THE DESIGNOSAUR AND THE FURNITURE FACTORY

simple software for fast fabrication

YEONJOO OH, GABE JOHNSON, MARK D. GROSS, ELLEN YI-LUEN DO
{yeonjoo, johnsogg, mdgross, ellendo}@cmu.edu
CFA 201, Carnegie Mellon University, Pittsburgh PA 15213 USA

Abstract. We describe two domain oriented design tools that help novice designers design three-dimensional models that they can build using rapid manufacturing equipment. By embedding domain and manufacturing knowledge in the software and providing a sketching interface, novice designers can acquire and practice skills in modeling and manufacturing, without first having to master complicated CAD tools.

1. From sketching to fabrication: simple design tools for making things

1.1 INTRODUCTION

We want to make it easy for ordinary people, especially children, to design and manufacture three-dimensional models using planar components as an entrée to learning to design. We believe that the experience of designing and making things is a powerful vehicle for learning. For many people, designing and making something can be rewarding and engaging, and, we think, can motivate more general learning in science, technology, engineering, and mathematics. Further, we think that in its own right, designing is an intellectual capacity that, once acquired in one domain, can be widely applied. Sadly, many people view design as an innate talent for creativity that they lack. Our project to build lightweight software to engage young people and naïve users in designing and manufacturing simple 3-D models aims to open the door to design and the rich universe of learning that design affords.

Three-dimensional physical models are powerful devices that help people see and understand designs. One can hold a physical model in the hand, take it apart, and reassemble it, perhaps in different ways. This ability to interact physically with a model and its parts is important, we think, for thinking about a design; and the experience of designing with 3-D models teaches spatial skills that designers cannot easily acquire through other means such as drawing or computer graphics modeling.

Making models in the traditional way demands considerable manual skill and dexterity, for example cutting wood parts with a razor knife. The advent and adoption of rapid prototyping and manufacturing (RPM) machinery has made it possible for ordinary designers, students, and even children, to produce physical artifacts using computational means. Although most RPM hardware is as simple to use as a printer, the software tools that designers use to produce representations for output require a great deal of expertise. To produce a 3-D model designers must create a computer graphic representation. Typically designers do this using powerful general-purpose CAD modeling tools that impose a significant learning curve. Requiring of professional designers this degree of sophistication and expertise may be acceptable; however, the tools bar entry to casual and novice users.

Our goal in this work is on one hand to explore and populate the space of computationally enhanced craft activities for teaching and learning design; and on the other hand, to develop interaction techniques appropriate to the domain and the intended users. We aim to exploit characteristics of specific design domains to build relatively simple special-purpose modeling tools that novice users can easily and quickly learn. The computational design environment can help a designer by building in some requirements of the manufacturing method (e.g., the method of joining parts and sizing of joint features) as well as circumscribe the design domain through the methods and options the tools present. Our method is to explore this territory through constructing working prototypes that illustrate ideas and possibilities.

1.2 FURNITURE AND DINOSAUR MODELS

We describe two projects that explore this territory of easy-to-use domain-oriented tools that help naïve users design three-dimensional models. Using the Furniture Factory users sketch, model, and produce a class of furniture models. The Designosaur enables its users to make models of dinosaurs, both real and imaginary. Both projects enable users to begin by sketching a design and both employ a laser cutter as the output medium, to cut model materials from flat material such as wood or plastic sheets. Both use embedded knowledge about the design domain and manufacturing process to implicitly help users make design decisions or add detail to the model.

Figure 1 shows a store bought model furniture kit. It is packaged as flat panels with pre-cut parts that can be punched out and assembled into furniture models. A simple joining system (note small holes and slots in the laid-out parts in Figure 1) keeps the assembled parts together without glue or additional hardware. If permanence is desired, the parts can be glued and the model can be painted. Some furniture models are cleverly designed. For example, the cabinet doors in the piece shown below are hinged to open and close: The small tabs on the top and bottom of the two door parts (bottom right) fit into holes in the two lower horizontal shelves (bottom middle).
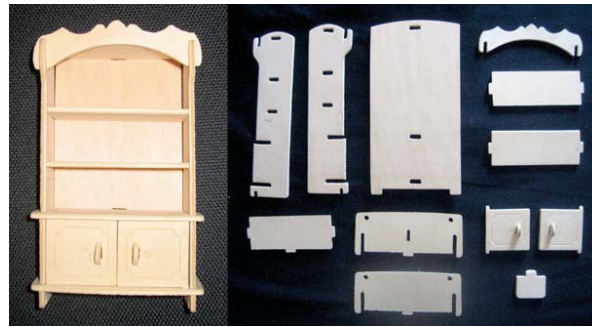

Figure 1. A furniture model kit packaged as flat panels of punch-out parts with notches and tabs for joints.

Figure 2. Dinosaur model made from a kit.

Similar in manufacture to the miniature furniture kits are off-the-shelf dinosaur kits. These consist an assortment of wooden "bones" that can be put together as though it were a puzzle. (Figure 2) It is satisfying to bring form to a three-dimensional creature by putting together bits of wood. Like the furniture kits, all the parts are planar; they join orthogonally using a paired notching scheme to make a three-dimensional structure.

As cute (or terrifying) as these models are, a drawback to both these kits (and other off-the-shelf wood models) is that you are limited to assembling a figure that someone else has designed. When you are finished, you have a dinosaur skeleton that looks good sitting on a bookshelf, or a set of furniture that you can put in your dollhouse.

