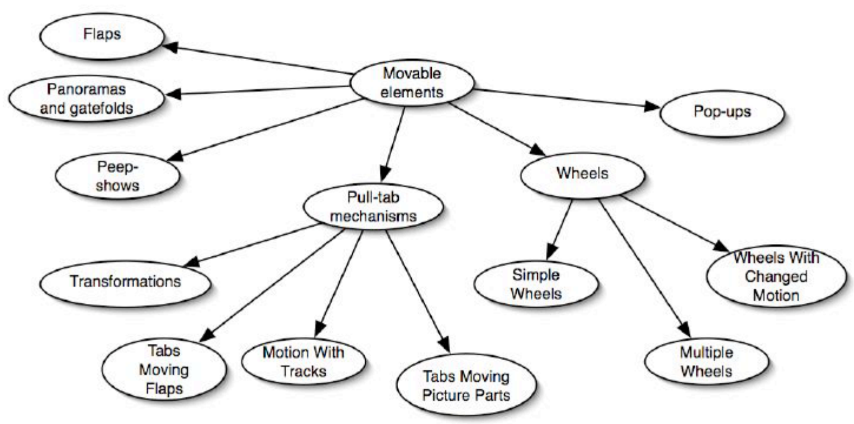

Figure 4.1: A Partial Taxonomy of Movable Devices: This illustrates the relationship between some of the more commonly used devices. Note that pull-tabs can animate multiple types of devices.

Figure 4.1 presents a partial taxonomy of devices used in movable books, most of which were introduced in Section 3.1. It can be seen that pop-ups are a distinct type of movable device. When talking about movable books the discussion focuses on the device and devices are composed of elements (even if a device is but one element.)

In the above taxonomy, both wheels and tabs can be used to produce motion in several ways, and are therefore divided into several types. There can be one or many wheels in a mechanism, and the wheel's motion can further be transformed by a lever into other types of motion. Similarly, pull-tabs can move the parts of transformations to change one picture to another or animate flaps or parts of a picture (as seen in Figure 3.12) or even pull or push a piece along a track.

Besides being pulled manually, pull-tabs can also be attached to other parts of the page[2] , or the opposite side of the gutter (the center fold of the page) to allow the tab to be pulled automatically when the book is opened.

Figure 4.2: A Gatefold Producing Side-Pages: A page from *Raggedy Ann and Andy and the Camel with the Wrinkled Knees* [41]. The gatefold is folded into a panorama and the folds are glued on the back side to produce the effect of an additional book on the left side of the page.

Carousels are a form of book and not simple devices, and have therefore been left out of this taxonomy, but peep-shows have been included, as they can be present in small size on a single page. Panoramas can also exist on a single page, most commonly in the form of *gatefolds*. Gatefolds are made by folding over one end of the page that has been made wider than the book itself (the *Playboy* centerfold is a perfect example of a gatefold.) Gatefolds can also be multiply-folded to produce a panorama effect, or glued together on the back side to produce the effect of another book attached to the edge of the page. An example of this device can be seen in Figure 4.2. This has become quite common in contemporary pop-up books, as the extra pages produced

[2] The term *page* is used in this dissertation to refer to the combinations of left and right leaves of a book or card when it is opened.

by the gatefold can accommodate more text or, as seen in this example, additional small pop-ups. It also allows the book to close into a flatter form. Most pop-ups are thicker near the gutter, and the gatefold adds thickness to the outer edges of the left or right sides of the page.

4.1.2 Pop-up Elements

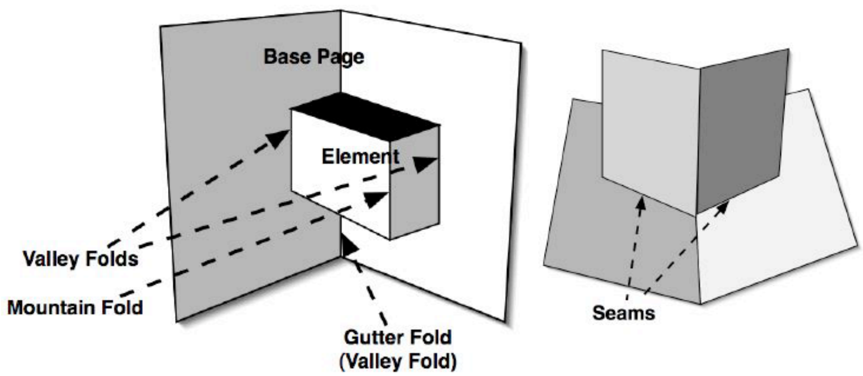

Figure 4.3: Parts of Pop-up Elements on the Base Page: Shown on the left is an element without extra pieces (a 90° element). The element on the right demonstrates seams on an element with an extra piece attached to the page (a 180° element).

The features of elements arise from the three operations that can be performed on paper to produce pop-ups: folding, cutting, and gluing. A feature common to all pop-up elements is the fold. For the purposes of introducing pop-up elements, assume that the element is sitting alone on the *base page* (that is, the open page of the book)[3] . Folds are of two types, valley folds and mountain folds. A *valley fold* is a fold in which the fold line is farther from the viewer than the paper on either side. The opposite fold is a *mountain fold*. The center fold or seam of the book or card is called the *gutter*, and is most commonly a valley fold. Cutting creates a *cut*, and gluing a piece onto the page creates a *seam*.

Figure 4.3 illustrates folds and seams. The element on the left is called a *step*, and is composed of two cuts and three folds. The step will be used for many examples in this chapter, as

[3] Combinations of elements will be discussed in Section 4.1.3, and, as it will be seen, are not very different from isolated elements, except for some additional terminology.

it is a simple element both to make and to visualize. The element on the right, used to illustrate the seam, is composed of an extra piece glued onto the base page and is called a *v-fold*.

For an element to fold and unfold properly two conditions must be met. First, its geometric constraints must be satisfied. Second, it must not collide with any other element.

To illustrate how geometric constraints affect an element, consider the two example step elements shown in in Figure 4.4. As previously mentioned, a step consists of two cuts, indicated by the solid lines, and three folds (plus the gutter over which the step is placed) indicated by the dashed lines. While the shape of the cuts is unconstrained, so long as the cuts intersect the folds and do not intersect each other, the folds are constrained such that all of the folds, including the fold in the underlying page, must be parallel and the distances between them must allow for the formation of a parallelogram cross-section.[4] That is, $\overline{AB}||\overline{CD}||\overline{EF}||\overline{GH}$, and the distance from $\overline{AB}$ to $\overline{CD}$ = the distance from $\overline{EF}$ to $\overline{GH}$. While the step has a fairly simple set of constraints, others, like the v-fold are more complex. Detailed information about the geometric constraints of five common elements (those shown in Figure 4.6) is presented in Section 5.4.3.

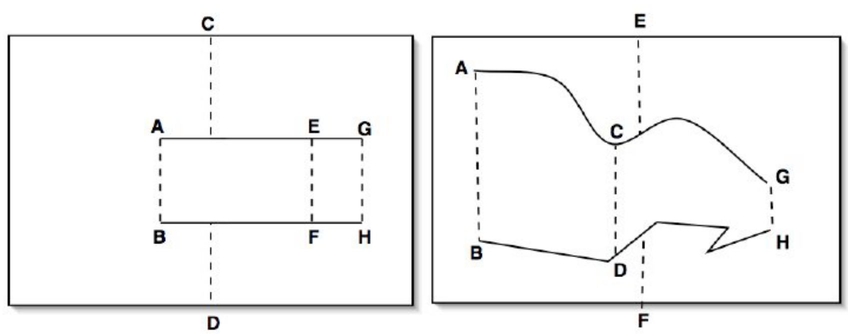

Figure 4.4: Two examples of the step element that follow the constraints on this element. Regardless of the shape of the cuts, the folds are parallel, and the distances allow both elements to fold.

Variations can be made to elements so long as the constraints are not violated. Cuts can be of any shape and material can be added or removed from elements. For instance, a door or

[4] Most pop-up elements rely on the collapsibility of the parallelogram (the cross-section of the element) or, in 3-dimensional terms, the collapsibility of a four-sided prism or pyramid.

window could be cut into a step that represents a house, or a v-fold could be trimmed into an animal's nose.

