ParaShape

A Parametric Approach to Personal Design

Master's Thesis

Shadan Sadeghianborojeni

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Armin B. Cremers

Bonn. September 2013

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.
Shadan Saeghianborojeni
Bonn, 30. July 2013

Abstract

With the rise of personal fabrication, 3D printers found their way from industry to offices and homes, getting more affordable and smaller in size. This grew the range of 3D printer users from manufacturing experts, engineers, and designers to almost everyone with every level of knowledge.

Personal Design, which is the process of designing for personal fabrication[a], aims at making almost everyone with every level of knowledge capable of visualizing their design ideas into 3D models to be fabricated. The high learning curve of existing 3D modeling tools is a problem for novice home users to 3D model their desired object design. Despite the fact that in recent years several user-friendly 3D modeling tools such as SketchUp or 123d Design were released, we observed that people still find it difficult and time consuming to learn them and get a satisfying result. Another approach to achieve a 3D printable model is browsing the online 3D model libraries such as Thingiverse. However, to manipulate a downloaded 3D model, the user still needs to be acquainted with 3D modeling tools. Another problem concerns expert users who are familiar with 3D modeling. Despite what is mentioned, the problem is not just limited to the novice users, sometimes experts in 3D modeling spend a lot of time to understand the code of the model they have downloaded from a library just to make a small change, and this makes it difficult for them to reuse each others' work and they often would rather start everything from scratch.

ParaShape, which is a parametric approach to personal design, has solved these problems by providing users with a component-based modeling tool, where each model is made of its parametric composing components. Using this system, novice users can create complex 3D models by composing various components that they choose from a library, and customize them by manipulating parameter values. Moreover, ParaShape has defined component code templates, which increase 3D model code readability and code reuse.

[a] The process of creating what was produced before in factories by the consumers themselves using machines such as 3D printers or laser cutters

Contents

Figures And Tables

List of Figures

List of Tables

Acknowledgements

First of all, I would like to thank my supervisor Dipl.-Inform. René Bohne for his support and feedback.
Secondly, I would like to thank Prof. Dr Jan Borchers for giving me the chance to write my thesis at the Media Computing Group and his time and support.
I would also like to thank Prof. Dr. Armin B. Cremers and Dr. Pascal Bihler, my second supervisors at the University of Bonn for their great contribution and valuable ideas.
Besides, I would like to thank all the testers, who participated in my user studies conducted in this thesis for their time and feedback.
Special thanks to my parents and Eike for their full support during my studies.
Shadan

Chapter 1

Introduction

Creativity is not the finding of a thing, but the making something out of it after it is found.

James Russell Lowell

1.1 Industrial Revolution

Industrial revolution turned agrarian life to city life and domestic production to mass production in factories

Before 1760, which is known by many historians as the starting point of industrial revolution, people used to live in agrarian societies where all needs were met by the products of the farms or hand products of families. Homes were the place where all the products were manufactured at a scale that was always a response to the needs. The work load was divided among family members. Production process was often time consuming and slow, and required skill and great amount of muscular and other sorts of energy like water.

From 1760, machine inventions started and small factories were established. Machines replaced people and production was centralized in factories rather than houses. The lifestyle of people changed totally, they mostly were workers living in cities around factories. The mass production made everyone needless of having the skill and consuming the time and energy to create their own things.

Having access to the technology of making things by machines, set a ground for thinking how things should look and be designed to be more applicable, less costly, and more good looking. People started to expertise in designing things and developed a profession out of it "the designer". However, in the time of industrial revolution this profession was limited to a few people, who were mostly engineers working at the factories.

1.2 DIY Movement

DIY movement encouraged people to make their things without the help of experts

In the beginning of the 1970s, a new movement was started by the release of some publications[1], which were known as the response to the consumer culture, which was spread through the society as a result of the industrialization. These publications were encouraging people to repair, make and customize their things without the help of experts. DIY, which is an abbreviation for the phrase "Do it Yourself" is the term that was used from 1950s for home improvement movement, when people started to improve their houses by themselves[Tur10].
Everything that was used by DIY people to make their own things, was considered as a "tool". A tool could be anything, from gardening and carpentry tools to books, magazines, courses or any other information resource. In other words, anything that would enrich your power or skills was considered a tool.
People started to like the idea behind DIY, since most of the time it was economically efficient and it also gave them a feeling of satisfaction. People enjoyed making their own products with their own ideas using their preferred material and tools. Having the result in their hands after hours of work fulfilled their desire for having something done and gave them an internal pride of their skills.
With the spread of video cassette recorders (VCR) in homes, a lot of people started to get video lessons for DIY skills, other resources were newsletters, magazines, TV shows, and books. Nowadays, there are thousands of websites dedicated to DIY on the internet, people share ideas and write about their experiments on blogs; there is at least one example of almost everything to be done by DIY on the web.

1.3 Digital Fabrication

1.3.1 Numerical Control

Numerical control is the process of controlling a machine by coded instructions

"Numerical control (NC) is the operation of a machine tool by a series of coded instructions consisting of numbers, alphabet letters, and symbols that the machine control unit (MCU) can understand."[KGS00]
The first forms of NC goes back to beginning years of industrial revolution. In 1725 the knitting machines used punch cards to knit different patterns on the cloth.
In 1952, John Parson from Parson Corporation cooperating with Servomechanism Laboratory of MIT, released the first milling machine, which was capable of three dimensional movements of parts simultaneously. One big problem at this time was that each user was using her own specific code thus, sharing programs was impossible. This problem was solved when in early 1960s, the Electronic Industry Alliance settled a standardized version of G-Code, which was originally implemented in MIT Servomechanisms Laboratory in 1950s.
At the 1960 Machine Tool Show in Chicago, over a hundred NC

[1] Stewart Brand - The Whole Earth Catalog
Popular Mechanics -American magazine

machines were presented. Most of these machines had relatively simple point-to-point positioning, but the principle of NC was now firmly established [KGS00].

1.3.2 Computer Numerical Control

CNC is controlling machine tools by computer programs

Computer numerical control (CNC) appeared for the first time in 1950s, when John Runyon from MIT invented a method to produce the punch tapes under computer control. Given a list of points and speeds, punch tapes were automatically produced [Ros78].

In 1960s, with the reduction of the price of computers, the number of CNC machines grew rapidly. In 1970s, software based controls were introduced, which made it possible to revise and edit a part of program whenever needed. The improvements made in the interfaces of CNC machines such as menu-selected displays, advance graphics, and ease of programming made them more popular [KGS00].

1.3.3 Computer Integrated Manufacturing

CIM is is designing and manufacturing by using computer hardware and software

The term "Computer Integrated Manufacturing" was first used by Joseph Harrington in 1975, in his book called "Computer Integrated Manufacturing". CIM is the process of applying computer hardware and software to design and manufacture products. Programmability of computers promoted manufacturing to a higher level of flexibility, accuracy and less erroneous through using CIM subsystems such as CAD (Computer Aided Design), CAE (Computer Aided Engineering), and CAM (Computer Aided Manufacturing). Some historians have named this promotion an "industrial revolution" [YLYW06] [Ayr90].

1.3.4 Digital Fabrication

Digital fabrication is the conversion of digital designs to physical objects

Digital fabrication is a new face of industry, which converts digital designs in to physical objects. Digital fabricators turn bits in to atoms by getting a CAD model, and building a physical object from the material fed into them with the help of CAM; the advantage of this industry is, that the scale of production can change from few to many without a change in price. Digital fabricators are divided in to two categories: subtractive and additive [Mot11].

Subtractive Manufacturing

Subtractive manufacturing is the process of removing pieces from a block or sheet of material with a CNC machine until the desired shape is reached. This process is also sometimes called "machining". There are four categories of subtractive manufacturing: CNC milling, CNC routing, CNC waterjet cutting, and laser cutting.

Additive Manufacturing

Additive manufacturing, is the process of making physical objects from digital models, but unlike subtractive manufacturing, additive manufacturing is creating solid objects by adding layers of material upon each other. This process is also called 3D printing, as it is similar to what happens in 2D printers; the additive manufacturing tools print the 3D digital model layer by layer, where each layer is a cross section of the model. Various technologies are applied in 3D printing such as Fused Deposition Modeling (FDM), Selective Laser Sintering (SLS), direct metal Laser Sintering (DMLS), Powder bed and inkjet head 3D printing, Laminated Object Manufacturing (LOM), Electron Beam Melting (EBM), and Selective Heat Sintering (SHS).

1.4 Home Factories: The Next Industrial Revolution

with the help of home factories one can fabricate almost everything at home

As DIY people were still focusing on bringing their ideas into products, the proliferation of personal computers equipped them with many useful tools that increased their ability in implementing their ideas. With the help of CAD programs one could design anything that comes to her mind in detail. In 1990s, with the rise of the internet, everyone was able to share their ideas or learning new skills from experienced people all over the world. Plus, digital fabricators have decreased in size and in price and day by day, they are getting more affordable to a larger number of the population. Neil Gershenfeld in his book "Fab: The Coming Revolution on Your Desktop-from Personal Computers to Personal Fabrication" points to a new industrial revolution that brings factories to homes and makes everybody skilled enough to make almost anything with home fabricators [Ger05].

1.4.1 Personal Fabrication

Through personal fabrication people can induce their taste and ideas in their everyday objects

People buy thousands of things in their everyday life, admitting the price and design the market offers carrying the thoughts "I wish it was cheaper" or "I wish I could change this part" in their minds and sometimes they bump into the case that they need something for a specific application they cannot find in the market. Another case could be the spare parts of their stuff, which are outdated or not provided by the producing company. The answer to all of these, is a personal computer and a digital fabricator. After a century of mass production, many people now have access to sophisticated production tools and the knowledge to manufacture everything they need in the desired amount.
With the rise of digital fabrication, the number of online fabrication services grew, designers and artists can easily benefit from providers such as Shapeways, Ponoko or i.Materialise, which give online 3D printing services such as proving 3D model libraries, 3D modeling customizer and creator tools, and delivering printed objects to customers. They are also able to sell their designs on marketplaces offered by Shape-

ways and Ponoko. Distributed manufacturing networks like CloudFab give users the chance to find the closest shop owning the fabricators they need. Fablabs and production shops, like TechShops also provide a wide range of fabrication tool for the users to bring their ideas into things they want to fabricate [Mot11].

1.4.2 Home Fabrication

In 2004, Dr. Adrian Bowyer, professor at the University of Bath started a project with his graduate students called RepRap (replicating rapid prototyper). The goal of this project was to produce an affordable personal 3D printer, which was able to create its own parts. This was the start for 3D printers to find their way to homes and offices. In the following years many other companies released personal 3D printers with affordable prices. Most of these 3D printers are able to print objects from different types of plastic, wood, and food. Home fabrication is the process of making physical objects out of a design with a personal 3D printer at home. One can easily make her designs or the design she has downloaded from an online service provider at home in a couple of hours needless of having to wait days for the thing to be delivered to her home.

1.4.3 Personal Design

Being involved with personal fabrication or home fabrication, one always needs a design to be manufactured. Designers are skilled enough to bring their ideas on a paper and use tools like CAD applications to implement them. This process is called *personal design*. However, this is considered as an issue in home fabrication; home users are usually not skilled enough to be able to implement their ideas by CAD applications and it takes quite a long time to professionalize enough to be able to 3D model anything. The complexity of UI in CAD tools and the high learning curve is an obstacle for every user to involve herself in personal fabrication. Similar to the PC revolution, which had its breakthrough after Graphical User Interface, the personal fabrication can only find its way to people's lives under the condition that the lack of easy-to-use tools are filled with the manifestation of new interaction concepts, software technologies, and user-friendly tools.

1.4.4 Motivation

Complexity of CAD tools is a problem in personal design

As it is mentioned in 1.4.3, there exist several problems, which limit users to use personal fabricators. Complex interface of the existing CAD tools and their high learning curve has limited the users group to the experts in 3D modeling. Moreover, online 3D model libraries like Thingiverse, cover a limited range of object models.

Online 3D customizers designed for novice users cover a low range of objects

Some service providers like Shapeways have offered simple 3D customizers, which provide a template of an object that can be changed in shape easily by dragging and dropping predefined points on the model, or adding parts; but they still cover a narrow range of objects.

Another problem is that, sometimes we find the desired model online but the size and parameters have to be changed to fit to our application. Although there exist parametric models online, to be able to change the parameters, one needs to be acquainted with CAD. Our goal is to present an approach in which, we are able to solve the problems mentioned in Personal Design.

Professional CAD users also have problems interacting with other designers' code

The problem does not bound to novice users, professional CAD users also have difficulties for editing the designs that they have downloaded from the internet. For example, if they want to change dimensions of a specific part of a model, they need to know if there is a variable defined for that and where in the code it is defined.

In this thesis, we are going to introduce an approach, which aims to ease personal design for users and solve the mentioned problems. To do so, first we need to know what has been done so far to ease the personal design process. In the next chapter we will present a number of related approaches and products that have been created to help users to implement their design ideas in 3D geometry.

1.5 Thesis Structure

- **Chapter one:** In this Chapter we give the motivation of our work after presenting a brief history of personal design and fabrication. Then we overview different parts of the thesis.

- **Chapter two:** Chapter two covers theoretical background and related work to our approach, we give short descriptions of several CAD tools and 3D customizer and 3D modeling libraries and by comparing their attributes, we extract the requirements of our system.

- **Chapter three:** This chapter includes our work. We start this chapter with an initial survey, solution concept definition, and system usability scale test on Makerbot Customizer, which results in requirement elicitation of our system. We continue with describing our approach and the conducted studies. After designing and testing the paper prototype of the system, we design the final mockup of ParaShape.

- **Chapter four:** In chapter four we evaluate our mockup in a qualitative user study and a Post Study System usability Questionnaire. Finally we verify if we have met the requirements of our system.

- **Chapter five:** In the last chapter we summarize our work and the concepts learned from it. We follow with directions to future work from what is learned in studies.

Chapter 2

Theoretical Background and Related Work

As we advance in life we learn the limits of our abilities.

Henry Ford

The first step in 3D printing is designing a 3D model. The designs are created by using computer aided design (CAD) tools, or animation modeling software. One normally has to be acquainted with mathematical representations of 3D objects and experienced enough with working with a CAD software to be able to represent her ideas and design into a 3D model. As 3D printing technology is heading to find its way to homes, some companies started to release user friendly 3D modelers and creators with have less complex interfaces. They are aimed to make the design process almost possible for anybody with a basic computer knowledge. Moreover, there exist 3D model libraries where people can share their designs or download existing 3D models, which they can edit and manufacture later. In this Chapter, we are going to present the file formats used for personal fabrication. Following, we will have an overview on some of these tools and services and at last, we are going to introduce an approach on collaborative tagging, which we will use in our work later on.

2.1 File Formats

There are various file formats supported by 3D printers. Two of the most popular ones are explained in this section. In our approach, we are going to study the strength and weakness points of each of these file formats to verify which one fulfills the requirements of our users more.

2.1.1 STL

STL format describes the object by an unordered list of triangular facets, which make the outer skin of it

To make the geometric description of CAD software packages understandable for the personal fabrication systems, an interface is needed. The STL (StereoLithography) file format created by 3D systems, is used by almost all of these systems.

The STL format describes the object by an unordered list of triangular facets, which make the outer skin of it. There are two STL formats: ASCII and binary format; although ASCII format is longer, but is human readable. Each of the triangular facets in STL is described by the coordinates of the three vertex and a unit normal vector to show whether the facet is facing inside or outside the object.

One benefit of using STL is that, it is a simple way to represent CAD design for manufacturing systems, it is also a *de facto*[1] standard already used by CAD packages and manufacturing systems. Nevertheless there are also drawbacks, for a given accuracy parameter, STL file is many times bigger than the CAD file. It also carries redundant information such as the coordinates of duplicate vertex. Moreover, due to lack of robustness in tessellation algorithms used by CAD vendors, there are geometry flaws which need to be fixed with repairing software, which increases the production time [CLL10].

2.1.2 AMF

AMF is a XML-based format, which represents geometry and material of objects independent of the resolution

The new Additive Manufacturing File Format (AMF), which is known as a replacement for STL has the capability of defining regions geometrically in triangle mesh, voxel (volumetric pixel) bitmap or functional representation. AMF represents geometry and material of objects independent of the resolution. AMF can be simply converted to standard STL format.

AMF is a compact XML-based file format, which makes it capable of adding features to when needed to be adopted by CAD programs or additive manufacturing processes.

One of the weaknesses of STL format is lack of support for different materials, AMF has solved this problem by its <*Palette*> tag, where different type of materials can be defined and other attributes also may be added to each material [HL09].

2.2 CAD Software

CAD software one of most important 3D modeling tools

CAD software packages are considered as the most important tools to create 3D models by some users. In this section, we are introducing some of them, which are used more often for 3D printing. We are going to give a brief introduction of the characteristics of each tool and extract the most important ones which have an important role in decreasing

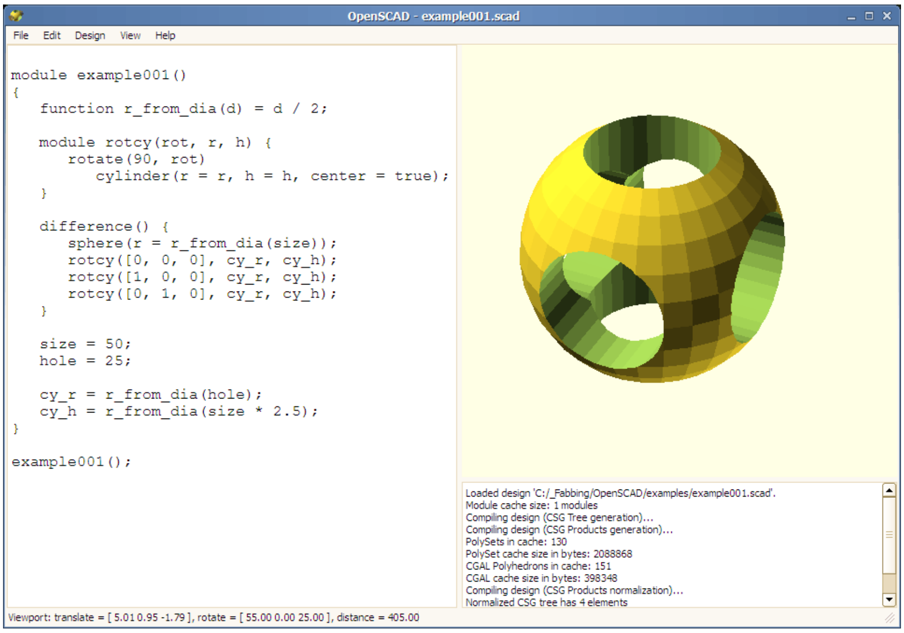
[1] "De facto means "existing in fact," and de facto standards develop either with the inordinate success of a single company or through a consortium of private corporate partners."[SSS05]

the learning curve. Later on, we are going to compare them to have an overview on which tools to choose according to our application and to see if everything we need is there, and which characteristics an ideal easy-to-use modeling tool should have.

2.2.1 OpenScad

OpenSCAD is a script based CAD tool, which focuses more on the structure of design content

OpenScad is a free 3D CAD application for making solid 3D models. Unlike most of the CAD software, OpenScad does not follow WYSIWYG (what you see is what you get) concept, which describes a system, in which the content (text, graphics) appears on screen while editing as they would be in the final product, which can be a web page, printed document or, set of slides. OpenScad focuses more on the CAD aspects of 3D modeling rather than its look, it follows WYGIWYM (what you get is what you mean) concept, which means that the user focuses on the structure of the model, and writes the content according to its meaning in the model. Here, the editor should know the semantics of the content before, to be able to export the final result. OpenScad acts like a compiler that reads through the script and renders the 3D model described in it. Then, it exports it to different standard 3D formats like STL, OFF, and DXF. The advantage of this system is that it separates the focus on structure and appearance of the content. OpenScad is known to be a good application for designing machine parts, but not animated movies [VGAS12]. This tool is being widely used by DIY people working with 3D printers who share their 3d models on libraries like Thingiverse. (see Figure 2.1).



```
module example001()
{
   function r_from_dia(d) = d / 2;

   module rotcy(rot, r, h) {
      rotate(90, rot)
         cylinder(r = r, h = h, center = true);
   }

   difference() {
      sphere(r = r_from_dia(size));
      rotcy([0, 0, 0], cy_r, cy_h);
      rotcy([1, 0, 0], cy_r, cy_h);
      rotcy([0, 1, 0], cy_r, cy_h);
   }

   size = 50;
   hole = 25;

   cy_r = r_from_dia(hole);
   cy_h = r_from_dia(size * 2.5);
}

example001();
```

```
Loaded design 'C:/_Fabbing/OpenSCAD/examples/example001.scad'.
Module cache size: 1 modules
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
PolySets in cache: 130
PolySet cache size in bytes: 2088868
CGAL Polyhedrons in cache: 151
CGAL cache size in bytes: 398348
Compiling design (CSG Products normalization)...
Normalized CSG tree has 4 elements
```

```
Viewport: translate = [ 5.01 0.95 -1.79 ], rotate = [ 55.00 0.00 25.00 ], distance = 405.00
```

Figure 2.1: OpenScad acts like a complier that reads through the script and renders the 3D model. [2]

[2] Picture credits to http://wiki.imal.org/howto/3d-modeler-cad-softwares

2.2.2 AutoCAD

AutoCAD is a tool fo drawing and editing 2D and 3D models where one can set material, camera view and light

AutoCAD is said to be one of the most popular CAD packages around the world, which offers tools for 2D a 3D design. The first 3D representation of AutoCAD was in 1988 in Release 10, which displayed 3D objects in wire-frame. The variety of tools offered makes AutoCAD able of drawing whatever can be drawn on a board. However, the learning curve is high and most users find it difficult to start working with AutoCAD [McC99].
3D modeling has evolved in AutoCAD rapidly, AutoCAD 2010 includes many 3D modeling, viewing, controlling, and printing commands and now the users can do whatever is possible in other software packages such as Maya, Cinema 4D, and 3DS. AutoCAD users can draw meshes and solids, edit them, set up views, cameras, materials, and lights and use view ports for 3D printing [Ham10].