To be sure there is satisfaction in successfully assembling a model; and some understanding to be gained in seeing how the flat parts go together to make a 3-D form. Yet after a few successful attempts, one begins to think about how a model could be changed: What if you could create something that might not have ever existed, such as a three-headed raptor with wings? What if you could design your own cabinet or table, instead of accepting the stuffy conventional models that come in the box? In short, with a store-bought model you are limited to the manual labor of assembling the parts leaving no room for you to participate as a designer.

Nothing prevents an enterprising young designer from going to the local art-and-craft supply store and buying a sheet of basswood, and cutting out a model of his or her own design. However, the step from purchase-and-assemble to building-from-raw-materials is large: The designer must decide and draw the parts, determine how they will join, then carefully cut them from the sheet. Tolerance errors have severe consequences; a small mismatch between notch size or position and material thickness means the model won't assemble correctly. The process also requires considerable dexterity with a razor knife, a skill that takes some time to acquire. All these are, of course, part of designing and manufacturing a wood model in the traditional way. It may be argued that eliminating these components "dumbs down" the process and eliminates learning opportunities. On the contrary, we believe that at least for beginning designers it is important to put design decision making first, and by providing scaffolding for decision making and manufacturing, we can draw them into the game.

1.3 RELATED WORK

Our project relates to research in several areas: learning through design, feature modeling and design assembly, and sketch-based 3-D graphics.

Activity learning theory stresses the importance of doing things to learn (Blust and Bates, 2004). People engage more in learning when they act than when information is presented for their passive consumption. Problem-based learning is a widely cited example, in which teams of students work together to address real-world problems. The studio model of learning in

architecture and product design depends on hands-on experience for students to acquire knowledge and skills with the guidance of a design instructor. Others argue that design is a powerful vehicle for learning mathematics, using Escher's drawings or quilt pattern design as domains (Shaffer, 1997, Lamberty and Kolodner, 2002).

We follow in the "constructionist" tradition of mathematician and educator Seymour Papert and his students and colleagues at MIT. In developing the Logo programming language for children Papert argued that programs children write function as mathematical 'objects to think with' and that their concreteness affords a powerful kind of learning (Papert, 1980). Later work of Papert's former students and colleagues (e.g. Mitchel Resnick's group at MIT) has explored and extended this idea, moving beyond the inner world of software into the hybrid world of physically embodied computation (Resnick et al., 1998). Michael Eisenberg and his craft technology group, our collaborators at the University of Colorado, have explored a variety of computer assisted physical construction and craft activities, beginning with computational tools for designing and making polyhedron models out of paper (Eisenberg and Eisenberg, 1997), and more recently mechanical automata toys (Blauvelt et al., 2000) and electronically embedded quilts and clothing (Buechley et al., 2005). We have seen an explosion of interest in professional architecture circles and among architectural educators in "digital fabrication", the use of rapid prototyping and manufacturing to produce buildings and models (Kolarevic, 2004, Sass, 2005, Iwamoto, 2004). Neil Gershenfeld (Gershenfeld, 2005) discusses the more general implications of these technologies.

The second area of related work is feature modeling and design assembly. The two projects described here are reminiscent of the feature-based modeling and design for assembly or design for manufacturability research prevalent in the late 1980s and 1990s in the AI and Design community (Dixon et al., 1987, Shah, 1991, Shah and Mantyla, 1996). Jung and Nousch (2000) describe a system for designing and fabrication of furniture using computer graphics modeling. Their BEAVER program allows users to construct closets of various types and sizes in a 3-D computer graphics environment using parametric modeling. It details all necessary parts and hardware and it creates assembly instructions with text and step-by-step drawings. Also in the domain of furniture parts assembly, Agrawala et al. (Agrawala et al., 2003) studied people following printed assembly instructions and diagrams and based on this study described principles for better diagrams. Their system automatically generates appropriate assembly diagrams according to these principles.

The third area of related work deals with creating 3-D models from 2-D sketches. Recently software has been developed that can construct a 3-D object from a 2D sketched drawing. For example, Teddy (Igarashi et al., 1997) models 3-D curved surfaces (such as a Teddy bear) from 2D freeform strokes. It creates a closed polygon from a stroke and finds a spine in center of the polygon. Next, it creates a 2D triangular mesh between the spine and the polygon perimeter, and then raises each spine vertex and smoothes the mesh. Digital Clay (Schweikardt and Gross, 2000) converts 2D representation of edges into a 3-D Model using the Huffman-Clowes scheme to label each edge in a drawing as a concave or convex junction of faces. 3-D Journal (Lipson and Shpitalni, 2002) constructs a 3-D model from 2D strokes and allows a user to change viewpoint while sketching. First it calculates the angles of all segments in a sketch and identifies a prevailing axis system. It then assigns all lines with the identified axis system, tolerating inaccurate vertex positions, inaccurate connections between segments and sketches with missing lines. The axially-aligned planes approach of (Varley et al., 2005) inflates a wireframe drawing to a 2-1/2 D model by labeling lines as convex and concave, determining pairs of parallel lines, and calculating a Z-coordinate value for each vertex. In Furniture Factory, we assume a prevailing 3-D axis system and match lines in the sketch to axial lines—similar to the techniques of Lipson and Varley.