Figure 4.5 shows a partial taxonomy of pop-up elements. Those elements that are part of Popup Workshop are shown shaded. There are other elements in the pop-up domain, but some are combinations or variations of these elements and others are seldom used. The elements included here serve to illustrate the general categories of pop-up elements.

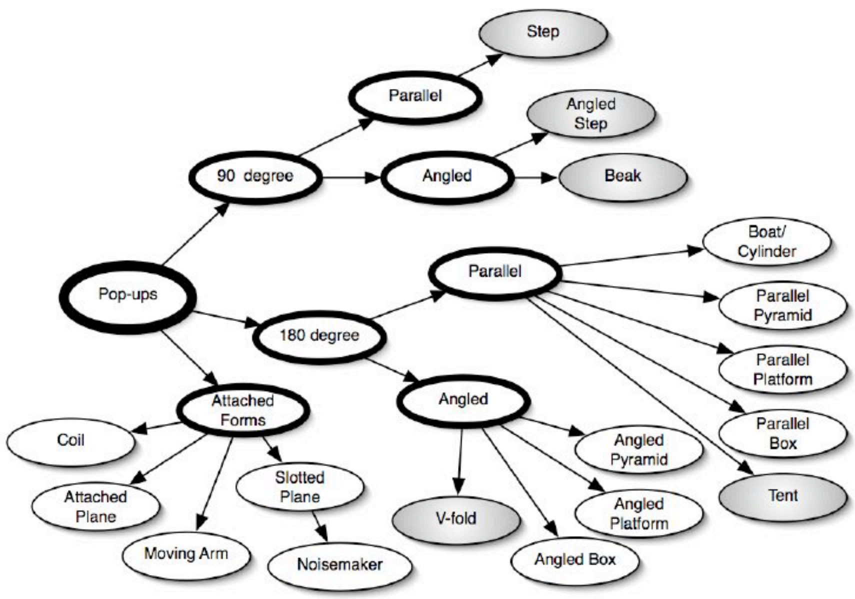

Figure 4.5: A Partial Taxonomy of Pop-up Elements: This illustrates the relationship between some of the more commonly used basic elements. Shaded elements are available in Popup Workshop.

Most pop-up elements can be divided into 90° and 180° elements. As their name suggests, 90° elements are best viewed when the page is halfway open at 90°, and folding the page flat results in these elements rejoining the page from which they are cut. Another distinguishing feature of 90° elements is that they are made without additional pieces. That is, they are composed only of cuts and folds made to the base page. 180° elements, while they can have cuts and folds, will also have extra pieces glued onto the page and are best viewed when the page is fully opened. Since the pages of a pop-up book are usually opened fully when viewing the pages, professional pop-ups are made almost entirely from 180° elements.

Each of those categories is further divided into angled and parallel elements. A parallel element's folds or seams are parallel to the fold or seam on which it is placed (the *parent fold*), while in an angled element, the folds and seams are at some angle to the parent fold. Many elements exist in both angled and parallel forms.

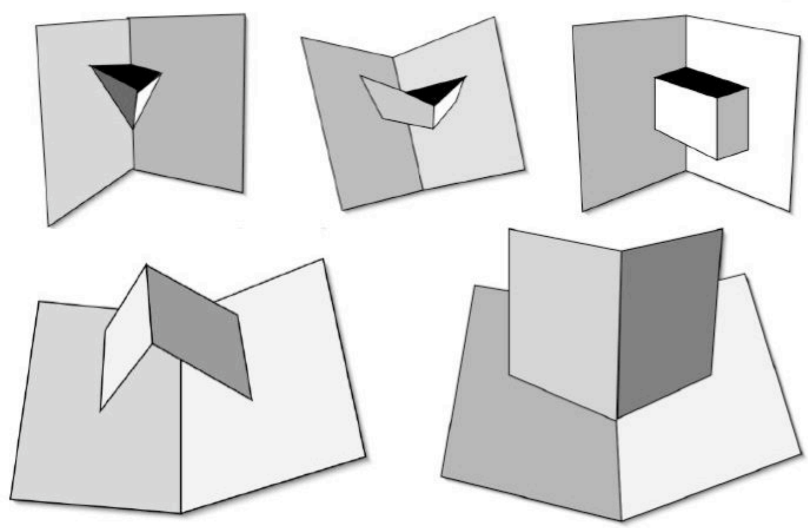

Figure 4.6: Five Commonly Encountered Pop-up Elements: Top from left to right, three 90° elements: beak, angled step, step. Bottom from left to right, two 180° elements: tent, v-fold. These are the elements used in Popup Workshop.

In this taxonomy the three 90° elements are closely related (Figure 4.6). The *beak* consists of a single cut across the parent fold. Three folds are made from a single point lying on the parent fold (or an extension of the parent fold) to the cut, forming a pyramid. Since the pyramid's base is a parallelogram, the pyramid can be flattened, and will therefore open and close. The *angled step* is essentially a beak with the point removed by another cut. (The folds, if extended, would still all meet to form the pyramid.) The *step*, on the other hand, is the case where all of the folds, including the parent fold are parallel. This produces a four-sided prism that is also flat-foldable. Although the step is produced with parallel folds and the angled step with a hidden intersection point of all the folds, they are similar in appearance.

The geometric constraints on the step have been discussed. Those of the beak are similar but as it is an angled element, the constraints are on the angles between the intersecting folds rather than on the distances between parallel folds. See Section 5.4.3.2 for more detail.

With respect to 180° elements, two of the simplest and most common are the *tent* and the *v-fold* (Figure 4.6). The tent is a relative of the step and relies on the production of a prism with a parallelogram cross-section. However, two of the sides of the prism are produced by an added piece of paper. The tent and the step provide a good example of the difference between 180° and 90° elements. If the page is opened to 180°, the tent stands up from the page (if the sides are long enough) while the step lies flat. The tent, like the step, is a parallel element, since the sides that are glued to the page must be parallel to the parent fold. The v-fold, on the other hand, is an angled element since the sides glued to the page are not parallel to the parent fold. The v-fold is in fact the 180° equivalent of the beak, with the same pyramidal cross-section and similar geometric constraints, even though the added piece involved makes its constraints more complex.

There are two ways to produce collapsable pop-up elements. One is for all folds to meet at a point on the parent fold. This is seen in the beak, the angled step, and the v-fold. The other method is for all folds to be parallel as in the tent and the step. A proof of the necessity that one of these conditions must exist is presented in Lee, Tor and Soo [66]. The existence of these two conditions provides the basis for the differentiation between angled and parallel elements. In all cases, the geometric constraints on angled elements include the intersection of the folds. In all cases, the geometric constraints on parallel elements require parallel folds. Several elements exist in both angled and parallel forms, and three of the most illustrative are boxes, pyramids and platforms (Figure 4.7).

Boxes are frequently used in professional pop-ups, often to represent houses or other structures. The boxes illustrated here are open on the top, but a top can be added to pop up or down into place, or even produce a peaked roof effect. The parallel form of the box has extra folds along the sides that intersect the parent fold in order to allow the box to fold. The angled

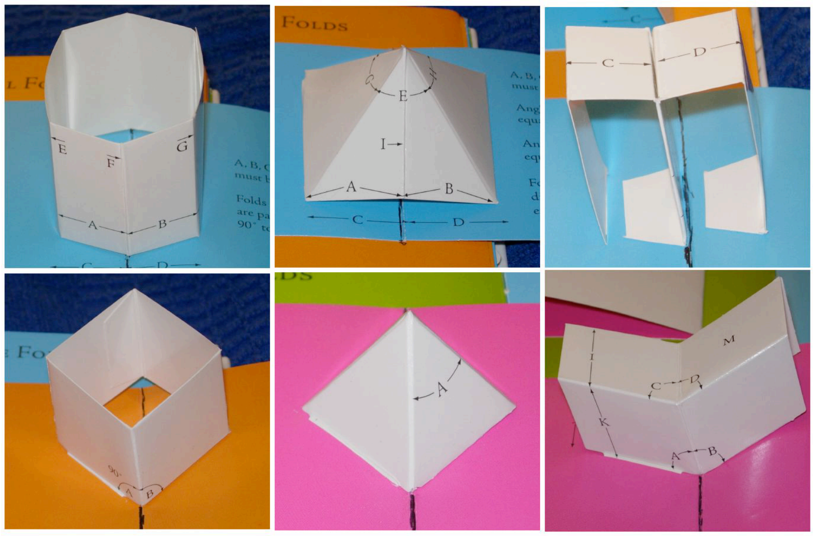

Figure 4.7: Box, Pyramid, and Platform Elements: Each element is shown in parallel (top) and angled form (bottom). Photos of elements included in Carter and Diaz, *The Elements of Pop-up* [14]

box, on the other hand, acts much like a v-fold as one side on either side of the parent fold is attached to the page, while the other sides are not attached and lift up from the page as the page is closed.

