Patterns express generic solutions to a well-described problem. In parametric modeling patterns can be used to describe a “tactical” level of work, above mechanics and below design. We describe three patterns and show how they can be used through an example model.
1. PATTERNS
The term “design pattern” originated with Christopher Alexander (1979) to describe an established architectural configuration and the related explanation of its development and affordances. Among its antecedents are Palladio’s “Il Quattro Libri dell’Architettura” and numerous Victorian design and construction pattern books. There is a large literature on patterns and several good review papers. The term has been adopted by the software engineering community to describe abstract, well-established forms of program construction. Patterns express design work at a tactical level, above simple editing and below overall conception. A pattern typically comprises a name, a problem description, an abstract solution, and a discussion of consequences.

We are developing patterns to help designers learn and use propagation-based parametric modeling systems. We have largely focused on the Generative Components system as this allows us to access a large group of designers who are currently learning both the system and the computational concepts underlying propagation-based systems. While we expect that our results could generalize to other parametric modelers, such application has yet to be done.

A propagation-based system represents a design state as an acyclic directed graph and two algorithms, one for ordering the graph, and one for propagating values through the graph. The nodes of the graph are schemata, that is, they are objects containing variables and constraints amongst the variables. The constraints are realized in an update algorithm that computes values for the dependent variables of a node. A graph models a usually infinite collection of instances, each of which is determined by assigning values to the independent variables of the graph. To compute an instance, graphs are ordered and then values are propagated through the graph. Both algorithms are simple and efficient, enabling interaction with large models. A typical system provides a limited set of update algorithms for nodes from which users must compose designs. Users often spend considerable time composing models with a small number of nodes that perform specific tasks. They use these small models as components in larger designs. We intend that our patterns capture these acts of authorship, above nodes but below designs. Patterns appear to have utility in learning. We have taught parametric modeling to several hundred professionals and graduate students. Over time we noticed that our instruction has increasingly focused on this tactical level. We have begun to use patterns as explicit elements of learning.

2. EXAMPLE PATTERNS
A pattern is a generic solution to a well-described problem. Its description includes the problem and solution, as well as other contextual information. Patterns have become a common device in explaining systems and design situations (Week 2002; Tidwell 2005; Evitts 2000; van Duyne et al. 2002; Gamma et al. 1995). Authors express patterns in different ways. Here we adopt Tidwell’s Use the internal structure of the JIG to capture its direct and self-explanatory style comprising Title, What, Use When, Why, How and Examples (Tidwell 2005). The Title should be a brief and memorable name by which the pattern will be known. What uses an imperative voice to describe how to put the pattern into action. Use When provides context needed to recognize when the pattern might be applicable. Why gives motivation for using the pattern and outlines the benefits that accrue to its use. How gives the mechanics of the pattern. For us, a distinguishing feature of a pattern is that it has an explanation of mechanism, that is, all instances of the pattern have similar symbolic structure. Samples provide concrete instances of the necessarily abstract pattern descriptions.

The following example patterns apply to the description of a hypothetical roof following a curved centre-line over a variable landscape. More patterns can be found at www.designpatterns.ca

2.1. JIG
What. Build simple abstract frameworks to isolate structure and location from geometric detail.

Use when. You are developing a complex model that will be controlled by a set of parameters. This pattern really is this general! Trying to build complexity and parametric control without a jig usually leads to confusion.

Why. Most models contain many elements and few controls. The logic of the controls is typically simple. A JIG is a simplified model that allows you to understand how the controls work without the distracting detail and slow interaction implied by a larger model. JIGs help make complexity understandable: by isolating the essential logic of a model into a small number of elements, they help both you and future users author and analyze a model. JIGs can be changed easily compared to complex models. Since the propagation algorithm must visit every node of a model, a JIG, being simple, can increase the interactivity of a model as it is being authored. Once developed, a JIG can be reused in other contexts, but only if it can be isolated from the rest of the model.

How. A JIG should appear and behave like a simplified version of your end goal. A physical example is the strongback and stations used in building a small boat. The stations locate and support the hull when it is being
constructed. Fairing, the process of making the hull smooth and continuous, can be done much more simply with stations than with a complete hull. JIGs are like construction lines in that they help locate elements. They are unlike such lines in that they are linked to the controls that enliven the parametric model.

To make a JIG, you have to understand the parametric behaviour you want and how the JIG will be used to define the complete model. A good JIG typically has relatively few geometric inputs (for example, points, lines, planes, coordinate systems) and each of these is carefully named. The small number of geometric inputs allows you to easily locate the JIG. The names are the primary means by which you will understand the JIG when you (or someone else) reuse it in the future.

Use the internal structure of the JIG to capture intended logical behaviour. For example, if depth of a truss is proportional to its span, a JIG might comprise a line and a variable whose value is proportional to the length of the line. Samples are given in figures 1 and 2.