2. Furniture Factory

The Furniture Factory program is a proof-of-concept working prototype built to explore how to help designers construct physical prototypes using rapid prototyping and manufacturing machines. It provides a sketch-based design interface that a designer can use to draw furniture in 3-D. The design is then displayed in an isometric viewing window where the designer can view it and edit it. The program then decomposes the 3-D model into flat panels that are displayed in the parts window. Furniture Factory adds joints where one panel connects to another according to connection conditions. These added joints enable designers to construct the physical model easily and quickly. The program then generates HPGL code to cut the furniture parts on a laser cutter. Designers construct their furniture by assembling the cut parts.

Furniture Factory has four components that share a data structure: Sketch Interface, Geometry Analyzer, 3-D Representation, and Joint Creator. Figure 3 shows the information flow among these components After a designer sketches the isometric view of furniture in the Sketch Interface, Geometry Analyzer analyzes the sketch and computes faces, edges and 3-D coordinates. These computed faces, edges, and their 3-D coordinates are stored in the data structure. 3-D Representation creates a 3-D model using information extracted from the designer's isometric sketch and displays it. The Geometry Analyzer also decomposes the 3-D model into 2D geometries and analyzes the connection conditions. These decomposed geometries are also stored in Data Structure. Joint Creator adds necessary joints according to the identified connection conditions.
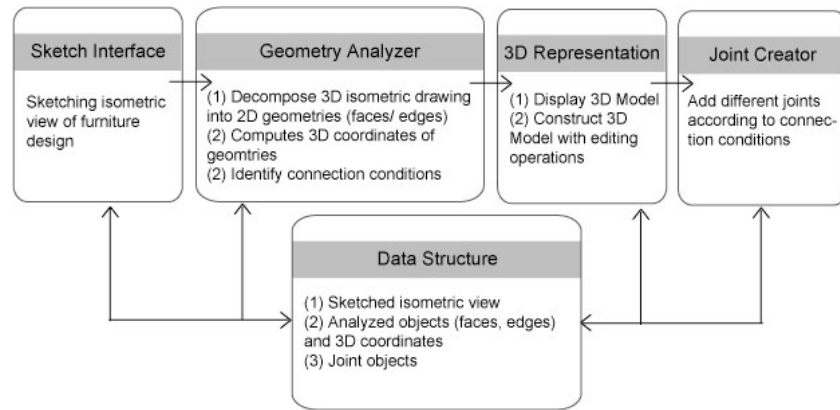

Figure 3. Components of Furniture Factory

The Furniture Factory user interface employs three windows. The <u>Sketch</u> window displays freehand sketches the designer draws with a tablet and stylus (Figure 5-a). The <u>3-D</u> window has two functions: to enable the designer to view the design and to edit its parts. Figure 5-b shows a 3-D view of the bookshelf generated from the sketch. These parts can also be moved and rotated. The <u>Parts</u> window displays the parts and their joints (Figure 9-a). The rest of this section describes these components and their roles in Furniture Factory.

2.1. SKETCHING INPUT

In Furniture Factory's freehand sketching environment designers use a stylus and tablet to make freehand isometric sketches to create 3-D models (Figure 5-a). Our simple sketch interface recognizes three types of lines: horizontal, vertical, and diagonal. Following isometric conventions each type of line is assigned to one of the Cartesian coordinate axes: horizontal line to the Y-axis, vertical line to the Z-axis, and diagonal line to the X-axis. Figure 4 illustrates the relationship between 2D directions of lines and the 3-D coordinate axes.
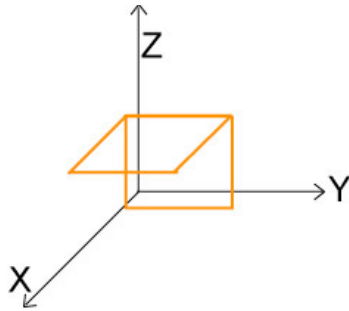
Figure 4. Relationship between three types of lines and 3-D Cartesian coordinates.

2.2. 3-D REPRESENTATION

Furniture Factory provides the designer with a 3-D display window in which the sketched model converted to 3-D data objects is displayed using an OpenGL viewer (Figure 5-b). The 3-D representation displays a 3-D model of a sketched artifact. The designer can rotate the model to visually evaluate the furniture design (Figure 5-b).
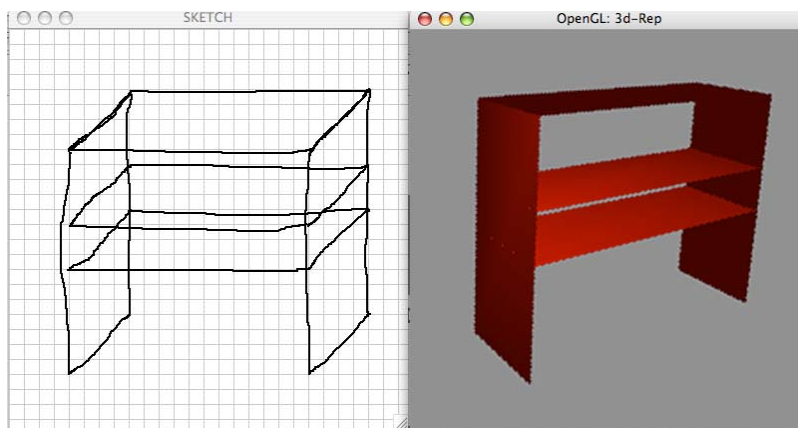