While *pyramids* are seldom used in professional pop-ups they are included here as an example of unusual shapes that are possible using the same principles as boxes. Once again, the parallel form of the element folds in the center, and the angled form lifts up to allow the page to close.

The *platform* is an interesting element. It is commonly used to hold up a flat, table-like sheet of paper, and the platform and the attached sheet bend in the center to allow closure. This supported sheet or plane can be part of the design, a lily-pad or table for instance, and is often used to hide elements that are needed to produce motion but are best kept invisible. The most important difference between the platform and the box is that the platform allows the center of the element to collapse to allow the folding of the attached paper surface. There are several variations of the parallel platform, that can be made more or less stable depending on the paper engineer's desires. The parallel platform pictured in Figure 4.7, for instance, can sway from left to right when pushed, while the angled platform shown is much more rigid. The angled platform is, at its heart, two v-folds connected in the center while the parallel platform is based on uprights placed parallel to the parent fold.

Figure 4.8: Boat and Coil Elements: On the left, a boat element, a 180° parallel element that bends the paper. Photo of element included in Carter and Diaz, *The Elements of Pop-up* [14]. On the right is a coil, illustrated in Valenta, *Pop-o-Mania* [123]. The coil is here classified as an attached form, as it has no folds.

It is also possible to have elements that rely on the bending of the paper. One of these elements is the *boat*, shown in 4.8. The boat is pulled from two sides, and the attachment points are parallel to the parent fold, making the boat a parallel element. If the pull is large enough, a cylinder can be formed. As with the box, a top can be attached if a closed form is desired.

One group of elements that is not so easily placed into 90° or 180° categories is the attached forms. These are flat pieces of paper placed onto an element to either extend it or to provide some desired motion. They can be attached to any form of element: 90°, 180°, angled or parallel. Attached forms provide great latitude in creativity for the paper engineer. Attached forms can be divided into several categories.

One type, the coil (Figure 4.8), is extremely simple to make but can produce a striking effect. It is simply a spiral cut from paper, with one end glued to each half of the page, between

Figure 4.9: Attached Planes: Two methods of attaching extra planes to an element. On the left, an arm holds the attached plane away from the element, a parallel box in this case. Photo of element included in Carter and Diaz, *The Elements of Pop-up* [14]. On the right, the planes are directly attached to the side of step elements, from Valenta, *Pop-o-Mania* [123].

elements, or between an element and the page. The coil spreads as the page is opened, and the spring-like form curves and bends.

Attached planes are flat pieces that are attached to elements in order to extend them. There are two main ways in which this is accomplished as illustrated in Figure 4.9. First, an arm can be used to hold the plane away from an element and parallel to it. This can be done with either a modified tent or a doubled set of arms for additional stability as shown in the figure. Second, pieces can be glued to the side of an existing element. So long as the attached plane does not interfere with the motion of other elements, any plane of an element can be extended and the element will fold correctly. In the figure, the sides of two steps are extended with attached planes to model a house and a tree.

A special case of the attached plane is the *moving arm*. There are several methods of constructing a moving arm, but the most common is to add a beak to another element and connect the arm to the beak. This is one place where a 90° element is used in professional pop-ups. Figure 4.10 shows the attachment and motion of one such moving arm.

Finally, attached forms can be slotted into other elements. Slots are made in the piece to be attached and part or parts of the elements are pushed through them. Conversely, the elements

Figure 4.10: Moving Arm Element: The motion of the lamb is surprisingly large, and the motion is driven by a beak to which the lamb is attached. From Steer, *Snappy Little Farmyard* [109]

can be slotted and the attached form pushed through. Figure 4.11 shows a slotted attached form and a special case of the attached form called the *noisemaker* that rubs small teeth across an element to produce a sawing noise as the page is opened and closed.

While the taxonomy of Figure 4.5 includes virtually all pop-up elements, it gives no indication of how common each element is in practice. Table 4.1 summarizes the number of occurrences of all of the elements found in three example pop-up books. These books are ones that were shown to the children during user testing in order to observe whether children learned to identify the elements used in professionally produced pop-ups (see Section 6.2.2). These particular books were chosen because they varied in the complexity of their construction and in the age level of their intended audiences. *Snappy Little Farmyard* [109] is a simple picture book aimed at pre-school or younger elementary school aged children. *Haunted House* [85] (previously discussed in Section 3.1.6) started the trend of including many pop-up elements on each page and is intended for older elementary school aged children. Finally, *Raggedy Ann and Andy and the*

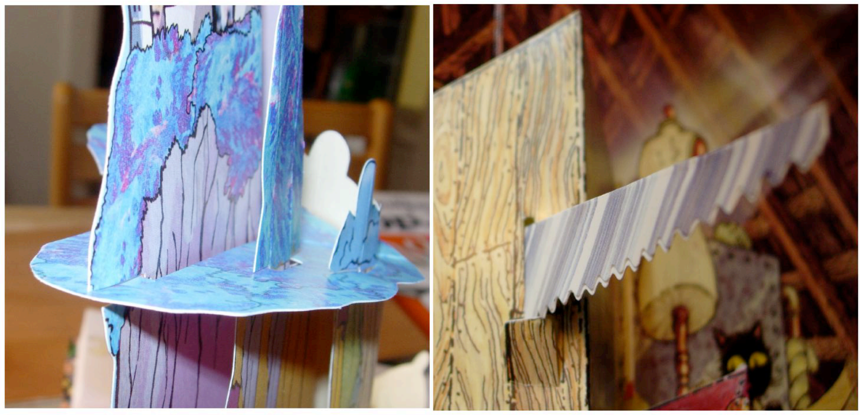

Figure 4.11: Slotted and noisemaker elements: On the left is a slotted plane with three v-folds positioned through the slots to produce a floating castle effect, from *Raggedy Ann and Andy and the Camel with the Wrinkled Knees* [41]. On the right is a noisemaker. The plane, slotted through a tent, rubs on the tab and creates noise, from *Haunted House* [85]

.

Camel with the Wrinkled Knees [41] is intended for young people or adults and presents sophisticated variations on a variety of pop-up elements. Example pages from each of these books are shown in Chapter 6.

In counting elements in these books, variations were included along with the simple form. For instance, a v-fold can be bent into a beak shape or the center of the added piece can be cut out to make an arch, but both would be counted as v-folds. When moving arms were attached to beaks, the beak and arm were counted separately. Wheels were counted once for each page on which they were displayed so a wheel serving two pages counted as two wheels.

Attached planes, v-folds, and tents were the most commonly used elements and accounted for 65% of the total used. Examples of attached forms vary widely, being used in a multitude of ways and attached by different means. The only 90° elements found in these books were beaks, used both alone and as supports for moving arms.[5] The books intended for an older audience contained not only more elements, with the number of element roughly doubling with each book, but also displayed more variety in the elements chosen. The large number of attached planes in

[5] All of the moving arms were supported by beaks.

Element Type	Element	Farmyard	Haunted House	Raggedy Ann	Total
90°	beak	5	8	4	17
180°	v-fold	8	5	15	28
	tent	15	11	36	62
	parallel box	0	0	1	1
	angled box	0	1	0	1
	angled platform	0	0	1	1
Attached piece	moving arm	4	5	8	17
	coil	0	2	0	2
	slotted piece	0	3	5	8
	noisemaker	0	1	0	1
	attached plane	2	9	47	58
Movable device	gatefold/panorama	0	0	12	12
	transformation	0	1	1	2
	pull-tab	0	7	1	8
	wheel	0	2	0	2
	flap	0	8	0	8
	Total	34	63	131	228

Table 4.1: Elements used in three commercial pop-up books of increasing complexity and age level. The intended age of the reader increases from left to right.