2.2. ORGANIZED COLLECTION OF POINTS

What. Organize collections of point-like objects to locate repeating elements.

Use when. Most designed artifacts have repeating elements. These may vary by both their absolute position and their spatial relationships with nearby repeating elements. Use this pattern when you are able to think about the size and location of repeating elements in terms of a set of defining points.

Why. A collection of points organized to capture the intended spatial relationships can greatly simplify the process of further model development. This saves time and effort in both modeling and in the reuse of a model in new contexts.

How. Make a collection in either Euclidean or parametric space. Point-like objects may be located in Euclidean space or parametric space. Euclidean space is the familiar space of everyday life. It can be represented through Cartesian, polar or spherical coordinates. Most curves and surfaces (those defined internally by parametric equations) also define a “coordinate system” that defines locations on the curve or surface. Unlike those of Cartesian space, these parametric formulations may not preserve constant distance, either geometrically or along the defining object. Use this collection as the input to define repeating elements.
The logical structure of a collection is important—it provides the relationships through which points can be used to define objects. For instance, a collection structured as a 2-D array provides, for each point $P_{ij}$, easy access to the points $P_{gh}$ where $g \in \{i-1, i, i+1\}$ and $h \in \{j-1, j, j+1\}$. A collection structured as a tree provides for each point $P$, easy access to $\text{parent}(P)$ and $\text{children}(P)$. Samples are given in figures 3 and 4.

2.3. PLACEHOLDER

What. Use proxy objects to organize complex inputs when making collections.

Use when. You are defining collections of complex objects and each object in a collection requires multiple inputs. You can describe the multiple inputs to an object through a single abstract object.

Why. You often want to array a complex module across a surface or along a set of curves. If this module requires multiple point-like inputs themselves defined on the target, organizing these inputs is complex and error prone. On the other hand, if you can define the inputs to the complex module through a simple construct such as a polygon, it is often much easier to arrange polygons on the target and then use them to place the complex module.

How. The mechanism for this pattern has two parts. The first is the proxy object itself: a simple object that carries the inputs for the module. For example, a rectangular module may require four input points, one for each corner. A four-sided polygon can act as a proxy for these points: each of the vertices of the polygon provides one of the points. The proxy simplifies the arguments needed for the module: instead of four points, you only need one polygon. The second part of the mechanism is to relate the proxy object to the model. For example, a polygon proxy can be placed using a rectangular array of points by relating the polygon’s vertices to the points $P_i$, $P_{i+1}$, $P_{i+1}$, and $P_{i+1}$. The code required to place a generic object such as a polygon is more simple and reusable than the code for a specific module. A sample is given in figure 5.

FIGURE 3 Geometric (a) and parametric space (b)

FIGURE 4 A parametric diagrid is an example of an ORGANIZED COLLECTION OF POINTS. It describes a collection of points rotated in parametric space.

FIGURE 5 PLACEHOLDERS. (a) Curves as PLACEHOLDERS for struts. (b) Shapes as PLACEHOLDERS for covering surfaces.
3. AN EXAMPLE
We present the use of patterns through an exercise for an abstract roof (Figure 6) originally developed by one of us (Kilian) for a workshop taught in 2005. It has since served as the lead introduction to many introductory tutorial sessions in Generative Components. It combines several key parametric modeling concepts and quickly yields a form with some architectural credibility. Many versions exist, and the one presented here has been edited for simplicity and clarity. It is typically created in stages: curve, abstract frame, surface, point array, shape array, struts, feature compilation and feature array. These correspond to (and are partly the source of) some of the patterns presented above. It typically requires an entire morning session at the start of an introductory Generative Components tutorial to demonstrate and have students create a version of this model.

The instructional intent is to introduce several core associative parametric modeling principles and to show how they can be linked to create a whole. The key goal is to extend the notion of parametric constructs beyond simply describing form to that of interdependent design constructs that support exploration of possibilities. The objective is to motivate participants to make the investment of learning and effort in order to gain the benefits of design insight and discovery.

The exercise is structured into several steps with increasing complexity and increasing dependencies between simple constructs. Over time, we have begun to generalize the constructs in the exercise into patterns. It is clear to us that patterns do not suffice for either learning or modeling. If they are to work, it seems that they must do so in combination with clarity of explicit composition—the parts of the model and at least the abstract steps required to create them must also be described. Further, these parts and steps must be explained abstractly: a surface to carry a collection rather than this B-spline surface that represents a roof and is sectioned by this array of planes.

In the multi-page Figure 7 we present both the original intent (labeled Composition) of the exercise and note (using the label Pattern) where patterns may apply to generalize both explanation and model.

