B-Shelves: A Web Based Mass Customized Product

Lauren M. Johnson

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Architecture

University of Washington
2012

Committee:
Brian R. Johnson
Kimo Griggs

Program Authorized to Offer Degree:

Department of Architecture

University of Washington

ABSTRACT

B-Shelves: A Web Based Mass Customized Product

Lauren M. Johnson

Chair of the Supervisory Committee:
Professor Brian R. Johnson
Department of Architecture

B-Shelves (Box-Shelves) explores the relationship between design intent, generative design, mass customization and digital fabrication. B-Shelves is a mass custom furniture product that is customized through a web based system. The project looks at generative design as an enabler for different design and manufacturing processes and different designer/consumer relationships. A process of mass customization, as described by the B-Shelves thesis, is able to support consumers who are interested in customizing a design through a guided process. The design-to-manufacture process of many customizers (the consumers) to many products has shifted the traditional role of the architect. In this case, the architect must consider the design of multiple end products and focus on a framework that is able to generate both explored designs as well as numerous other customized versions.

The B-Shelves project explores three parts of this mass custom process: how form is generated (including codifying the architect's knowledge base in a parametric framework), how form is customized (using an interactive web interface), and how numerous unique forms are fabricated (using efficiencies of mass production). Each part of the file-to-factory process is discussed in depth, as is the construction of a built prototype.

TABLE OF CONTENTS

LIST OF FIGURES

AKNOWLEGEMENTS

The author wishes to express gratitude for all of the support and guidance received throughout the thesis. A special thanks is given to the thesis chair, Brian R. Johnson, and committee member, Kimo Griggs. This research would not have been possible without the support of the Design Machine Group [DMG], friends and family.

INTRODUCTION

Introduction

The B-Shelves (Box-Shelves) thesis explores the role of the architect and the possibility for the role to shift as the designer/ client relationship diversifies. Today both designers and consumers are thinking and acting in new ways, in part due to the current digital mindset and available computing resources. Computing resources, both in the form of web based communication and design and manufacturing processes are cheaper and more accessible than ever before.

Today, consumers have a great interest in personal choice and, with the rise of the DIY (do it yourself) culture, they are involved in processes of making. This is visible through custom printed tee shirts, HGTV, Food Network, Etsy.com, maker spaces, hacker spaces etc.(Ganter et al. 2011; Gilmore and Pine, 2000). Architects are also designing in new ways, specifically utilizing parametric modeling and generative design methods.

This thesis addresses the possible shifts in consumer and designer behavior by developing a design framework from which many consumer customizable products can be created. B-Shelves is a widely customizable file to factory shelving product.

1. ROLE OF ARCHITECT OVER TIME

Medieval Architect

In order to introduce the evolutionary role of the architect, a brief history will describe the architect's influence over time. The role of the medieval architect was closely linked to the role of the builder. Architects guided design decision making on-site (fig 1.1). Given the tool set they had to work with, physical models and tangible building blocks, their method of working was closely tied to the actual method of construction. The architect was aware of the material properties and assembly processes needed for building construction. Design was carried out with specific building components in mind. This awareness made for a more direct transfer of ideas into built artifact (Schodek et al., 2004).

Fig. 1.1 Medieval architect builder

Renaissance Architect

With the invention of the orthographic drawing, the role of the architect shifted. In the Renaissance the architect became more removed from the building site and design intent began to be communicated through abstract drawings. The role and interests of the Renaissance architect was tied to individual knowledge and artistic exploration. Concepts of interest such as ideal proportions and symbolism were able to be described in the architectural plans.

In the design process facilitated through orthographic drawings, designers' thoughts could be transferred onto paper in a process that provided quicker feedback than building physical models or prototypes (Kalay, 2004; Kolarevic, 2008). On paper, the architect is able to evaluate and revise the design. The simplification of measured drawings and architectural models teaches and guides the designer toward decision making as well as permits him to ignore details that do not have to be designed at a particular time. Scale is a guiding factor to what level of building detail is designed at what stage of the design process. The physical task of redrawing a plan at a larger scale (for instance, increasing the scale of the representational drawing from 1/8" = 1'-0" to 1/4" = 1'-0") causes the architect to both reevaluate the overarching decisions made at the original scale as well as design finer details appropriate for the new scale.

20th Century Architect

For much of the Twentieth Century, the role of the architect continued to be to design buildings and draw orthographic plans that were then delivered to the builders. Yet, a higher level of coordination and understanding of complex design requirements became part of the job description. Architects collaborated with engineers and consultants for their expertise. The general understanding of architectural concepts also became more complex as an architect typically paid attention to: high rise construction, circulation, egress, active and passive heating and cooling strategies, environmentally efficient construction, etc.

Role of Architect Today

Today the role of the architect is capable of shifting again. Two potential shifts are identified. In one scenario, the role of the architect is tied to an understanding of material properties and processes of making, such as has not been the case since the Renaissance. Today the link to material properties is experienced through digital means including Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) process and digital fabrication (Corser, 2010; Iwamoto, 2009; Kolarevic, 2003; Perez, 2008 Schodek et al, 2004). In another scenario, the architect's role is able to shift towards designing systems instead of artifacts (Alexander et al., 1977; Barros et al. 2011; Flat Clock, 2012; Huang and Krawczyk, 2007; Nordin et al. 2010; NikeiD, 2011). It is possible for a designer today to take part in both of these transitions. Parametrics and quick digital modeling and visualization, together with digital manufacturing, are changing the way architects do design as well as their possible role (Archea, 1987; Kieran and Timberlake, 2003).

Computing has revolutionized the design process. Architects are able to develop and visualize design schemes more rapidly than was possible with only pencil and paper (Kalay, 2004). Digital three dimensional modeling has allowed architects to model buildings, perhaps focusing on elements that would traditionally be associated with plan or section, and then instantly view the building in perspective. All aspects of the design representation are in one place. For example, shifting a wall six inches to the left while modeling in plan view will effectively change the wall placement in elevation and perspective views also; there is only one geometry model. Because of this, digital 3D models can be built upon in a steady progression. There is no longer a need to redraw the entire plan, section, etc. each time a change must be implemented or the building is to be visualized at a different scale. In addition, the ability to either array an object or create an instance of an object, significantly speeds up the time it takes to construct an architectural model.

Computing's contribution to the design field is not solely in the form of visual representation. Alternate methods of designing are facilitated with computing. Parametrics, for example, enables multiple design iterations to be visualized and analyzed almost instantly. A parametric process alters the schedule of the design phase. Instead of designing broadly at the beginning and honing in on details as the project is able to be visualized at progressively larger scales, details can be acknowledged from a very early phase and expressed as a range of possible parameters. Early in the design process, much time is devoted to develop the parametric framework, yet once it is developed a large variety of iterations can be generated (Barros et al. 2011; Kolarevic, 2003; Terzidis, 2006). Given a functional parametric model, many design schemes can be explored in a short amount of time and a good understanding of the specific design problem can be gleaned from the exploration of its various functional and formal possibilities.

2. SHIFTED ROLE OF ARCHITECT

Shifted Role of Architect

The thesis is focused on the way in which the architect is capable of shifting his role from the designer of one-off artifacts, to the designer of systems for the production of many artifacts. In the design of systems, the architect is concerned with experiential knowledge of material properties and construction processes. A design scenario is explored for the B-Shelves thesis in which a consumer is able to customize a product to their specifications through a guided framework (fig 2.1). The framework is developed by the architect and is an embodiment of their design ideals for the particular customizable product.

USER CUSTOMIZATION – FRAMEWORK – CUSTOMIZED PRODUCTS

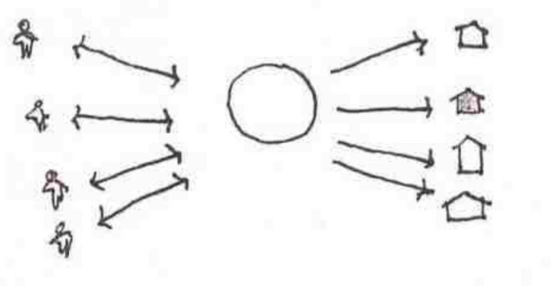


Fig. 2.1 Consumer customization framework

Existing and New Models of Manufacture

From a consumer's perspective, there are two traditional patterns of design and making, purchasable products can be either custom or mass produced. Custom products are considered desirable because specific individual needs are able to be met in the design of the product. For some, owning a custom furniture piece is able to offer a certain level of prestige because of the knowledge that there is no other entirely like it (Gilmore and Pine, 2000). Consumers are often willing to pay higher prices for a custom product because they know they are receiving a unique product that is designed and made with care. Yet commissioning a piece requires time as well as money; time for meeting with the designer, for the design itself, and for one-off making.

Mass produced products are appealing to the consumer because they are easy and relatively inexpensive to obtain. The look and quality of the finished object, as well as its price, is apparent in an off-the-shelf item. In addition, while the actual build quality of the product depends on the goals of the production company and the product itself, mass production offers the capability of uniform, high-quality fit and finish. Variety, however, is lost with industrial production (fig 2.2).

A process of mass customization aims to combine the variety of custom made products with the production efficiencies of mass production. While the definition of mass customization is in flux, it is often considered to be a reaction against mass production. Consumers are reacting to the fact that mass produced objects lack variety, but the alternative, custom made objects are both expensive and time consuming to obtain (Gilmore and Pine, 2000). Mass customization is not a new design and manufacturing process, but it has been made more feasible with the accessibility of computing and Computer Numerically Controlled (CNC) fabrication (Kieran and Timberlake, 2003). It does not claim that it will replace either custom or mass produced ways of making, rather that it will fill a gap in what is made available to consumers.

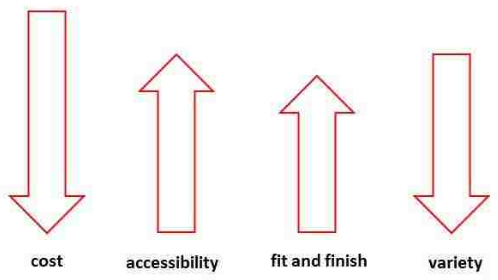


Fig. 2.2 Tradeoffs of mass manufacture

For the B-Shelves thesis, consumers have control over the customization of their own product. It is noted that consumer control is not always the norm with mass customized products, in some cases the designer is in charge of making the final decision regarding how each product varies from the next (Lynn, 1999). Each B-Shelves customized product shares commonalities with other B-Shelves customized products. In other words, every customized product can be thought of as a version of an expansive parametric design. The architect is able to identify what parts of the product make most sense to customize and where standard parts or standard fabrication processes can be used, if used at all.

Efficient manufacturing procedures of mass production, such as assembly lines and mechanization, make it possible to produce inexpensive, precision-made objects and do so quickly. In order for a mass customization process to output products faster and for less fabrication costs than a traditional custom process, procedures of mass production must be implemented (Davis, 1987; Kieran and Timberlake, 2003).

A parametric design framework, for B-Shelves or similar customizable products, should take into consideration the different types of manufacturing procedures for the efficient production of highly variable form. Certain fabrication operations allow for customization more easily than others. For instance, a CNC router is built to cut out any shape that can be described as a vector line, while a chop saw just cuts through material at a straight or mitered angle. In many situations, using the tool that has the most built-in functionality for primary customization procedures (the CNC router in this case) will result in a product with a high level of variability and an efficient manufacturing process (Kieran and Timberlake, 2003). For the thesis, I would argue that not every fabrication and assembly procedure within a process of mass customization must be mechanized; yet, an understanding and acknowledgement of process that is patterned after mass production is necessary to distinguish between mass customization and traditional custom manufacture.