Raggedy Ann is particularly striking and may be partly a factor of the paper engineer's (Kees Moerbeek) style. Of these books, *Haunted House* accounted for the majority of the non-pop-up movable elements. Gatefolds were only seen in *Raggedy Ann* where two are found on each page.

4.1.3 Combining Pop-up Elements

So far, the discussion has centered on individual elements attached to the base page. Construction techniques for combining elements are much the same although they can introduce difficulties when trying to visualize or calculate the resulting motion. Combining multiple levels of elements also presents challenges when considering computer aided design of pop-ups, as the points to which an element is anchored change in ways that depend on changes to the elements beneath it.

Combining elements can introduce dependencies since the removal or change of one element can affect those elements placed on it. Because of these dependencies, it is important to

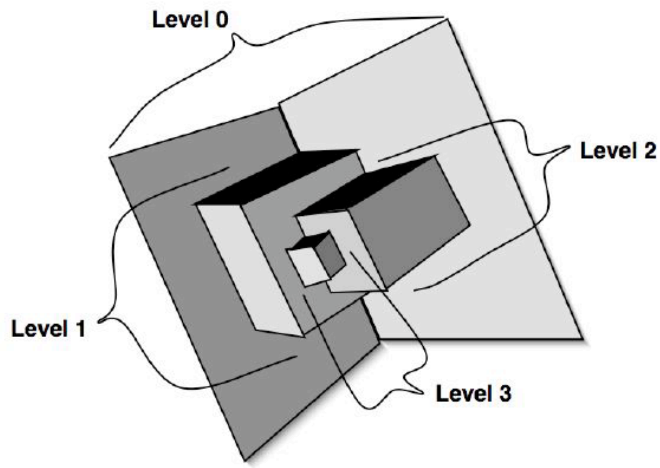

Figure 4.12: Three step elements combined into one pop-up. The base page is at level 0 and each added element is at a level one greater than the highest level to which it is attached.

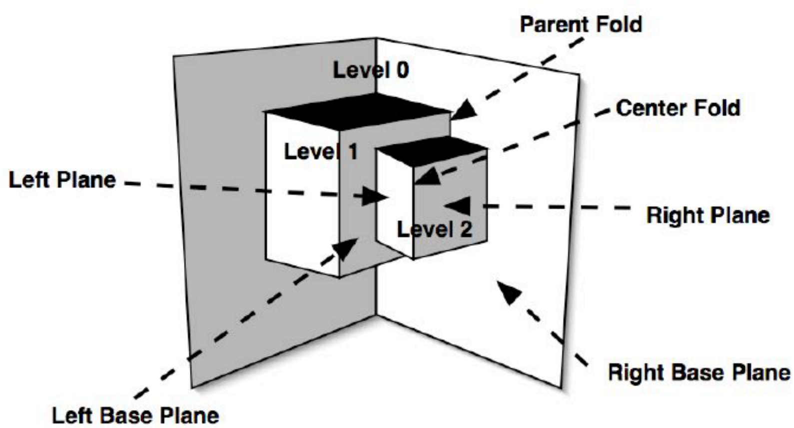

Figure 4.13: Terms describing a pop-up element at Level 2. Note that every element of level 1 or greater has these parts and that Level 0 lacks base planes and a parent fold. Note also that the parent fold for elements of level 1 is the gutter of the base page.

know the level of each element whenever elements are to be placed on one another. As an example, Figure 4.12 shows the result of combining three step elements. In this example the base page is designated level 0 and the level of each added step is always one more than the highest numbered level to which it is attached. Additional terminology also becomes necessary when describing combinations of elements. Figure 4.13 illustrates these terms for a step at level 2. This step has both a left and right plane. The fold over which the step sits is its parent fold (if the element is at level 1 its parent fold will be the gutter). Additionally, there are right and left base planes, to which the right and left planes of the step are attached.

Constraints on individual elements also apply to combinations of elements. So long as the constraints in relation to the parent folds are met for an element and all the elements below it, the pop-up will fold. One additional constraint for higher-level elements is that they must be small enough to sit on the planes of the elements below them. Therefore, elements become smaller at higher levels.

Figure 4.14: An Inverted Element: Placing the same element over valley and mountain folds changes the appearance of an element. Three steps have been placed on another step. The center step, placed over a mountain fold, looks different than the steps over the valley folds.

All elements can be placed over either mountain or valley folds, and 180° elements can also be placed over seams (the edge where a tab is glued). 90° elements cannot be placed over seams, as they rely on a cut being made across the parent fold and in the case of the seam, not a simple fold, this is not possible. Whether a 90° element is placed over a mountain or valley fold directly affects its shape. The normal form of a 90° element can be considered to be one placed over the gutter or any other valley fold. In contrast, a 90° element placed over a mountain fold has all of its own folds reversed in direction from that normal form. 90° elements which have folds reversed from their normal form will be termed *inverted elements*. To illustrate, in Figure

4.14, steps have been placed over each of the folds of a step at level 1 with the center step being an inverted step. Inverted 180° elements are seldom encountered, as they must be placed on the back of the page or other elements in order to have their folds reversed.[6]

4.2 Previous Research on Computationally Enhanced Pop-up Design

While the literature dedicated to the study of using computers to design pop-ups is quite small, common threads of inquiry nonetheless emerge and will help to focus the discussion in this section. Four core topics appear repeatedly: the mathematics underlying the operation of pop-ups, the animation of virtual pop-ups, design tools for creating pop-ups, and techniques for bringing virtual pop-ups to physical realization.

The work reviewed here comes from the only research areas known to be considering computational tools for the creation of pop-ups. First, the work from Nanyang Technological University (NTU) in Singapore focuses primarily on the mathematics of pop-up creation and the design of tools for professional paper engineers that would facilitate the production of mass-produced pop-up books. Second, the literature from the origamic architecture community [7] concentrates on the development of computer based design tools to allow adult hobbyists to create one-off pop-up cards in the origamic architecture style. Finally, Andrew Glassner's work describes a software tool to help adult hobbyists create multiple copies of pop-up cards that could be given for special occasions.

There is other related research not included here. For example, general research on the modeling of paper and the extensive literature of computational origami are not included. Although these areas of inquiry relate to paper-crafts, they do not involve the production of pop-ups. The production of virtual books, such as the digitization and display of movable books for digital library purposes (see Cubaud, Dupire and Topol [21] for one such study), has been eschewed, as the focus here is on creating physical objects.

[6] The v-fold may be considered to have an inverted form. For a description of the inverted v-fold, see Section 5.4.3.4.

[7] See Section 4.2.2 for a description of origamic architecture and examples of pop-ups produced in that style.

4.2.1 Mathematics of Pop-ups

Two papers from NTU focus on the mathematical properties of both 90° and 180° pop-ups elements. The authors' ultimate goal is stated to be the production of a CAD system for creating professional pop-ups. As they point out, only a small number of skilled people are paper engineers. In addition, pop-ups require a great deal of time to design, and the publishing process requires a large number of steps:

> ...pop-up book manufacturers are looking to computerization, which can capture some of the tedious and time consuming parts of the process and perform them automatically and efficiently. Ultimately the goal is a computer-aided design system with integrated manufacturing facilities. [66, p. 21]

Much of their emphasis, therefore, is on efficiency. Their concerns are not those of this work, focused as it is on keeping the craft aspects of the design and building of pop-ups alive and particularly on helping children learn the craft. However, the authors are clearly focused on paper engineering as it is practiced, and their work is required reading in this area.

Lee, Tor and Soo [66] focus on 180° elements–tents and v-folds–both of which they call v-folds. In fact, a v-fold is an angled form of a tent or more precisely, the angled tent is a variation of the v-fold as shown in Figure 4.15. Angled tents were not included in the taxonomy of Section 4.1.2 for that reason.

The authors present an informal proof that either all four folds or seams of these elements must be parallel (the tent), or all must meet at a single point (the v-fold/angled tent) in order to produce a foldable element.[8] In addition, they explore the requirement that all attached pieces must be long enough that the page can completely open. They also develop equations for the intermediate locations of the planes of the pop-up as it opens and closes in order to animate the pop-up (see Section 4.2.3).

[8] It also can be shown that this restriction extends to the 90° elements (step, angled step and beak), although the authors do not do so. Their method of proof relies on the properties of planes bounded by pairwise intersection lines, and the geometry in question is identical in the case of a step and a tent, for instance.