Figure 5. Sketching Input and 3-D Representation: (a) Sketched isometric model of furniture; (b) 3-D Representation window displays 3-D representation

2.3. DATA STRUCTURE

Furniture Factory stores three kinds of data objects: faces, edges, and joints. Figure 6 shows the objects, their attributes, and their relationships. As the designer makes the sketch drawing, the Geometry Analyzer generates descriptions of the 3-D edges and faces and links them in the data structure to reflect the configuration of the furniture object.
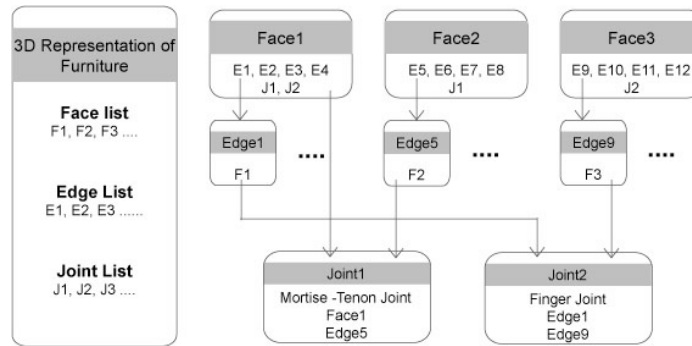
Figure 6. Objects and relationships: Each 3-D representation stores interconnected faces, edges, and joints

2.4. GEOMETRY ANALYZER

Geometry Analyzer performs three functions: analyze the designer's isometric sketch, decompose it into basic elements (faces and edges) and determine how the faces and edges connect. First, Geometry Analyzer takes the isometric wire frame drawing from the Sketch Interface. Considering the wire frame drawing as a graph in which lines are edges and nodes are vertices of the 2D drawing, the Geometry Analyzer identifies simple cycles that contain no edges as faces and assigns them 3-D coordinates, using the isometric axis assignment heuristic mentioned above. Second, the Geometry Analyzer transfers each part elements into 2D coordinates to cut on a laser cutter. Third, the Geometry Analyzer identifies where and how these faces and edges connect to each other. The current Geometry Analyzer identifies two kinds of connection conditions: face – edge and edge – edge. Figure 7 illustrates two connection conditions.
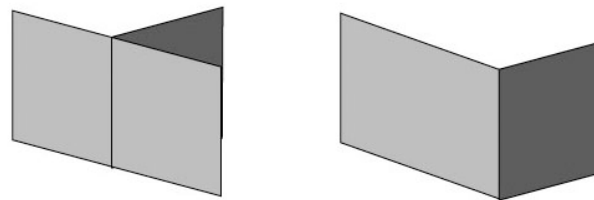

Figure 7. Two Connection Conditions: (a) Face::Edge: one face abuts an edge of another face (b)
Edge::Edge: when one edge forms a corner with an edge of another face.

2.5. JOINT CREATOR

Joint Creator adds joints into the furniture parts according to the identified connection conditions. Currently the Joint Creator supports two types of joints: (1) mortise-and-tenon joint and (2) finger joint. When one face connects to one edge of another face, the Joint Creator adds a mortise-and-tenon joint. When two faces connect at their edges in a corner, the program adds finger joints. Figures 7 and 8 illustrate connection conditions and joint types.
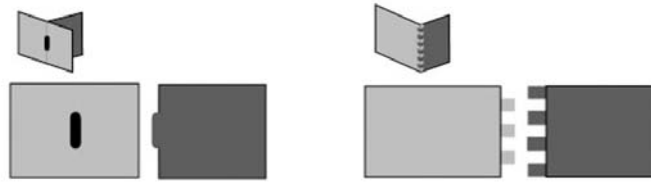
Figure 8. Connection Conditions and Joint Types: Where a face and an edge connect, Furniture Factory creates a mortise & tenon joint. Where two edges connect in a corner the program creates a finger joint.