Reducing total design and production time is part of creating an efficient mass customization process (Davis, 1987). Time is able to be cut at both the item's customization phase and its fabrication phase. At the customization phase, each consumer is able to make final design decisions (though manipulating the parametric model) on their own time and without the designers' physical presence. At the fabrication phase, fast manufacturing is possible with CNC tools that are indifferent to the variations between jobs. For example, different CAM files can be milled out of different pieces of stock material in approximately the same amount of time, as long as the tool and material properties remain the same.

Depending on the complexity of the product, it is possible to achieve further time savings by simultaneously manufacturing different parts in different factories. This is an approach taken by the car and computer industries in facilitating customization (Kieran and Timberlake, 2003; Schodek, 2004). In this scenario, the manufacturing process is able to take advantage of factories that specialize in specific techniques (powder coating, welding, building electronic parts, etc.) to speed up the overall fabrication time and utilize mass production assembly line techniques where applicable. Finally, a file-to-factory process is capable of diminishing lag time between manufacturing stages. Less time can be spent between fabrication tasks because the sequence and strategies of operations has been prototyped and predetermined.

3. DEVELOPING A FRAMEWORK FOR CUSTOMIZATION

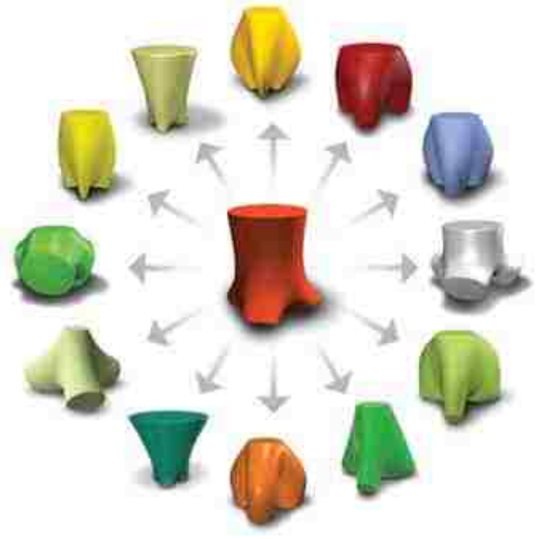


Fig. 3.1 Common framework enables custom product design and manufacture

Designing for User Customization

This chapter addresses what is needed in order for a designer deliver a product that is made for end-user customization.

In order for the parametric framework to support the semi-automated production of customized products, explicit rules that encompass the designer's intent must be described in the system (Alexander, 1964; Davis, 1987; Duarte, 2006). The framework can be set up to generate solutions that, for the most part, fall within a range of solutions that the designer deems fit (Simon, 1969). The solution set may be similar to the type of designs that the architect would create if he were to explore numerous alterations using a traditional process without user customization. A framework for user customization should include three parts: 1) a method for generating form (where the architect imparts knowledge), 2) a consumer interface (a way for people to customize), and 3) a method for efficient manufacture. The process of designing for customization will be broken down into each of the three parts, starting with form generation.

Form Generation

Different approaches exist for automating the creation of form. In a kit of parts approach, the designer develops a set number of components in a top down fashion (Corser, 2010) that is then given over to the user for configuration. Legos are an example of a kit of parts generation system. With only a few original modular parts, it is possible to create a high number of forms. The number of formal possibilities will always be able to be calculated due to the combinatorial nature of the system. In designing for a kit of parts system, the architect is able to exercise control over configuration possibilities by enforcing parameter combinations that enhance each other and at the same time by not permitting combinations that are infeasible to build or fail to meet the aesthetic goals of the system.

Examples of consumer customization systems that utilize kit of parts selection include NikeID, Mini Cooper and FlatPak houses. (Nike, 2011; Mini 2011; FlatPak; 2005). Both NikeID (fig 3.2) and Mini Cooper are examples of how companies which have a background in mass production have transitioned to offer user customizable products that are manufactured in a mass customization process. While kit of parts systems are often implemented at the initial product design phase (FlatPak, 2005; Tetrad 2005), the switch from assembling entirely standard parts to assembling a combination of standard and custom parts can go fairly smoothly.

An architect can also employ a bottom up approach to developing methods for creating form (Barros et al. 2011; Duarte, 2003; Hensel et al., 2004; Flat Clock, 2012; mTABLE, 2008; Vita, 2009). Instead of building from a set of Lego-like components, the geometry itself can be freely manipulated as would be the case when modeling clay. By using an algorithmic approach for form generation and including randomness in the system, infinite formal possibilities are achievable through the manipulation of certain parameters. Using this type of methodology, geometry is able to be manipulated without requiring the user to perform 3D modeling tasks. The consumer can experience an element of the unknown (uniqueness) in a generative process of creating form. Like the kit of parts method, the architect is able to embed rules into the generative framework to guide the user toward controlled outcomes.

Fig. 3.2 NikeID, kit-of-parts selection example

Designing for form generation, whether a kit of parts or algorithmic system is being implemented, is an exercise in determining a form finding methodology that suits the particular project as well as an exercise in how much control the designer is willing to give to the consumers, and in what ways. Additionally, gauging how much control the average consumer desires is helpful to creating a marketable system.

Interface

Instead of face to face communication, the interface is the place where the architect can communicate his intent to the user. The traditional architect/client role remains, only in a shifted form. Decisions about the interface affect the design of the form generation system. The designer is able to provide the user with controls for customization (in the form of sliders, for example) as well as make certain decisions on his own. Consumer customization may be intentionally limited for reasons including aesthetics, maintaining constructability, or maintaining a clear and simple set of interface choices (Hernandez, 2005).

The interface becomes the paper and pencil, or the 3D modeling software, for the consumer; it is their design and visualization aid (Archea, 1987). It has the option to be web based, as is the case with the thesis, or not. In the case of a web interface, whether the website is intended for touch or mouse and keyboard interaction will also affect what type of geometry manipulation model makes the most sense. An interface that is based on selection and slider control or limited geometry model manipulations, as opposed to an interface that would allow the consumer to freely model the geometry themselves, can allow for a less overwhelming consumer experience and a more straightforward approach to following framework rules (Huang, 2007).

Fabrication

Fabrication is the third aspect included in a parametric framework for customization. In developing a semi-automated fabrication process, it is necessary to design a smooth transition from geometry model to CNC manufactureable files. Also, borrowing concepts of mass production manufacture is integral to the development of an efficient mass customization process; this includes understanding the available fabrication resources and designing customization options to match those resources (fig 3.3).

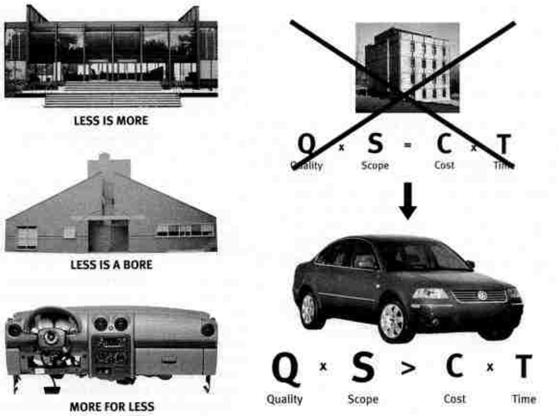


Fig. 3.3 Diagrams depicting time and cost savings of mass customization manufacturing techniques

4. CASE STUDIES

Mass Customization Case Studies

This chapter will describe several examples of mass customized products that are individually customized by their respective consumer. In particular, a shelving case study will be described since the customizable product of the thesis is shelving. Examples with different form generation, interface and fabrication methods will be described. Finally, the Flat Clock case study will be discussed in depth because of its approach to the entire customization to fabrication process.

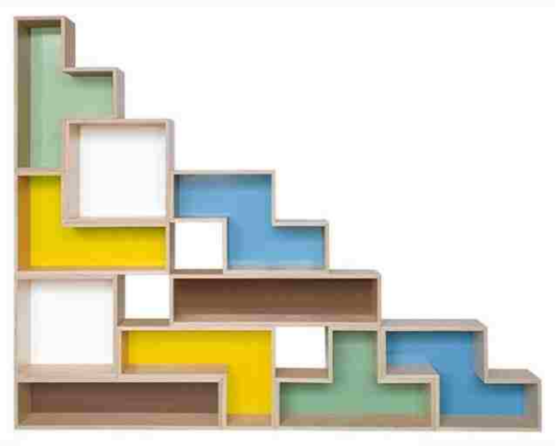


Fig. 4.1 Tetrad shelves, kit-of-parts selection example

Fig.4.2 M-Table, form generation example

Tetrad Case Study

Tetrad is a customizable modular shelving system developed by Brave Space Design (Tetrad, 2005). Tetrad employs a kit of parts strategy for creating form, and the shape of each shelving module is built in reference to the Tetris game (fig 4.1). The consumer customizes the product by printing out a template provided by Brave Space Design on paper, cutting out the module shapes, and arranging them by hand. From the desired paper configuration, the user has the option of ordering the specific kit of parts. In this case, consumer design interaction and customization is achieved by hand rather than through a digital interface.

Each of the modules is able to be pre manufactured in a process of mass production. Upon ordering, the shipping and assembly can differ in accordance with users individual desires.

mTABLE Case Study

The mTABLE by Gramazio and Kohler (mTABLE, 2008) is able to be built to custom dimensions and with numerous possibilities for tabletop surface topologies (fig 4.2). The system allows for full geometry form manipulation; it is not based on kit of parts selection. Customization is accessed through a combination of mobile phone, the system was built for the Nokia 60 series, and website interfaces. Users are able to influence the position and type of surface indentation though applying virtual forces by repeatedly pressing a button on the phone.

Although the design for each tabletop is unique, manufacturing is limited to a small set of materials and tools. Custom variations in the tabletop surfaces are milled using a CNC router.

Flat Clock Case Study

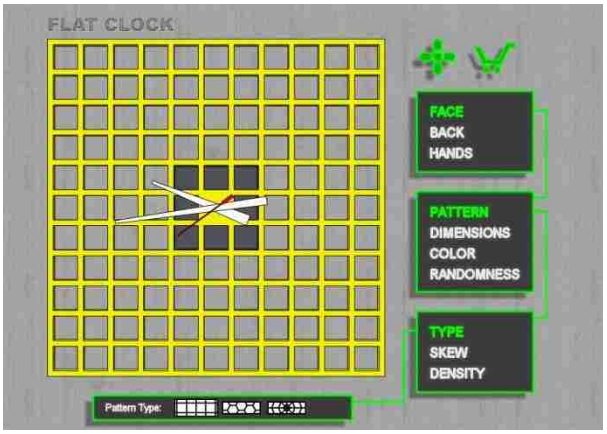
Flat Clock is a customizable acrylic wall clock designed and manufactured by Johnson House Design (fig 4.3) (Flat Clock, 2012; Johnson House Design, 2012). Flat Clocks are customized and ordered through the website Flatclock.com. Unique and randomized patterns are generated in accordance with the user's intent and are laser cut from acrylic sheets to form the clock face. Clocks measure approximately 16 inches square, although the actual size depends on the individual pattern. Multiple color options are available.