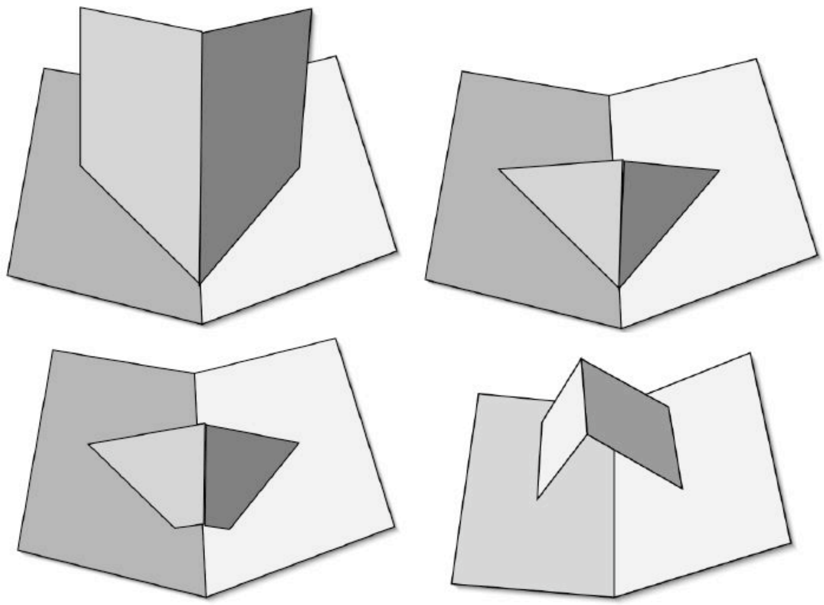

Figure 4.15: The Relationship Between V-fold, Angled Tent, and Parallel Tent: A v-fold (upper left) can be shortened (upper right), and the point trimmed to produce an angled tent (lower left). The angled tent is therefore a modified v-fold. The parallel tent (lower right) is the other possible form of a tent.

90° elements are the subject of a second paper by Tor, Mak and Lee [117]. Here they are concerned with the numbers of valley folds and mountain folds necessary for the elements to be flat-foldable, examining both angled and parallel forms. They present a set of relationships between the numbers of mountain and valley folds and establish that in order to be foldable, a 90° element must satisfy these relationships. However, the reverse is not true: an element can satisfy their relationships on fold numbers and still not be flat foldable, as other constraints may not be satisfied (the angles or distances of the folds from each other, for instance). This paper is much less useful to the designer of pop-up software, as the geometric constraints derived are a subset of those required to produce a foldable pop-up.

Most of the papers by Glassner investigate pop-up mathematics as a tool for animating the pop-up while it is being designed (see Section 4.2.3). However he does derive the major constraint on a beak [36]: that the angles between one outer fold and the parent fold and the other outer fold and the center fold must remain equal. This constraint is ubiquitous in the pop-up literature but this appears to be the sole proof published.

Groups working in the field of origamic architecture are primarily interested in the relationship of the 2-dimensional pattern to the 3-dimensional folded card in order to store the card in only one form and convert easily from one to another, and in the relationships of the moving points in order to animate the card. These relationships will be discussed in more detail in Section 4.2.3.

The papers from NTU are certainly important work in the area of the mathematics of pop-ups. They present informal proofs of constraints of both 90° and 180° elements as well as a discussion of methods of animation. In contrast, work by groups in origamic architecture and by Andrew Glassner explores the mathematics of pop-up elements primarily as a prerequisite for producing a software tool for pop-up design. That is, they are concerned with the points of elements in three dimensions as the pop-up is opened and closed, but not in the general mathematics of pop-up foldability, with the single exception of Glassner who presents a proof of the constraints on asymmetric beaks.

4.2.2 Computer Based Pop-up Design Tools

The only available software for the design of pop-ups is produced for the construction of origamic architecture. Origamic architecture (OA) encompasses a subset of pop-up elements, and has been developed by Masahiro Chatani [17] in a series of books of origamic architecture designs. Technically, origamic architecture cards are produced by cutting and folding one sheet of paper. It differs from traditional pop-up making in that there are no pieces added by gluing and therefore most origamic architecture consists entirely of 90° elements. Three examples of origamic architecture designs are shown in Figure 4.16. Buildings are a popular subject of origamic architecture cards, but lovely abstract designs can be produced as well.

Origamic architecture designs are simple in concept, but that simplicity can be deceptive. As Mitani says:

Figure 4.16: Examples of Origamic Architecture: Three designs by Masahiro Chatani. Clockwise from left are Rokuzan Museum, Todaiji Temple-Daibutsu-Den, and Wedding Cake. Picture courtesy Masahiro Chatani [17].

> Because of the limitations that the paper can be cut but not added to, designing OAs requires a great deal of experience. Traditionally they have been designed by a process of trial and error. [75, p. 93]

Computer enhancement of the design of origamic architecture is aimed at the adult hobbyist of this style and the production of single cards.

Mitani, Suzuki and Uno [76] provided the first description of software to aid the origamic architecture designer. The software described was called 3D Card Maker [77] and was later distributed by Tama Software as Pop-Up Card Designer[9] [113]. This is the only currently available commercial software for pop-ups, albeit for a limited subset of elements. In fact, the only pop-up element available is the step.

[9] Pop-Up Card Designer is available for Windows only at a current price of US$18.

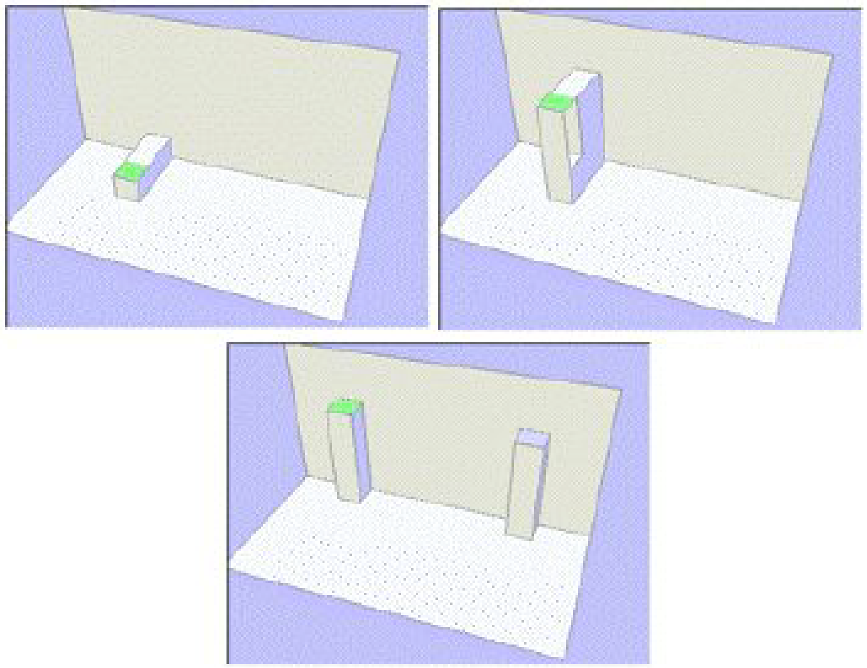

Figure 4.17: Making a Pop-up with Pop-Up Card Designer: The cursor (green square) is controlled with the arrow keys and space bar. A mirroring mode is available (below) for making balanced designs. Picture courtesy Tama Software [113].

The user interface for Pop-Up Card Designer is a simple one. As shown in Figure 4.17, the design is made on the 3D representation of the card. Keyboard keys are used to position the cursor and build the element. A mirroring function is provided as well, since origamic architecture designs are often symmetrical. The design can also include "windows" cut out of the vertical planes.

One limitation of Pop-Up Card Designer is that it only produces designs in which an edge is parallel to an axis. That is, all cuts are straight, and the design is built by stacking cubes.[10] Diagonal lines must be approximated with staggered-height cubes, and a design like the Todaiji Temple-Daibutsu-Den in Figure 4.16 cannot be made with this software as it contains curved cuts. In a later paper, Mitani and Suzuki [75] describe a method to overcome this restriction. As with the

[10] In the 2 dimensional form in which the design is stored, this equates to covering the design with squares.