Composition: Define the site. It starts simply by placing unrelated points, using x, y, and z parameters. A curve using the points as control points introduces associativity in a basic form. The lesson here is the reduction of an otherwise very complex geometry of a landscape to abstract easy-to-manipulate geometric entities that contain sufficient information to act as a stand-in.

Pattern: A Jig. Four points define the control polygon for a curve that represents the centre-line of the roof as it would lie on the site.

Composition: Define the location(s) for placing Jigs. Placing a plane by parameter normal to the curve introduces the concept of parametric space. Its intersection with another plane parallel to the global xy-plane defines the direction for the roof support. The direction models...
the intent of a baseline horizontal and perpendicular to the path. The key concept here is that modeling must often be the result of combining several independent entities.

Pattern: This and the next figure comprise a second JIG. A pair of planes define the base on which a JIG will be built.

Composition: A simple T sectional rig defines the roof cross-section. The T rig relates its equal length sides to the height of the rig. The height is defined by a vertical line that starts where the JIG is located and ends at a global datum for the ridget of the roof. The result is that each section of the roof is geometrically similar and the overall roof widens as its height increases.

Pattern: JIG (concluded). The JIG comprises a vertical line from the curve to a given global elevation and two outriggers with lengths proportional to the length of the vertical line.

Composition: Replication of the parametric element along the path line demonstrates a key concept of robust, reusable element definition; each instance is geometrically similar to all others.

Pattern: The JIG is REPLICATED along the curve at regular parametric intervals (REPLICATION is another pattern. Due to page length restrictions, we do not describe it here).

Composition: The roof surface becomes the “site” on which further modeling occurs. Once the surface is in place, the modeler need no longer attend to the prior structure of the JIG.

Pattern: ORGANIZED COLLECTION OF POINTS. The abstract roof surface uses the endpoints of the JIG lines as its control net. Sections through the surface perpendicular to the centre-line inherit the geometric similarity of the JIG.
**Composition:** This and the following four figures document a PLACEHOLDER. The structure (and steps) are to place a collection of point for arraying the eventual proxy object, to define the proxy object and the module that uses it, to array the proxy object across the point collection and to instance the module using the proxy array.

**Pattern:** An Organized Collection of Points defines the geometry of the eventual roof structure.

**Composition:** A shape uses the four points as input. The structural module comprises two cones (with constant radii) using the shape (not the underlying points) as their inputs.

**Pattern:** PLACEHOLDER (continued). The indirection provided by the shape will be crucial to arraying the module across the roof.

**Composition:** Built on top of the roof surface, a collection of points will form the basis for further modeling.

**Pattern:** PLACEHOLDER (continued). Four points form the base for the structural module that will cover the roof.

**Composition:** Shapes are defined on every cell of the array of points on the roof. This sets the stage for applying the module across the roof.

**Pattern:** PLACEHOLDER (continued). A PLACEHOLDER requires that definition and target be of the same kind of object.
Composition: The module is arrayed using these shapes as its input. One module now fits automatically into many contexts.

Pattern: Placeholder (concluded). One definition, mediated through a placeholder, can yield multiple parametric realizations.

Using patterns, the model can be easily transformed. By replacing the instances of the Jig, the Organized Collection of Points on the surface and the Placeholder, the roof can be radically changed, yet the model structure remains largely unaffected. In such transformations, patterns act as informal objects—they provide a structure whereby edits can be more readily understood than by examining an undifferentiated complex model. They are unlike objects in the sense that patterns under-specify mechanism—they show approaches to problems, rather than specific data-structures and algorithms for solving them.

4. THE FUTURE

The goal of teaching parametric design through patterns is to identify approaches to structuring design problems in a general way so that they can be adapted to any design problem. Each representation has its limitations and current implementations of parametric software are no exception. In many ways, parametric modeling environments are more rigid than traditional digital modeling applications. But there can be synergy in combining design patterns to yield elegant and robust design variations that would not be otherwise achievable. Patterns are a way to identify successful general strategies that exemplify a key concept in a memorable fashion that can easily be taught. A pattern can stand as a synecdoche for more complex concepts necessary for real life design scenarios. Good examples ideally combine several simple patterns to demonstrate the next level of complexity in capturing design in a parametric and associative way.

To date we have used the concept of patterns as a heuristic around which to structure both modeling and instruction. Like many pattern authors, we have not accounted for how our patterns are developed, nor for their external validity. The companion paper (Qian, Chen, and Woodbury 2007) describes use of a participant observer research method to discern the validity of the pattern approach and to account for the origins and applicability of specific patterns.

5. ACKNOWLEDGEMENTS

This work was partially supported by the National Sciences and Engineering Research Council of Canada, the Networks of Centres of Excellence Program of Canada, and BCcampus.

REFERENCES