2.2.3 Blender

Blender is mostly used for making 3D animation by creating 3D models, giving them texture and animates them with predefined functions

Blender is a free and open source software used for 3D modeling, making 3D animations, 3D games, or editing videos. All of these are done by the same software. Blender is an application, which creates 3D models, gives them textures, and makes them move and act according to animation functions defined, lights the scene with its rendering engine and mixed movie shots, by its composition module [Fla10].

Blender displays the model according to the camera location and lighting defined by the user

There exists a camera and a light when Blender renders a scene. Considering the camera's point of view, Blender scans the scene and calculates how far should objects on the scene be, where on the scene they should be, which parts can be seen by the camera and how big should they appear. Moreover, according to the position of the light, it figures out the shadows and how the surfaces look like [Bla12].

Blender can be run on different operating systems such as Windows, Linux, Mac OS and Solaris (see Figure 2.2).

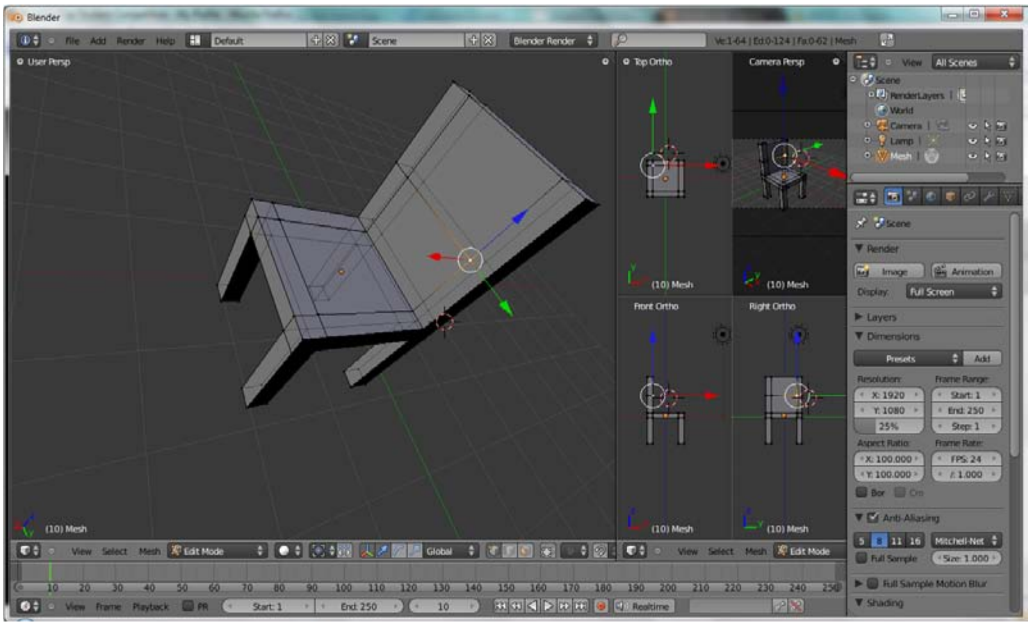


Figure 2.2: Considering the camera position, Blender renders the 3D model from different points of views [3]

2.2.4 TinkerCad

Tinkercad[4] is a browser CAD. This 3D modeling tool makes everyone capable of 3D modeling. One can simply pick up meshes, merge them together, remove shapes from a mesh to make holes, and do linear transformations on them. There is also the possibility to make models from imported SVG or STL files. Each user has a profile, where she can save her work for later manipulation. After a model is created, the user can download the STL file for desktop fabrication. She can also use the 3D printing services from providers, which hold partnership with TinkerCad such as Ponoko, Sculpteo, and i.materialise. To share the model online, there is also a direct link to Thingiverse from the TinkerCad web page [PFS13] (see Figure 2.3).

TinkerCad is a browser-based CAD tool, which is designed for novice users and supports file formats for personal fabrication

[3] Picture credits to http://azerdark.wordpress.com/2010/02/23/first-time-using-blender/

[4] https://tinkercad.com/

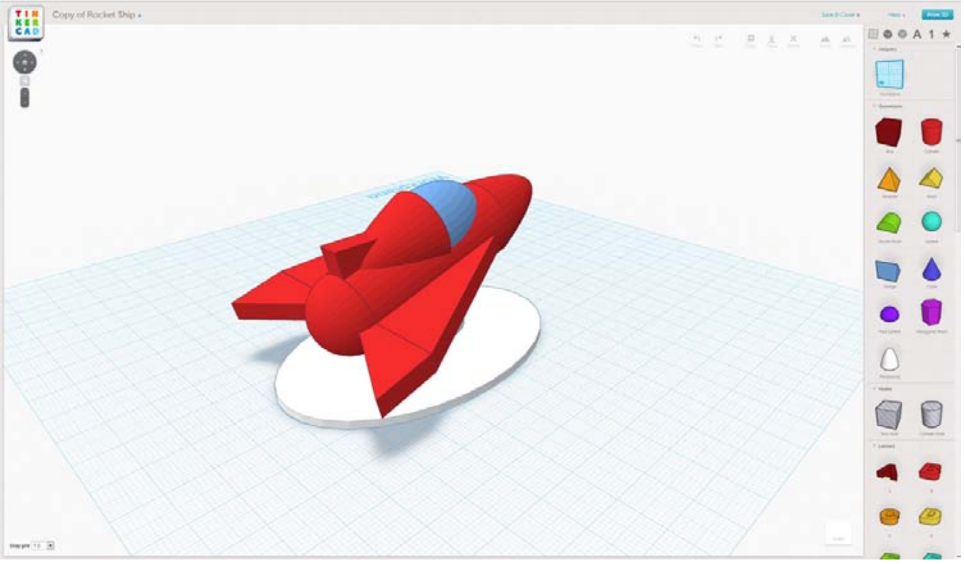


Figure 2.3: In TinkerCad models are made by picking up meshes, merging them together or subtracting them from each other. [5]

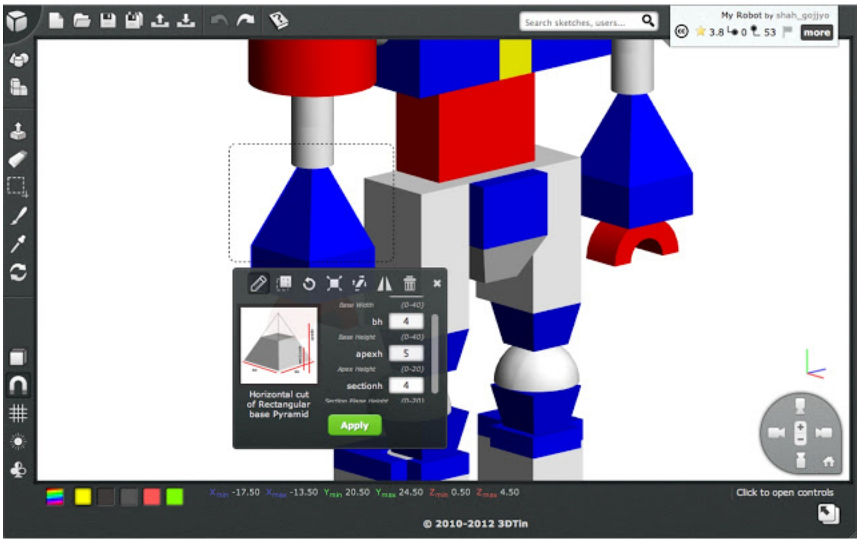
2.2.5 3DTin

3DTin is a browser-based tool, which enables users to create 3D models by putting predefined geometric shapes, and supports transformations such as rotation and scaling

3DTin[6] is a web browser WYSIWYG standard modeler that can be run in browsers like Google Chrome and Firefox with WebGL support. One can create 3D models from predefined geometric shapes, voxels (Volumetric Picture Elements), which is like putting Lego bricks together, or 2D imported images in JPEG or PNG format. It also offers tools to induce rotation, scaling, changing colors, and grouping of objects [PFS13]. 3DTin was first developed in India by Jayesh Salvi in March 2010. The user needs to create an account to save her models. 3DTin is free if the user shares all her designs under Creative Commons licensing. Models can be exported in PNG, OBJ, DAE, and STL format. After the model is exported, the user has the chance to download it from the web browser, or upload it on the 3DTin library in Thingiverse or send it to i.materialise to have it printed in the desired material and color [Fra12] (see Figure 2.4).

[5] Picture credits to http://solidsmack.com/cad/3d-printers-never-fear-the-brand-new-tinkercad-is-finally-here/

[6] http://www.3dtin.com/



Base Width (0-40)
bh 4
Base Height (0-40)
apexh 5
Apex Height (0-20)
sectionh 4
Section Plane Height (0-20)

X_{min} -17.50 X_{max} -13.50 Y_{min} 20.50 Y_{max} 24.50 Z_{min} 0.50 Z_{max} 4.50

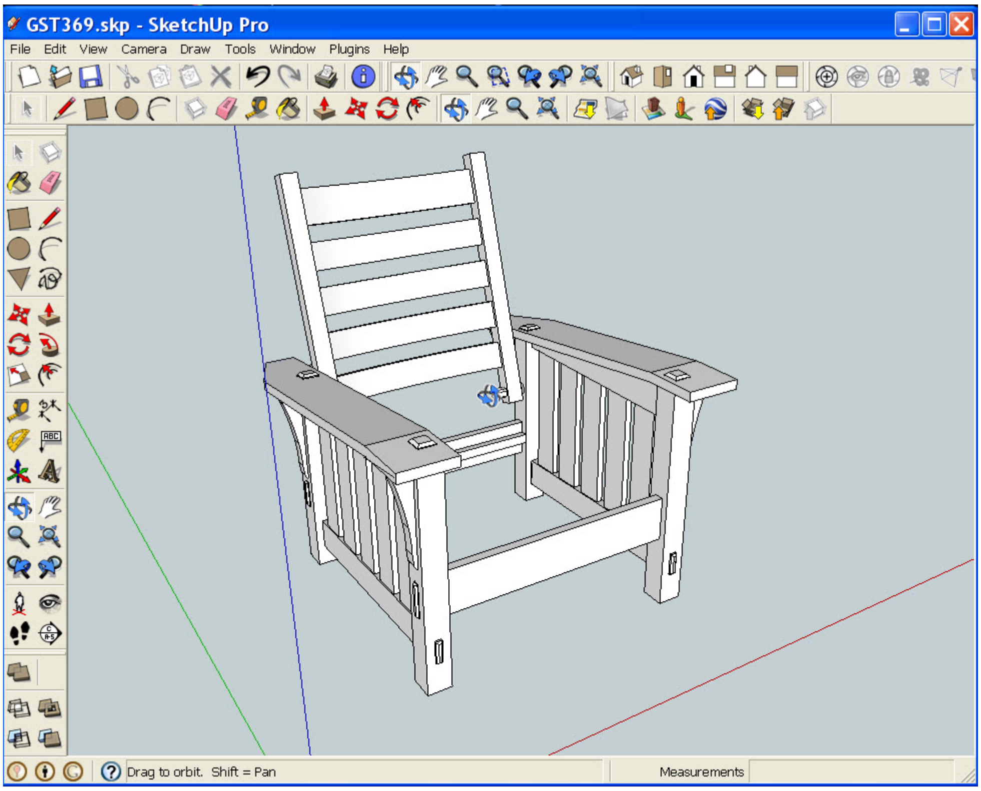
Figure 2.4: In 3DTin models are created from predefined voxels, which can be rotated, scaled, modified in color, and grouped together.[7]

2.2.6 Google SketchUp

One of the most popular 3D modeling tools among users with little or no designing knowledge is Google SketchUp[8], which was introduced by Google in the year 2000.
SktechUp, unlike 3DTin and Tinkercad, is not a web based 3D modeler. It is a downloadable software, which exists in two versions: the free SketchUp version and, Sketchup Pro. What made SketchUp popular is its low learning curve in comparison with other software packages such as SolidWorks and AutoCAD. The interface is simple and the tools offered are similar to those of other software like Microsoft Paint and Photoshop, including lines, arcs, rectangles, polygons, and freehand drawing, plus some other powerful tools for modeling [Sin10] (See Figure 2.5.

Google SketchUp is a popular modeling tool due to its' low learning curve and familiar menus similar to Microsoft Paint and Photoshop

[7] Picture credits to http://educarymotivar.blogspot.com.es/2012/04/3dtin.html
[8] http://www.sketchup.com/



Drag to orbit. Shift = Pan

Figure 2.5: In 3DTin models are created from predefined voxels, which can be rotated, scaled, modified in color, and grouped together.[9]

2.2.7 123D Design

123D Design is an easy to learn software for non-engineer users

123D Design is one of the 3D modeling software released recently by Autodesk with the aim of modeling for 3D printing. One of the characteristics of this software is its low learning curve, which is also considered lower than Google Sketchup. It is mostly used by non-engineer users who want to 3D model complex objects [Chu11].

123D Design is free and can be run on Windows, OS X, and iPad. It also has an online version. Although the iPad version does not support all the features, one can take the design from iPad to a computer to clean it up [PFS13].

2.2.8 Comparison

So far we have reviewed the characteristics of different CAD tools. In this part, we are going to make a comparison between them to see which one fulfills the needs of our application. Table 2.1 presents this comparison in detail. In this table we compared the CAD tools described before from six different aspects that we found important to our personal fabrication users. Below, we describe three aspects briefly:

- *Free:* A free tool is preferred by users, since some users think that

[9] Picture credits to http://readwatchdo.com/sketchup/

they need to check the software before they start using it. There are also users, who use these tools rarely (once a year), and would rather not to spend money on them.

- *Supports STL:* STL is currently the most common file format used by 3D printers. If the CAD tool is capable of producing STL file itself, it would reduce the cost of converting files for users.
- *Open source:* Users prefer open source software due to various reasons. It is more secure; since the user can see what is written in the code. It is flexible and customizable; the user can change the code to make the software fit to her requirements. The quality and support is higher; as there are many users developing the software, they are more acquainted with the user needs, there are also many support pages for the software online. One can try the software before buying it and even verify if by making changes to it, it would fulfill her needs.
- *Low learning curve:* As mentioned before, not all 3D printer users are professional 3D modelers. Therefore, they need an easy-to-use software, with which they can 3D model objects in the minimum amount of time.
- *WYSIWYG:* A tool, which follows WYSIWYG concept is easier for users to interact with. Seeing the 3D model appearing on the screen while editing it as if it is the final product, helps users to know if they are going the right direction.
- *OS Compatibility:* The more various Operating Systems a tool is compatible to, the higher number of users it supports.
- *Parametric:* If a tool supports parametric modeling, users can derive new instances from existing models needless of creating new models.

	free	supports STL	open source	low learning curve	WSIWYG	OS compatibility	parametric
OpenScad	+	+	+	-	-	Windows Linux Mac OS X	+
Blender	+	+	+	-	+	Windows Linux Mac OS X Solaris	+
AutoCAD	-	+	-	-	+	Windows Mac OS X	+
TinkerCad	limited time	+	+	+	+	Browser-based	-
3DTin	+	+	+	+	+	Borwser-based	-
SketchUp	+	plugin	-	+	+	Windows Mac OS X	-
123D Design	-	+	-	+	+	Windows	-

Table 2.1: comparison of most popular CAD programs for 3D printing

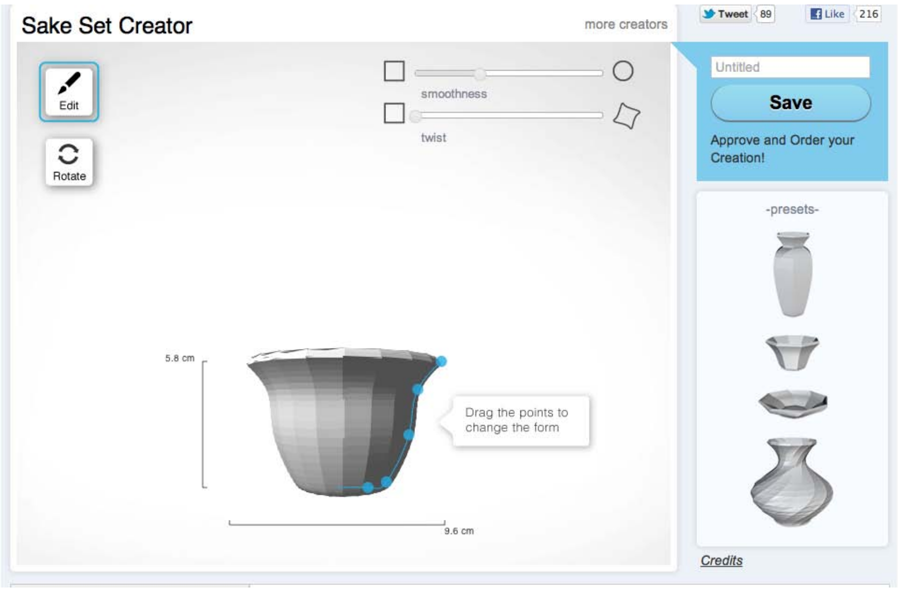
2.3 3D Creators and Customizers

To ease the design process for novice users, some companies started to release tools, which are known as creators and customizers. Creators are tools, which help users have 3D models by giving a 2D image or the scan of an object as an input and get the resulting 3D model as output. While with the help of customziers, users can start from existing 3D templates or parts, customize them, and then merge them together. Below, some of these creators and customziers are described in detail.

2.3.1 Shapeways Creators

Shapeways has creators, which enable users to make 3D objects from 2D images, or reforming already existing templates

To ease the design process, Shapeways[10] offers a range of creators for different objects such as iPhone cases, sake sets, vases, and jewelry (see Figure 2.6). These creators allow users to make designs by reforming a template by dragging and dropping defined points, or uploading a 2D black and white .jpg image, and turn it to a 3D model by extrusion; this method is used to make earrings, pendants, and other accessories.
Having limited the user to a sequence of tasks to go through to reach the design, reduces the chance for confusion with tools and the flow of the process.



Drag the points to change the form

Approve and Order your Creation!

Figure 2.6: Shapeways sake set creator- the primitive model can be customized by changing the position of predefined points[11]

[10] http://www.shapeways.com/creator

[11] Picture credits to http://ideacious.wordpress.com/2011/11/24/discussion-open-fabrication-the-future-of-3-d-printing-pt-2/shapeways/

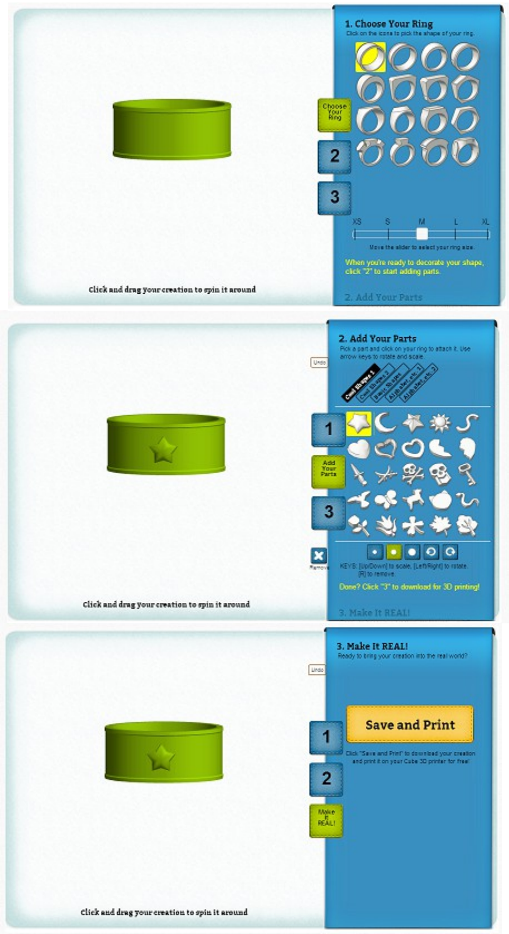
2.3.2 Cubify Easy Apps

Cubify[12] is a 3D printing web service, providing easy 3D modeling web-based apps to create toys like Android figurines, space ships, and accessories. The creation process in these apps consist of three steps:

1. Select the basic template of the model

2. Select parts to be added on your model

3. Save and print



1. Choose Your Ring

Click on the icons to pick the shape of your ring.

Move the slider to select your ring size.

When you're ready to decorate your shape, click "2" to start adding parts.

2. Add Your Parts

Pick a part and click on your ring to attach it. Use arrow keys to rotate and scale.

KEYS: [Up/Down] to scale, [Left/Right] to rotate, [R] to remove.

Done? Click "3" to download for 3D printing!

3. Make It REAL!

Ready to bring your creation into the real world?

Click "Save and Print" to download your creation and print it on your Cube 3D printer for free!

Click and drag your creation to spin it around

Figure 2.7: Cubify Easy Apps-3 steps of creation process

[12] http://cubify.com/apps/apps.aspx?tb_create_apps

2.3.3 Sculpteo Workshops

Users can make 3D models by picking existing models customizing and merging them

Sculpteo is another 3D printing service. In Sculpteo's online workshops, one can choose from a variety of easy apps to design her own object by picking already existing models, customizing them and merging them together. It also provides apps for iPhone and iPad, where the user can scan her face in 2D and define bounds for it; then by rotate-extruding the 2D image, make a 3D object such as a vase (see Figure2.8).