Pop-Up Card Designer method, the process is simplified by origamic architecture requirements confining designs to a single sheet of paper without added pieces. This provides two primary simplifications in the software. First, there is a one-to-one mapping between points on the 2D sheet of paper and the 3D construction since all points exist on only one sheet. Second, every design can be defined on the paper sheet as a set of non-overlapping polygons.

To generalize from the original design using squares to the new design in which any combination of polygons can be used, Mitani and Suzuki propose a system in which the plane of the new polygon is overlaid on previous polygons The previous ones are cut away, creating a non-overlapping set of polygons that completely cover the paper. Since this can lead to a design that satisfies the condition of having no overlapping polygons but will not be foldable, a "test for pop-up condition" is used to allow the user to verify the design. This is a simple connectivity test, as the pieces must connect in order to be lifted by the opening process. The authors have created prototype software that implements this method.

There are several remaining issues with this method. For instance, the authors have not implemented the creation of curved cuts. Also, the interface is not as simple as that in the existing software, although it is incompletely described. In addition, it is not clear when the "test for pop-up condition" is made, or what the software does when the test fails. If the test is made on design completion it seems that it might be painful for the user to finish a design, then find that it does not open. If the test is made at each change operation, it may be intrusive to the design process. Moreover, the authors indicate that a successful test does not guarantee that the pop-up will open and they do not indicate how often or under what conditions it would fail to predict pop-up behavior.

In Chen and Zhang [18] and Zhang and Chen [130], this method is extended. These papers draw heavily from Mitani and Suzuki but simplify the user interface and incorporate smooth contoured faces via curved cuts.

The most complete software for the design of pop-ups is described by Glassner in a two-part paper [36, 37], a technical report [34], and a patent [35]. These four sources describe the same system, although each concentrates on different details. The software is not publicly available, although Glassner claims to have used it to design cards for personal use. The two-part IEEE article is required reading for anyone desiring to model pop-ups computationally, as Glassner is an experienced paper engineer who is familiar with the existing literature, range of elements, and problems involved in the computer generation of pop-up elements.

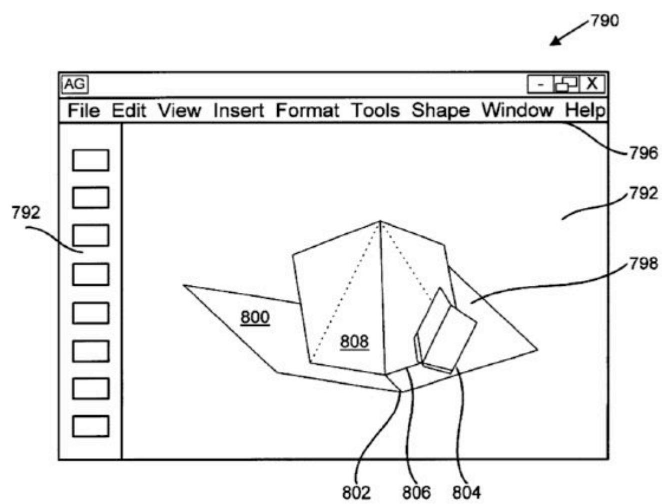

Figure 4.18: Pop-up Assistant Interface. Picture from the patent [35].

Glassner's focus is on the hobbyist who wants to make multiple copies of a single pop-up card, such as could be used for a party invitation for example, something that becomes time-consuming if the cards are made completely by hand. In addition, Glassner notes that some elements are difficult to make, and take a considerable amount of trial and error to produce–in particular he mentions v-folds where the designer desires a particular slant when opened, and his software is aimed at reducing the effort to make such elements.

Glassner describes a wide variety of pop-up elements that he has implemented, but excludes those that require bending (boats for example) as do the other authors in this section. He works with both 90° and 180° elements but only talks in depth about beaks and v-folds, using beaks as an example to develop the mathematics of his animation method (see Section 4.2.3).

Glassner has developed his *pop-up assistant* to enable the user to design cards using an array of elements, which include v-folds, beaks, tents, steps, moving arms, pull-tabs, and wheels. The interface is incompletely described, and as the software is not available, the reader is required to make inferences about how it might function. Figure 4.18 shows the interface as illustrated in the patent, which is the best illustration available. The pop-up assistant as described appears to manipulate the 3-D representation, with the elements added to the screen using drag-and-drop from a menu. The corners of elements appear to be equipped with handles to change their location, however Glassner also mentions that the user is able to specify the angle for a v-fold separately. The pop-up assistant checks the constraints for each element and modifies the card to preserve them:

> The designer may create mechanisms, open and close the card, and interactively drag points around; the rest of the mechanism is automatically adjusted as necessary. [34, p. 4]

The pop-up assistant also incorporates some useful error detection for the designer. Glassner locates colliding elements as well as elements that protrude from the cover when the card is closed.

The team at NTU, although their long-term intent is to produce software for professional pop-up publishing, has no software system for design as yet. The pop-ups used in animation tests are hard-coded into a graphics package, which allows them to produce an animation of the pop-up and images of the result.

Although two systems have been described in the literature, only one actual system is available and this system allows only the production of designs in the origamic architecture style, a limited subset of elements not including those used in professional pop-up books. Both systems allow manipulation of the elements, including changes to elements already placed, although Pop-Up Card Designer uses a keyboard interface while Glassner's interface appears to be primarily mouse-driven. In both cases, changes are made to the 3-dimensional representation of the card being designed.

4.2.3 Viewing the Operation of Virtual Pop-ups

One important part of any design system for pop-ups is some method for viewing the operation of the virtual pop-up by animating[11] the 3-dimensional representation. That this is necessary, or at least highly desirable, follows from the importance of both the motion of the design (the aesthetics) and the need to verify that the opening proceeds smoothly with no pieces colliding (the functionality). All of the researchers in the field have addressed this problem.

There are two parts to the problem. The first, and most easily solved, is that the change in the positions of elements on one level causes a corresponding change in the positions of elements on the levels above it when the pop-up is opened and closed. It is necessary to consider how these changes are propagated through all of the levels of the pop-up. Second, and the more difficult, is to determine exactly how the points defining each element move in 3-space.

In Lee, Tor and Soo [66], formulas for calculating the positions of the elements when the pop-up is partially open are derived trigonometrically. The authors consider only 180° elements in the form of tents and v-folds. In the case of the tent this is the calculation of the distance between the two side folds for a given angle of opening, while in the case of the v-fold, the formulas produce the angle between the pop-up and the page on each side fold. They also discuss the necessity for calculating the positions of elements on each level in order to provide input to those calculations for the next level. This requires the element information to be contained in a list of N elements, where N is the total number of elements in the pop-up with the elements added to the list starting with the lowest level. If calculations for a given angle of opening proceed in list order, the opened points can be calculated in O(N) time. Because the authors have no software for generating pop-up designs, examples of pop-ups are given pre-computed starting points and are displayed on the screen at various opening angles.

In origamic architecture the animation is easier, as there is a simple trigonometric relation-

[11] The word *animation* is the most accurate to describe this process. Wrensch [128] provides a good description of the problems with simulation and suggests the term animation to describe "lightweight simulation [128, pp. 96–99]" of the kind presented here.

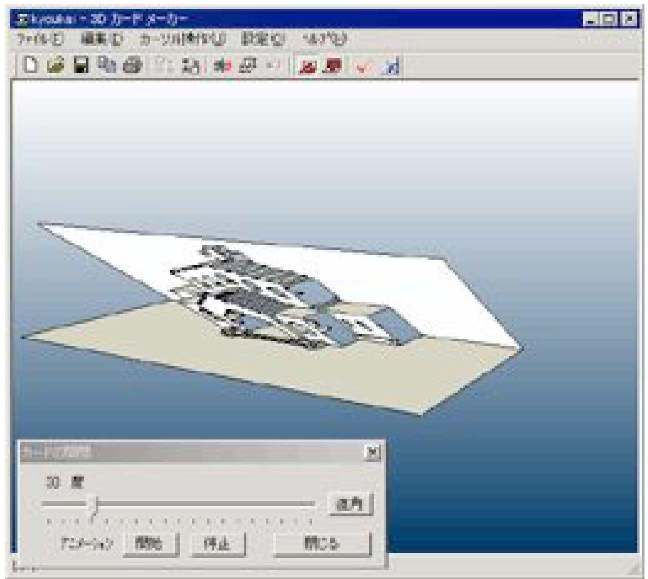

Figure 4.19: Animation of the Pop-up with Pop-Up Card Designer: A slider controls the opening angle. Picture courtesy Tama Software [113].

ship between a point on the pattern and its position in space as the paper is lifted, as described in Mitani and Suzuki [75]. If the point is stored with the values (x, y, z) when $\theta = 90°$, the final point in 3-space (X, Y, Z) for any other angle θ is $X = x$, $Y = y - z\cos\theta$, and $Z = z\sin\theta$. This is the method most likely used in Pop-up Card Designer where, once created, the design can be animated to show the opening and closing (see Figure 4.19). There is no enforced order in which to calculate the 3-space version of the points, as there are no dependencies between them when stored in the (x, y, z) form.