2.6. CUT AND ASSEMBLY

The Furniture Factory then displays the 2-D parts and their newly created joints in the Parts window. The Furniture Factory labels the faces and edges to help the designer assemble their furniture. For example, it labels two faces of a bookshelf "Bottom" and "Back". These two faces meet at a joint the system names "e". The system labels the two meeting edges – "e1" (Bottom) and "e2" (Back). Users can understand how to assemble parts easily using these labels. Then Furniture Factory generates HPGL code to cut the furniture parts on a laser cutter ready to assemble. Figure 9 shows the drawings of parts, the cut panels and assembled furniture.
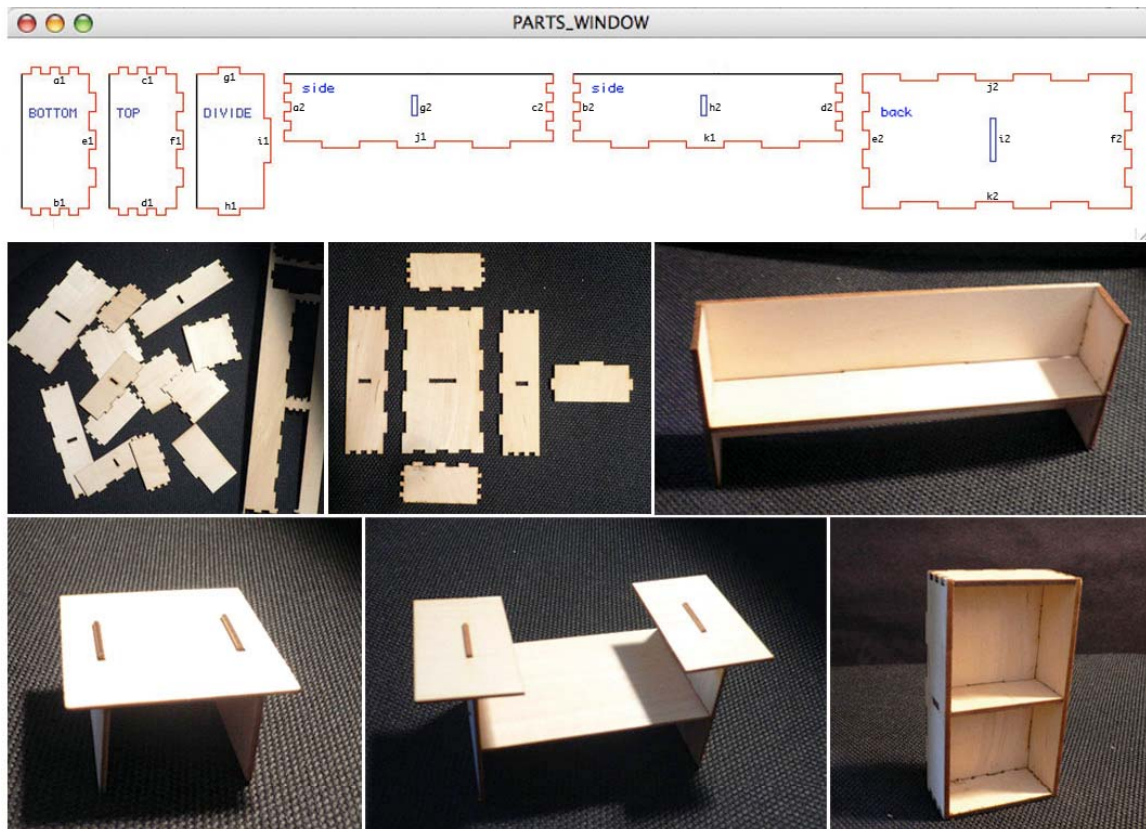

Figure 9. Drawing and Parts on a Laser Cutter and Assembled Furniture: (a) Cut Drawing is composed by labelled parts and joints (b) The user assembles these parts and construct physical furniture model.

3. The Designosaur

3.1 SKETCHING BONES

The Designosaur is a simple modeling program that empowers people to create their own models of dinosaurs or dinosaur-like skeletons. A Designosaur creature is assembled out of pieces of thin wood, plastic, foam, or other material. The pieces fasten together using a notching system with identical notches; pieces slide together in 'X' fashion such that their planes are orthogonal. Figure 10 shows a dinosaur leg bone sketched freehand in the Designosaur modeling program.
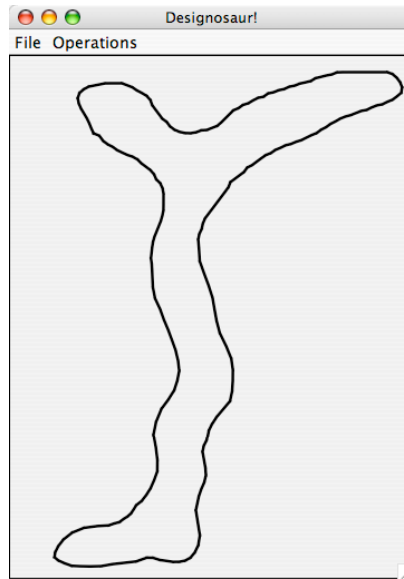

Figure 10. A dinosaur hind leg bone sketched in the Designosaur modeling program.

3.2 BONES ARE CONNECTED BY NOTCHES

In Designosaur, designers draw individual bones and indicate where other bones attach by specifying notch points. A notch is a joint that connects exactly two bones. Notch parameters (width and depth) are provided later on, when the designer is ready to produce a physical model on the laser cutter. For this reason, notches are displayed only as thin markers. In figure 11-a, the designer has drawn a vertebra bone, and inserted a notch along the bottom surface. The notch symbol points inward, indicating the direction of the cut. Figure 11-b shows the notch detailed with width and depth appropriate for the chosen material, ready to print in an HPGL file.
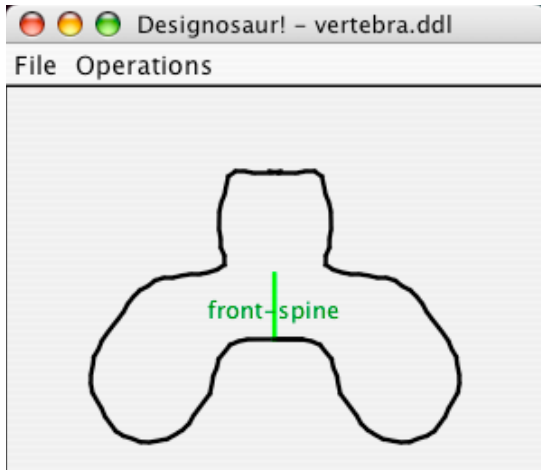
Figure 11-a: Using the Designosaur's Bone Editor to create a vertebra. A thin line marks the location of the notch.
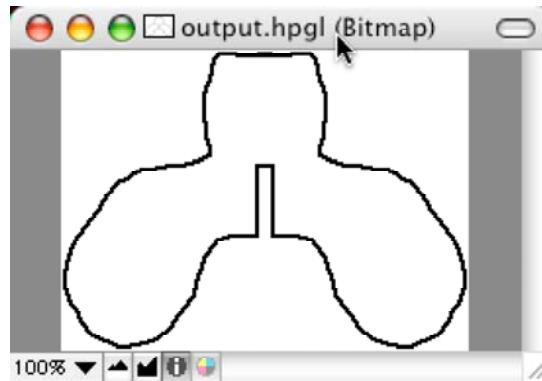
Figure 11-b: The same vertebra rendered in HPGL for cutting. The notch has been assigned a width and depth according to the chosen material.