One can also upload text or images in Sculpteo and the app converts them to 3D models. After the model is created, the user can order it to have it printed in different materials and delivered.

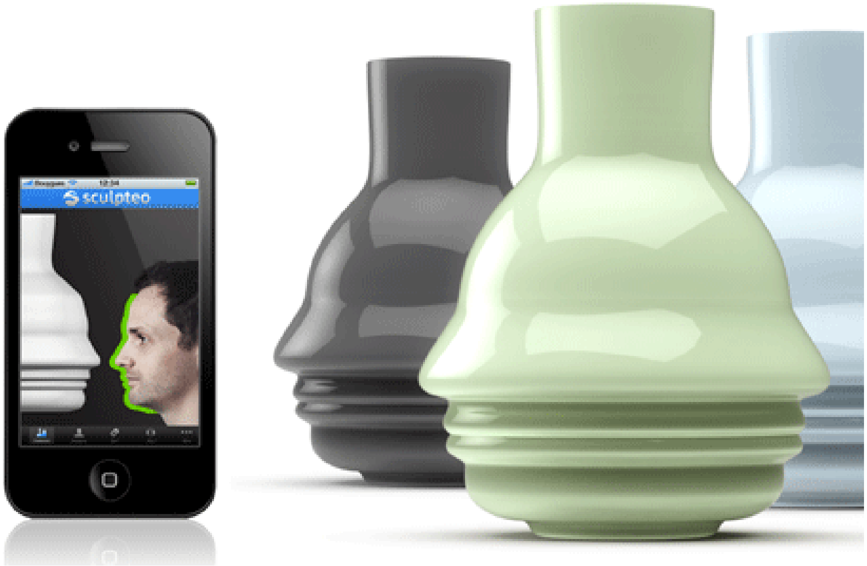


Figure 2.8: Sculpteo app gets a 2D image as an input and creates a 3D model out of it.[13]

2.3.4 Autodesk 123D

Autodesk 123D is a set of modeling tools with the aim of making design for fabrication easy for novice users

Autodesk 123D is a set of modeling tools, designed for people with basic design skills, and hobbyists. Despite providing simple drawing and modeling capabilities, it supports making physical objects from the design by supporting the STL file format.
To ease the printing process for the users, Autodesk 123D cooperates with three companies (Ponoko, Techshop and 3D Systems) where they can upload their designs and order them to be printed in different materials.
There are more than just designing from scratch one can do in 123D; there exist other apps, which also support 3D printing services:

[13] Picture credits to http://www.develop3d.com/blog/sculpteo-expands-its-services-using-the-cloud

123D Catch creates 3D models by scanning several photos from different viewpoints of an object

1. 123D Catch: Is an image based 3D scanning and rendering software, which uses 3D scanning, cloud computing and rendering techniques to transform digital photos to 3D models. The process is to take several photos from different viewpoints of an object, then a 3D modeling program from the cloud identifies unique points (provided that there is enough overlap in the photos) to reconstruct relative positions from which the photos were taken. Having this information, a 3D model is rendered, which can be imported into CAD programs such as AutoCAD to be manipulated (see Figure 2.9). The app is now available for iPhone and iPad.

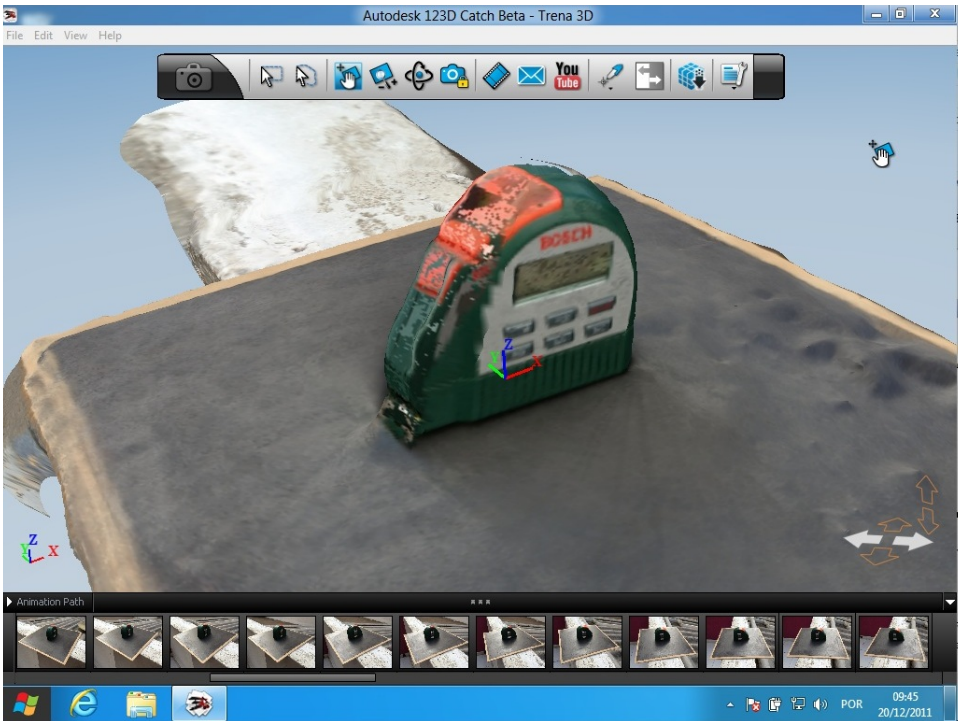


Figure 2.9: 123D Catch creates a 3D model by identifying unique points on several photos of an object taken from different viewpoints and reconstruct the positions from which the photo was taken .[14]

one can interact with virtual clay in 123D Sculpt to experience virtual sculpting

2. 123D Sculpt: Is an iPad app, where one can experience sculpting as it is in reality. To start sculpting, one should choose a shape from the library (creatures, humans, vehicles, etc.) and interact with the virtual clay by pushing, pulling, and painting it. There are even more actions to do on the sculpt model, such as enlarging, shrinking, adding shades, and hues, or taking areas of an image and rubbing it on the model.

123D Make provides various cutting patterns of an uploaded model

3. 123D Make: In this app the user uploads a 3D model or chooses it from the library and receives LOM (Laminated Object Manufacturing) solid models. The app generates 2D plans and provides

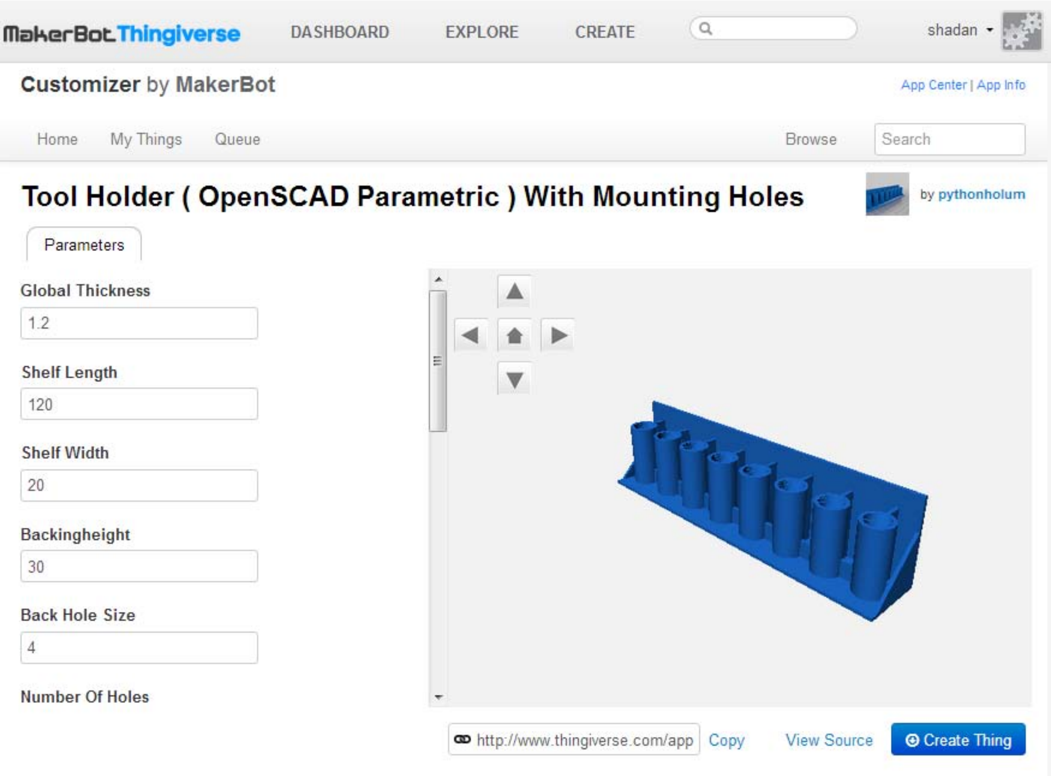
[14] Picture credits to http://geekorama.com.br/autodesk-123d-catch-transforme-suas-fotos-em-modelos-3d

various cutting patterns with the assembly instructions to make the 3D model.

2.3.5 MakerBot Customizer

makerbot customizer gets the OpenScad code of model as input and creates an interface with customizable controls on model parameters

One of the most recent apps released by MakerBot is Customizer, which is a tool for users to change dimensions and parameter values of a model by manipulating controls like sliders, or entering values in text fields. As the value is entered, the Customizer requests an image of the model with the defined parameter values to its server, and displays the image of the new model in its originally predefined orientation. Thingiverse users can make a Customizer out of their models by tagging them as Customizer. The app takes the OpenScad script of the model and makes a Customizer out of it, considering the defined parameters and their values in the script.



Tool Holder (OpenSCAD Parametric) With Mounting Holes

Global Thickness: 1.2

Shelf Length: 120

Shelf Width: 20

Backingheight: 30

Back Hole Size: 4

Number Of Holes

Figure 2.10: Makerbot Customizer-the parameter values of the model on the right side can be modified by changing the values in the text inputs on the left side

2.3.6 SketchChair

SketchChair is an application, which allows novice users to design their own chair to be made of material sheets cut by a laser cutter or CNC milling machine. Users are provided with a 2D sketch interface, where they can make different types of chairs (stools, office chair, *etc.*) and have them customized to their size (ergonomics). To design a chair, the users sketch the profile of the chair using a mouse or tablet PC, which is extruded to make a frame out of interlocking planes. There exists a leg tool that eases the process of sketching the leg profile. To test the stability and ergonomics of the chair, a virtual human figure is placed on the chair, so that the final design is checked, whether it is stable enough to stand (see Figure 2.11). After the design is complete, the system generates the profiles and pieces that fit together. The user can cut the pieces and construct her chair, using a laser cutter or CNC milling machine [SLMI11].

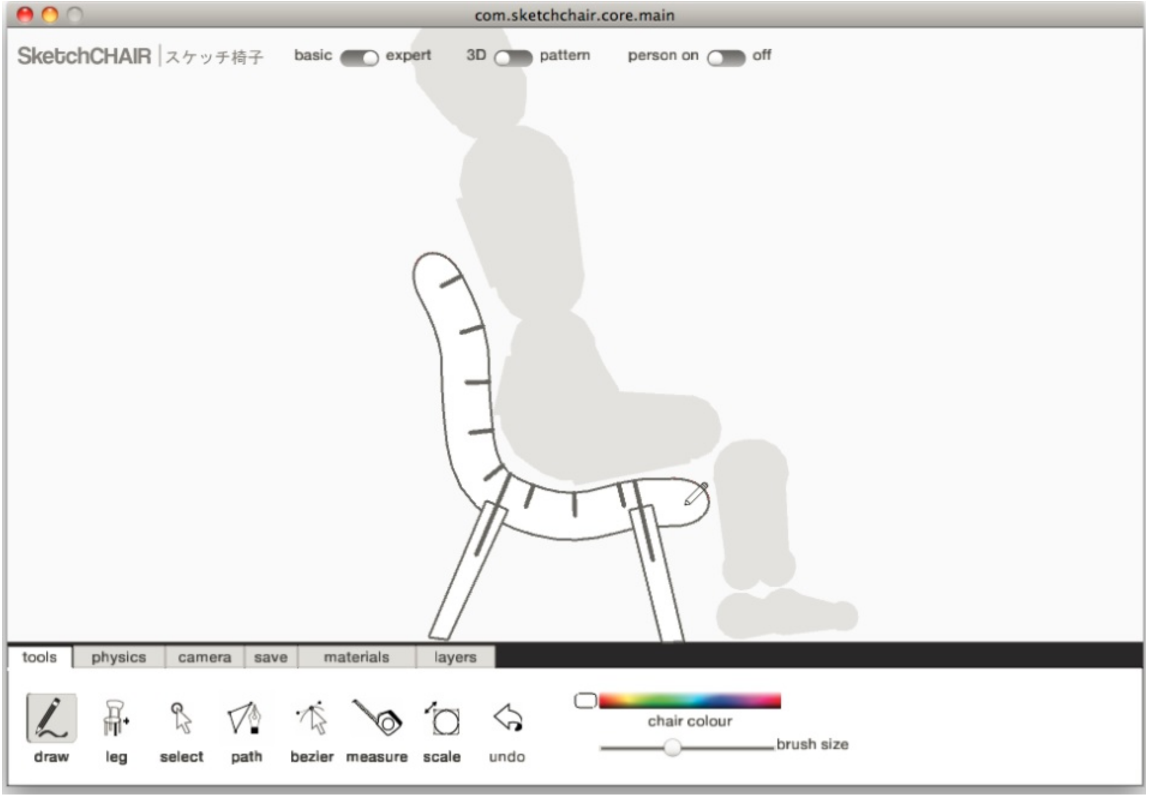


Figure 2.11: SketchChair-to verify the stability and the ergonomics of the chair, a virtual human figure is placed on it.[15]

2.3.7 Summary

In this section, we had an overview on some 3D creators and customizers. Despite the fact that the interaction techniques used in these systems to output the resulting 3D model has reduced the amount of work for the users considerably, they still cover a limited range of objects. Most of the customizers are defined for specific primitive 3D models and the flexibility for manipulation and customization is low.

The existing creators are also mostly using the scan and extrusion technique in which, they get the scanned 2D image and give it a 3rd dimension by extruding it. Although this technique is easy for novice users to learn, it does not support customization and individualization

[15] Picture credits to http://educarymotivar.blogspot.com.es/2012/04/3dtin.html

of the models.
Among all the customizers and creators we introduced, MakerBot Customizer covers the highest range of models and supports customization. However, the system can be improved in the number and the flexibility of functionalities for customization.

2.4 Primitive Instancing

In primitive instancing, the system uses a set of predefined 3D solid shapes, which are primitives. Primitives are usually parametrized, having parameters for transformation aspects, dimensions, and also the number of elements like faces of a pyramid. Each parametrized primitive can be assumed as the defining object of a family of solids, which is derived from it by modifying the parameter values, and share the same parameters with it. This concept is called *grouping technology* and is of high importance in CAD. When it is not possible or difficult to define a complex object by a combination of simpler objects, primitive instancing is of excellent use.
Despite having a hierarchical presentation of instances, each node instance is an independent defined object itself. However, there still is no procurement for combining instances together. Moreover, for creating a new instance and also having drawing methods to manipulate mass properties of the solids, the code should be written for each primitive [Fol96].

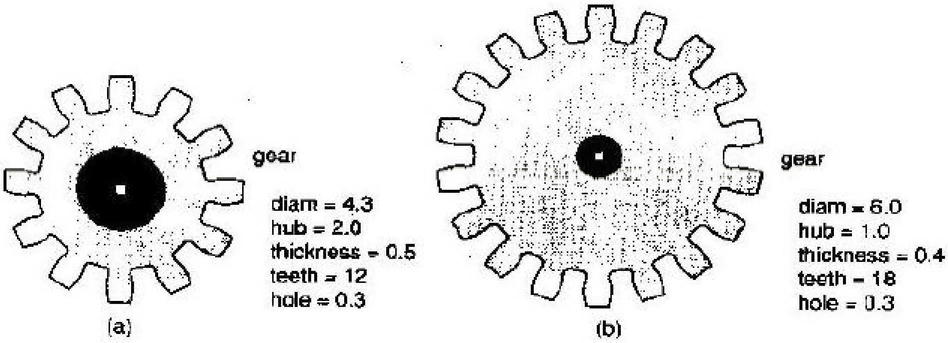


Figure 2.12: Primitive instancing: two different instances derived from the same primitive. Although both gears come from the same hierarchy, each one holds its own attributes independent of the other [Fol96].

2.5 Part-Based Modeling

Part-based modeling is an approach used for 3D modeling, in which each object is made up of its composing parts. This method simplifies modeling complicated objects and reduces the time spent for it. In this section, we are introducing three different approaches on part-based modeling.

2.5.1 Modeling by Example

Modeling by example is the process of making new models by cutting parts from similar models and composing them together

Modeling by Example is a data-driven approach to construct 3D models. In this system, the user uses a database of 3D models as input to create her desired model as output: first, the user chooses a model from the database and selects a part of it to be edited. Then, she searches in the database to find parts similar to the selected one. After choosing the desired model from the search results, some editing operations are induced to the model. The part is cut from the original model by intelligent scissoring, and is composited into the final model.

This approach is easy to learn as the user needs to learn only a few commands such as open, select, cut, copy, paste, and undo, which are familiar to all computer users. Moreover, while the concept of the design comes from the user, the geometric details are taken from the existing models in the database. This approach consists of three steps:

1. 3D surfaces segmentation
2. Shape-based search to find models, which match the queried part
3. Forming the new model by composition of parts [FKS+04]

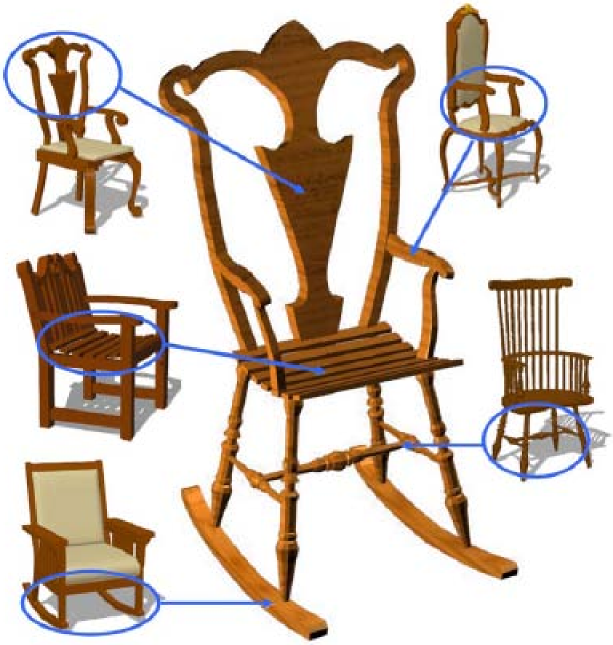


Figure 2.13: Modeling by Example: The chair in the middle is made up of the circled parts of the other chairs. In this method the model is composed of geometric parts extracted from a database of 3D models [FKS+04].

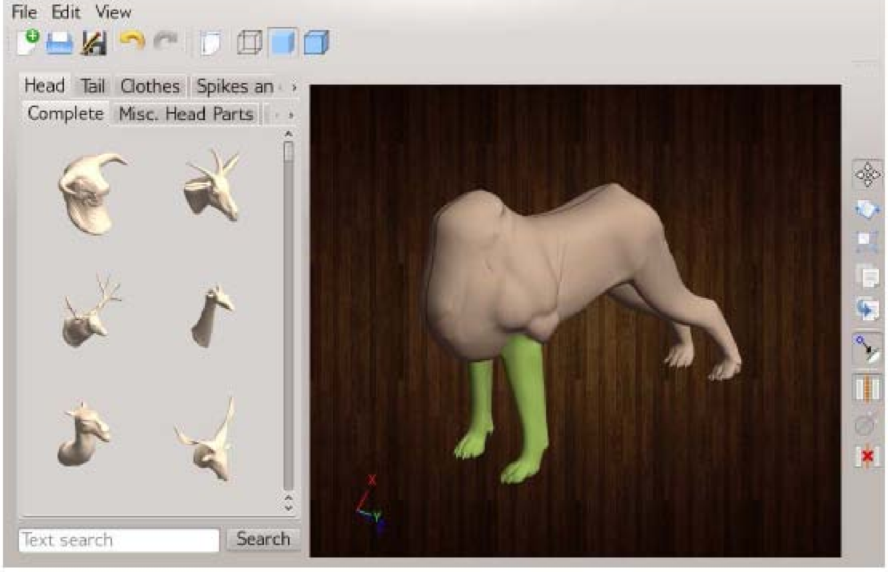


Figure 2.14: Assembly-Based Modeling: the model is made up of components in an intelligent list, which are extracted from a repository. Every time a component is added, the list is updated with the components which are related to the model [CKGK11]

2.5.2 Assembly-Based Modeling

User chooses components from an intelligent list, which displays updated model parts each time the model is changed and compose them together

Assembly-based modeling is another approach to ease 3D modeling for the user. In this approach, components that are either designed specifically for this use or taken from a repository, are assembled to form a new model.
The approach consists of two stages: preprocessing and interactive stage. In preprocessing stage, the input is a repository of segmented and labeled components. Labels define the categories components belong to, such as head, arm, torso, etc. Labels can also show the component hierarchies by defining their subparts; for example, upper torso and lower torso are subparts of a torso. Components are also clustered based on their geometric style. So there might exist several clusters in each category based on the vectors of the features. In the interactive stage, a probabilistic inference is performed to generate a list of components, which are ranked most to be compatible to augment the current model. There are tabs in the interface, each belonging to a specific component, which is related to the model and contacting a list of ranked models (see Figure 2.14). Each time the user chooses a component and adds it to the model, these lists are re-ranked according to the updated model. This process goes on until the desired model is achieved [CKGK11].