In Glassner [36, 37], the locations of the moving points during opening and closing are found using equations derived for intersection points of three spheres. This calculation is illustrated for the beak and the v-fold. Glassner states that other elements (including parallel elements that he mentions but does not describe in detail) can use the same method, but he does not provide an example of how this can be done. He uses, as do Lee, Tor and Soo [66], a list to keep the elements in order so as to process the lower before the upper. However, the statements:

> To process the risers [his element data structure], I start at the beginning of the list and look for a riser that can be positioned..I compute its points and mark it as positioned...When I reach the end of the list, if I positioned any riser on that pass, I go back to the start and go through the list again... If efficiency is an issue, you can preprocess the list and build a tree structure that you can later traverse in a single pass. [37, p. 75]

are somewhat puzzling, as a list with the elements added in order of construction should only need to be traversed once, as established by Lee, Tor, and Soo [66].

The subject of pop-up animation is an important one as witnessed by the fact that all the researchers involved have been concerned with the subject. Animation provides visual feedback to the user about the the pop-up, including the motion produced and the interaction, including collisions, among the elements. Paper engineering is a craft in which the final motion of the object is of paramount importance. Not surprisingly, among the three approaches, there are three solutions. The simple case is origamic architecture, in which the points can be calculated with a simple set of formulas in no particular order. For the more complex 180° elements NTU and Glassner employ a similar method, a linked list of elements, although the methods by which the points for each are calculated are derived differently. And in all cases, once the points defining the elements are calculated, they may be displayed by use of a graphics package.

4.2.4 Creating Physical Pop-ups from Virtual Representations

The two complete design systems examined have been those of Glassner and Pop-Up Card Designer. In these systems, the 3-dimensional representation of the pop-up is the one on which all additions and changes are made. Some method needs to exist to transform this representation into two dimensions so that construction of a physical pop-up can be undertaken. Neither system presents a 2-dimensional representation to the user during the design phase.

Figure 4.20 shows the original design page of a complex design and the pattern page produced from it in Pop-Up Card Designer. While the user can view either representation, she must switch between them. The pattern shows fold and cut marks in the usual form for origamic

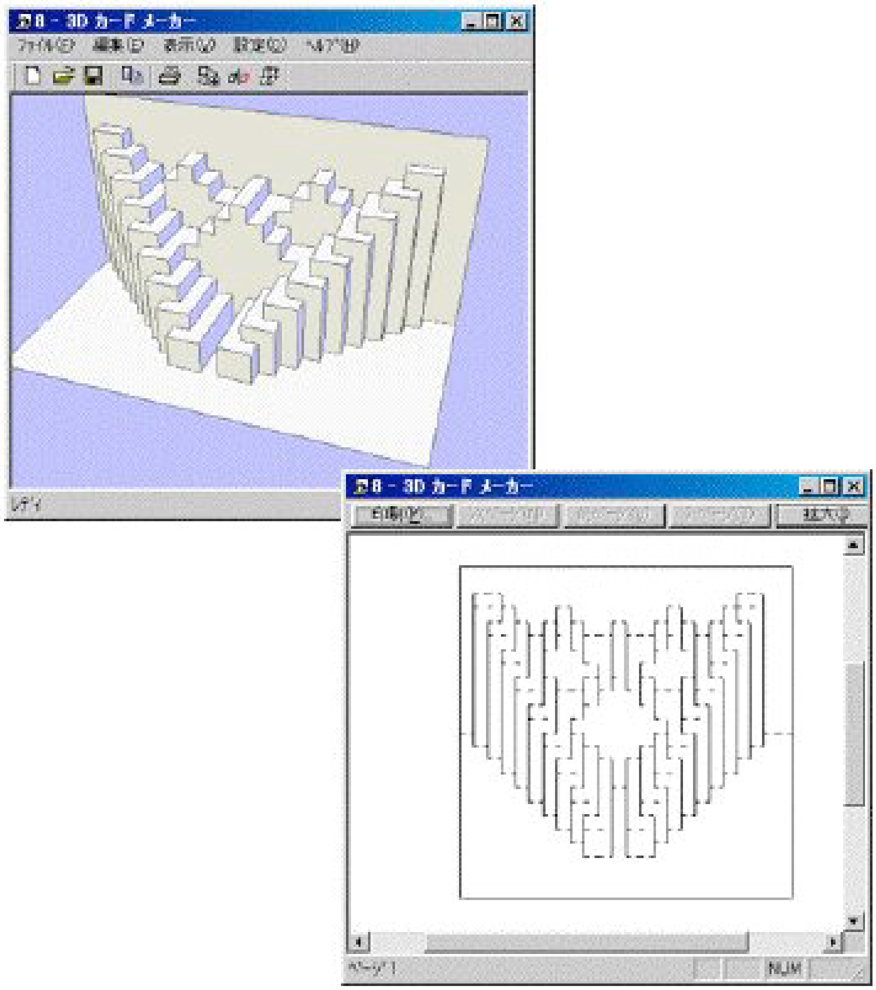

Figure 4.20: Printing a Final Pattern with Pop-Up Card Designer: After designing the pop-up, a pattern with cut and fold marks can be produced and printed. Picture courtesy Tama Software [113].

architecture patterns, with folds as dashed lines, and cuts as solid lines which are transferred to the printed sheet.

Glassner's system presumably acts in the same way, although no details are given. It seems likely that a window displaying the separate pieces to be printed is not displayed, since one is not shown. He does have one concern which does not arise in origamic architecture in that he must provide tabs for pieces that will be glued onto the pop-up, and guide lines to position the pieces.

Glassner also considers one additional detail that becomes important when printing many cards. He proposes a simple method to pack pieces on the page to reduce paper waste. Such algorithms are used for applications in clothing manufacture where patterns for cutting must be packed onto the smallest amount of cloth. (See Milenkovic, Daniels, and Li [74], for instance). Glassner places bounding rectangles around each piece and uses a greedy algorithm to position the rectangles onto the paper.

NTU has not yet reported developing a system which is capable of actually designing or producing physical pop-ups, rather their work has focused exclusively on the mathematics such a system would require.

Of the three research areas, the two which can produce physical pop-ups have chosen to present the pop-up in progress for editing in 3-dimensional form and either show the pattern separately for printing, or simply print the resulting pattern. Having the pattern (the notational representation) displayed with the 3-dimensional representation would seem to be particularly useful for the learner. However, even the practiced maker of pop-ups could benefit from seeing the 2- and 3-dimensional forms change together as the pop-up is designed or decorated.

4.3 Introduction to a System for Children's Use in Pop-up Crafts

The research discussed in the previous section focuses both on mathematical and theoretical work on computational enhancement of paper engineering, as well as actual systems. But for the purposes of this research, there are some things that have not been examined in this literature. First, children are never considered as users of these systems which are designed with adults, either hobbyists or professional paper engineers, in mind. Second, the systems described or envisioned are not seen as learning tools. The users would be employing these systems to design pop-ups on a continuing basis, not learning how to make pop-ups without the system.

There are, however, important similarities among these works that are useful to consider for the design of a system for children's pop-up construction learning. All of the researchers discuss the importance of presenting a 3-dimensional view of a pop-up in progress, and of animating this view in order to show the motion produced. All have considered pop-up elements as the basis for their design system, although they differ on which elements they feel are most important. Finally, all propose to produce output that can be used to make the final pop-up in physical form.