If a notch knows which two bones it joins together, we can use this information to assemble a list of bones for printing, and also to construct a 3-D graphics model of the skeleton. The designer can identify the mating parts (by selecting from a menu of already-known parts) when notching a bone, or later in the process if the mating bone part has not yet been designed. The label appears as soon as the mating information has been added. Figure 14-b shows a sketched dinosaur spine, with five notches. One notch is labeled "head connector" and the other four are labeled "vertebra". The label next to each notch indicates the name of the mating part that the notch will join in the assembled model. Similarly, Figure 14-c shows a symmetric vertebra with a notch labeled to show that the vertebra connects to the front spine bone

3.3 SYMMETRY / MIRROR MODE

Many bones are symmetric, so the Designosaur supports symmetry in bone design by providing a 'mirror mode'. When the designer enters mirror mode, a vertical line is positioned down the centre of the drawing area. All the designer's strokes are duplicated on the opposite side of this mirror line. This helps with making bones such as ribs or vertebrae. The hip bone in Figure 12 was sketched in mirror mode.
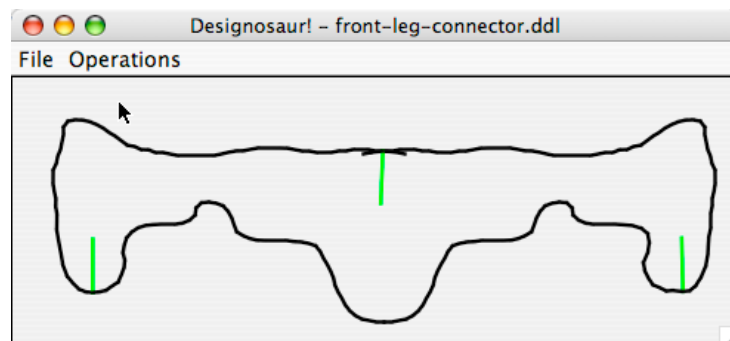

Figure 12. A front leg connector (hip), drawn using the mirror mode.

3.4 CONNECTION LINKS

The connection information that the designer provides by labeling notches with their mating parts

enables the Designosaur program to construct a graph data structure that represents the connectivity of the bones in the model (Figure 13).
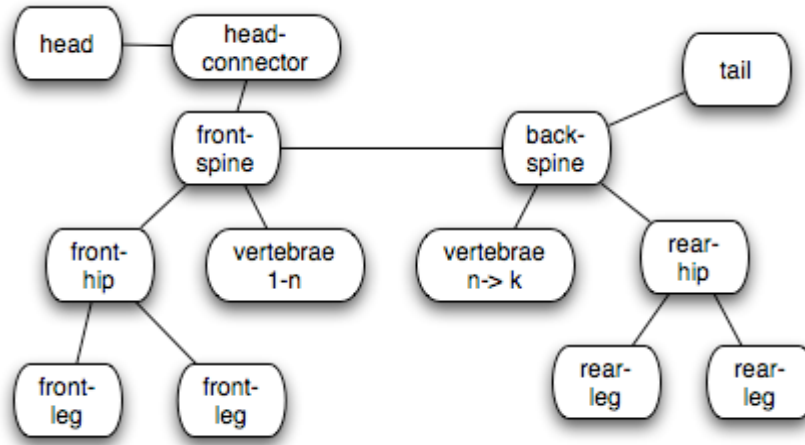

Figure 13.Structure of a dinosaur model

3.5 NOTCH REGIONS

A dinosaur spine generally has a number of attached vertebrae. In order to help evenly distribute notch points for attaching these bones to the spine, the designer can specify 'notch regions'. The designer indicates how many notches to place along a given path, and the Designosaur automatically distributes the attachment points with even spacing. Figure 14-b shows a spine to which the designer has applied four notches using a parameterised notch region.
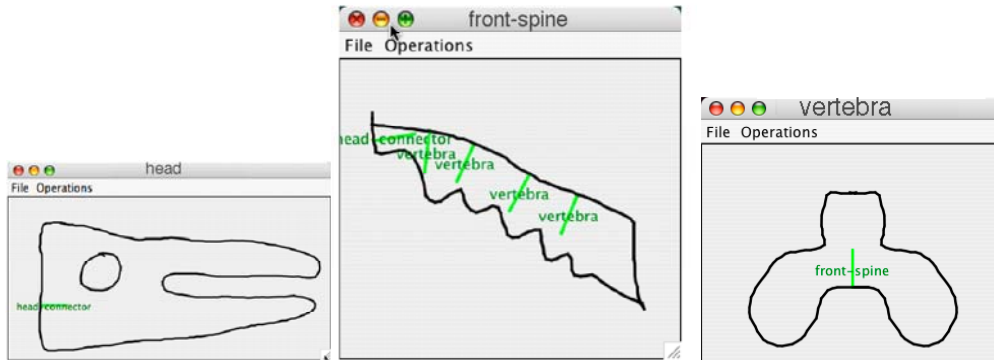