2.5.3 Model Composition from Interchangeable Components

users can shuffle interchangeable components of models that belong to the same model class

One of the approaches done at the University of British Columbia to ease 3D modeling for novice users, is composition of models from interchangeable components. In this approach, researchers based their work on an observation, which showed that most users create models, which belong to a small set of model classes such as humans or quadrupeds. Considering this, they express that members of each model class share a common group of components. Based on this knowledge, a modeling system is introduced, which allows users to make new models by shuffling interchangeable components of already existing models. The shuffling system consists of a modeling interface and a pre-processor, which has the responsibility to segment input models into meaningful components. This process is either performed at run time, after the user chooses the models she wants to work with, or beforehand, on the complete database of models [KJS07].

2.6 3D Model Libraries

One of the most popular services used by DIY people and 3D printer users, are online 3D model libraries, where designers can share their designs and get inspired by each other's work, and beginners have the chance to download the model of the object they want to make. As 3D model libraries cover a larger range of object models in comparison with other 3D printing services, it is used widely by novice users, who do not have 3D modeling experience. However, there still exists a drawback; when there is a need to customize a model downloaded from the libraries, one needs to be acquainted with 3D modeling.
In this section, we are introducing some of these popular libraries to verify how the 3D models are presented, shared, searched and downloaded there, and which other services are offered by them. Later on in the next chapter we are going to present a solution to customization of 3D models in online libraries.

2.6.1 Thingiverse

Thingiverse[16] is a website owned by MakerBot Industries, where people can upload their design files to share them with others. The designs are mostly used in digital fabrication by DIY users working with 3D printers, laser cutters, and milling machines. 3D printer users can download the .stl files and have them printed or edit them (they should be familiar with 3D modeling) and print their own version. To prevent abuse, the company has an intellectual property policy according to which, all unauthorized and infringing materials or users who repeat uploading them are removed (see Figure 2.15).

Thingiverse provides designs for 3D printing, laser cutting and CNC milling

[16] http://www.thingiverse.com/

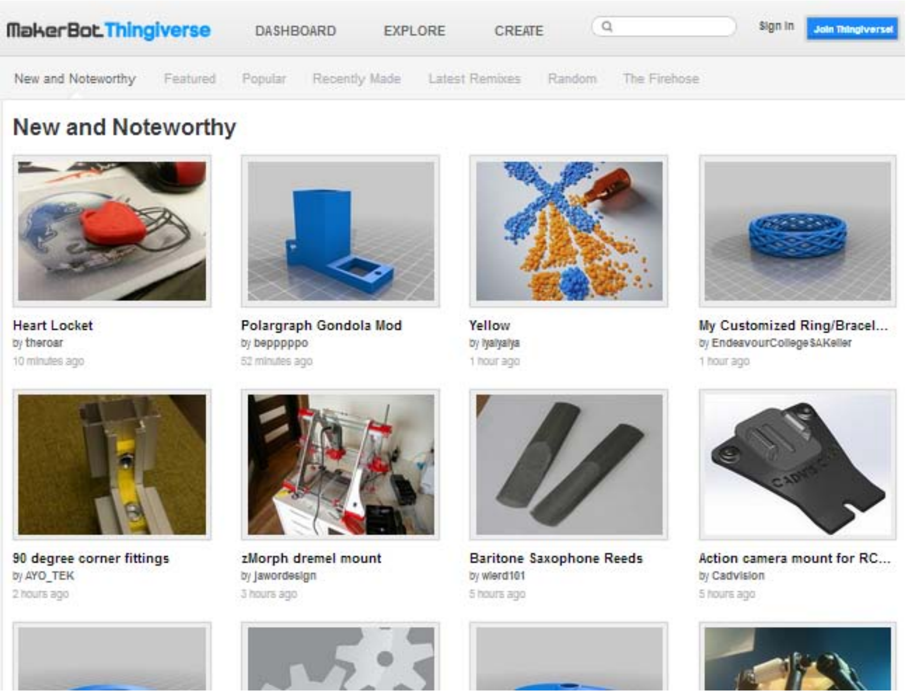


Figure 2.15: Thingiverse web page

2.6.2 Shapeways 3D Parts Database

Users can choose models, have them checked, choose material, check the final price before before submission of 3D print order

Shapeways[17] was one of the first websites giving services for 3D printing, especially for those who find 3D printers expensive or difficult to use. The website has a repository for 3D models for 3D printing. One can find models in the following categories: Art, For Your Home, Fashion, Gadgets, Games, Jewelry, Maker/DIY, Miniatures. Users can choose models from the repository or upload their models in their accounts, have them checked and approved by Shapeways, choose material for the model to be printed, and submit for purchase and delivery. The price on Shapeways is reasonable although the time to have the object in hand (10 to 21 days) is not comparable to the time it takes if we have it manufactured at home [Sin10]

2.6.3 Google 3D Warehouse

Google 3D ware house provides a big repository of 3D models with a search engine like Google's

Tadros in his book "Real World Google Sketchup 7" describes Google 3D Warehouse[18] as a "YouTube for 3D models", which is a web-based service to upload and download models for free in Sketchup (.skp), Collada (.dae) or Google Earth (.kmz) file formats.

One can find a wide range of models of different things in the Warehouse. Searching the Warehouse to find our desired model is quite similar to searching in Google, the more specific your words or phrases

[17] http://www.shapeways.com/3d_parts_database

[18] http://sketchup.google.com/3dwarehouse/

are, the more concrete the result is; for example a search of the word "car" gives around 25,000 results while if you mention the model of the car it can be reduced to 200.[Tad09]

2.6.4 GrabCAD

GrabCAD is community for mechanical engineers to share their CAD files and download others' public designs

GrabCAD[19] is a community for engineers (especially mechanical engineers) to share their ideas and expand their knowledge. GrabCAD provides an online CAD model library for users registered on the site to upload their CAD files and download others' public designs.
GrabCAD also offers other services such as hosting designs privately so that, if a user wants to share a design on her smartphone with someone else, she just needs to run GrabCAD and send an invitation, and the receiving person is needless of having a design software on her computer to see the design due to the built-in 3D viewer, which can be run on Chrome and Firefox. Another service from GrabCAD is, when a company has a problem editing a design, it pays GrabCAD to host a challenge among site members to submit their solutions and win prizes.
Most of GrabCAD users are new companies and students. Experts and old companies have enough experience to start a new family of the existing products, but students and new companies, who need to start everything from scratch can use models on the library as a guide to find something similar to what they need.

2.6.5 Ponoko

Ponoko is a popular service for sharing, buying and selling products and designs

One of the online personal manufacturing service providers is Ponoko[20], which offers direct contact between designers and consumers. It provides a platform for selling, buying, making, and sharing products and product designs. The designs on the site are both commercial and open source. There are tens of thousands of designs available on the site to be customized and printed at homes or by Ponoko to be delivered at customer's door. To estimate the price of the final product, Ponoko has a pre-production system, which calculates the price according to the desired material [Ved11].

2.6.6 Comparison

So far, we had a review on some characteristics of some 3D model libraries. In this part, we are going to have an overview on them and compare them. In Table 2.2 we have compared from four aspects, which we consider important to the users:

- *Free models:* if a library provides users with free 3d models

- *3D print:* if a library 3D prints models on users' request
- *Selling models:* If users can sell their 3D models on a library

[19] http://grabcad.com/
[20] https://www.ponoko.com/

- *Has customizer:* If a library provides a customizer for its 3D models

	free models	3D print	selling models	has customizer
Thingiverse	√	–	–	√
Shapeways 3D Parts Database	–	√	√	–
Google 3D Warehouse	√	–	–	–
GrabCAD	√	–	–	–
Ponoko	–	√	√	–

Table 2.2: Comparison between 3D model libraries

2.7 Collaborative Tagging

collaborative tagging is giving annotations to data

As the amount of information grows every year, the need for metadata is recognized more. Metadata helps identify, describe, and manage information. In digital libraries and databases, one can browse and discover data related to specific topics easier by using metadata [GH05]. Collaborative tagging, which is the process of giving descriptive keywords as metadata for shared content, enables information organization. In collaborative tagging, users can use their own keywords as tags or use the most popular tags. The more a tag is used, the more popular it is recognized. The most popular tags are defined as "tag clouds" [MM06].

In tag clouds, tags are prioritized by their popularity. The more a tag is used in the cloud the most popular it is. Tags also can be clustered according to their similarity and abstractness. Having two tags *i* and *j*, $Co(t_i, t_j)$, which shows the number of times the two tags were commonly used to annotate the same object, indicates how similar they are. Abstractness of the tags also can be evaluated in the same way; having the tags *i* and *j*, if we consider $D(t_i, \neg t_j)$ as the number of times that tag *i* is used for an object but tag *j* not, then for each tag *i*, $D(t_i) = Co(t_i, t_j) + D(t_i, \neg t_j)$ indicates the total number of objects tagged by *i*. knowing this, if $D(t_i, \neg t_j)$ is bigger than $D(t_j, \neg t_i)$, then we can assume that tag *i* is more abstract than tag *j* [SQF11].

Although these numbers and methods give quite a satisfying result in clustering, browsing, and searching data, there still exists a drawback. As there is no supervision on the annotations, sometimes tags that are not relevant to an object are used as search keyword due to their popularity (number of times they are used). This, makes an issue in reaching the data we need. To solve this problem, we are presenting a method in the next chapter.

2.8 Future Use

In this chapter we reviewed some of the existing tools, services, and approaches to ease the design process in personal fabrication. Having noticed that the learning curve for most of the presented CAD packages is high, and the existing creators do not support all functionalities to make anything, and the models on the libraries are not easily customizable, we conclude that a flexible framework for creating parametric 3D models with a low learning curve is missing. With what we learned from the strength and weakness points of the existing tools, we will extract the requirements for an easy-to-use 3D modeling tool. To make our tool more customizable and flexible, we are going to use part-based modeling methods. And finally, we define a solution for having better metadata on models shared on the internet. In the following chapter we will discuss in detail how we will use the concepts defined here.

Chapter 3

Own Work

> Every act of creation is first of all an act of destruction.
>
> Pablo Picasso

The rise of personal fabrication in the last decade has attracted a large range of users to personal design, which is the process of converting design ideas to objects by personal fabricators. However, as we learned from our related work, there exist limits in this process. The existing 3D modeling CAD tools are too complex and time-consuming to learn, and 3D creators/customizers cover a low range of objects. 3D model libraries, which are the mostly used tool by expert and novice users contain a large variety of models, but as all models are static, users need to be acquainted with 3D modeling to be able to customize them to fulfill their needs.

As a solution to this problem we are introducing *ParaShape*, a parametric approach to personal design. ParaShape proposes a parametric component-based modeling tool with low learning curve and easy-to-use interface. The system enables users to reuse components of other existing 3D models, customize them, and make complex models without needing to have 3D modeling experience. It also provides expert users with a standard code structure which gives predefined templates for model components with GUI controls, to increase code readability and reuse and speed up the 3D modeling process. In this chapter, we are going to discuss the studies and the work done to design and implement the final mockup of ParaShape, in detail.

3.1 Initial Survey

Having studied and reviewed some characteristics of several 3D modeling CAD tools in related works, we understood that the main problem of users to interact with them is high learning curve. The more a tool is available for the users, concerning its price, supporting operating systems, and file formats for fabrication, the more the users are attracted to it. Having understandable interface and commands, and following WYSIWYG concept are other aspects that can influence the duration of learning process. Choosing the tool to visualize the idea of a design seems to be a matter of taste, but considering each user's background and experience, users with similar level of experience are assumed to have similar choices. To verify this assumption and get more information for finding out the requirements of our solution system, we decided to conduct a user survey.

3.1.1 Defining User Groups

Before starting the survey, we had a review of the problems of personal design. The main problem, is the high learning curve and the complexity of already existing 3D modeling tools. This, automatically divides the users in to two groups:

1. Professional Users: users who have experience in 3D modeling or design
2. Novice users: users without 3D modeling or design background

The reason that we also considered users' experience in design other than 3D modeling, is that generally users with design background, as we have observed, are faster in learning 3D modeling due to their power in visualizing the 3D geometry of objects.

3.1.2 Design

According to the defined user groups, we designed the survey in 3 parts:

1. Background
2. For designers
3. For non-designers

The survey contained 26 questions, four for the first part, which asked about the participants' background, depending on their answer to this part they would either choose 'for designers' or 'for non-designers' part. A participant could answer a minimum of 12 and a maximum of 16 questions. The survey ran for 10 days and had a record of 40 full responses. The complete question set of the survey is provided in Appendix A.

3.1.3 Results

In this section, we are going to present the survey questions and the analysis of participants' answers to them.

Background

In this part, we asked about participants' background in four questions. The questions were about their age, gender, occupation, and if they have any design background. We had almost the same number of participants from each gender (60% female and 40% male), and were in the age range of 21-35 years old. 48% of the participants were students, but diverse with regard to their study field or field of occupation, 38% of them had design background.

For Designers

In this part, we asked participants with design background about their experience, their favorite modeling tool, their evaluation of the tools they have used so far, and also the average time they have spent using these tools.

Experience and Frequency of Design

The participants with design experience, were asked about the duration of their experience and the frequency of it. Our median participant is a 29 year old female architect, who has seven years of design experience for several times a week.

Evaluation of CAD Tools

In this part, we asked participants whether they use a CAD tool to model their design, and in case of positive answer, we asked them to name their favorite modeling tools and evaluate them.
First, the participants were asked to name their most used CAD tools, the results are shown in Table 3.1. As it can be seen CAD tools like AutoCAD, Rhinoceros 3D, and 3D Studio Max are the mostly used tools among our expert users (around 20%), which were mostly architectures and industrial designers. All these three packages offer a wide range of commands and menu options related to their application. On the other hand, SketchUp, which is an easy to use package, but with a smaller number of commands and tools, has the least popularity (2%) among experts.

CAD Tool	Usage Percentage
AutoCAD	23%
Rhinoceros 3D	20%
3D Studio Max	20%
Revit	10%
SketchUp	2 %

Table 3.1: The usage of different CAD Tools according to designer participants of the survey

Script-based Vs. Non-script-based CAD

The participants were asked whether they prefer script-based CAD tools such as OpenSCAD or non-script-based ones like SketchUp. As assumed before, most of them (92%) preferred non-script-based tools. They hinted to the wide range of menu options, functions, and user friendlier interface in non-script-based tools. Furthermore, most of are participants pointed out to not being familiar with programming as another reason for not choosing script-based tools.

Usability of CAD Tools

The participants were asked to evaluate the usability of four different CAD tools. These tools were chosen in a way that each of them emphasizes on an attribute of all existing CAD tools more than the others:

- OpenSCAD: script-based modeling
- AutoCAD: functions for 2D and 3D modeling, mostly used by architects
- SketchUp: easy to learn, almost no previous knowledge needed
- Blender: used for creating animation movies, visual effects, or video games

We asked participants to rate these tools as 'useful', 'not useful' or 'never used'. As it is illustrated in Table 3.2, AutoCAD with 81% and Blender with 75% are chosen as he most useful and the least useful tool respectively. SketchUp is the second most useful tool chosen, following AutoCAD. Concerning these results, we can conclude that, users consider having various functionalities for 2D and 3D modeling, and learnability as two important factors for usability of a modeling tool.

CAD Tool	Useful	Not useful	Never used
OpenSCAD	33%	8%	58%
AutoCAD	81%	9%	9%
SketchUp	58%	16%	25%
Blender	16%	75%	8%

Table 3.2: Perceived usability evaluation of CAD tools with different attributes

To ease comparison of the data in Table 3.2, we decided to present data in a diagram, to do so we mapped 'useful' to +1, 'not useful' to -1, and 'never used' to 0. Dividing the result by the number of participants, the means of CAD tools' evaluations are shown in Figure 3.1.

We also asked participants whether they have ever evaluated the usability of any other CAD tool, the most common answers were addressing Rhino and 3ds Max as useful tools.

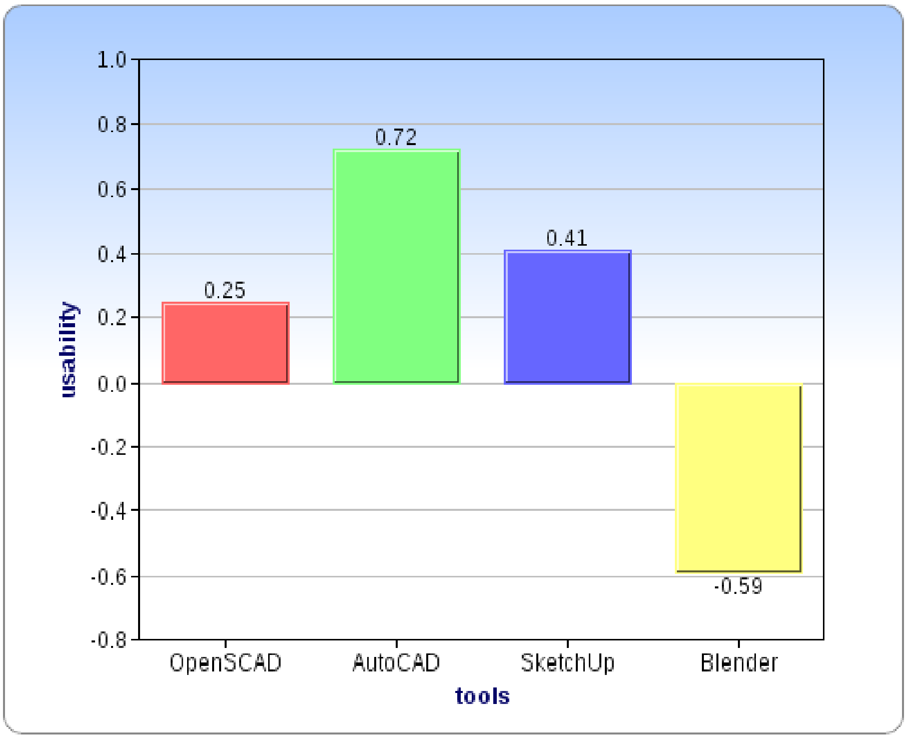


Figure 3.1: Comparison of different CAD tools by mean of evaluated usability from not useful(-1) to useful(1)

Design Purpose and the Tools Used

As these questions were asked from participants with design background, we needed to know on which purpose they design. The perceived results demonstrate that 45% of the participants design objects because it is their job, 27% do it as a part of their job or studies, 18% do it as a hobby, and 9% design for all the mentioned reasons.
Another question, which the participants were asked, was about the tools they use for designing. Table 3.3 shows the answers to this question; 38% of the participants chose using CAD, and 30% preferred to have the combination of CAD and pen and paper for visualizing their design ideas.

Tool	Usage
Pen and paper	15%
CAD	38%
Both pen and paper and CAD	30%
None	7%
Other	7%

Table 3.3: Results achieved after asking designer participants about their favorite tools to visualize their design idea

Afterwards the participants were encouraged to answer how satisfied they are with the tool they use; we reached 63% of satisfaction among them.

Time Spent

To verify the efficiency of the tools, we asked the participants how much time they use in average for visualizing a simple object with their preferred tool. The answers were as Table 3.4; Most of our users (63%) have estimated the time to visualize a simple object between five to 30 minutes.

< 5 min	14%
5-10 min	35%
15-30 min	28%
30-60 min	14%
> 60 min	7 %

Table 3.4: Average time used to visualize a design with users' preferred tool

Comments

Finally, the participants were asked if they have any comments on tools or if they would like anything to change. Having more menu options such as predefined polygons, polylines, and shapes, higher range of material, and more mathematical functions, were some of the answers. Others wrote about the gap between what human mind visualizes, and what the tools display, they thought the result displayed should be similar to what they implement there.

For Non-designers

This part is dedicated to our participants without design or 3D modeling background. We tried to get information about their preferences on visualizing their ideas by asking some questions. Initially, the participants were asked about their experience; 50% of our participants had tried to make things such as furniture pieces, ornaments, toys, and games for their every day use or as a hobby. Of this population, 72% have tried to design objects such as ornaments, toys, furniture and home interior decoration.

Visualization Method

To have an estimation of how a non-designer user visualizes her idea if she wants to design something, we asked them to choose between some methods that had been observed to be popular among them, or name their own method. The results are shown in Table 3.5. As it can be seen, the majority of our participants prefer to visualize their design ideas by pen and paper (38%) or computer software (23%), or using examples from similar objects (15%).

Drawing(pen and paper)	38%
Computer software(Paint)	23%
3d modeling tools(SketchUp)	7%
Explaining by words	7%
Making examples from similar objects	15%
Other	3%

Table 3.5: visualization methods for non-designer users

Interaction with Parametric 3D Modeling Tools

In this section, we introduce two different methods for parametric 3D modeling and ask participants, which they think is easier to use and which one is more interesting to them. The methods are:

1. Having a One-piece model and being able to change its dimensions.
2. Having several model parts that can be changed in dimension and attached together (like Lego).

Having observed the results, we could see that there was quite the same amount of answers on each of the methods (58% for having several parametric model parts and 41% for having a one-piece model). However, there was a noticeable difference in the participants' answers on which method they find more interesting; 83% of the participants announced the several parametric model parts method as more interesting.

Time Spent

As the non-designer users do not have a specific professional tool, we asked them about the average time they spend to visualize their design ideas to have an estimation of their expectation of the efficiency of a tool to be provided to them. As it is shown Table 3.6, almost 90% of our participants prefer to spend less than one hour to visualize their design ideas, and among these 50% wish to have their design visualized in less than 30 minutes.