A framework was previously established to investigate the acquisition of craft ability. This framework was used in Chapter 2 to explore the use of computers in providing tools for learning craft in general and in Chapter 3 for a similar exploration for learning paper engineering. This framework can also serve as a guide to the design of software for teaching the craft of pop-up making by helping to focus on ways that knowledge, skill and appreciation of pop-ups can be enhanced through the use of software.

Some general principles should be recalled here. First, the desired operation of the software is a great deal like that of the toy looms used by children of Mayan weavers. The software should help the learner bypass some of the more difficult tasks involved in the design of pop-ups, in particular the geometric constraints required for the page's opening and closing. Second, the software should be based on the actual practice of paper engineering. For instance, elements are at the heart of the pop-up design process and should be introduced explicitly. In addition, the notation should be as standard as possible. Finally, the software should be simple to use. It should not present barriers of its own, but be an easy guide to the craft of paper engineering.

4.3.1 Supporting the Growth of Knowledge

Section 2.4 mentions several ways in which computer software can aid in the acquisition of knowledge. The first was to enforce standards so that it is easier to design a working item, thus reducing frustration. This suggests that it would be helpful to design a pop-up system where the geometric constraints on elements are always enforced thereby making it more difficult to design a pop-up that will not open and close properly. Second, every craft has a notation, and the simultaneous presentation of both notation and final craft object can help one learn how the notation works and how to predict the form of the final object from the notation. For this reason, it makes sense to present both the notational representation (also called the pattern) and the 3-dimensional representations of the virtual pop-up to children, not just the 3-dimensional representation alone. In addition, since a paper engineer commonly works on the pattern sheet of paper directly and not a 3-dimensional form, it also makes sense to have the user draw elements directly on the pattern in preparation for a time when she is no longer using the tool. Third, it is important to be able to experiment quickly with many designs. For this reason the user should be able to change and remove elements easily. Fourth, vocabulary learning can be aided through the use of standard vocabulary in the software and documentation. It has been noted that the names used for elements are not standardized across the domain. The taxonomies in this chapter (Figures 4.1 and 4.5) use names that are meaningful and child-friendly: *step* rather than *90° double slit* for example. It is also important that these names be used consistently in help tools, the interface, and the documentation. This promotes learning a standard vocabulary from the beginning and improves understanding. Finally, a useful set of elements should be included. However, these should be a basic set, leaving the user room in which to experiment with variations of these as well as other elements on her own.

4.3.2 Supporting the Growth of Skill

The desire to have an incomplete set of elements present, so as to force the user to modify and innovate, is also an example of an aid to skill development. By offering the most commonly used elements but not restricting the user solely to them, the user is encouraged to make changes to the elements and to add elements without the use of the software.

Section 2.4.2 cautioned that computational enhancement of a craft can take too much away from users in terms of actual fabrication practice. For this reason, the actual construction of the pop-up, including cutting, folding and gluing, should be accomplished by the user. To that end, it is also advisable to have the documentation cover some of the details on how to assemble final pop-ups, as children may be working without adult or experienced help. Flagging of errors, if it is included, could include collision detection as in Glassner's pop-up assistant. This is another way in which the computer can easily perform a task that would be otherwise impossible until the time that physical construction occurs.

4.3.3 Supporting the Growth of Appreciation

The growth of appreciation is linked to the opportunity to see the designs of others and compare them to one's own. This part of craft learning is harder to support computationally than the other competencies, but several design decisions can be made to allow these opportunities.

Such facilitation is promoted by the ability to share designs among users of the software. For example, printed copies of the designs can not only be used to construct the pop-up, but can be shared among users. This ability to print multiple copies allows several children the opportunity to own a copy of a book that they have designed together. In another possible scenario, a simple pop-up produced in multiple copies for a group of children can be decorated and varied in numerous ways by each, with the addition of color, craft decorations such as sequins, or additional pieces of paper. A simple pop-up could also be made to tell a different story by each child. The ability to print multiple copies of a pop-up therefore becomes an aid to the acquisition of appreciation. Johnson [58] uses lessons such as these frequently, for example, his exercise on a step element (that he calls a *basic box*) as described in Section 3.3.2.

Another method of sharing designs is provided by writing files from the software that allow the user to save her work. This not only allows the user to return to the design later, but to share it with others who have the software and want to make their own changes to it. The user is therefore able to use the designs of others directly, or to revisit her previous designs to improve them. It is important that this format produces files that are small in size for easy transmission. It is also useful to have a format which can be parsed without the help of the software making a text-based format desirable.

Yet another way of sharing designs is by creating pictures of the pop-up pages. These can not only be sent to others who may not have the program, but can be imported into graphical tools to add textures, photos, or other decorations.

To further support the growth of appreciation, the documentation for the software can be an important adjunct to the software itself as it can include places the user can go for more information, pictures of other pop-ups, and tips on how to alter and decorate pop-ups. This provides the learner with other sources, in particular on-line sources, for ideas and pictures of pop-ups by other artists.

4.3.4 Other Design Considerations

There are a few design considerations which may be gleaned from the research in this area but are hard to place in a particular competency.

When printing the final copy to construct the pop-up, Glassner makes a good case for putting as many pieces on one sheet as possible to decrease paper waste, and suggests a possible simple algorithm for doing so. However, Glassner is designing software for the use of the hobbyist making multiple copies of each pop-up, a case in which a great deal of wasted paper could result. For the young learner of the craft of paper engineering it seems likely that only one copy of a given design will be made at any time. Paper waste would be minimal. Even if multiple pieces are placed on one page, it might be best to have the option for printing one piece per page in order to allow printing of a given piece on differently colored paper, or to easily replace a piece damaged in the cutting process.

A mirroring function is provided in Pop-Up Card Designer. This is particularly useful in the origamic architecture style as such designs are often symmetric. In addition, this is a task that is easy for the computer to provide and difficult for the paper engineer to accurately produce and could be a useful addition to such a system.

4.4 Summary

In order to come to an understanding of the ways in which computation can be added to pop-up design, it is necessary to understand how paper engineers construct these objects with an eye toward mimicking those methods. Pop-ups and other movable forms are composed of elements that can be combined and modified to produce the desired results.

Pop-up elements are a subset of movable book elements, and can be divided into 90° elements that are best displayed when the page is half-opened, and 180° elements that are shown to best effect on a fully-opened page and most professional pop-ups are composed of 180° elements. Elements can further be divided into angled and parallel forms depending on their relationships with their parent fold (the underlying fold over which the element sits).

Elements can be transformed by the addition of attached planes. Each type of element has its own set of geometric constraints that must be met to allow it to fold properly when the page is opened and closed. These constraints continue to hold when elements are placed on top of one another and this poses a number of problems when attempting to animate such structures in software.

The literature on the computational enhancement of the animation and design of pop-ups focuses primarily on mathematical descriptions of pop-ups and work on simulating the opening and closing of pop-up elements. The main problem, determining the locations of the points of a pop-up at any degree of opening, has been approached mathematically both trigonometrically and by calculating the intersection of three spheres. It has been established that a list of elements can produce these locations in O(N) time, where N is the number of elements.

Several pieces of design software have been produced, only one of which is available to the public. This program, Pop-Up Card Designer, is easy to use but is not designed as a learning tool for children and designs only a limited set of pop-ups in the style called origamic architecture (OA). Origamic architecture consists of single-sheet pop-ups that are produced by cutting and folding, but not gluing on additional pieces. Andrew Glassner has designed a more flexible system, but it is only minimally described and not available. All systems so far rely on the user making changes on the 3D representation of the pop-up, and none are aimed at children learning the craft.

Using the framework described in Chapter 2, the research results from Section 4.2, and the analysis of paper engineering as a craft as seen in Section 3.2, general rules for the design of pop-up design software for children can be established. First, it is desirable to have the editing done on the 2D pattern, rather than on the 3D representation to facilitate learning about the notation of paper engineering, and to mimic the design methods of most paper engineers. The interface should be kept as simple as possible, and unneeded features should not be included. Part of the purpose of these choices is to force the user to develop skills of folding, cutting, gluing, altering the elements into new shapes, and decorating. Elements included should be common, versatile, and provide a range of types: 90°, 180°, angled and parallel. In addition, users should be able to share designs.

Now that the basic criteria used in the design of such a tool have been established, Chapter 5 provides the details of the design of Popup Workshop and the implementation of the program.