Flat Clocks use a voronoi algorithm to generate highly varied patterns. All Flat Clock face designs start out as having a square grid pattern and then can be adapted to a personalized pattern of solids and voids by changing various parameters (fig 4.4 – fig 4.9). Altering a parameter will update the FlatClock 3D model and visualization in real time.

Interaction with the customization interface occurs through slider controls and selection palettes. Different menus populate the screen depending on the type of manipulation the user is performing. At the highest level of customization, three parts of the clock are able to be customized: the face, the back, and the hands. The primary customization options occur within the face menu. Under each part, more options and respective slider controls are available.

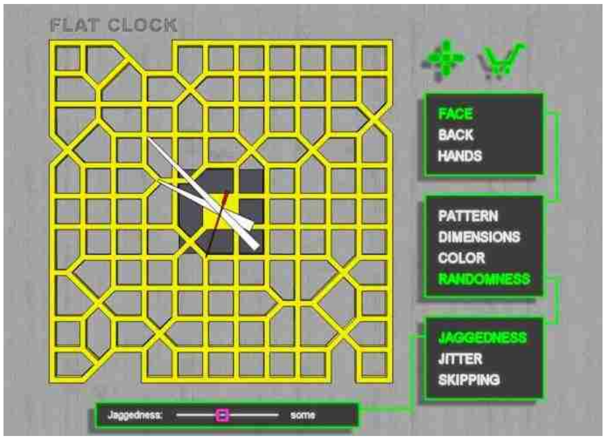
The clock is visualized via a live 3D renderer, developed in HTML5 by Johnson House Design. The 3D model is generated to scale and navigable. The model visualized on the website is the same vector model as used for the fabrication drawings. All geometry is generated in real time, and every manipulation is cumulative. As long as some degree of randomness is chosen to be included in the design, clicking on a parameter several times in a row will produce a different design each time. The probability of two of the exact same designs being generated is extremely low.

Fig. 4.3 Flat Clock customizable acrylic wall clock



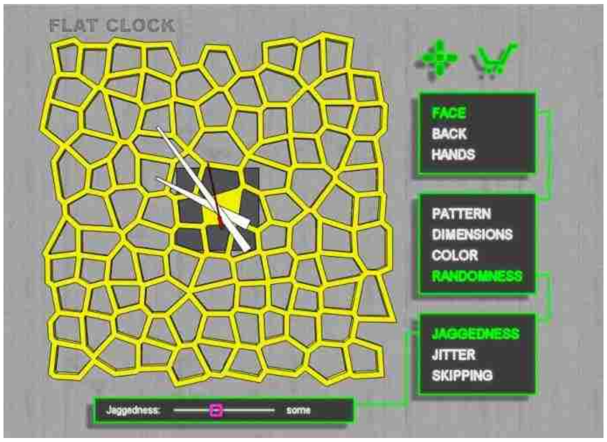
Pattern Type:

Fig. 4.4 Customization sequence 1



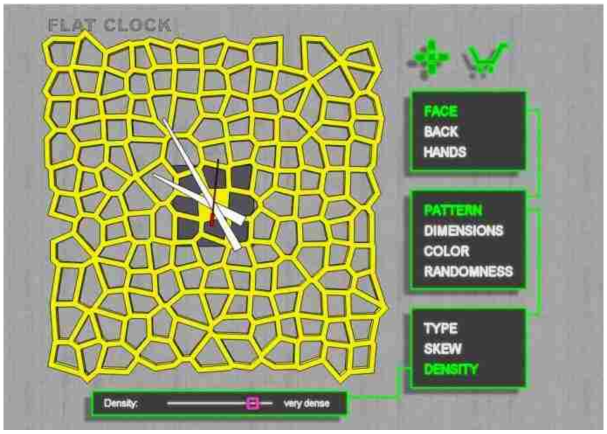
Jaggedness: some

Fig. 4.5 Customization sequence 2, randomness



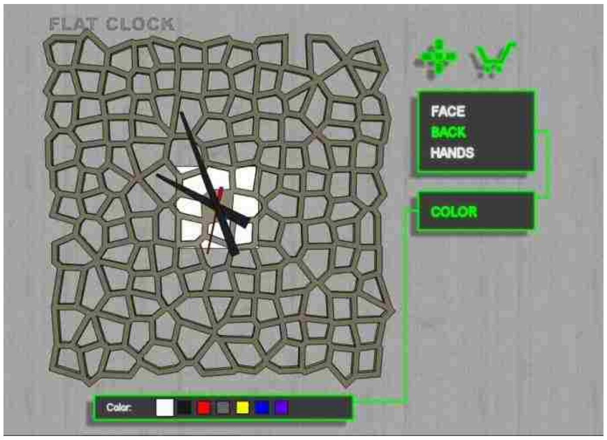
Jaggedness: some

Fig. 4.6 Customization sequence 3, randomness



Density: very dense

Fig. 4.7 Customization sequence 4, density



Color:

Fig. 4.8 Customization sequence 5, color

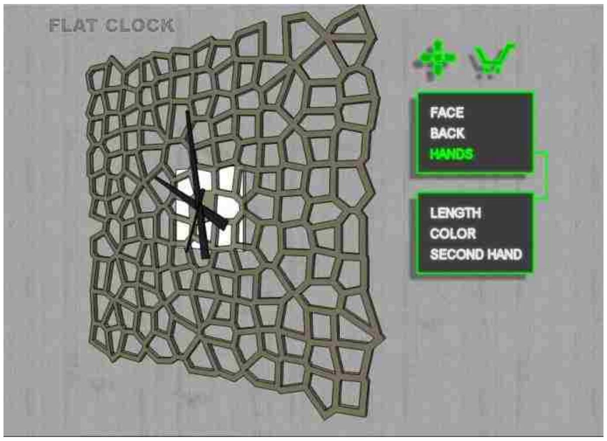


Fig. 4.9 Customization sequence, orbit 3D model

5. MASS CUSTOM SHELVING

Shelving as a Furniture Type

In order to design for customizable shelves, shelving was first examined as a furniture typology. The primary function of shelving is the storage of objects. To meet a variety of storage needs, shelving is often available for purchase in a range of sizes and sub types. This is true of mass produced and custom made shelves. Shelving unit dimensions can be based off of different functional uses as well as different interior spatial arrangements.

Distinctly named subtypes of shelving suggest that different geometry relationships tend to accompany different uses. For example, a shelving unit described as an entertainment center tends to have low and wide overall dimensions; a bookcase on the other hand, is often tall and narrow compared to the entertainment center (Ramsey and Sleeper, 2000).

Although different shelving subtypes exist, geometry relationships and overall size are more flexible than many other furniture types. For instance, the width of a sofa is typically sized for three people sitting side by side. The depth and seat height of a sofa is designed to fit a person seated comfortably with their knees out in front of them and their feet touching the floor. Sofa size parameters are variable, but not to the degree that the piece becomes uncomfortable (i.e. too tall) or completely fails to function (fig 5.1). The sofa is engineered to human dimensions and, like many furniture typologies; ergonomics is a major factor in its design (Ramsey and Sleeper, 2000).

Shelving, on the other hand, is engineered for the size and placement of the objects it will hold; ergonomics plays a lesser role in the design. Because of this, shelf compartments tend to be described in more detail than the dimensions of the overall size. This attention to the smaller parts allows the width and height of the shelving unit to vary greatly and consequentially, to conform to the custom dimensions of the interior space it is set in. A shelf is able to cover the entire length of a wall, or occupy an otherwise unusable corner space. Yet ergonomics is still a consideration; for example it would be impractical to design a shelf where all of the compartments are smaller than the size of a person's hand because of the necessity to retrieve the stored contents. Similarly, designing a very deep shelving unit, unless designed for a specific item, would result in much wasted space.

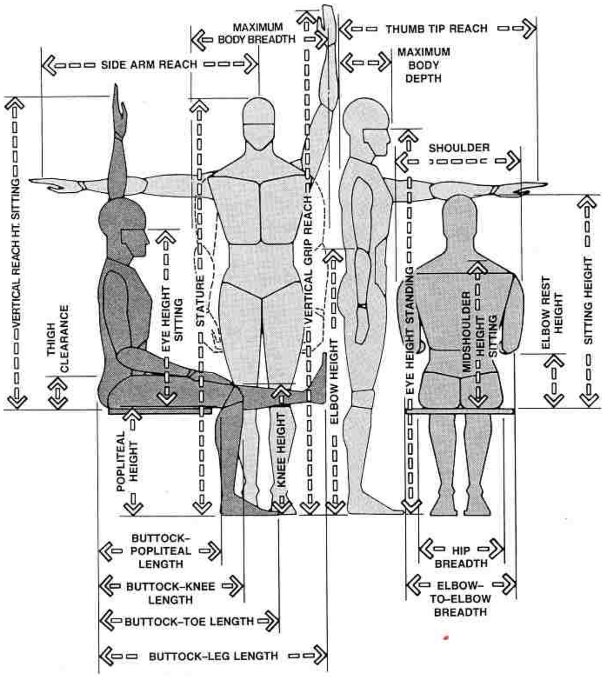


Fig. 5.1 Architectural ergonomics guidelines

Shelving Unit Physical Components

For analysis purposes, shelving units can be broken down into their physical components (i.e. bases and sides). Individual components and the relationship between components, as they create the overall shelving unit configuration, will be discussed.

Shelving units are made up of bases, the surfaces that hold stored objects, and connectors, pieces that structurally support the bases. The exact orientation and specifications of each of these components will differ if the shelf is freestanding or wall mounted. They can also differ depending on the shelf's intended use, materiality, or aesthetics.

With a few exceptions, notably wine racks, bases are commonly oriented in a horizontal position (parallel to the floor). In this orientation, each item is able to be stored upright, regardless of its shape or size. For wall mounted shelves, a horizontal base orientation is able to support bracket style connectors. It is also capable of supporting numerous styles of vertical connectors for the creation of freestanding shelves. Connectors can be oriented perpendicular to the shelves (exactly vertical) or at a slightly skewed angle. They can be positioned at the ends of the horizontal shelf (end caps) or as interior supports. Examples of different ways of configuring bases and connectors where components meet at right angles will be described. From the example shelf configurations, it is easy to imagine more complex arrangements where components meet at skewed angles or have different joint connections.

In one example configuration, shelving can be thought of as a composition of planes. Horizontal base components are the primary compositional pieces and vertical components act as secondary connector members. Basic examples include a boards connected to the wall with brackets or a series or parallel freestanding boards held up by cinder blocks (fig 5.2).