< 5 min	15%
5-10 min	23%
15-30 min	15%
30-60 min	38%
> 60 min	7%

Table 3.6: Average time used to visualize a design estimated by non-designer users

Measurement Units

At last, the participants were asked how they prefer the parameters of a model to be defined, they were given an example of a ring with size parameters defined once by length of the diameter of the ring in millimeters in three sizes and another time by the market size units for wearables (small, medium, large). The achieved results showed that 58% preferred the market units and 41% would like to have the parameters defined in metric units.

3.1.4 Summary

Having collected the results of our survey, we have a brighter view on what users require from a modeling tool. Based on their background and design and 3D modeling experience, the requirements vary.
An average expert user who does 3D modeling on a daily basis for her job or as a hobby, is more comfortable with working with tools such as AutoCAD or 3D Studio Max, which follow the WYSIWYG concept and displays the model on the screen while editing it, and provides a high range of functions and menu options for creating complicated models. Since they are mostly not familiar with programming, they would rather work with non-script-based modeling tools. Our participants considered having various functionalities and learnability of a tool as two factors, which increase its usability. To visualize a design idea they mostly prefer using CAD and expect to do it under 30 minutes.
On the other hand, our non-designer participants visualize their design ideas by tools such as pen and paper and computer software, they also use examples from similar objects to describe their ideas. To 3D model an object, they find it more interesting to have several model parts that can be changed in dimensions and attached together. Comparing the units of measurement, more than half of our participants would rather use market units of measurement than metric units. The estimated time to visualize design ideas, is less than 30 minutes.

3.2 Problem and Proposed Solution

After reviewing the related works and having considered user preferences from the survey results , we came to a conclusion, which leads to our solution concept. In this section, we are going to present our solution concept briefly, but before that, we are going to re-define our target users.

3.2.1 Target User Groups and Their Problems

According to what has been done in the survey, we can categorize our users into two user groups:

1. Expert CAD users: users who are acquainted with 3D modeling with CAD tools

2. Novice users: users who are not familiar with 3D modeling with CAD tools and need a noticeable time (more than one month) to professionalize in modeling with a tool.

Having two user groups with different levels of 3D modeling background, eases the process of addressing the problems to their related user group and thus giving a better solution. For our novice users we have a target group of people who are the users without design and 2D/3D modeling background and are willing to make their everyday objects via a 3D printer; but on the other hand, we have the experts, who are either designers and 3D modeling experts or non-designers and 3D modeling experts. The difference between these two is having specialized in designing and using it as a profession (designers), or simply visualizing an idea through a 3D model (non-designers).
As our goal is to use the knowledge and information provided by experts to help novice users, first we had to have an estimation of how eager experts are to share their designs. To do so, we had interviews with professors in FH Aachen-University of applied sciences- Faculty of Design. We talked about our approach and the concept of sharing 3D models of designs online and having them modified by other users. Our results from these interviews expressed that, most of the professional designers do not tend to share their designs since they believe that it is their financial source, due to economic reasons, they would not share designs for free on the web. Moreover, they also do not desire to have their designs changed and customized by others, since they think the customized object is not their design anymore and violates their idea.
Having collected these results, we decided to limit our target expert users group, to those who share their 3D models online (e.g. on Thingiverse).
Another important fact that had to be considered, is the 3D modeling tool used by our expert users. Having observed 3D models shared on online libraries such as Thingiverse, we came to the conclusion that most of our target users use OpenScad (script-based modeling) to visualize their ideas into 3D models; and to download a model, one has to download the script, run it and have it rendered on the OpenScad desktop application on her own computer. Although this result, negates what we have achieved in our survey about script-based tools, but as we have to use the shared knowledge on the internet, we are obliged to follow it and limit our experts to people who share their OpenScad models online.
Pursuant to what is mentioned above, we define our target user groups again:

1. Expert CAD users: users who are acquainted with 3D modeling with OpenScad and tend to share their models online.

2. Novice users: users who are not familiar with 3D modeling with CAD tools and need a noticeable time (more than one month) to professionalize in modeling with a tool.

The first group, has to usually deal with the problem that after downloading the model code from a 3D model library, it takes a considerable amount of time to induce a small change in the code or read the code in general, due to lack of structure. One solution to this problem is to provide our expert users with a structure, where they can create new models by just modifying a piece of code, needless of coding everything from scratch.
The significant problems of the second user group are, the high learning curve of most of the existing CAD tools, the low range of objects covered by easy-to-use 3D customizers/creators, and the lack of flexibility in the models in 3D libraries. An application with a user-friendly interface, and providing the flexibility for the users to be able to customize their models to a high-level, can be a solution to this problem.
To have a better view on the requirements of our system we have created a list of the requirements of our system:

- High range of object 3D models
- High flexibility of the existing 3D models for customization
- High learnability of the system
- Capability of making complex 3D models without having 3D modeling skill (novice users)
- Understandable interface
- Understandable units of measurement
- High readability of 3D model code
- Capability of code reuse (expert users)
- Time efficiency of the system
- Easy and fast search and browsing for the models

In this thesis we have decided to start an approach, which aims to provide a structured coding system that, eases code reuse and readability for expert users and also use their shared models online as an input to a system, which creates an easy-to-use interface for each model with controls providing functionalities to customize the model. Moreover, we are defining a metadata structure for our models to ease browsing and comparison between them. In the following sections, we will describe these concepts in detail.

3.3 System Usability Scale on MakerBot Customizer

After running the survey, we gathered some information about the characteristics of a good 3D modeling tool based on user's knowledge, and background. As the concept behind ParaShape is a parametric customizer with a user-friendly interface, we found that MakerBot Customizer, which is introduced earlier in our related work, is the closest existing product to our solution concept. Therefore, we decided to do a qualitative user study on this tool. To verify how much it fulfills the requirements of the users.

3.3.1 Design of the Study

The subjects of the study were ten students with the age range between 22 to 30. From the aspect of 3D modeling background, our most professional user had tried making 3D models in Google SketchUp a few times, thus all the subjects are assumed in the novice-users group. Our goal was to discover the interaction flaws and strengths in the system for later use in our own solution. Doing so we decided to use the scavenger hunt method [CS01] coupled with users observation and think aloud technique to have a close inspection on the system use. At last, we handed in a System Usability Scale [Bro96] questionnaire to the participants, to evaluate system usability. The System Usability Scale questionnaire is enclosed in Appendix B.

Defining the Red Routes

Red routes are the critical tasks that the user carries out interacting with a system, which have to be completed as quickly and smoothly as possible to promote system usability [Jur10]. For using scavenger hunt method, the first thing is to define the red routes of the system. For MakerBot Customizer the red routes are as below:

- Launch the Customizer
- Explore 3D model Customizer for a specific object[1]
- Navigate the 3D model
- Change tabs
- Change parameter values
- Verify the result model
- Create STL from the model

3.3.2 Starting the Test

Having defined red routes of MakerBot Customizer, the participants were asked to carry them on after getting a brief introduction on the Customizer and its use.
After answering a few questions on their age and background in 3D modeling, the participants were given a sample 3D model and were asked to change the default model on the customizer to look similar to the sample and finally create the STL file of it. These tasks were chosen according to the data achieved from previous observations, where users had a model in mind or had seen an object before and preferred to customize it. To avoid bias in this study, we asked each participant to

[1] http://www.thingiverse.com/apps/customizer/run?thing_id=40703&code=cbb4240f777fd07a6fe11730b19241ab
http://www.thingiverse.com/apps/customizer/run?thing_id=45877&code=b2521234e4b0b16e5a41145c272d31c2

customize two different models, one being easy and another difficult; so that, the functionality of the customizer would not be judged by the variation of the models.
Participants were thinking aloud while carrying on the tasks. Hence, we could obtain data about their cognitive process while interacting with the system. Moreover, to avoid losing information that users do not mention in think aloud we also observed their interaction with the system.
After the test was done , the total time for completing the tasks was measured, and participants were asked to fill in the system usability scale questionnaire.

3.3.3 System Usability Scale Score

Conforming to the data gathered from the SUS questionnaire, which was given to the participants right after they had completed the test, we estimated the usability score of MakerBot Customizer 52%, which is a below average usability score (the acceptable usability score is above 70%).

3.3.4 Comments and Observation Results

In this part, we are going to present the comments given by the users when thinking aloud, and also the perceptions achieved from observing them. Six out of our ten participants managed to create a model by Customizer similar to the sample they were given. Two participants gave up trying after 15 minutes working with Customizer, giving the reason "It was exhausting", and the other two finished the work up to around 80 percent similar model, saying it was not clear enough for them which control to use for making the model better. Following, we declare some common negative points of the system mentioned by participants or observed in most cases, which we believe are the reasons for the below average usability scale.

- **Slow system feedback:** the main problem of the system mentioned by all participants, was the delayed feedback of the system after changing parameter values. The problem is due to the shortage of a 3D renderer to induce changes in the geometry of the model and render the updated model.
- **Poor 3D navigation:** users need to be able to do principle navigation acts like zoom in/out, rotation, and moving to have a clear image of the object they want to 3D print. In MakerBot Customizer, navigation control in 3D model is implemented in 5 buttons, two for rotation around the y axis, two for rotation around the x axis, and one to turn back the model to the default view 3.2. There are 3 main problems on the navigation system of the 3D display of MakerBot Customizer:
  1. Discrete rotation views: view points of objects are only defined in a few angles and users can not have a complete rotation around the object. This problem stands out while work-

ing with asymmetric detailed models.

2. Zoom function not implemented: There is no zoom function in Customizer to give users a more or less detailed view; hence, users have problems interacting with complex detailed models.

3. Moving function not implemented: sometimes the user needs to see the model from other viewpoints or in other location on the panel. Therefore, she needs to be able to move the model to her desired coordinates.

- **Vague mappings between controls and their functionalities:** One of the problems mentioned by most of our participants and also seen in observations, was the unclear labeling of controls and their functionalities. Users had to scroll up and down in the list of controls and reread the labels several times and do try and error to understand which control adjusts which parameter value. This took a considerable time and was one of the main reasons that two of our participants gave up. Even having a good code with well-annotated labels, the novice users , who are not familiar with geometric terms and semantics, might have problems in finding the mappings between controls and their functionalities. A solution to this problem could be real-time feedback of the system when adjusting a value to show which parameter has changed, another one could be highlighting the area in the 3D renderer, which will vary in case of value adjustment. Having labels with pictures showing the related parameters, also could be a solution.

- **Default values cannot be retrieved:** In all input controls, one cannot retrieve the default value after changing it unless refreshing the whole page, which refreshes all other customized values. Existence of a control that provides the default value is vital.

- **Reverse action not implemented:** there are situations when users want to change the value of the parameter from a value other than the default to a new value, in these cases if a reverse action like an "Undo" button is not implemented it would be complicated for the user to remember the previous value and return the model to its previous state.

- **Numerical inputs are read in digits:** in text input controls, when a numerical value is entered, it is read digit by digit. For example, entering the value 65 for a parameter, first the value 6 is taken and queried and then 65. This reduces interaction speed. The presence of an event handler, which for example reads input after the Enter key is pressed would eliminate the problem.

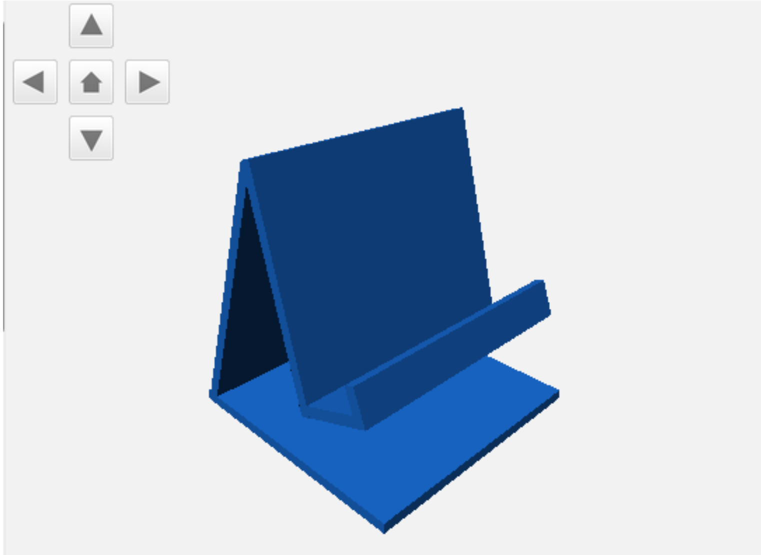


Figure 3.2: MakerBot Customizer 3D view panel and its navigation control buttons

3.3.5 Meeting the Requirements

Earlier in this chapter, we defined the requirements of a good customizer. Having run the study on MakerBot Customizer, we recognized that despite having a good concept behind it, our participants evaluated it as a not usable system. The reasons for this result are mostly mentioned in the comments given by the participants. Table 3.7 is a review on which of the requirements of our proposed system is met by MakerBot Customizer.

	Requirement	**met**
1	High range of object 3D models	√
2	High flexibility of the existing 3D models for customization	√
3	High learnability of the system	√
4	Capability of making complex 3D models without having 3D modeling skill (novice users)	–
5	Understandable interface	√
6	Understandable units of measurement	–
7	High readability of 3D model code(expert users)	–
8	Capability of code reuse (expert users)	–
9	Time efficiency of the system	–
10	Easy and fast search and browsing for the models	√

Table 3.7: Requirements met by MakerBot Customizer

We can conclude from Table 3.7 and the comments given by participants that Makerbot Customizer has met user requirements from the aspects of covering high range of models, learnability, flexibility of models, and understandable interface to some extent. But there is a lot to improve in this tool; MakerBot Customizer does not provide novice users with functionalities to make complex 3D models without being skilled, and the units of measurement used in this system are mostly metric units, which are difficult to estimate for the novice users. It also does not support code readability and code reuse for expert users, and there is no structure that guides experts to write a code in a way that makes the resulting customizer easier to use. The participants have also evaluated the system as not time efficient in case of complicated models.

3.4 Component-Based Models

As already mentioned in the last section and also in the related work, an obstacle in 3D modeling is that the tools designed for novices users are either too complex or cover a low range of object models. In the best case, a novice user can use an easy-to-use modeling tool and create a simple model, but when it comes to complex models, these tools lack many functionalities and menu options, or become difficult to learn. One of the approaches followed to solve this problem, is the component-based modeling, which is also defined as "Modeling by Example" [FKS+04], "Assembly-Based Modeling" [CKGK11] or "Model Composition from Interchangeable Components" [KJS07] in different literatures. The concept behind all these titles is to create complex models by composing model parts/components, which are taken from a database of existing models. This process is done either by cutting parts from already existing models in the same class of the final model (for example, if you want to create a chair model you take parts from already existing chair models in the library), or take parts from a component list, which is updated each time your model is changed to the components more relevant to you model, or shuffling interchangeable predefined parts.

In this thesis, we decided to use a parametric component-based modeling approach, as it has been seen that composing models from different components simplifies the process and also in our survey, participants mentioned to find it more interesting to use. To reduce processing cost, we decided to avoid cutting the parts from already existing models and rather have independent components. The basic assumption in our approach is that every object is made up of one or more components, and to create a model of each object one needs to be able to reach all needed components and attach them together. The advantage of this method in comparison with having one-piece-models, is the variety of choices provided by the range of components, which belong to the same category.

3.4.1 Hierarchical Presentation of Component Classes

To follow the component-based modeling approach, we encountered the need for having clusters to put relevant components in the same class, hence due to having the knowledge of which component class to look up, searching for a specific component would also be easier. We start with a simple example of an object, which is popular on 3D modeling libraries to be printed: a lamp. A simple electrical lamp is made of the components below (see Figure 3.3[2]):

- lamp base
- shade
- lamp socket
- bulb
- wire

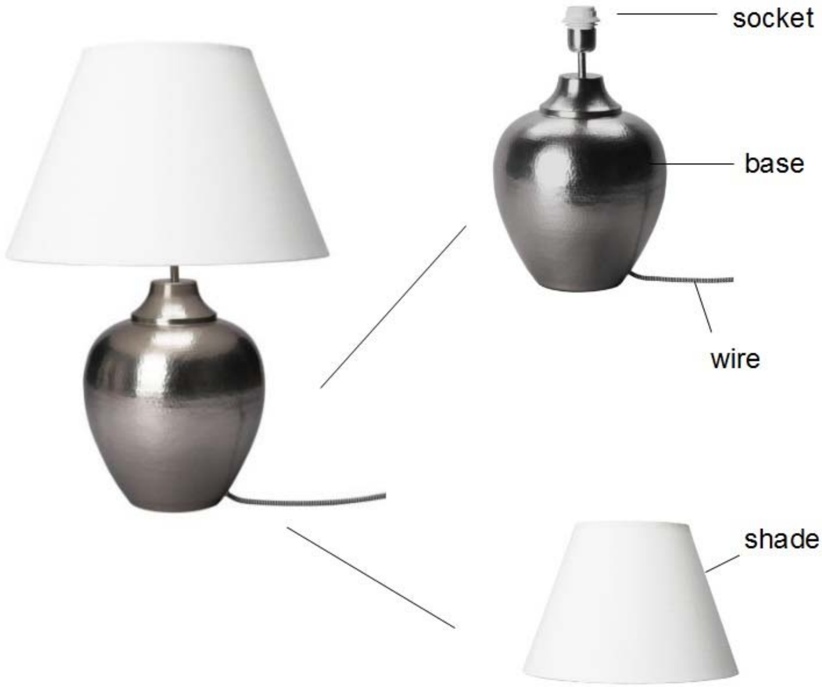


Figure 3.3: A simple lamp with its composing components

Except for the bulb and the wire, the three other parts can be printed by a home 3D printer. If we want to print a lamp, the first step would be to have a 3D model of the lamp we would like to print. We can simply search the word lamp in libraries like Thingiverse and obtain several models by different designers. But what if we would like to have the base from one model, the shade from one and the socket from another one? Here, we need a system to provide us these components separately and classified. Therefore, we dedicate classes to different components, for example here we can have a lamp shade class, a lamp socket class and a lamp base class.
Now imagine the situation we want to choose our lamp shade, due to the variety of designs we can choose among different styles such as Uno shades, bell style, coolie style, drum style, empire, hard back, hexagon style, oval style, rectangle style, square style and so on. It is obvious that having lamp shades of each style in one cluster is much

[2] Picture Credits for http://secure.business.ikea.com/pl-pl/products/s79861119/

more efficient than having them all in the lampshade class. Here, we define each of these clusters as a subclass of the lamp shade class, which has its own attributes other than the attributes of the lamp shade class (see Figure 3.4). We will follow the same structure for all components. This hierarchical tree structure of components, which is familiar to almost all computer users (e.g. windows tree view) enables them to have a clear view of component classes and their specificity, the more specific a component is the closer it is to the leaf nodes.

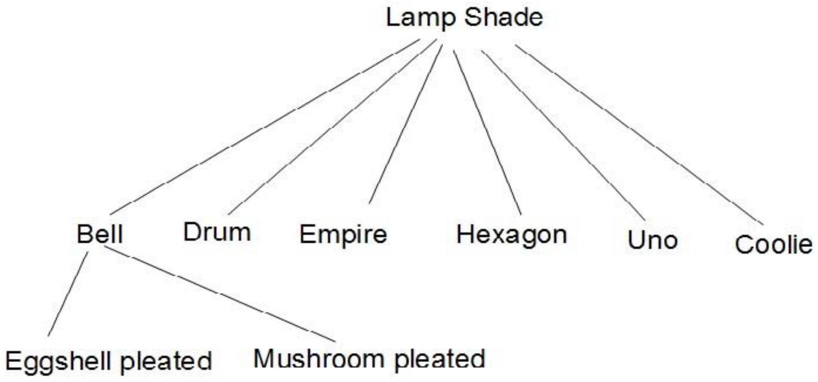


Figure 3.4: Lamp shade hierarchical representation

Considering our two user groups, this structure can ease searching and browsing components for both user groups, especially for novice users when they want to find and compose components to create a model, this can help them see all subclasses of a component at once. The question is if this structure can change anything in the 3D modeling process by expert user group.

As mentioned before, a problem, which designers and expert 3D modeling users face, is the lack of structure in the model code they get from 3D model libraries and its readability. Each user writes the code in her way and names the components according to her background knowledge and taste. Lack of structure in code can cause waste of several minutes finding a code element (e.g.variable). We decided to take advantage of the hierarchical structure to solve this problem and similar ones.

Our proposal is to use the root node of each component hierarchy as a template for all sub-components, for example for the lamp shades, we can use the root lamp shade class as the template, with a specific code structure and variables. Following this, to create a bell style lamp shade model template, one first needs to have the template code of the lampshade class with defined parameters and then add specifications to convert it to a bell style shade, and so on it works for an eggshell pleated bell style shade model, which is derived from bell shade template.

Having a template for all components provides a standard structure for coding them, which prevents users from facing non familiar code structure. Moreover, if an expert user wants to modify a model to some extent, needless of writing everything from scratch, she can insert her change in the relevant code element. The idea behind this method is not limited to standardizing the code structure for increasing readability, but providing error prevention by limiting user actions. The fewer the user's options are at a specific state of a system, the less her chances are to commit errors. Having a template for a component, limits you to follow the structure and prevents forgetting the fundamental elements in the code. We will discuss this more in Component Code Templates section.