Figure 14. Three bones and their relationships (shown by labelled notches), (a) head, with a notch for a head connector (not shown); (b) front-spine with four vertebrae notches and a notch for the head connector; (c) symmetric vertebra with notch for spine.

3.6 SYSTEM ARCHITECTURE

The Designosaur interprets sketch input, storing each stroke as custom point objects in a bi-directional linked list. The linked list structure provides a convenient method for traversing each stroke forwards or backwards. The custom point objects store location as well as notch data and they provide methods to get information about the stroke at that point, such as curvature or distance along the line to another point.

When the designer positions a notch, the point closest to the indicated location is flagged as having a notch that extends inward from the bone edge at that point. A notch is always normal to the edge. As there are two possible orthogonal directions to the stroke at any point, the user

disambiguates direction by specifying a left- or right-handed normal vector. If the designer later edits the surface of the bone and moves the notch point, the left- or right-handedness remains constant, and the notch moves along with the stroke in an intuitive manner.

3.7 MANUFACTURING

After the bones and their connections have been designed, it is time to print the dinosaur using a laser cutter. The width of notches depends on the material of manufacture. For example, we have experimentally determined that 4mm-thick foam core media (a rigid art material consisting of a sandwich of foam plastic between two paper faces) requires a notch that is 18 pixels wide on-screen. When printing bones, the Designosaur asks how thick the material is and uses that number to "notchify" (add notches to) the shape.

Figure 15. Physical dinosaur parts: The foam core sheet from which the pieces are cut sitting on top of the laser cutter. Below, unassembled dinosaur bones.

After notches have been inserted into bones, the Designosaur generates an HPGL file that specifies points for the laser cutter to begin, where to move, and where to stop cutting. Figure 15 shows the cut pieces on the laser cutter. We have found that spongy media such as foam core works better than rigid media such as wood because the pieces tend to adhere more tightly. Figure 16 shows these pieces assembled into a dinosaur model made of foam core.

Figure 16. Made with the Designosaur: The authors created this dinosaur using the Designosaur system and a laser cutter. Compare with store-bought model, Figure 2.

4. Discussion & Future Work

4.1 DISCUSSION

We do not intend the Designosaur or the Furniture Factory as general-purpose design tools. Both are limited to constrained domains (furniture and dinosaur models) and a constrained manufacturing universe: assembling pieces cut from flat material using a small number of jointing techniques. For beginning designers, the limited domain and the constrained manufacturing method offer a design space that is at the same time rich in possibility yet not overwhelmingly large and complex. Both, of course, are specific instances of a more general design system in which 2-D parts are assembled using mating joints, which suggests a future effort to build a meta-modeler (a language for describing design systems) in which one can specify the Furniture Factory, the Designosaur, or other instances of flat single-material component assembly systems.

There is a delicate balance between doing the work for the designer, and requiring the designer to do the work. On one hand, if our goal were merely to enable people to produce 3-D models, we could provide a catalog of parameterized designs. A user would select a desk or a dinosaur, choose parameters, such as the overall dimensions, and then print (cut) the model. Although this satisfies the requirement of producing 3-D models and in our limited universe it might be sufficient to cover the design space, this would deny users the experience of designing.

On the other hand, a designer could simply draw model parts directly, using a simple 2-D drawing program. As the models are all cut from flat material and the files sent to the laser cutter are 2-D drawing files, this wouldn't be too difficult, at least in principle. However, the designer would need to add jointing details to the parts. Although conceptually simple, it is a tedious task to get the sizes and positions of the joints exactly correct. The Furniture Factory also provides the user with a 3-D view of the assembled design unavailable in a 2-D drawing program.

Our projects contribute to the field of personal fabrication by providing sketching interfaces and intelligent design support, and we hope to gain insight into creative processes and learning through developing personal fabrication technology. Striking the balance between preserving simplicity while supporting creativity is a challenging goal for this research. We described here two example systems that can be considered instances of a more general design system. This general design system comprises three parts: (1) sketching input and recognition, (2) inference,

interpretation and design decision support, (3) transformation and preparation for machine fabrication of parts. Our projects are still proof-of-concept working prototypes that we hope to develop into more robust systems that we can evaluate with end users; at this point, though, our assessment of these prototypes has been solely in the nature of formative evaluation leading to incremental improvements of the human-computer interface.

4.2 EXTENDING THE FURNITURE FACTORY AND DESIGNOSAUR

We plan to add additional jointing conditions to Furniture Factory, which will allow designers to construct more complex objects than those allowed by the current finger and mortise-and-tenon joints. A richer vocabulary of joints could also allow the designer to select among alternative jointing methods: in the current version the program determines the jointing scheme, although the designer can override the system. We also plan to add an error-checking and critiquing component, building on our previous design critiquing work (Oh et al., 2004). For example, if the designer sketches a table with different length legs, the Furniture Factory's critics could warn the designer that the table will be unstable. Beyond simple error checking, the Furniture Factory's critiquing component could retrieve interesting or relevant cases from a library of stored models.