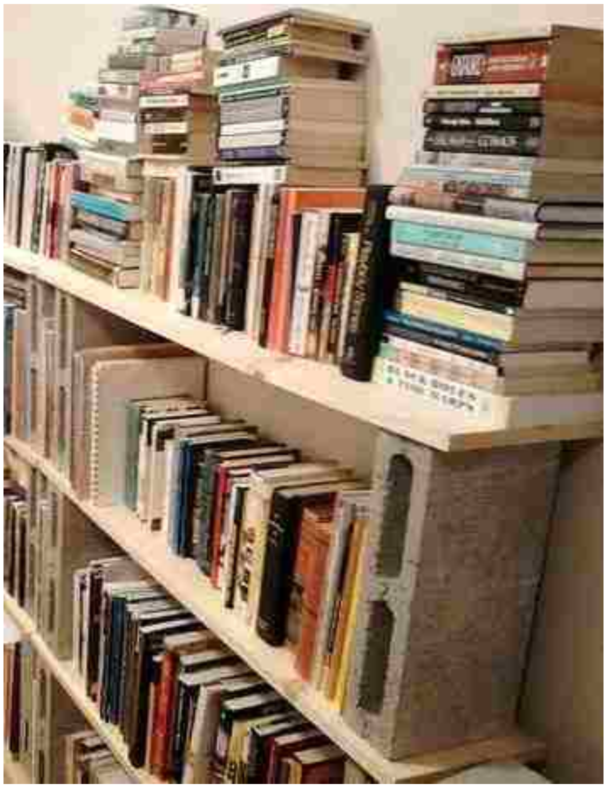


Fig. 5.2 Board and block shelving

In another configuration, shelving can be constructed from a series of boxes. In this case, the horizontal and vertical components are connected. Each box compartment is structurally stable on its own, as in stacked milk crates (fig 5.3). A system of boxes can be configured from stacked adjacent boxes (fig 5.3), or rigid interlocking boxes that overlap in elevation view.

Fig. 5.3 Milk crate box shelving

A hybrid component configuration system can also be created by combining configurations of the planar and box examples. Other right angled forms, U or L-shaped forms, can also be included in a hybrid design language (Vita, 2009).

Shelving as a Mass Custom Product

Shelving, specifically shelving unit configurations that have horizontal bases and perpendicular vertical connectors, is suitable for mass customization. Geometry arranged in this way is flexible. Much variation is possible in determining where each component is placed. Components can be arranged to create small or large storage spaces, ideal for different uses. In addition geometry is somewhat limited formally (rectangle pieces, right angle connections) which is desirable for efficient mechanized manufacturing procedures. Shelving configurations that have more specific and complex geometries will be limited in the ways they will able to be customized. For example wine racks with v shaped shelf configurations will not be as flexible towards accommodating other uses or various sizes (if the use is intended to be maintained for wine storage).

Shelving designed to be freestanding is desirable for a mass customized product. Freestanding shelving units are repositionable, unlike wall mounted shelves, and do not permanently alter the space they occupy. For many users, apartment dwellers in particular, making permanent alterations to their space is not an option. A user customizable shelf is able to fit specific of dimensions of an interior space. Unless the desired dimensions happen to be an even number, such as 36 inches, this is not something that is typically available with a mass produced product even if the furniture line offers several sizes.

6. B-SHELVES

Fig. 6.1 B-Shelves built shelving prototype

B-Shelves

For the thesis, B-Shelves are designed as customizable freestanding shelving units made up of a series of interlocking boxes. The manner in which boxes intersect each other creates two spatial divisions, original boxes (boxes with four distinct corners) and emergent boxes (box shapes created by the intersections). Inside original boxes, large items can be stored while smaller items can be stored in the emergent shapes. A large variety of spatial configurations are possible with the system. B-Shelves are customizable though a website where the consumer is able to interact with their particular shelving product. They are able to do so by determining the shelf size, the specific box arrangement, and finishing options (color, doors, backs, etc.). Shelving units are constructed from 3/4” thickness Baltic Birch plywood. Side pieces are slotted together at the box intersections, and glued at the corners.

By designing the system with the intent that it can be reached by a wide consumer population, the evolving architect and consumer roles have begun to be addressed. A web interface is used as the customization platform so that B-Shelves can be made customizable my many. Since the form generation system is made for the internet, it was programmed in JavaScript and makes use of the canvas drawing element for HTML5. In this case, Grasshopper, or other similar graphical programming software, would not suffice as a methodology for encoding designer intent. A Grasshopper interface would only be accessible to the people who have purchased Rhino and installed the plugin on their computer, a relatively small consumer group. Also, Grasshopper is fundamentally set up to execute code in a linear progression, while the force/physics generation system is reliant on recursive and looping operations. Website visualization occurs in 2D. This decision was made partially because a 2D physics simulation simplifies the form generation calculations. Physical box components are represented as rectangles where the screen (2D representation) is the front elevation view of the shelving unit. Rectangles are drawn at the centerline of the box thickness, 3/4 inch Baltic Birch in reality. This decision was also made because developing a 3D visualization on the web is neither fully supported nor easy to execute. Options for live 3D visualization include utilizing WebGL for HTML5 and making use of built in rendering libraries (however still more complex and time consuming to program than 2D drawing) using Processing.js, (a comparable process to programming in JavaScript) , or building one’s own 3D renderer to run on canvas. For the thesis, time was spent developing the methodologies for form generation rather than expanding the system for full 3D visualization.

7. B-SHELVES FORM GENERATION

Encoding Designer's Intent

The majority of time in the 10 week thesis term was devoted to developing a form generation system. Form generation is the part of the three part framework where the designer's intent is able to be expressed. Interface and fabrication, the other components of the system, are necessary to create a complete demonstration of a user customizable product system. Yet, without a fairly complete exploration of methods for generating form, the amount of time spent on the other parts of the system will be irrelevant to the big picture.

An algorithmic approach to generating form was taken for the B-Shelves thesis. This type of approach, as compared to a kit of parts system, is able to produce a large variety of spatial configurations and generate forms in a way that provides the user with an element of surprise. This also works well with the typological considerations discussed in the previous chapter. The B-Shelves design is flexible enough to be manipulated in many ways and simple enough to be efficiently manufactured. Finally an algorithmic provides a challenge in analyzing and encoding design intent, a primary interest of the thesis.

For the thesis, the first step in encoding design intent was developing a broad concept for what later became B-Shelves. The case study analysis and shelving type study prepared me to develop a set of goals and a conceptual architectural model (fig 7.1) that manifest these goals in physical form. The goals are as follows. The system must be able to generate box-type shelving configurations that:1) have a high degree of formal variation, 2) are able to fit the overall shelf size to the consumers' desired dimensions, 3) are reliably structurally stable as freestanding shelves, 4) create box sizes and configurations to sizes that are functional the majority of the time, 5) create boxes intersections in locations where joint details are constructible, 6) have joints that are designed for efficient fabrication and assembly procedures, 7) have meaningful and varied finishing options (in addition to formal arrangement), and 8) conform to the aesthetics of the designer.

In the case of B-Shelves it was important that a system for algorithmically generating form could be found or developed that produced results similar to the conceptual model. The form finding methodology was not chosen only for the interest in the methodology itself but for the type of results it tended to produce.

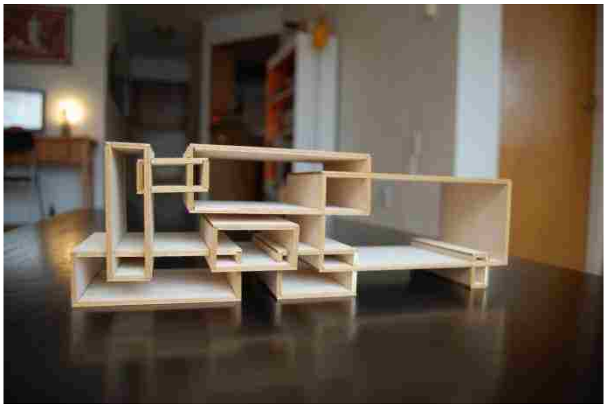


Fig. 7.1 Original box configuration concept model

Once an initial form generation system was implemented, generated geometry was evaluated by me for its fitness in relationship to the conceptual model. Rules and relationships of the form generation methodology were then modified in order to produce solutions that were a more optimum fit with the concept. The cycle of evaluating the generated solutions then modifying the system code was continued until forms were produced that reliably adhered to the concept goals.

Of course, the nature of an algorithmic system is that the designer cannot plan for every condition. The act of visualizing the type of geometry produced by the system can cause the designer to rethink the architectural concept model in order to best embody the intent. It is also possible to rework the goals themselves. For example, an original goal for B-Shelves was the ability to organize the shelf configurations in a way that could store specified items (a particular vase brought back from Europe, books, dvd player, etc.) in appropriately sized compartments. In other words, retain a few specified voids in the configuration of overlapping boxes. This goal turned out to be more complex to achieve than expected, and was removed from the set of requirements. It remains a promising functional addition to the B-Shelves system and it is discussed further in the section on future work.

Shape Grammar System

I began the development of a form generation system by implementing a shape grammar approach. In other research, DeStijl paintings have been analyzed and generated using shape grammar systems (Kirsch, 1986). DeStijl paintings have a similar arrangement pattern to the B-Shelves concept. Shape grammars are often employed for the division of space and they work particularly well for dividing rectangular spaces. Kirsch, as well as others, has also worked on methods for analyzing and recreating generic forms with a shape grammar system (Duarte, 2003; Stiny, 1980). In the thesis exercise, a simple rule set was developed and executed. Progression though the shape rules was intended to be manipulated by consumers in order to involve the users in the customization process and to produce a large degree of variation that is not always captured in shape grammar implementations. This differs from implementations where progression of form manipulation happens though one descriptive notated path (Knight, 1994). Experiments highlighting the differences between alternate starting situations were carried out. Whether the grammar started with the shelf bounding box, or interior box(s), or a combination of the two was tried. Stopping rules and methods for cropping were also explored.

As it turned out, box overlaps were difficult to achieve with the rule set. The shape grammar encouraged a clear division of space but did not produce overlapping conditions as described in the initial concept (fig 7.2-fig 7.4). Very small boxes, created after the recursive shape rules had run for several cycles, tended to overlap; yet this provided overly dense arrangements for the desired shelving application. Ultimately a different method for generating form was utilized for the B-Shelves thesis. That is not to say that forms similar to B-Shelves configurations could not be created with a shape grammar provided further analysis and the development of a suitable rule set.

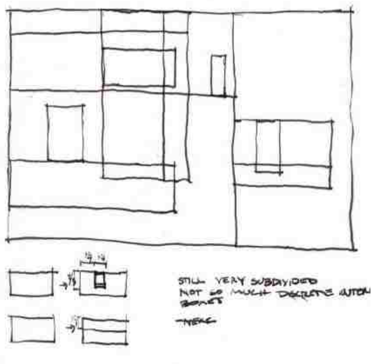


Fig. 7.2 Shape grammar exploration first rule set

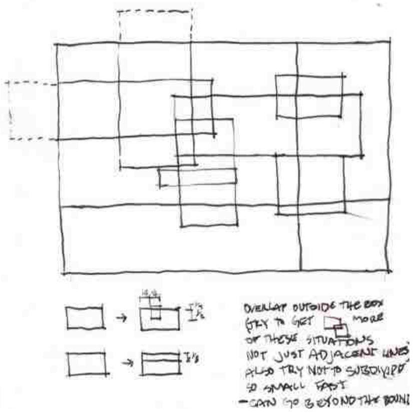


Fig. 7.3 Shape grammar exploration second rule set

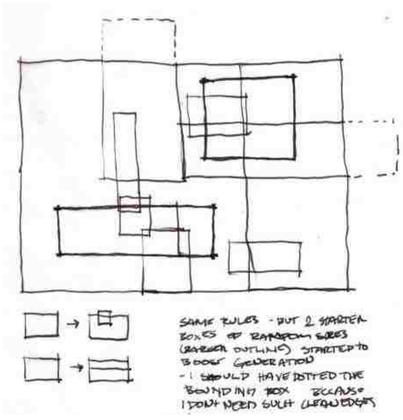


Fig. 7.4 Shape grammar exploration third rule set

Movement/Force/Physics Simulation System

Since the shape grammar methodology tended to produce gridded forms rather than the overlapping box forms described in the concept model, I turned to a type of system that would maintain a lifelike organization of the generated boxes. I developed a highly simplified 2D movement/force/physics based simulation that is able to generate and arrange boxes in realistic looking scenarios. The algorithmic form generation system on the B-Shelves website uses this simulation. It is programmed in JavaScript using canvas drawing element for HTML5. The system will be broken down into descriptions of simulation components (boxes, attraction point, etc.) and the animation forces (how boxes are arranged to generate shelving units). Several iterations were designed before reaching the current status of the system. Iterations will be explained along with the progress made toward encoding the designer's intent at each step.