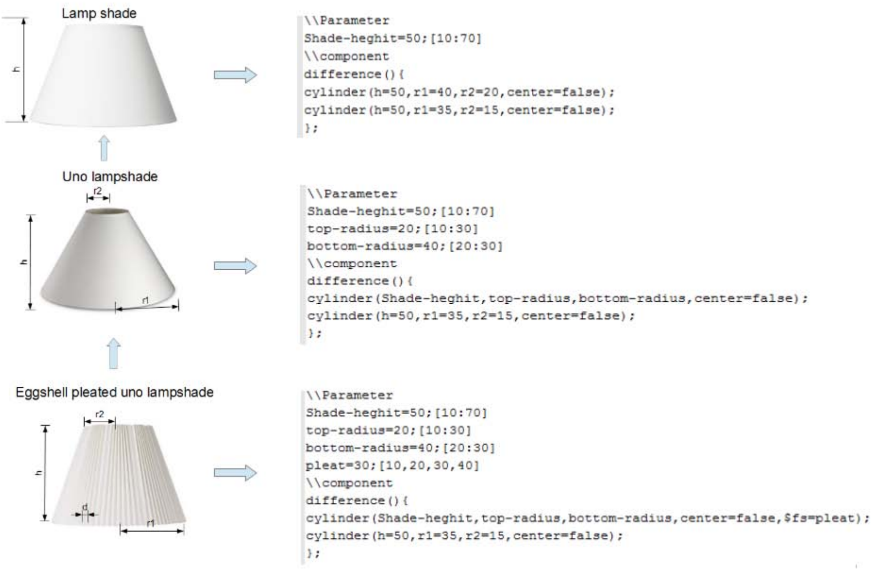
3.5 Primitive Instancing of Model Components

So far, we have defined our hierarchical structure for component models and making a complex model out of them to help novice users form new models, and for designers to get inspired by the work of others and reuse their code. Let us continue with the lamp creation example. We already know, which components a lamp consists of and we also know where to browse for a specific type of component. Now assume that we have found the proper components, the shade, the base, and the socket. But we would like to have a smaller shade with the same design as the one we have found or a base with a bigger bottom radius than the one we have. Therefore, we have to go through the code again, find the parameter definition of the element and modify the value. However, this method also needs the user experience in CAD modeling and makes novice users incapable of accomplishing this task.

As a solution to this problem, we decided to use primitive instancing of components. In this method, we define each component model as a primitive with a set of predefined parameters, and every model derived from them by changing these parameters is considered as an instance. Having parametric parts helps creating many instances from one single primitive.

One question is how can primitive instancing help our users? To response to this question, we propose having all components parametrized. Having a predefined finite set of parameters in each component model template gives the users the chance to customize components to their desired dimensions and then compose them together. As mentioned before, in our hierarchical representation of components the sub components inherit the attributes of their super components, parameters are of these attributes. Hence, every parameter defined in a component should be seen in all components, which are below it in the hierarchy. For example, the lamp shade component has one height parameter h, the Uno style lamp shade component, which is a sub-component of lamp shade has also the height parameter h and also two other parameters $r1$ and $r2$, which are the radiuses of the bottom and top circular cross sections of the shade, respectively and the eggshell pleated Uno style lamp shade has an extra parameter *d*, which is the width of each pleat (see Figure 3.5).

An advantage of primitive instancing is that, when expert users use



```
\\Parameter
Shade-heghit=50;[10:70]
\\component
difference(){
cylinder(h=50,r1=40,r2=20,center=false);
cylinder(h=50,r1=35,r2=15,center=false);
};
```

```
\\Parameter
Shade-heghit=50;[10:70]
top-radius=20;[10:30]
bottom-radius=40;[20:30]
\\component
difference(){
cylinder(Shade-heghit,top-radius,bottom-radius,center=false);
cylinder(h=50,r1=35,r2=15,center=false);
};
```

```
\\Parameter
Shade-heghit=50;[10:70]
top-radius=20;[10:30]
bottom-radius=40;[20:30]
pleat=30;[10,20,30,40]
\\component
difference(){
cylinder(Shade-heghit,top-radius,bottom-radius,center=false,$fs=pleat);
cylinder(h=50,r1=35,r2=15,center=false);
};
```

Figure 3.5: Each component inherits the parameters from its super-components in hierarchy and the code is extended each time a new instance is derived.

templates to create new components having the parameters set defined in the template, they will not forget them and the existence of already defined parameters prevents possible errors in parameter definition.

Up to now, we have presented a method to enable users to change parameter values of component model primitives and creating new instances of them. Nevertheless, this has not yet solved the problem for our novice users, since they still have to go through the code to modify parameter values. What is required here, is an easy to learn tool, which can enable users to modify the parameter values without having to go through the code. We decided to have a framework, which enables users to change the parameter values in the code through a simple interface with familiar controls.

ParaShape, which is our proposed solution for all problems mentioned so far, is a framework, which receives 3D model code as an input and converts it to an application with a 3D renderer, which displays the model and several controls and menu options, which enable users to change parameter values through them.

One problem with having parametric components, is that giving the users the freedom to modify parameter values might cause design violation or cause errors in the code. In ParaShape, we ask our expert users to insert parameter value ranges for each parameter they define in the code. So, the controls in the interface are defined in this value range and consequently, the user is limited to only choose values included in this range.

3.6 Units of Measurement

In the previous section we proposed having parametric parts with predefined parameter ranges to prevent design violation. This make 3D modeling process less complicated for our novice users. However, we still wonder if there is a way to make this process easier.
Considering a novice user without any 3D modeling and design background. It would be more laborious for her imagining how exactly big is a cylinder with a cross section radius of 10 millimeters and height of 44 millimeters, than imagining the magnitude of a AAA battery. This led us to the direction that we should think of more familiar terms for defining units of measurement for parameter values.
To ground our assumption on some approved information, before giving any proposal, we decided to conduct a study to verify if this problem exists among our subjects and which units of measurement they prefer.

3.6.1 Design of Study

To understand the users preferences between these two systems for defining units of measurements, we decided to have a within subject group study in which the same group of subjects was asked to try both systems and say which system they would prefer to work with. Here, we named the first system the *metric measurement units* (millimeter, centimeter, meter) and the second system *market measurement units* as it uses the units that users are familiar with, in the market.
Subject: Ten novice users without 3D modeling experience in the age range of 23 to 30 years old
Independent Variable: the measurement unit (market vs. metric)
Dependent Variable: acceptability of the system for the user
Hypothesis: market units of measurement are more acceptable for non-expert 3D modeling users.

3.6.2 Test and Results

To verify the correctness of our hypothesis, we decided to conduct the study only working on non-expert users. Having ten participants between 23 to 30 years old, we started the test by giving them 4 different object models and asking them which units of measurement they would prefer for each object. The objects were a battery case, a ring, a lamp socket, and scissor handles. For each of these objects, we brought an example of the market unit and one from the metric unit. For example, for battery case we used 10mm diameter of cross section and 44mm height as metric example, and AAA battery size as market size, for the ring, the diameter dimension in millimeters and ring size (e.g.size 16), for the lamp socket the diameter of the socket and the lamp socket model (e.g. E17) and for the scissor, which was aimed to be printed for a child we proposed handle ring diameter as metric and "child size" as market unit of measurement. We provided users with real objects (a battery, a scissor, a lamp bulb , a lamp, and a ring) in case they needed to have them in their hands. We also gave the participants a tape measure to be able to measure things if they needed to. For each object we asked participant to rate the acceptability of the measurement units a 1-5 Likert scale.

As participants were rating the measurement units, we asked them to explain why they prefer them to the other units. As they completed the test, we asked them to give comments on their preferred units or any suggestions to improve them.

According to our results, 76% of our answers rated the market units of measurement as a more acceptable system. Our participants declared that these units were more familiar to them and they found them easier to estimate dimensions of the objects rather than the metric units. Moreover, they claimed that using standard sizes of the objects like batteries or lamp sockets, prevents measurement errors that occur when using metric units and measuring them, themselves. However, some users believed that for wearables such as rings, it would be better to have metric units besides market ones. Because, the reason that some users would like to print their wearables, might be the case that they have a size between two defined market sizes and therefore they would need to customize their model to the appropriate size using metric units.

3.7 Component Code Templates

One of the main goals of ParaShape, as mentioned before, is to help expert users group to use each other's work. Assuming all our expert users start using the hierarchical structure of model components, there always exists a risk of forgetting a parameter in the code or inducing one's own taste in the structure of the code, which prevents other users to take advantage of the code easily.

Error prevention has always been a target for UI designers and various approaches have been proposed to overcome this problem; however, we decided to go through principles of human computer interaction to find a solution. Schneiderman in his eight golden rules of interface design, has pointed out to "strive for consistency" as the first rule [SB98]; this rule says that similar situations should require similar actions. Lack of consistency always violates users' expectations and memory and thus makes them confused.

An approach proposed for preventing errors and inducing consistency, is applying constraints. Norman in his book "The Design of Everyday Things" indicates that the way to make a product less erroneous is to make it impossible to do it otherwise, in other words by limiting the choices of the actions user can take, we can lead her to the right direction [Nor02].

Having the concepts above in mind, we thought of using a standard structure and template for the 3D model code to prevent users from spending their time for comprehending the code, finding elements in it, or making loss of activation errors by forgetting important elements in the code. For each 3D model component, there exists a user-defined template, which other 3D model instances are derive from. All components derived from the same template, share some parameters and methods. Therefore, there should be the equivalent structure in the code body of them. Moreover, the extra parameters or functions added to the derived components should also have a similar structure to ease scanning through the code. And, finally all 3D model components should look similar in the structure of the code, so the users' expectation is not violated moving from one component code to the other one. She also would not need to browse through the code to find one element. Providing this template, we make users limited to follow this structure and thus the chance to make errors is reduced.

3.8 AMF File Format

After creating the entire 3D model out of its components in the proper dimensions, there are still some points left to induce one's taste into it: the color, the material, and the texture. STL file format, which is widely used for additive manufacturing does not support these features, thus despite that the user's 3D printer is capable of printing in different colors and materials, the file format does not support it.

As in ParaShape, we are aiming to provide users with a flexible 3D modeling tool, we decided to use AMF file format, which is more supportive in comparison with STL. Color, texture, and material of each object can be defined in AMF. As information on AMF is saved in XML format, it is supported by a high range of tools for creating, editing, and viewing AMF files. Moreover, XML files are human-readable that makes it easy to debug errors in the file. Furthermore, XML format is flexible so that missing or extra elements do not cause problems for parser as long as the XML standard is not violated. Last, but not least, one of the important specifications of AMF is, that we can define the physical measurement units (millimeter or inch), as we assume our users to be from all over the world, using different units, would be problematic in changing parameter values and understanding the dimensions of the object to be printed [HL09].

In ParaShape we save each component of 3D model and the whole model in AMF format so that our expert users can go through the XML file and change attribute tags (color, material, texture) of the component and the novice users can perform the same actions by manipulating dedicated controls on the interface.

3.9 Collaborative Tagging of 3D Models

As already mentioned, ParaShape targets two main goals:

- code reuse for expert users, and
- creation of complex models form parametric parts for novice users.

For both of the objectives above, users first need to search for the model components that fit into their work context. Therefore, they need to have component names. As assumed before, all components are created by our expert users and thus named by them, without using a framework obliging them to choose preassigned keywords.
Adding metadata to content in the form of keywords, labels, or tags ease searching process. Collaborative tagging, which is the process of annotating data by assigning tags to them without the supervision of any librarians to verify the validity of tags, has grown popularity among web users in recent years [GH05]. As in ParaShape all the models are created and shared by users, we decided to provide them with a tagging system to annotate models with the most relevant keywords; thus, models will be searched and categorized by the tags users have given them. However, due to lack of supervision, there exists the risk of appearance of irrelevant tags. To avoid irrelevant tags being used as keywords for search due to their popularity, we decided to define a rating system for the tags. So, not only users can add their own tags, but also can they rate existing tags. Having such a system, the most relevant tags will be ranked higher and used for searching.

3.10 Paper Prototype

3.10.1 Design of Study

After having all concepts of our work defined, we decided to test the combination of all of them in a low fidelity paper prototype to see how they work together. Due to having two user groups, our test target is also divided into two categories. As we have observed and mentioned before, most users of 3D printing service providers such as Thingiverse use script-based 3D modeling tools like OpenScad. Thus, we tried to simulate a paper version of a tool, which provides our expert users an script-based 3D modeling environment, to verify whether our system eases the process of script based 3D modeling and if code reuse is supported. Moreover, we had to test whether complexity of navigation through the code is reduced, and how much effort is needed for creating an instance from a primitive 3D model. The target to test for novice users group, was how easy they can find a 3D model, browse and change its components, and modify parameter values for each component.
We created a paper version of ParaShape, which is assumed to be a web application. The views are simulated by using A4 (297mm × 210mm) sheets of paper. To ease process for both user groups, we dedicated an independent view to each group: a general view for our novice users group, and a professional view for expert users; in our professional view, we have an interface similar to what they have seen in OpenScad plus extra controls for the added functionality for modifying model components and adding parameters or tabs to the interface for novice users. And, for novice users, the interface is similar to MakerBot Customizer, but with edited and extra features derived from our results

Use case	create a 3D model
Actor	expert user
Pre-condition	user has a design idea in mind and needs a 3D model of it
Post-condition	user has a 3D model of her design idea
Main flow	1. Log in to ParaShape 2. Go to Create page 3. Choose professional view 4. Find an object model or model template 5. System displays the template or list of object models 6. Add components to the template 7. Add parameters to the template 8. Add parameter tabs to the interface 9. Export AMF of the model 10. Add model to "My things"
Alternative flow	4a. Find an object model or model template 4a.1. Tag model 4a.2. Rank Tags 5a. Component template does not exist 5a.1. Search again 5a.2.Create a new component 5a.2.1 System brings the general object template with a minimum of 1 component

Table 3.8: Create a 3D model use case by an expert user

from the system usability scale study and the think aloud observation that we ran earlier in this chapter.

3.10.2 Use-case Scenarios

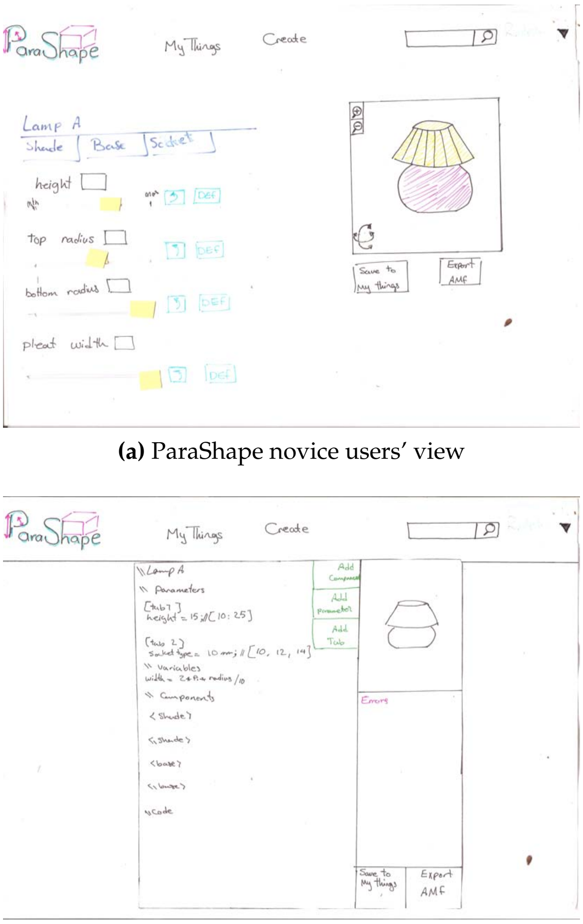
Before presenting the test results, for the reader's comfort, in this section, we present a detailed description of the scenarios of the main use case of ParaShape. Tables 3.8 and 3.9 show the use-scenarios for expert and novice users respectively.

Use case	create a 3D model
Actor	novice user
Pre-condition	user has a design idea in mind and needs a 3D model of it
Post-condition	user has a 3D model of her design idea
Main flow	1. Log in to ParaShape 2. Go to Create page 3. Choose general view 4. Find an object model 5. System displays the list of object models 6. Add components to the model 6.1. Find proper components 6.1. Add an locate them to the existing model 7. Modify parameters of the model 8. Navigate with the 3D model and zoom in and out the model 9. Export AMF of the model 10. Add model to "My things"
Alternative flow	4a. Find an object model 4a.1. Tag model 4a.2. Rank Tags

Table 3.9: Create a 3D model use case by a novice user

3.10.3 Test

The testers were initially asked about their occupation and 3D modeling (especially OpenScad) background; and depending on their answers, they were given a questionnaire to fill in after the test. So that, the information about their satisfaction of the system, which would not be received by observation and think aloud during the test with paper prototype was documented.
After having the questions about their background answered, we showed testers their own home page of ParaShape, which showed their username on the top right corner as if they are logged in; users were able to access their own 3D model archive "My things" and the "Create" page, through which, they could make new 3D models. There also existed a search box, which allowed users to search existing models. After entering the Create page, the testers where guided to two views: general and professional, which led them to the appropriate interface according to their knowledge; a script-based 3D modeling environment for experts and a view of the model with input controls for changing the parameter values for the novice users. Images of these environments are presented in Figure. 3.6 .



```
\\Lamp A
\\ Parameters
[tab1]
height = 15;//[10:25]

(tab 2)
socket type = 10 mm;//[10, 12, 14]
\\ variables
width = 2*Pi*radius/10
\\ Components
<Shade>

<\shade>

<base>

<\base>

\\Code
```

(a) ParaShape novice users' view

(b) ParaShape expert users' view

Figure 3.6: We defined separate views for each users group

Testers from both user groups were asked to create a specific type of lamp model with ParaShape (imagining they have seen the lamp somewhere before, and they want to have something similar but not exactly the same). After they searched the existing models of a lamp, they were asked to customize the lamp by exchanging its composing components, changing parameter values, color, texture, and material of the 3D model.
To show real-time rendering of the 3D model after each modification, various images of the object were shown to the testers, which were relevant to the change they have performed. Input controls, such as sliders and drop downs were built, in a way that users could interact with them; so, it would give them a more real feeling of interaction than an inert drawing. For 3D transformations, such as rotation, moving, zooming in and out, we offered controls on the display and mouse-and-keyboard shortcuts similar to what is seen in other 3D renderers(left click + drag = rotate, right click + drag = move, middle click + scroll down and up respectively = zoom in and out respectively) (see Figure 3.7 and Figure 3.8). More images of the paper prototype are presented in Appendix C.
An important point, which had to be considered designing the paper prototype, was that if we give complete freedom to novice users to change parameter values to any possible value, there would be the risk that the design is violated or the object would not be stable and printable anymore. To overcome this problem, we ask expert users to opt a finite set of values for input control of each parameter, so that the novice user is obliged to choose among the offered values and the design will not be violated.

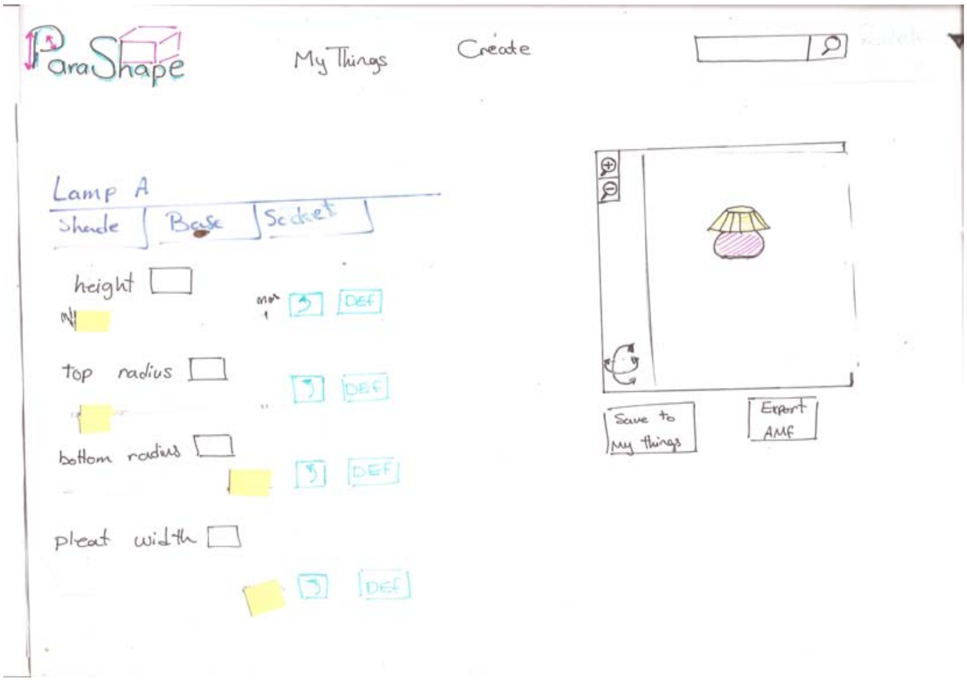


Figure 3.7: Model navigation and transformation on Parashape - zoomed out model

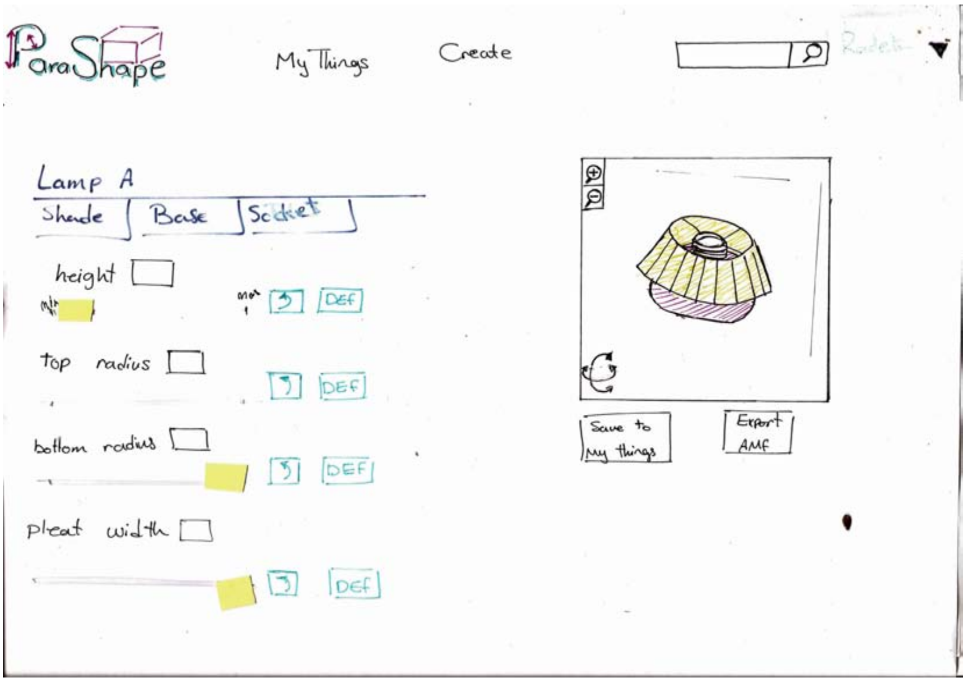


Figure 3.8: Model navigation and transformation on Parashape - rotated up model

3.10.4 Results

Participants

Our participants were divided into two groups of expert and novice users. In the expert users group, our median participant is a male (90%) student with the age 24 , who is semi-professional in 3D modeling with tools like Google SketchUp and OpenScad. On the other side, our median novice user is a 26 year old male student, who has not 3D modeled before or have just worked a few times with Google SketchUp.