The current Designosaur provides the ability to draw simple bones, indicate where they fit together, then generate HPGL code needed to produce the physical output. As with the Furniture Factory, simple critiquing both about the domain (the dinosaur won't stand up: move its center of gravity) and about fabrication would be a valuable addition. Beyond small improvements to the interface that we expect to identify when we test the system with users beyond graduate students in our laboratory, we plan to add a 3-D modeling graphics view, similar to that offered in Furniture Factory. Looking further ahead, we envision a sketch-based system that allows users to draw the outside shape of a creature and let the system infer internal skeletal structure. For example, a drawing of a dinosaur that has a long neck, a fat midsection, a tail, and two nubs for legs is enough for a shape recognition system to determine the general features of the creature and make some intelligent guesses about what the designer wants. The size of the ribs, for example, will conform to the curvature of the belly, so that the ribs in the middle are longer and wider than those towards the front and hind legs. Individual bones will also be malleable; the designer will be able to grab the tail and yank it away from the body in order to make it longer, and the Designosaur will add vertebrae to the spine as necessary. More speculatively, it would be possible for the designer to add mechanical joints to the dinosaur, test the kinematics in an animation on the screen, and then print out the parts and assemble a moveable model. One could even imagine adding electronics and servomotors to the model to make the dinosaur walk on its own. But that is another story …

Acknowledgements

References

Agrawala, M, Phan, D, Heiser, J, Haymaker, J, Klingner, K, Hanrahan, P and Tversky, B (2003) In SIGGRAPH 2003, pp. 828-837.
Blauvelt, G, Wrensch, T and Eisenberg, M (2000) Knowledge-Based Systems, 13, 471-478.

Blust, RP and Bates, JB (2004) Activity Based Learning - Wagons R Us - A Lean Manufacturing Simulation, In American Society of Engineering Education (ASEE) Annual Conference & Exposition: Engineering Education Reaches New Heights, Salt Lake City, UT; USA.

Buechley, L, Elumeze, N, Dodson, C and Eisenberg, M (2005) Quilt Snaps: A Fabric Based Computational Construction Kit, In IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2005), Tokushima, Japan.

Dixon, JR, Cunningham, JJ and Simmons, MM (1987) Research in Designing with Features, In Proceedings IFIP TC 5/WG 5.2 Workshop on Intelligent CAD,(Eds, Yoshikawa, H and Gossard, D) New York, pp. 137-148.

Eisenberg, M and Eisenberg, AN (1997) Journal of Computers in Mathematics and Science Teaching, 16, 477-511.

Gershenfeld, N (2005) FAB: The Coming Revolution on Your Desktop--From Personal Computers to Personal Fabrication, Basic Books.

Igarashi, T, Matsuoka, S, Kawachiya, S and Tanaka, H (1997) In UIST '97ACM, pp. 105-114.

Iwamoto, L (2004) Embodied Fabrication: computer-aided spacemaking, In Association for Computer Aided Design in Architecture: Fabrication: examining the digital practice of architecture (Eds, Beesley, P, Cheng, NY-W and Williamson, RS) ACADIA, Toronto, pp. 269-281.

Kolarevic, B (2004) Architecture in the Digital Age: Design and Manufacturing, Spon, London.

Lamberty, KK and Kolodner, JL (2002) Exploring Digital Quilt Design Using Manipulatives as a Math Learning Tool, In International Conference of the Learning Sciences, Seattle, WA.

Lipson, H and Shpitalni, M (2002) In AAAI Spring Symposium on Sketch Understanding, American Association for Artificial Intelligence, Washington DC, pp. 99-104.

Oh, Y, Do, EY-L and Mark D Gross (2004) Intelligent Critiquing of Design Sketches, In Making Pen-Based Interaction Intelligent and Natural (Ed, Randall Davis, JL, Tom Stahovich, Rob Miller, and Eric Saund) AAAI Press, Arlington, Virginia, pp. 127 - 133.

Papert, S (1980) Mindstorms: Children, Computers, and Powerful Ideas, Basic Books, New York.

Resnick, M, Martin, F, Berg, R, Borovoy, R, Colella, V, Kramer, K and Silverman, B (1998) Digital manipulatives: new toys to think with, In SIGCHI conference on Human factors in computing systems, Los Angeles, California, United States, pp. 281-287.

Sass, L (2005) Wood Frame Grammar: Cad Scripting a Wood Frame House., In CAAD Futures, Vienna, Austria.

Schweikardt, E and Gross, MD (2000) Automation in Construction, 9, 107-115.

Shaffer, DW (1997) Journal of Mathematical Behavior, 16, 95-112.

Shah, JJ (1991) Research in Engineering Design, 1991(2), 93-108.

Shah, JJ and Mantyla, M (1996) Parametric and feature-based CAD/CAM,, John Wiley & Sons, Inc, New York.

Varley, P, Martin, R and Suzuki, H (2005) In EUROGRAPHICS workshop on Sketch-Based Interfaces and Modeling (Eds, Igarashi, T and Jorge, J) Eurographics, Dublin, Ireland, pp. 99-108.