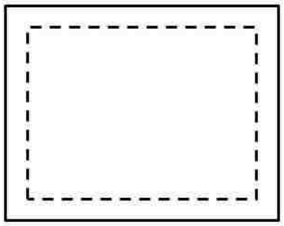


Fig. 7.5 Internal force offset, "sweet spot"

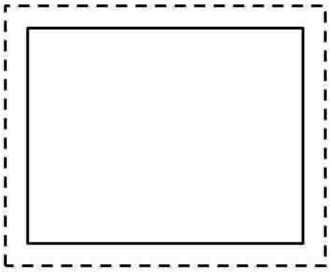


Fig. 7.6 External force offset, "aura"

Simulation Components

Each simulation component of the current system is listed and described in detail. Earlier versions of the system did not use as complex of component relationships as described here and did not implement all of the described components.

Origin Point: A single point that exerts pulling force on center of all boxes. Pulling force of the origin point is set globally by the designer.

Boxes: Boxes of a set width and height. Individual boxes have a few pieces of data, including: the box dimensions, the ideal intersection dimensions, the external force dimensions, and center point.
Box Dimensions: The width and height of the box to be physically built. Boxes are generated to random dimensions within a set width and height range that is parametrically variable by the user. Box dimensions are made up of a rectangle whose lines represent the centerline of the physical box sides.
Ideal Intersection Offset ("sweet spot"): This polygon is an internal offset of the box dimensions (fig 7.5). The internal offset is the same between all boxes and is set globally. Upon intersection with another box instance, the ideal intersection polygon serves as the focal point of various interactions that may occur. Simply put, the ideal intersection rectangle acts as a "sweet spot" that the edges of other boxes are generally attracted to.
External Force Offset ("aura"): This rectangle is an external offset of the box dimensions (fig 7.6). The offset is the same between all box instances and is set globally. Box instances will not interact with each other unless they intersect each others' external force dimensions. If two box instances intersect each others' external force offset, but do not intersect each others' box dimensions a mild repellant force is generated between the two boxes. This serves to gently push non-intersecting box instances away from each other.
Center Point: The center point of the box. Boxes are attracted to the origin point via their center points.

Ground Plane: A line that stops downward movement of the box instances.

Crop Box: A box that is used to crop off potentially unwanted portions of box instances. The cropping operation is optional and controlled by the consumer.

The crop box has two parts: the inner crop box and the outer crop box. Box instances that do not have at least one point inside the inner crop box are discarded. Box instances that are not discarded, but that intersect the outer crop box are cropped such that no points are outside of the outer crop box. When two box instances that intersect each other also intersect the same side of the outer crop box, one of the two box instances is cropped further in than the outer crop box, so as to avoid having concurrent edges.

Animation Forces

Calculations: The simulation runs at a constant simulated speed. For each time interval, forces on each box are added together. After all forces are accounted for, the resultant acceleration vector (in this simulation mass equals 1 so force and acceleration are equal to each other, F = ma) is multiplied by a globally fixed scalar equal to less than one. This serves as a rough approximation of friction, helping stabilize behavior and giving boxes an end velocity. Finally, the acceleration vector is used to update the velocity vector, which is used to find the box's new position. This process repeats itself at the next time interval. Naturally, velocity and position must be held on to for the next interval.

Forces: boxes interact with each other in a multitude of ways, depending upon the situation. Boxes that do not intersect each others' auras exert no forces on each other. Boxes that intersect each other's auras but that do not intersect each others' physical boxes exert pushing forces on each other. This was implemented to help prevent the edges of non-intersecting boxes from getting too close to each other (fig 7.10). It also helps prevent certain types of undesirable clumping (fig 7.9). For boxes that are intersecting each other's physical boxes, a number of scenarios can occur depending upon the way in which the boxes first intersect and how the boxes are interacting with other nearby boxes.

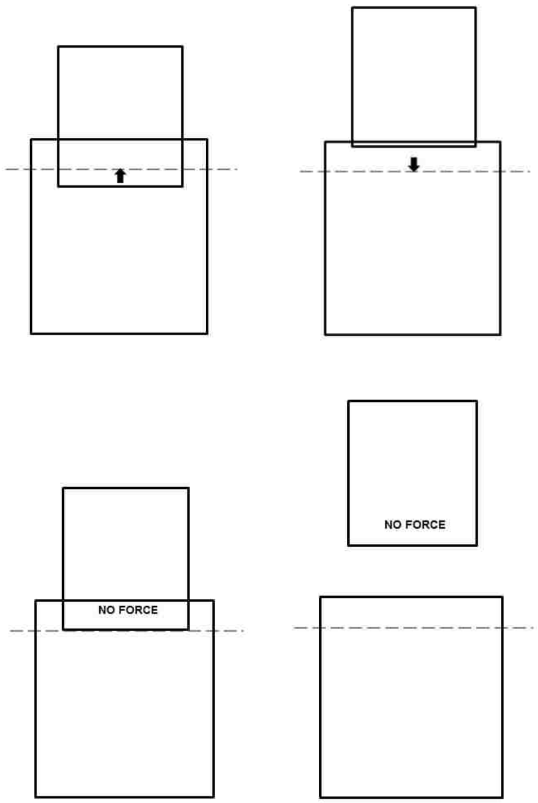


Fig 7.7 Simplified diagram of strong attractions
Top left, side attracted to the sweet spot. Top right, side attracted to the sweet spot. Bottom right no intersection and no attraction force. Bottom left side at sweet spot no attraction force

The most pronounced interaction is the so-called "strong force" interaction. When two boxes intersect, a particular side of one box develops a strong attraction to a particular side of the sweet spot of the other box, and vice-versa (fig 7.8). When strong forces develop, they will always develop between the opposite sides of the two intersecting boxes (fig 7.7). For example, the left side of box A can develop a strong force attraction to the right side of box B. The right side of box A would never develop a strong attraction to the left side of box B. Strong force attractions are the glue that hold the shelves together.

Weak force interactions occur between the same sides of intersecting boxes. For example, if box A is intersecting box B, there will be a weak force interaction between the top of box A and the top of box B, the left of box A and the left of box A, and so on. Weak force interactions are kept at zero unless equivalent box sides become closer than the sweet spot distance. When this occurs the weak force attraction (actually a repulsion) forces the two sides apart. Like the strong force interactions, this also is intended to keep intersecting boxes from clumping up or overlapping on equivalent sides in a manner that is unable to be constructed (fig 7.8).

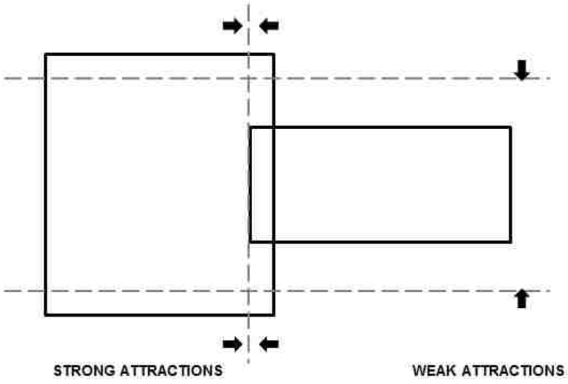


Fig. 7.8 Strong and weak attraction forces

Another force in play is the triple-intersect repulsion. In almost all cases when three boxes mutually intersect each other, the outcome is determined undesirable. To minimize this occurrence, each box keeps track of the boxes it intersects. If three boxes are found to mutually intersect, one box is chosen to be repelled.

Each type of force is scaled differently. Some of the force calculations are linear, but most are non-linear (using very simple quadratic functions), using the intersection depth or distance as the variable in the equation. Setting up the force equations is tricky. If a force equation is too weak it will go largely unnoticed. If it is strong it will overpower other important forces. If an equation is too abrupt, instability is introduced into the system (extreme jumpiness in the simulation) and if an equation is not abrupt enough, it can cause other unexpected effects that are difficult to decipher in the complexity of the system. Much time was spent adjusting the various force equations.

System Reflections and Designer's Intent

Using a heuristic movement/force/physics simulation for placement is different from using a shape grammar or a genetic algorithm. In a shape grammar based system, rules (what is fed to the generation system) are explicit, objectives (designers intent or the eight overarching goals I described in the B-Shelves concept model) implicit and upon analyzing the forms typically generated with the system, the given rules are more or less transparent. On the other hand, in a genetic algorithm solver, objectives are explicit, rules are implicit, and the way in which rules are applied can be vague (unless true and false situations are scripted into the system). A force based system, like shape grammars, has explicit rules, implicit objectives, but the given rules can be quite hidden in the type of forms typically generated.

In a force/physics based system, it is not obvious how to set rules to achieve desirable outcomes. Furthermore, the fuzziness of the outcomes makes tuning rules tricky, as the exact effects of individual rules isn't always obvious. Also, it may always be necessary to cull a certain percentage of outcomes that fail to meet objectives. None the less, the heuristic force based simulation has some major benefits over the other discussed systems. For example, unexpected and unique outcomes are possible, even given the same parameter set. One might argue that a genetic solver can produce similar results, but one might also argue that genetic solvers can be quickly bogged down by a high number of variables and a complex objective function. Because of the tendency of multivariate solvers to take a long time to solve, they may not be the best tool for real time generative systems.

Because B-Shelves is meant to be distributed via the internet and easily run on the average home computer (or touch device) computation speed is a concern. This is where a heuristic movement simulation shines. It is quick and, at least to the computer, mathematically simple. An added benefit is the fact that the simulation is animated and observable; it is something that the human mind can relate to and interact with on the spot.

Initial research on the B-Shelves project led me to the conclusion that a heuristic movement simulation was best suited to the task of mass customizing shelves in the tactile and interesting way that I envisioned. That is not to say that the developed system is perfect or ideal. Creating a stable system that implicitly enforces the proper objectives is difficult. An area for potential future research is the development of a hybrid system that uses both a shape grammar or solver and a force-based simulator. This may increase the stability and robustness of solutions without removing the playfulness and responsiveness of the current system.