Questionnaire and Observation

As mentioned earlier, after the test was done, the participants were asked to answer a questionnaire. There were separate questionnaires for expert and novice users, each containing five questions, which were answered on a 1-5 Likert scale. Tables 3.10 and 3.11 show the results in detail.

	Question	**Mean**
1	Did you find navigating in the existing code easier than in OpenScad environment?	4
2	Does having a template code for models reduce errors such as forgetting parameters?	4.6
3	Do you think having standard model code structure increases code reuse?	4.1
4	Do you think working with ParaShape is less time consuming than the usual process of 3D modeling?	4.1
5	If you had a tool like ParaShape with standard model code templates, would it motivate you to 3D print more things?	4.3

Table 3.10: Paper prototype questionnaire results for expert users. Questions were answered on 1-5 Likert scale

	Question	**Mean**
1	Did you find learning to work with ParaShape difficult?	4.2
2	Did you find the time you would spend for a model on ParaShape appropriate?	4.8
3	Did you find changing model components easy?	4.1
4	Did you enjoy working with ParaShape?	4.3
5	If you had a tool like ParaShape, would it motivate you to 3D print more things?	4.5

Table 3.11: Paper prototype questionnaire results for novice users. Questions were answered on 1-5 Likert scale

As it can be seen in tables 3.10 and 3.11 our testers were mostly satisfied with our prototype; however, there were still points to improve, which they mentioned in their comments or were observed by us.
Expert tester agreed on question 4 on their questionnaire that if there exists templates for the object they would like to create, it would make the process less time-consuming. They also agreed that the component templates reduce error and increase code readability; nevertheless, they preferred having a preview of the customizer as they are editing the code. Moreover, they offered having code blocks (areas of code) they can drag and drop so they wouldn't have to cut and paste for re-ordering items on customizer interface. Another comment was about adding tab blocks in the code, which create the tabs of the customizer's interface; users did not have a clear picture of what they are and they preferred having separated tabs each dedicated to one component and automatically created by the system without having an element in the code.
On the novice users side, despite the fact that all agreed on ParaShape being a usable tool, there were drawbacks on modifying the model:

some users had problems having some actions performed on input controls (sliders, drop down lists and text inputs) and the rest of the actions such as choosing a component to be changed or edited on the 3D renderer; this dichotomy confused the users at the first glance at the prototype, they expressed that they preferred both sets of actions on one side, or a hint on interface to make it more visible for them where to perform which action.

3.11 Balsamic Mockup

After evaluation and analysis of our paper prototype, we decided to redesign our system concerning the results. Our aim is to test the principles mentioned so far and see if both of our user groups would be satisfied enough to apply them in the future.

Before going through the explanation of the design of the mockup, we give a brief introduction of the principles that had to be concerned in the design and test for each user group:

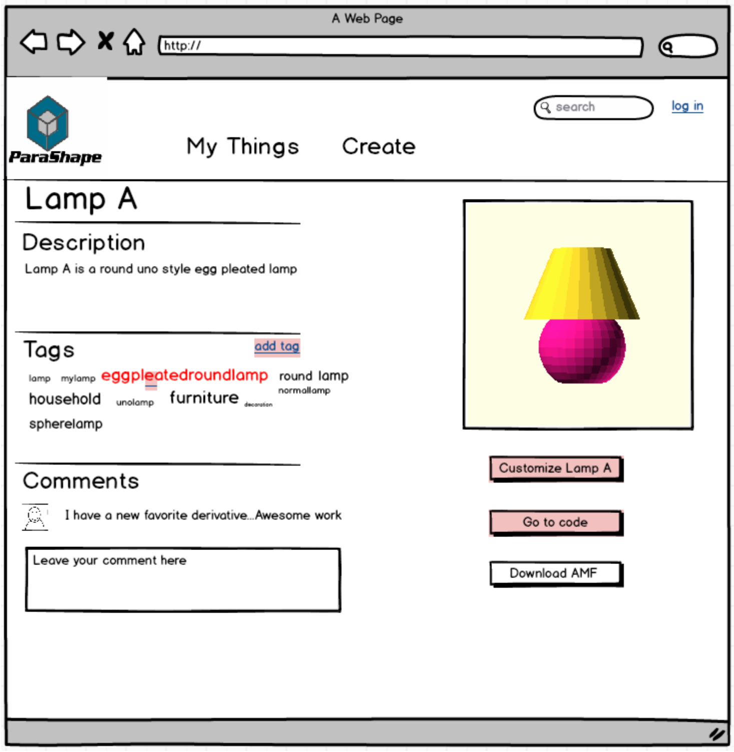
- **component-based modeling:** making complex models from composing components (both expert and novice users).
- **primitive instancing:** deriving customized instances out of existing primitive model components(both expert and novice users).
- **units of measurement:** changing parameter values by using market units of measurement(novice users).
- **component code templates:** understanding and using others' code easily; interacting with the new environment and code structure (expert users).

Our goal at the end of the evaluation of our mockup was to see if our system is efficient, effective, and satisfying after using these patterns.

3.11.1 Design

The results of the paper prototype evaluation and our observations led to some changes in the structure and flow of actions in ParaShape; some of these points are:

- Having two separate views for expert and novice users, prohibit experts from using the novice users view to customize a component of a model and then modify another component by manipulating its code. To avoid this problem, we decided to have one general view, which guides all users to the customizer's page. In the customizer page, there exists a button which links users to the coding environment, which was our former "expert view" page (see Figure 3.9).



Lamp A is a round uno style egg pleated lamp

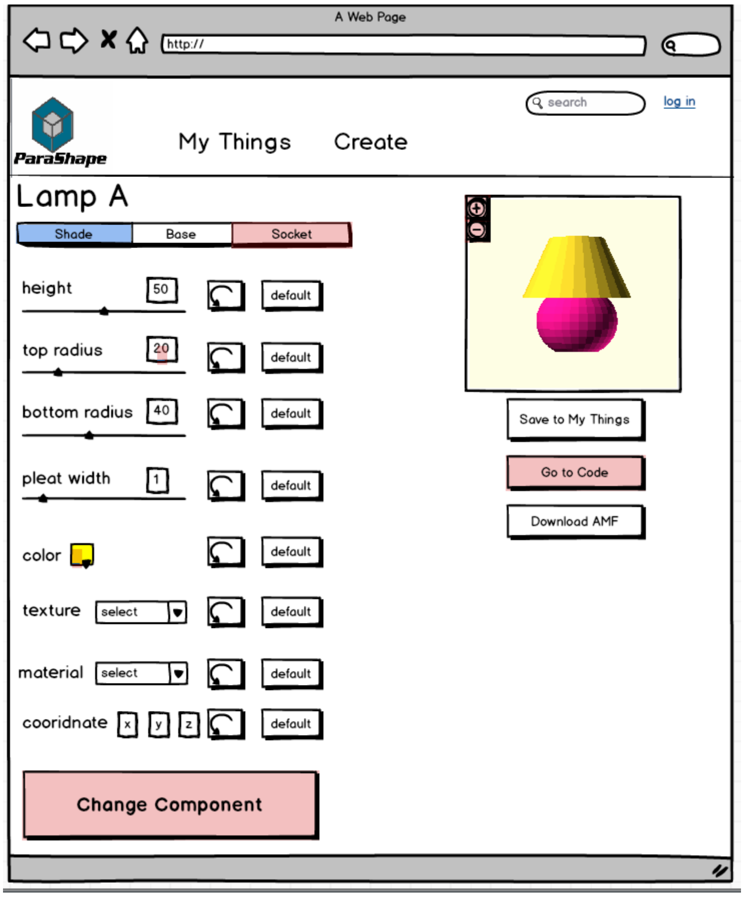
I have a new favorite derivative...Awesome work

Figure 3.9: The description page has links to both expert and general view

- In our paper prototype, we asked the experts to define tabs for the controls' panel. As they did not have a clear image of the customizer, which would be created from their code, they had no idea what are the tabs good for. Moreover, as they might have not worked with the customizer, they would not feel the necessity of having tabs to cluster the controls on the panel. Having observed these issues, we decided to remove the add tab button and instead, dedicate a separate tab to each component of the model automatically. In this way, the expert users will not get confused by creating tabs and choosing their contents (see Figure 3.10).

- Another problem observed in our paper prototype study was that in our novice user view (customizer) some of the actions were controlled by the input controls and others by interacting with the 3D renderer. Some of our testers were confused by this dichotomy and spent a noticeable time to find the appropriate controls for each action. As a lot of our testers expected all controls to be on the panel, we decided to move them all to the panel. So that, when the users enter the customizer view, they face a renderer and a controls panel, where a tab is dedicated to each composing component of the model and all dedicated controls of the component are presented there (see Figure 3.10).

- Component coordinates after changing a component were considered to be the same as the previous coordinate; after changing some components, it turned out that choosing a constant value



height 50

top radius 20

bottom radius 40

pleat width 1

color

texture select

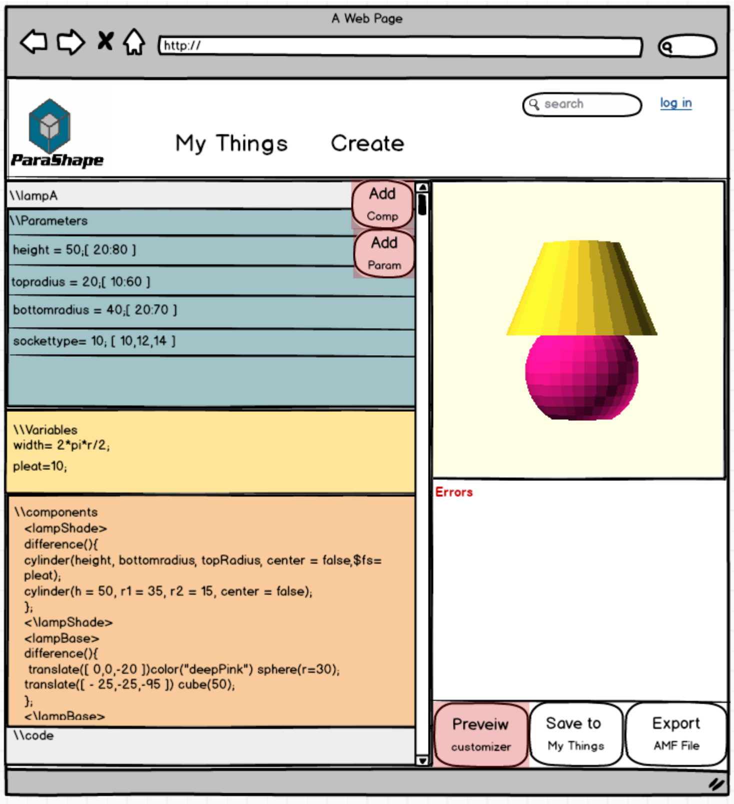
material select

cooridnate x y z

Figure 3.10: A tab is dedicated to each component containing all the relevant controls to customize it

for the component coordinates limits users to have the models which might not be their desired ones. Therefore, we decided to add an input control for each component's coordinates on the customizer.

- In our expert view, while writing the code, testers needed to have a picture of the final result and to know how the customizer looks; hence, we added a new button to this view which links the users to the preview of the customizer. Doing so, users can see the final product while working and can edit the outcoming customizer (see Figure 3.11).
- To increase readability in the text editor of the expert view, we added colored code blocks for each code element such as parameters, variables, and component codes. Having a different color in the background of each code block, helps users find the elements easier in the model code, and also moving from one model code to another (see Figure 3.11).



```
\\lampA
\\Parameters
height = 50;[ 20:80 ]
topradius = 20;[ 10:60 ]
bottomradius = 40;[ 20:70 ]
sockettype= 10; [ 10,12,14 ]
```

```
\\Variables
width= 2*pi*r/2;
pleat=10;
```

```
\\components
  <lampShade>
  difference(){
  cylinder(height, bottomradius, topRadius, center = false,$fs=
  pleat);
  cylinder(h = 50, r1 = 35, r2 = 15, center = false);
  };
  <\lampShade>
  <lampBase>
  difference(){
   translate([ 0,0,-20 ])color("deepPink") sphere(r=30);
  translate([ - 25,-25,-95 ]) cube(50);
  };
  <\lampBase>
\\code
```

Figure 3.11: Colored code blocks and preview button added to the expert view

Chapter 4

Evaluation

4.1 Balsamic Mockup Evaluation

Since we wanted to verify the results from our paper prototype test, we decided to conduct the Balsamic mockup test in the same way. However, we modified several points in our earlier user test; after asking users to perform the tasks while thinking aloud, they were given a Post Study System usability Questionnaire (PSSUQ) version 3 [SL12]. The reason we chose this test was that this questionnaire gives us the chance to evaluate the usability from four different aspects: overall usability of the system (Overall), system usefulness (SYSUSE), information quality (INFOQUAL), and interface quality (INTERQUAL). Using this questionnaire, we can specify the points of power and weakness of the system. The questionnaire consists of 16 items, which can be scored on a 7-point scale from *Strongly agree* to *Strongly disagree*. The complete questionnaire is enclosed in Appendix D .
The participants from both user groups (expert and novice users), were asked to start the test from the home page of ParaShape having the main task of creating a lamp model. During the test, after they reached specific levels, they were asked to perform some sub-tasks (such as zooming in and out the model). Tables 4.1 and 4.2 show the use-case scenarios of the test in detail.

Observing users while interacting with the system and extracting results from the PSSUQ, gave us quite an ascertaining information to verify the usability of ParaShape. However, one of our main goals of running this study, was to verify the admissibility of the principles derived from studies and observations that we ran earlier in this thesis. Following, you can find the hypotheses declaring these principles:

- **H1:** Users find it easier and more interesting to make a 3D model from existing components than making a complete model on

Use case	create a 3D model of a lamp
Actor	expert user
Pre-condition	user has a design idea of a lamp in mind and needs a 3D model of it
Post-condition	user has a 3D model of her design idea
Main flow	1. Log in to ParaShape 2. Go to Create page 3. Find an object model or model template 4. System displays the template or list of object models 5. Go to Customizer 6. Customize model by changing Parameter values 7. Change component 8. Go to code 9. Add components to the template 10. Add parameters to the template 11. Preview customizer interface 12. Export AMF of the model 13. Add model to "My things"
Alternative flow	3a. Find an object model or model template 3a.1. Tag model 3a.2. Rank Tags 4a. Component template does not exist 4a.1. Search again 4a.2.Create a new component 4a.2.1 System brings the general object template with a minimum of 1 component 9a. Desired component does not exist 9a1. Get an empty template and write your own code 9a2. Choose from one of the existing components and manipulate the code

Table 4.1: Create a 3D model of a lamp use-case by an expert user

Use case	create a 3D model of a lamp
Actor	novice user
Pre-condition	user has a design idea of a lamp in mind and needs a 3D model of it
Post-condition	user has a 3D model of her design idea
Main flow	1. Log in to ParaShape 2. Go to Create page 3. Find an object model or model template 4. System displays the template or list of object models 5. Go to Customizer 6. Customize model by changing Parameter values 7. Change component 8. Export AMF of the model 9. Add model to "My things"
Alternative flow	3a. Find an object model or model template 3a.1. Tag model 3a.2. Rank Tags 4a. Component template does not exist 4a.1. Search again 4a.2.Create a new component 4a.2.1 System brings the general object template with a minimum of 1 component

Table 4.2: Create a 3D model of a lamp use-case by a novice user

their own.

- **H2:** Users prefer deriving customized instances out of existing primitive model components rather than making their own model.

- **H3:** In case of objects with international standards for their size, users prefer using market units of measurement rather than metric units.

- **H4:** Expert users prefer the component code templates to the default coding environment of OpenScad (empty blank page).

To verify the accuracy of these hypotheses, we decided to implement another mockup, which does not follow our principles and investigate users' preferences after having worked with both systems. Our second mockup was a prototype for a non-component-based 3D modeling system, which used metric units of measurement and could not derive customized instances of other model components. Moreover, in the interface of the customizer of this system, there existed no tabs to classify the parameters based on the components they belong to. Figure Figure 4.1 demonstrates the customizer view of this mockup. .

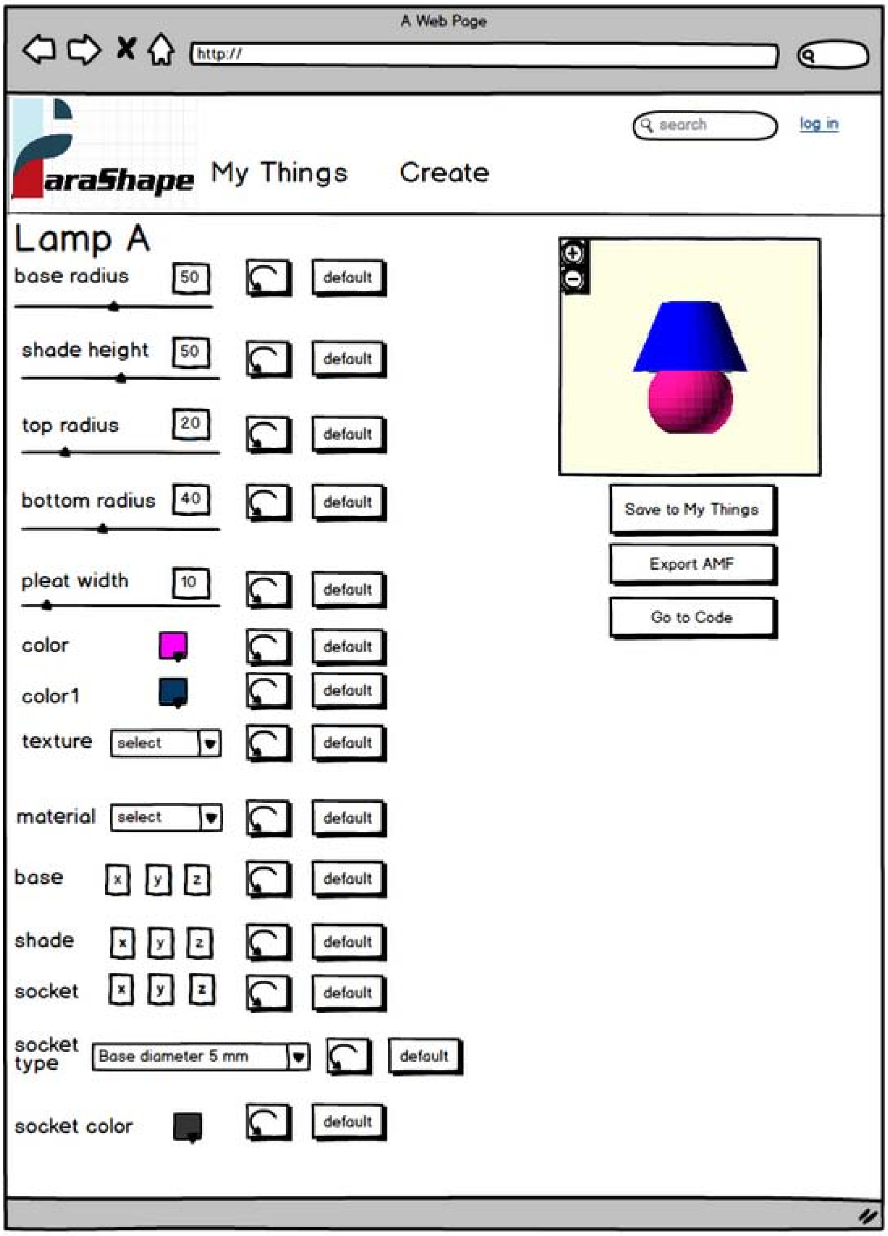
4.1.1 Results

Participants

Similar to our paper prototype study, we divide our participants into two groups of expert and novice users. The total number of our participants is 27, 15 novice users and 12 expert users. We had 33% female and 66% male participants. The participants were in the age range between 20 to 33 years old. Our expert users, according to what has been defined before, were chosen from people familiar with 3D modeling with OpenScad.

Time Used

For all participants, we measured the time it took them to complete their tasks. All expert users managed to finish their tasks in less than 20 minutes, and for our novice user the longest time it took them to do all the tasks was 10 minutes. According to what users mentioned in our initial survey about the estimated time that they need to visualize a design idea (30 minutes), and the feedback of our participants on the time efficiency of our system, we find these results satisfying.