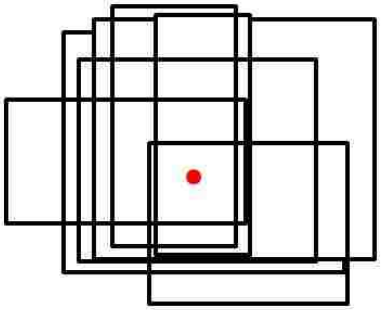


Fig. 7.9 Undesirable, no box attraction and repulsion forces in place

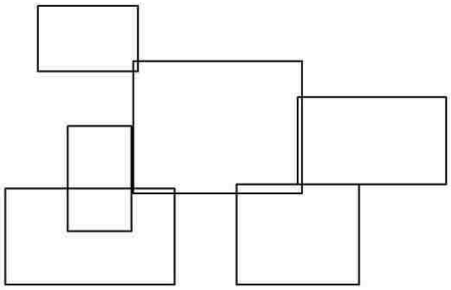


Fig. 7.10 Undesirable, edge-to-edge intersection conditions

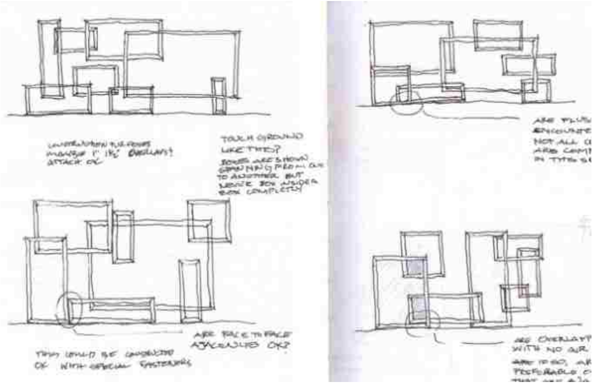


Fig. 7.11 Sketches of desirable and undesirable intersection conditions and their fabrication feasibility

System Iterations

Arriving at the current system was not a straight forward process. Various form generation systems were developed, tweaked, and modified until the system reached its present state.

In the first system iteration, boxes were initially generated at any location along the perimeter of the simulation space (the screen) and the origin point was placed in the center of the space. This resulted in boxes quickly settling around the origin point. The system was much simpler than the present system, with all force vectors being calculated between box center points (instead of box sides, as the present system uses). Box intersection depth was first calculated and if intersect distance was greater than zero, the unit vector between the two center points was then calculated and scaled by the intersect distance. Boxes were never attracted to each other; they were only attracted to the origin point. Although rapid and effective at generating form, this system tended to snap boxes into an offset grid that was boring to look at. It also configured boxes in a clump, lacking a flat bottom, in way that is infeasible to build as a freestanding shelf.

For the next implementation, a ground plane was added and complex box relationships were initiated (attractions to each others' sweet spots developed). While this produced configurations that were situated on the ground plane, box-to-box attractions led to excessively dense, impossible to build outcomes. To reduce the density of solutions, weak force interactions were implemented. Still, though, boxes formed in groups of triple or quadruple intersecting boxes similar to figure 7.9. To fight this, box intersections were tracked and boxes were repelled in triple intersection situations. This led to boxes piling up edge-to-edge, another undesirable situation.

In the following iteration, "auras" were added to boxes in order to avoid edge-to-edge arrangements. In addition, an attempt to size the shelf configurations to fit within the users' desired dimensions was carried out. Side lines were implemented in order to stop boxes from passing. The same methodology was used as used for the ground plane. However this tended to produce shelving results where boxes quickly stacked on top of each other and created few intersections. These configurations tended to be tippy, or structurally precarious (fig 7.12).

Finally, to create shelving units that adhered to the dimensional desires of the designer, an optional cropping function was built into the system. The initial cropping implementation would often produce boxes with concurrent edges. To ward off this problem, logic was added to the crop box to vary crop depth where necessary. Of course, parameters and force functions were constantly adjusted and tested throughout the entire process.

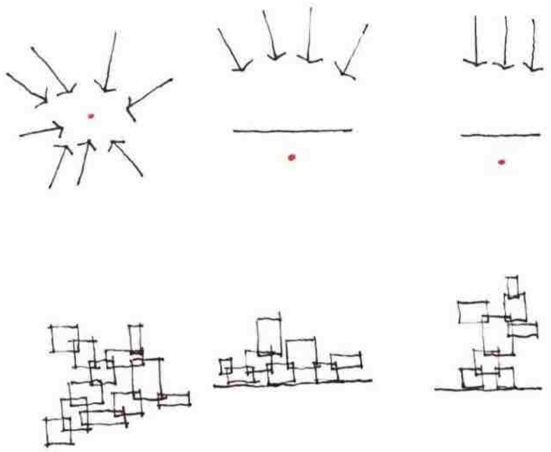


Fig. 7.12 Examples of boxes generated in different positions and corresponding configurations

Construction Details

Fig. 7.13 Rabbet joint detail

In addition to box spatial configurations, construction details are included in the form generation system. Yet designing for construction details is unlike designing for spatial configurations in the fact that the majority of design decisions are made by the architect himself without user customization. Construction details are closely linked to the overall form. Choices of details will affect means of construction and structural stability (fig 7.14, fig 7.15). During the B-Shelves design phase, several corner joint options were sketched and prototyped. They were then analyzed for their required production sequence, ease of assembly, structural stability, aesthetics, and ability to produce accurate and square joints. In the end, a rabbet joint was decided over an investigated miter or butt joint. The shelf example was built with a full rabbet for ease of assembly and a clean look (fig 7.13) although a stopped rabbet was also prototyped.

Prototyping was also conducted for the intersecting box joint conditions. With the side intersections, it was decided that each side piece would slot together in an interlocking fashion. Interlocking slots are easily cut to precise dimensions using the CNC router, a tool already being used for B-Shelves construction. Testing was carried out to find a slot dimension that snuggly fit the interlocking piece after sanding and finishing. All details were specifically designed for CNC cut 3/4" thick Baltic Birch plywood. If in the future B-Shelves were to be made with another material or another set of cutting tools, details would have to be worked out specifically for the new material and new machinery.

The detailing and prototyping process can take up a significant part of the design phase. Construction details must be precisely worked out so that they can be made in a repeatable fashion across the production of multiple furniture pieces. In addition, details must be designed to be generic enough that they function on every customized shelf. For instance the exact thickness of the slots should be detailed in a way that all slots will smoothly come together, including the last piece to lock into the shelf. Different B-Shelves form configurations will inevitably have their own load patterns and internal stresses. It is important that the joints are designed in a way that they can support the stresses for the different forms.

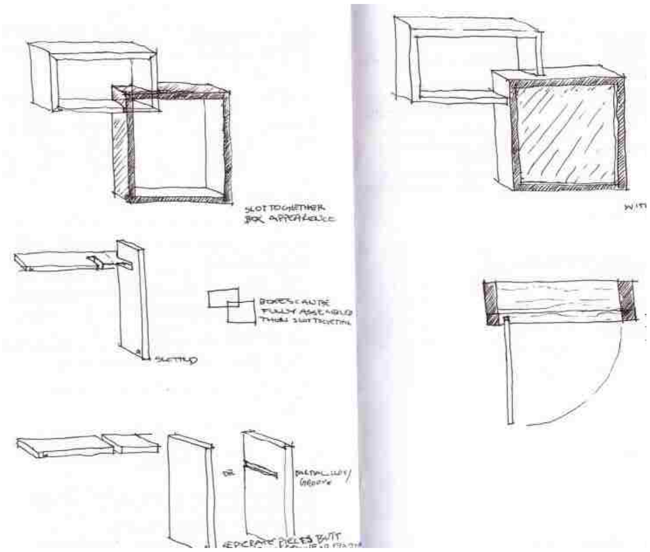


Fig. 7.14 Sketches of joint and door details

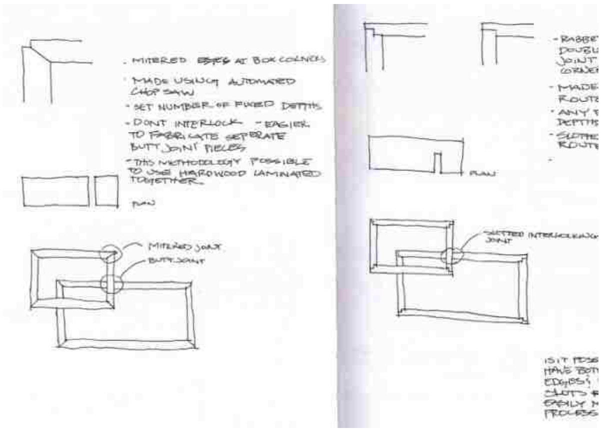


Fig. 7.15 Sketches of joint details as they relate to fabrication procedures

Finishes

Customizable finishing options are another way that one shelving unit can be made to look distinctly different from another. Finishing choices are intended to be partially generated and partially selected from predefined options. Generated finishing options have yet to be built into the force/physics system but the design intent has been clearly defined. Options for color, extruded boxes, doors, and backs are included as part of the B-Shelves system (fig 7.16 – fig 7.19). Accent colors are able to be painted on the inside walls of one or a few box components. To add accent color to a box in the already configured shelf composition, a consumer can pick a color from a predefined palette and watch as the color is applied to a randomly chosen box. Intelligence is built in to the system to generate finishing components in buildable locations. For example if a door is placed in relationship to a specific box, a back is automatically added to the same box. In situations where more than one door is added to the shelving unit, the system prevents door placement to occur at adjacent overlapping boxes; this is enforced because doors cannot physically overlap in the same plane. Also, the architect can maintain control of the amount of finishing that can be added to the shelf. This can be regulated by only allowing a percentage of the total boxes in a given shelf to have a particular finish. By exercising control over the number of possible finishing options, the designer can provide options that are not extremely materially intensive or expensive to make. This also allows the B-Shelves to be finished in a way that fits with the designer's ideals.

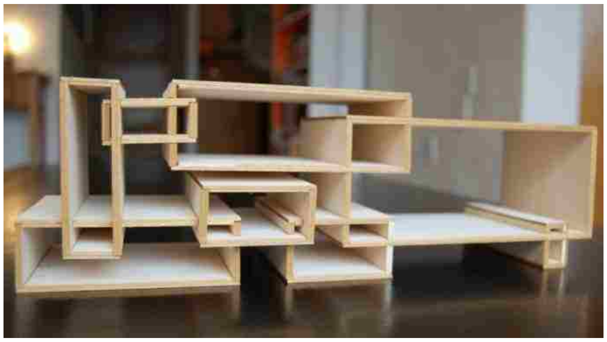


Fig. 7.16 Finishing options, extrusions

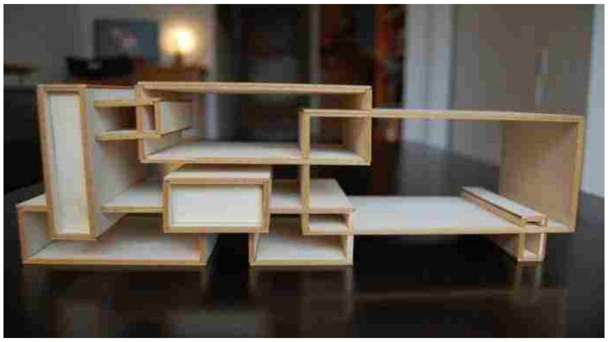


Fig. 7.17 Finishing options, doors and extrusions

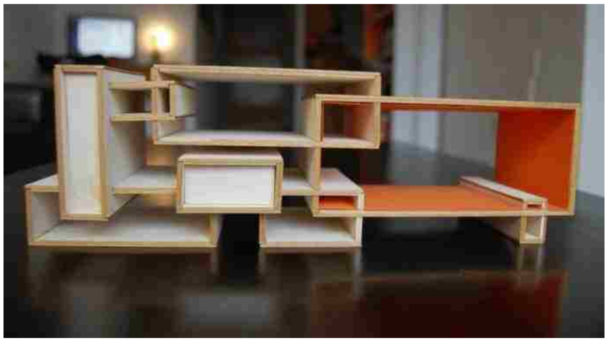


Fig. 7.18 Finishing options, doors, extrusions and color

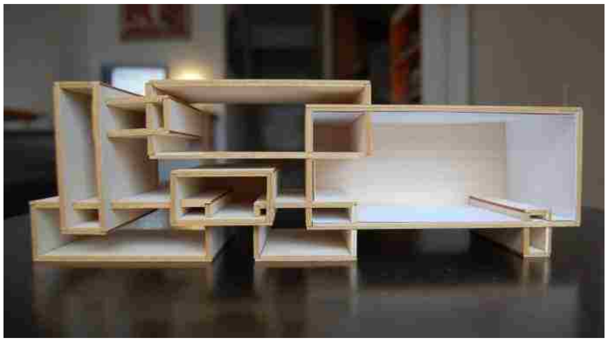
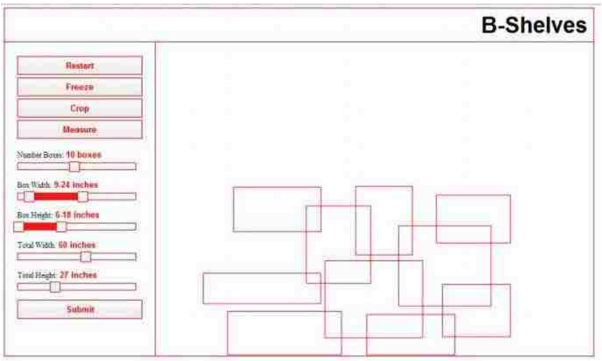


Fig. 7.19 Finishing options, color and backs

8. B-SHELVES INTERFACE



Number Boxes: 10 boxes
Box Width: 9.24 inches
Box Height: 6.18 inches
Total Width: 60 inches
Total Height: 27 inches

Fig. 8.1 Screenshot of B-Shelves web interface

Controls for Manipulation

Through the B-Shelves web interface, the user is able to customize and visualize their product in a simple and fun environment (fig 8.1). Parametric controls include the number of generated boxes, box width, box height, total shelf width and total shelf height. Users can alter all of the inputs through slider controls. The form finding animation is run when the website is first opened as well as restarted at the trigger of the restart button. Alteration to the parameter sliders will affect the animation once restarted. Options for freezing and cropping the animation are also possible.

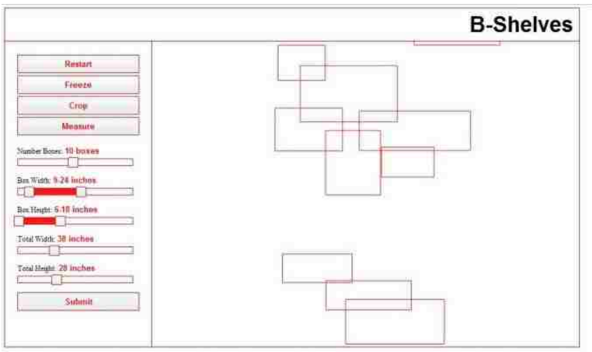
Cropping, will ensure that the generated shelf will fit inside the dimensions specified in the shelf dimension inputs. Cropping is a somewhat complex procedure that ensures box edges do not exactly overlap in the places where they have been cropped to size. Alternating intersecting boxes are cropped to a smaller size as described in the chapter on form generation.

In addition to manipulating slider inputs, the user is able to directly interact with the configuration by hovering over a box, picking it, and dragging it to a new location. Once the box is released in its new location, boxes will move to account for the new placement. The ability for the user to drag individual boxes can be advantageous in controlling finishing touches on a configuration. It is also possible to measure the dimensions of the shelf by clicking in the visualization window. A measuring tool allows the user to display dimensions between any two points (fig 8.5).

Finally, the user can click the submit button once they are satisfied with their B-Shelves shelving unit. This action saves the geometry data as an SVG (scalable vector graphics) file so that it can be prepared for fabrication.

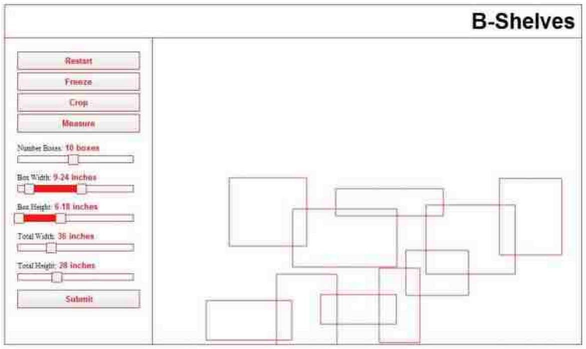
Product Visualization and Interaction

In the customization process, the consumer implements a change, restarts the simulation and sees the results. Parameters do not have to be changed in order to see a different result. A simple restart will generate each box in a slightly different location and at a slightly different size so the new configuration will always be different from the configuration before it. The B-Shelves customization plays like a game. The ability to reference the in progress design, through the visualization window, at all times helps to understand how changing a parameter affects form (fig 8.2 – fig 8.5).



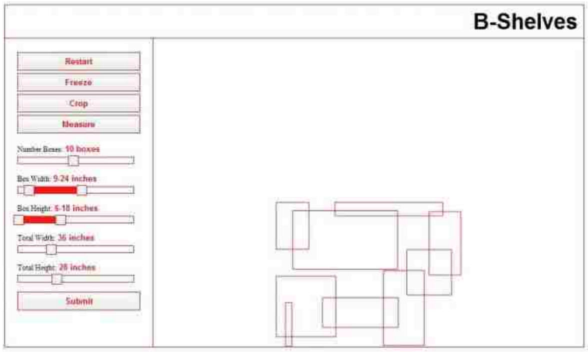
Number Boxes: 10 boxes
Box Width: 9-24 inches
Box Height: 6-18 inches
Total Width: 38 inches
Total Height: 28 inches

Fig 8.2 Animation sequence, 1



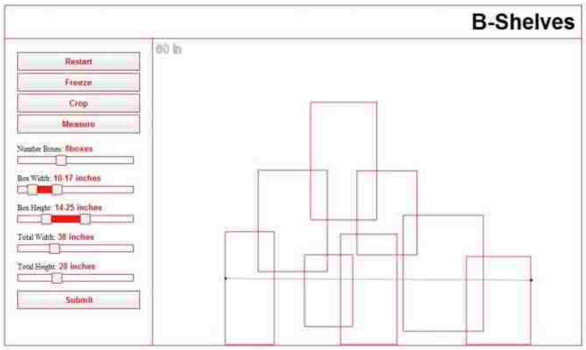
Number Boxes: 10 boxes
Box Width: 9-24 inches
Box Height: 6-18 inches
Total Width: 36 inches
Total Height: 28 inches

Fig 8.3 Animation sequence, 2



Number Boxes: 10 boxes
Box Width: 9-24 inches
Box Height: 6-18 inches
Total Width: 36 inches
Total Height: 28 inches

Fig 8.4 Animation sequence, 3 crop



Number Boxes: 8boxes
Box Width: 10-17 inches
Box Height: 14-25 inches
Total Width: 38 inches
Total Height: 28 inches

Fig 8.5 Measuring Tool

9. B-SHELVES FABRICATION

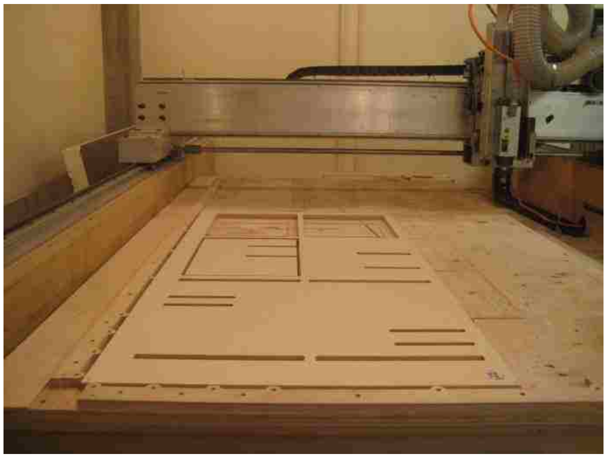


Fig. 9.1 B-Shelves pieces being cut on the CNC router

Fabrication File Preparation

Once B-Shelves forms are generated using the web based force/physics system, files are prepared for fabrication. Fabrication files are created in the Grasshopper generative modeling environment for Rhino software. The SVG geometry file is able to be exported as a DXF (drawing exchange format) and opened in Rhino and then Grasshopper.

In the automated Grasshopper environment, the scaled vector drawing is extruded to the desired depth, offset for material thickness, slots are created at the intersections, rabbets are created at the corner joints, and all pieces are labeled and laid out for fabrication (fig 9.2, fig 9.3). The Grasshopper definition does not need to be explicitly told the number of boxes that make up a shelf, in order to carry out fabrication preparation procedures. From Grasshopper, the laid out geometry is saved back into Rhino for final nesting and exporting (fig 9.4). Once the geometry is laid out, g-code is written using a CAM program. Rhino CAM was used in this case, however any CAM software for 2 ½ axis milling would be satisfactory.

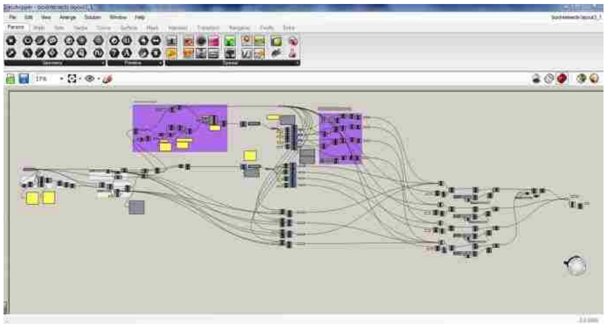


Fig. 9.2 Screenshot of fabrication preparation Grasshopper definition

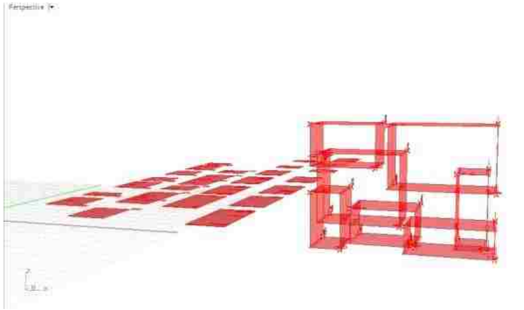


Fig. 9.3 Screenshot of extruded and laid out shelf geometry in Grasshopper/Rhino

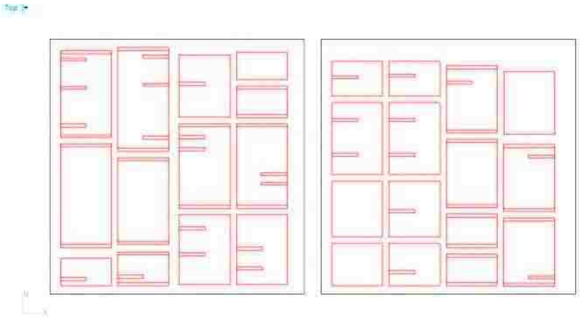


Fig. 9.4 Shelf layout on two 60” x 60” sheets of Baltic Birch plywood

Fabrication

B-Shelf pieces are cut out with a CNC router (fig 9.1). Labels are not milled into the wood but are written in pencil on each cut piece in accordance to the layout template. The pieces are then finished sanded until they are smooth to the touch. Following sanding, a natural finish is applied to the surfaces while flat.

Assembly

Each piece is slotted together and glued at the corners (fig 9.5, fig 9.6). This step of the B-Shelves manufacturing process involves a higher degree of hand craft than the rest of the process. Yet the manufacture of multiple shelves using an assembly line process is fast in comparison to a completely custom, traditional process. B-Shelves are entirely fabricated and assembled in a factory and then shipped to the individual consumer.

Fig. 9.5 Interlocking slot detail and assembly

Fig. 9.6 Shelf assembly glue up.

Fig. 9.7 Intersecting and emerging joinery conditions

10. CONCLUSIONS

Future Work

The B-Shelves thesis lays out a framework for a complete product customization scenario. The opportunity exists for B-Shelves to be transformed from a research project to a consumer grade product. If this route is taken, additional work can be done with each portion of the framework, form generation, interface and fabrication.

Additional work can be devoted to making sure the designer's intent is clearly described in the form finding program. Research could be continued in a number of directions. Earlier in the thesis, it was suggested that a hybrid system using both the force/physics simulation and a shape grammar system has potential for producing desirable configurations. This could be looked into. Currently the system produces structurally instable shelving configurations from time to time. This occurs partly because it is possible for a consumer to submit a final configuration before all boxes have settled (landed on the ground plane and intersected with at least one other box). A simple check to detect for settled situations could solve many of the structural instability problems. For other structural issues, methods to address highly cantilevered or unstable box configurations could be worked into the generation system. The structures, made up of stacking and interlocking boxes, are inherently robust and it would be possible to make sure that the majority of outcomes are stable without the need for complex systems such as a Finite Element Analysis (FEA) simulation.

Including an accurate visual description of the product in the interface would add to the customization experience. Expanding the form generation program to include live 3D modeling and rendering (programmed in Web GL for example) would be one option of how to achieve this. With a live 3D visualization, finishing details as well as material depth can be visualized in tandem with the customization process. Another way to achieve this would be to submit a final rendering to the customer once they have completed their B-Shelves configuration. Yet this solution would not have as great of an effect on the actual process of customization since feedback, from the 3D visualization, would occur late in the process.

For the manufacture of a consumer grade product, another round of prototyping as well as another look at construction detail designs could be carried out. Given more time, each step within the current fabrication and assembly process could be refined. This has the potential of creating a more automated manufacturing process that is suited for large scale production.

Additionally, the three part B-Shelves framework could be expanded to include a fourth component, shipping. Attributes of shipping are strongly connected to assembly but distinct enough that they could be explored on their own. Methods for shipping a large piece of furniture tie into decisions made about construction details and assembly. Tradeoffs between different shipping options (i.e. fully assembled versus flat pack) can be examined along with the way each option affects the design and manufacture of construction details. For instance an alternative method of shipping B-Shelves products where pre-assembled boxes would be nested inside of each other could be further analyzed. This scenario would lead to a redesign of slot details and assembly processes to allow for the initial gluing of individual boxes and for each box to slot into each other in a final assembly step carried out by the consumer.

Finally, additional functionality could be built into the B-Shelves process. This could include expanding the overall system goals. For instance, the idea for creating and maintaining storage spaces sized for particular items as described earlier in the thesis could be reevaluated.

Conclusion

Ideas of what constitutes a design experience are expanding. Different opportunities for designers and consumers are opening up as control over design decision making is no longer entirely in the hands of the architect (Huang and Krawczyk, 2007; Nordin et al., 2011). Who has control when, and how control is exercised in different situations is an interesting question. The thesis examines a methodology for designing a product, B-Shelves, which is able to be customized by many people. In doing so it considers the relationship between design intent, generative design, mass customization, and digital fabrication.

BIBLIOGRAPHY

Alexander, Christopher. 1964. *Notes on Synthesis of Form.* Cambridge: Harvard University Press.

Alexander, Christopher, Sara Ishikawa, Murray Silverstein. 1977. *A Pattern Language*. New York: Oxford University Press.

Ambrose, Michael, A., Benjamin Callam, Joseph Kunkel, Luc Wilson. 2009. “How to Make a Digi-Brick” In Proceedings of CAADRIA, Yunlin, Taiwan, April 22-25, 2009.

Archea, John. “Puzzle Making: What Architects do when No One is Looking.” In *Computability of Design,* edited by Yehuda E. Kalay, 37-52. New York: Wiley-Interscience, 1987.

Barros, Mario, Jose P. Duarte, Bruno Chaparro. 2011. “Digital Thonet: An Automated System for the Generation and Analysis of Custom- Made Chairs.” In Proceedings of eCAADe, Ljubljana, Slovenia, September 21-24, 2011.

Bonwetsch, Tobias, Daniel Kobel, Fabio Gramazio, and Matthias Kohler. 2006. "The Informed Wall: Applying Additive Digital Fabrication Techniques on Architecture." in Proceedings of ACADIA, Louisville, Kentucky, October 12-15, 2006.

Corser, Robert, ed. 2010. *Fabricating Architecture: Selected Readings in Digital Design and Manufacturing*. New York: Princeton Architectural Press.

Davis, Stanley, M. 1987. *Future Perfect.* Boston: Addison-Wesley.

Duarte, Jose, P. 2003. “A Discursive Grammar for Customizing Mass Housing: The Case of Siza’s Houses at Malagueira.” In Proceedings of eCAADe, Graz, Austria, September 17-20, 2003.

Flat Clock. 2012. Accessed February 15, 2012. www.flatclock.com.

FlatPak House. 2005. Accessed October 3, 2011. www.flatpakhouse.com.

Fluid Forms: Your Unique Designs. 2009. Accessed October 26, 2011. www.fluid-forms.com.

Ganter, M., Marchelli, G., Storti, D., Trainor, M. 2011. Accessed November 20, 2011. www.open3dp.me.washington.edu.

Gilmore, James H., Pine, Joseph. 2000. *Markets of one: creating customer-unique value through mass customization.* Cambridge: Harvard School Press.

Hensel, Michael, Achim Menges, Michael Weinstock. *“Emergence: Morphogenetic Design Strategies.” Architecture and Design 2004.*

Hernandez, Carlos. 2005. “Symetry, Rules, and Recursion: How to Design like Santiago Calatrava.” In Proceedings of eCAADe, Lisbon, Portugal September 21-24, 2005.

Hoisl, Frank, Kristina Shea. 2010. “A 3D Spatial Grammar Interpreter Applet” Paper Presented at Shape Grammar Implementation: From Theory to Usable Software Conference. Leeds, UK, July 11, 2010.

Huang, Joseph, C., Krawczyk, Robert. 2007. “A Choice Model of Consumer Participatory Design for Modular Houses.” in Proceedings of eCAADe, Weisbaden, Germany, 2007.

Iwamoto, Lisa. 2009. *Digital Fabrications: Architectural and Material Techniques.* New York: Princeton Architectural Press.

Johnson House Design. 2012. Accessed March 4, 2012. www.johnsonhousedesign.com.

Johnson, Lauren. 2011. “InterLattice: A Parametric Joinery System.” in Proceedings of PUARL International Conference, Portland, Oregon, October 28-30, 2011.

Kalay, Yehuda, E. 2004. *Architecture’s New Media: Principles, Theories, and Methods of Computer-Aided Design.* Cambridge: The MIT Press.

Kieran, Stephen, and James Timberlake. 2003. *Refabricating Architecture: How Manufacturing Methodologies are Poised to Transform Building Construction.* New York: McGraw-Hill Companies Inc.

Kilian, Axel. 2010. "Fabrication of Partially Double-Curved Surfaces out of Flat Sheet Material Through a 3D Puzzle Approach." in Proceedings of ACADIA, New York, NY, October 21-24, 2010.

Kirsch, Joan L., Russell A. Kirsch. 1986. “The Structure of Paintings: Formal Grammar and Design.” *Environment and Planning B* 13: 163 – 176.

Kirsch, Joan L., Russell A. Kirsch. 1989. “Computer Grammars for the Syntactical Analysis of Paintings.” In *World Art: Themes of Unity in Diversity, vol. 1,* edited by Irving Lavin. University Park: The Pennsylvania State University Press.

Knight, Terry, W. 1994. *Transformations in Design.* Cambridge: Cambridge University Press.

Kolarevic, Branko, and Kevin Klinger, eds. 2008. *Manufacturing Material Effects: Rethinking Design and Making in Architecture.* New York: Routledge.

Lynn, Greg. 1999. *Animate Form.* New York: Princeton Architectural Press.

Mini. 2011. Accessed October 3, 2011. www.miniusa.com.

mTable. 2008. Accessed October 17, 2011. www.gramaziokohler.com/web/e/projekte/17.

Nike iD. 2011. Accessed October 3, 2011. www.nikeid.nike.com.

Narahara, Taro. 2011. “Agent Based Computing.” in Proceedings of PUARL International Conference, Portland, Oregon, October 28-30, 2011.

Nordin, A., Hopf, A., Motte, D., Bjärnemo, R., & Eckhardt, C.-C. 2011. “Using genetic algorithms and Voronoi diagrams in product design.” *Journal of Computing and Information Science in Engineering.*

Nordin, A., Motte, D., Hopf, A., Bjärnemo, R., & Eckhardt, C.-C. 2010. “Complex product form generation in industrial design: A bookshelf based on Voronoi diagrams.” in Proceedings of 4th Design Computing and Cognition Conference - DCC'10, pp. 701-720. Dordrecht: Springer.

Perez, Santiago, R. 2008. “Crafting Complexity: Material / Procedure / Form.” In Proceedings of ACADIA, Minneapolis, Minnesota, October 16-19, 2008.

Ponoko. 2010. Accessed October 3, 2011. www.ponoko.com.

Ramsey, Charles, G., Harold R. Sleeper. 2000. *Architectural Graphic Standards.* 10th ed. Hoboken: John Wiley & Sons, Inc.

Schodek, Daniel, Martin Bechthold, James Kimo Griggs, Kenneth Kao, Marco Steinberg. 2004. *Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design.* Hoboken: John Wiley & Sons, Inc.

Simon, Herbert. A. 1969. *The Sciences of the Artificial*. Cambridge: The MIT Press.

Stiny, George. 1980. "Introduction to Shape and Shape Grammars" *Environment and Planning B* 7: 343 – 351.

Stiny, George, James Gips. 1972. “Shape Grammars and the Generative Specification of Painting and Sculpture” in Proceedings of C V Freiman (ed) *Information Processing 71*, Amsterdam. Republished in Petrocelli O R (ed) 1972. *The Best Computer Papers of 1971*. Philadelphia, Aurbach: 125-135.

Terzidis, Kostas. 2006. *Algorithmic Architecture.* Burlington: Architectural Press.

Tetrad. 2005. Accessed December 4, 2011. www.bravespacedesign.com/cat_tetrad.

Wonka, Peter, Michael Wimmer, Francois Sillion, William Ribarsky. 2003. “Instant Architecture.” ACM Transactions on Graphics. volume 22. number 3. pages 669 - 677. July 2003.

Vita. 2009. Accessed January 18, 2012. www.mdfitalia.it.