Lamp A

base radius 50

shade height 50

top radius 20

bottom radius 40

pleat width 10

texture select

material select

socket type Base diameter 5 mm

Figure 4.1: To verify the correctness of our hypotheses, we implemented a second mockup without the principles followed in.

Post Study System Usability Questionnaire

After completing all the tasks, our participants were asked to fill in the post study system usability questionnaire, in which they had to score the usability of ParaShape by rating 16 items from one (highest score) to seven (lowest score), or not applicable (NA) when an item does not apply to the system. After the questionnaire was filled in, we went over all the answers with them to make sure everything is understood completely. They were also asked to inscribe any further comments they had on the system.

Having collected all the questionnaires, we computed the results. As it was mentioned before, the PSSUQ has four scores, one overall and three sub-scales. Table 4.3 demonstrates the results obtained.

Overall	85%
SysQual	86%
InfoQual	87%
IntQual	88%

Table 4.3: POSSUQ results in four scales

As it can be seen in table 4.3, the scores obtained in all four scales are over 80%, which indicate sufficient reliability to consider ParaShape as a usable system. The comments given in the discussions over questionnaire items will be discussed later on in 4.1.1.

Hypotheses Verification

In this section, we are going to verify the correctness of our hypotheses. As mentioned before, we created a second mockup without the principles to give the users the chance to compare the system with and without the principles followed, and decide which one they find more usable.

- **H1:** 85% of our users announced that they prefer using existing components and compose them together to make a 3D model rather than making everything on their own. They believed that, this increases the chance of personalizing a 3D model according to their test.
- **H2:** 94% of the participants would rather derive customized instances of model components than creating the component on their own since, this method saves a lot of time and energy for them especially when having limited 3D modeling experience.
- **H3:** All our users prefer having market units of measurement for customizing their models in case of the existence of the standards. Having predefined standards and familiar units not only saves time for them, but also reduces the chance of measurement errors.
- **H4:** All our expert users mentioned that they would rather work with the component code templates than the default coding environment of OpenScad, saying it is more readable and less time consuming.

Comments and Recommendations

Here, we are going to review the comments we got during the test by participants thinking aloud or documented comments in PSSUQ.

- Some users mentioned that, while working with the customizer, they would rather have the component that they are customizing being highlighted in the renderer.

- Despite mentioning that having separate tabs per components in the interface was a good choice, some participants said that having an overall tab, which contains all parameters would be useful. Although in case of having many parameters, this tab would be difficult to browse, having a complete view of all the parameters will give the chance to adjust values which are related to each other easier.
- Labeling of some buttons had to be changed. For example, "Export AMF" might not be easy to understand for a novice user and a label like " Download Printable File" would have been understood better. The same goes for "Change Shade" instead of "Change Component".
- When the *Change Component* button is clicked, we have a pop-up window with a search box to search for other components to be replaced. Our participants imparted that, if a list of most popular components of the same kind is provided beside the search box, they would be more comfortable finding the component than having to think of a proper keyword to search for it.
- As different countries use different units of measurement (inch vs. centimeter), it is vital to have units next to the text boxes displaying the values, mentioned.
- Due to lack of geometry and design knowledge, some of our novice users had problems with understanding the concept of coordinates. Therefore, we decided to replace the coordinates with slider controls as in Figure 4.2.

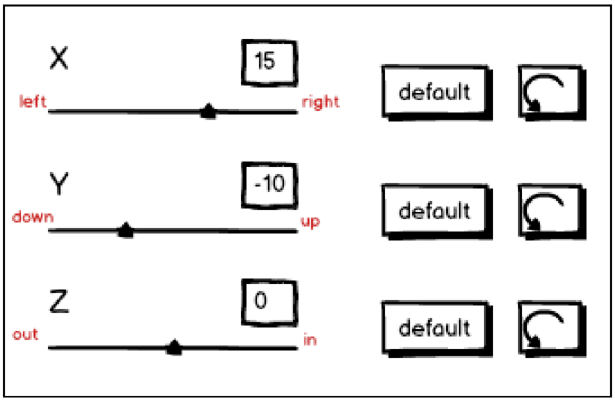


Figure 4.2: To ease customizing the coordinates we decided to use labels such as *up, down, left, right, out* and *in*

.

During our observations, we recognized that the participants, especially the novice users, describe the coordinates of a component in terms like left, right, up and down. Therefore, to ease interaction we decided to dedicate to labels to each component slider explaining the action direction by these words, which are more familiar to novice users. So that when they are looking at the 3D model from its default view, the labels map the direction that they would like to move the component to.

4.2 Meeting the Requirement

Earlier in last chapter, we defined the requirements of our system.Having collected the result from our user study, evaluation of the system and PSSUQ, here we verify whether these requirements are met by ParaShape. Table 4.4 displays the results of our verification. As it can be seen, we have met all the requirements defined to describe a good 3D modeling tool. Providing the expert users with the standard code template, the readability of the code and code reuse is supported. Moreover, having this structure results in having a customizer with an easy to learn UI for all 3D models, which makes it possible for novice users to make complex models by composing customizable components. Nevertheless, there still exist points to improve the system, which we will leave for our future work.

	Requirement	**met**
1	High range of object 3D models	$\surd$
2	High flexibility of the existing 3D models for customization	$\surd$
3	High learnability of the system	$\surd$
4	Capability of making complex 3D models without having 3D modeling skill (novice users)	$\surd$
5	Understandable interface	$\surd$
6	Understandable units of measurement	$\surd$
7	High readability of 3D model code(expert users)	$\surd$
8	Capability of code reuse (expert users)	$\surd$
9	Time efficiency of the system	$\surd$
10	Easy and fast search and browsing for the models	$\surd$

Table 4.4: Requirements met by ParaShape

Chapter 5

Summary and Further Work

5.1 Summary and Contribution

Research question: How to make everyone able to 3D model?

3D modeling has always been considered a complex task with a high learning curve. As the range of the 3D printer users grew from engineers and experts to home users, it is vital to make the design and 3D modeling process as easy as possible to make everyone benefit from personal fabrication.

ParaShape design goals

In this thesis, we presented ParaShape, a parametric approach to personal design. The main problems we tried to solve were, the high learning curve of the existing 3D modeling tools and lack of support for code reuse and readability of the 3D model code downloaded from 3D model libraries. After going over some popular 3D modeling tools and 3D printing services, we ran a survey to extract requirements of an ideal 3D modeling tool based on user experience. As we found MakerBot Customizer the closest product to our solution concepts we ran an SUS study on it to verify the usability scale and extrarct its points of power and weakness; the system suffered from many articles and users felt limited to make complex models with the system. Having defined our target user groups after interviewing expert and novice users, we came to the conclusion of creating a component-based parametric 3D modeling tool, where all models are composed of their parametric components which can be customized by changing parameter values. ParaShape uses the OpenScad model codes shared by expert users to provide novice users with a tool with an easy to use interface, where they can make their individual models by composing their desired components and customizing them in terms of dimension, shape, color, texture, and material. Using such a system, users are needless to create model components on their own and instead, they can derive instances of the already existing components, which make 3D modeling process time efficient.

creating 3D model code templates to improve readability and code reuse

While observing users during the 3D printing process, we recognized that our expert users are suffering from lack of structure in 3D model codes (OpenScad) that they download from the internet, which are provided by other users, who share their work online. To change a specific element in the code, sometimes users had to spend a considerable time to read the code and find the element; thus, many users preferred writing the code from scratch, themselves. To overcome this issue, we defined component code templates, which provides users with a new coding interface with separate code blocks dedicated to different code elements. We also equipped the interface with GUI controls to add parameters and components to the code; doing so, users could search existing component codes, edit them, and add them to their model code.

Another problem that we focused on in this thesis, was about the units of measurement. We ran a study on novice users, asking them about they preferred units of measurement in different cases. The results proved that most user would prefer the market units of measurement (units defined by international standards and offered on the market) more to metric units (meter, centimeter, inch, etc.).

Rating the most relevant tags to be used as search keywords

Another issue that we encountered, was about the tagging of 3D models. 3D models are usually searched by the most popular tags. But the most popular tags are not always the most relevant ones. Therefore, we decided to create a ranking system for the tags, so that, users rate the most relevant tags with higher ranks. Thus, they will be used as search keywords.

A qualitative user study, PSSUQ , and evaluation of meeting the requirements

The final mockup of the system was evaluated in three phases: A qualitative user study, the Post Study System Usability Questionnaire, and an evaluation on meeting the requirements. The system was proven to be usable and accepted by users. Having done the user studies and reviewing the comment of participants, we concluded that we have met all the requirements.

5.2 Future Work

5.2.1 Implementation

The most important step in future work is implementing ParaShape and verify the results

In the last chapter, we evaluated the Balsamic mockup of ParaShape and the results proved the usability of the system. However, there were minor points for improving the system that our participants mentioned. The first and most important step to take for our future work is, to implement a working system of ParaShape and see if the results are as good as the mockup test results.

Use participants' comments for improving the UI of ParaShap

We can improve the system in several aspects during implementation. As our participants pointed out in their comments after doing the mockup test, there existed minor points in the UI of ParaShape to be improved: the labeling of the buttons and controls can be more understandable and easier for users. Adding an overall tab to the customizer's interface to give the chance to the users to compare parameter values at a glance is also another point. To avoid confusion for customers, who are not familiar with the component names, we offer to have the component, which is being edited, highlighted in the renderer.

Receiving notifications when a primitive component is changed

As users are making their models from existing components, which are created by others, it would be useful to receive notifications from the system when the primitive component is edited. This can help novice users to be aware of modifications in the components and improve their models, when they needed to.

Supporting other 3D modeling tools than OpenScad

In this thesis, we only have considered users working with OpenScad who share their 3D models online as our expert group. Although this decision was made according to our interviews and observations, but having a system, which can support other 3D modeling tools, increases the number of expert users and thus enlarge the range of object 3D models covered by the system.

5.2.2 Evaluating the System with 3D Printers

After having the system implemented, we should run several tests to see whether it works properly with the 3D printers. The quality of the printed object is nevertheless, the main result of the personal fabrication process. Running long-term tests and collection of the results, can lead us to a system with a better definition of printable components. It also indicates how eager users get to 3D printing by having their problems solved with ParaShape.

5.2.3 Interaction Techniques

Through all our studies, we observed people visualizing their ideas of a 3D object. In ParaShape, we tried to reduce the complexity of 3D modeling to its easiest way. Nevertheless, the interaction technique between the user and the system can be improved to a more natural and familiar one. If the interaction technique of a 3D modeling system is close to the natural and everyday method used for visualization of an object by the user, it takes less time for the user to learn it. Below we discuss some of these techniques:

- All users, who find drawing easy, always use pen and paper to visualize their design ideas. A sketch-based modeling tool, in which users can draw the wire frame of a model with a digital or virtual pen and have it rendered in 3D after it is completed would be a good option for these users. Having the vertexes and edges of the wire frame model defined, one can manipulate the shape for edit purposes.

- A similar approach can also be used for users not familiar with drawing. Many computer users are good at creating 2D images with a software like Microsoft Paint, although they are not good in drawing. Having menu options and commands to draw lines, curves, and various shapes, enable users to do what they cannot in their real life. A tool with similar functionalities makes users capable of having a 2D wire frame of one view of the object they want to create. Thus, users are able to create 2D wire frames of the model from different views. The last step would be to map the points from different views with each other, and have a 3D wire frame to be rendered to a mesh.

- In chapter 2 we introduced 3D creators; these tools are mostly based on getting 2D images as an input and converting them to 3D models. Some of these tools such as 123D catch and Sculpteo apps use scanned photos as the 2D input of the system. Despite the fact that this method simplifies 3D modeling, there exist the lack of customization options. One easy-to-use interaction method can be having a 3D scanner, which outputs the 3D mesh in an environment, where users can manipulate it like working with virtual clay. This method gives users a feeling of touching and manipulating the 3D objects. Although this method might not be accurate enough for modeling objects with a lot of details, but it is appropriate for making crafts or DIY people, who do 3D modeling as a hobby.

5.2.4 Prevent 3D Printing Mistakes

The 3D printing process does not end with sending the 3D model to the printer and pressing the start button. The printed object must be usable. There are several mistakes that can prevent us from having our desired result. Many of these mistakes are made in the 3D modeling step. A good looking model does not definitely result in a good quality printed object. An ideal 3D modeling system should also support preventing these mistakes. Some of these issues are as following:

- There are issues that can make a 3D model not printable. One of the for example is non-manifold edges (edges at which more that two surfaces meet). An ideal modeling tool should recognize these issues and prevent them.
- The cost of the printed object is another matter, which should be concerned. A good tool is capable of creating models with the least amount of material needed.
- one important issue concerns ergonomics. For objects that fit human body such as furniture, tools, and wearables, this aspect should be considered in the modeling tool. A piece of furniture, which is not comfortable is useless, even though it looks perfect.

Appendix A

Initial Survey

In 3.1 we presented results of our initial survey. This appendix contains the complete set of survey questions.

A.1 Background

In this part we are going to ask some general questions about your background.

What is your gender?Female;Male(mandatory)

What is your current occupation? Free text(mandatory)

What is your age? Numerical input(mandatory)

do you have any design background? Yes; No(mandatory)
Help: Here by using the word "design" we particularly mean designing physical objects like everyday things such as furniture, etc.

A.2 For Designers

The questions of this section appear to the participants in case they have answered "yes" to the question asking weather they have design background.

For how many years have you been designing? Numerical Input

How regular do you design? Daily; Not daily, but several times per week; At least once a week; At least once a quarter; Less than once a quarter but occasionally

Do you use a CAD (Computer Aided Design)tool to model your

design? Yes;No (mandatory)

In case the answer the the previous question is "yes" the next four questions appear:
Which CAD tool(s) do you use? Free text(mandatory)

Which one do you prefer? Why? Script-based CAD tools(like OpenScad);Non-script-based CAD tools(like SketchUp)

Please rate the tools below:

	useful	not useful	never used
OpenScad			
AutoCAD			
SketchUp			
Blender			

If you have evaluated the usability of any other CAD tool(s) please name them. Free text

For what purpose do you design? It is my job; I sometimes need it for my job/ studies; Hobby To make things I need; Other(Multiple choice)

What tools do you use for design? Paper and pen; CAD; None; Other

Help: you can choose more than one option and if the tool you use is not mentioned write it in the text box in front of "other".

Does the result achieved by using the design tool you use satisfy you?

	not satisfied				very satisfied
result achieved					

How much time do you spend in average visualizing a simple object with your preferred tool? < 5 min; 5-15 min; 15-30 min; 30-60 min; >60 min

What would you change in the tool you use? Free text

A.3 For Non-designers

The questions of this section appear to the participants in case they have answered "yes" to the question asking weather they have design background.

Have you ever made anything for your everyday use or as a hobby?Yes; No(mandatory)
If "yes" what have you made? Free Text

Have you ever tried to design anything? Yes; No(mandatory)
Help:Here by using the word "design" we particularly mean designing physical objects like everyday things such as furniture, etc.
If "yes" what have you tried to design? Free text

How do you visualize your idea if you want to design something? Drawing(pen and paper); Computer software(paint); 3d modeling tools(sketchup); Explaining by words; Making examples from similar objects; Other(multiple choice)
Help: you can choose more than one option and if the tool you use is not mentioned write it in the text box in front of "other".

Which one do you find easier? Having a One-piece model and being able to change its dimensions; Having several parts that can be changed in dimension and attached together(like Lego)

Which one is more interesting? Having a One-piece model and being able to change its dimensions; Having several parts that can be changed in dimension and attached together(like Lego)

How much time do you usually spend on visualizing your idea? < 5 min; 5-10 min; 10-30 min; 30-60 min; >60 min

If you were supposed to change the dimensions of a (finger) ring which scale would you prefer? 6mm diametere/10mm diametere/12mm diameter; small/ medium/ large

Is there anything you miss when visualizing your ideas?(Free text)

Appendix B

System Usability Scale On MakerBot Customizer

In chapter 3 to have a clearer understanding of level of usability of MakerBot Customizer app, which was the closest concept to ParaShape we ran a user study on it. To collect some statistical data of user interaction except for the data gathered form observation and think aloud, we handed in a System Usability Scale questionnaire after the test to the participants. Table B.1 presents the questionnaire from which we estimated the usability of MakerBot Customizer.

	Statement	Strongly agree				Strongly disagree
1	I think that I would like to use this app frequently.					
2	I found the app unnecessarily complex.					
3	I thought the product was easy to use.					
4	I think that I would need the support of a technical person to be able to use this app.					
5	I found the various functions in the app were well integrated.					
6	I thought there was too much inconsistency in this app.					
7	I imagine that most people would learn to use this app very quickly.					
8	I found the app very awkward to use.					
9	I felt very confident using the app.					
10	I needed to learn a lot of things before I could get going with this app.					

Table B.1: System Usability Scale questionnaire

Appendix C

Paper Prototype

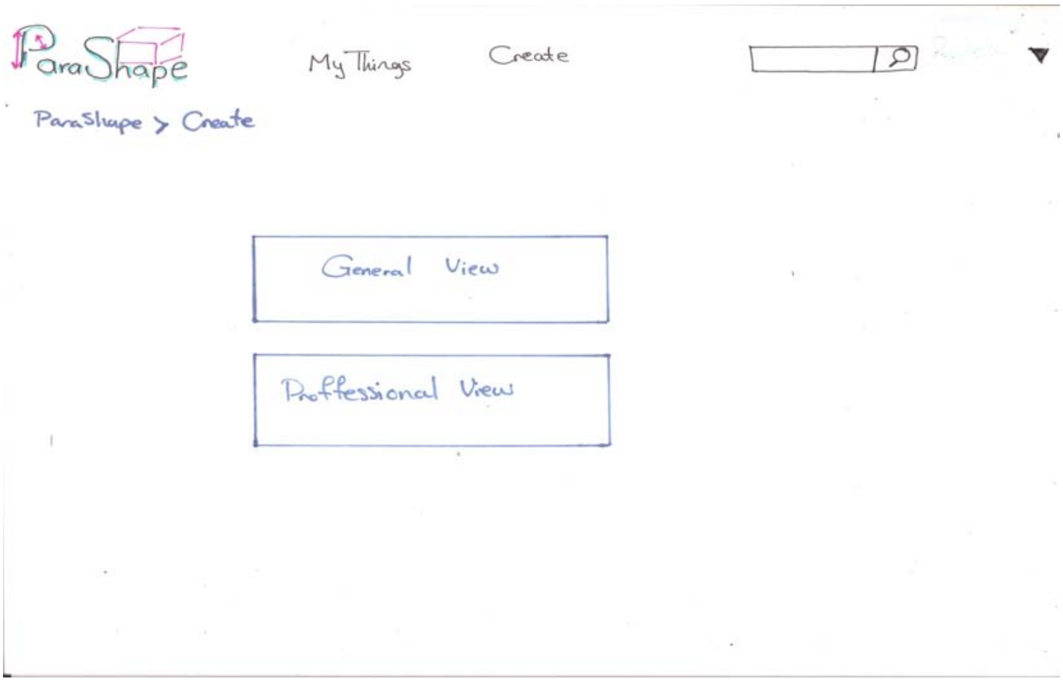


Figure C.1: ParaShape create page

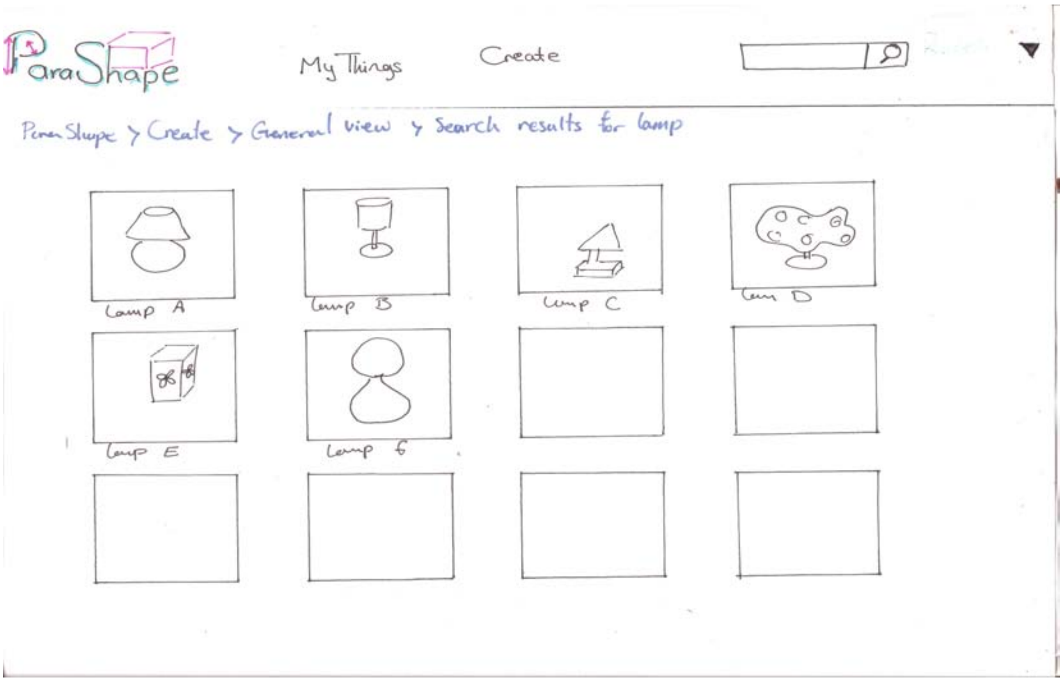


Figure C.2: ParaShape search results for "lamp"

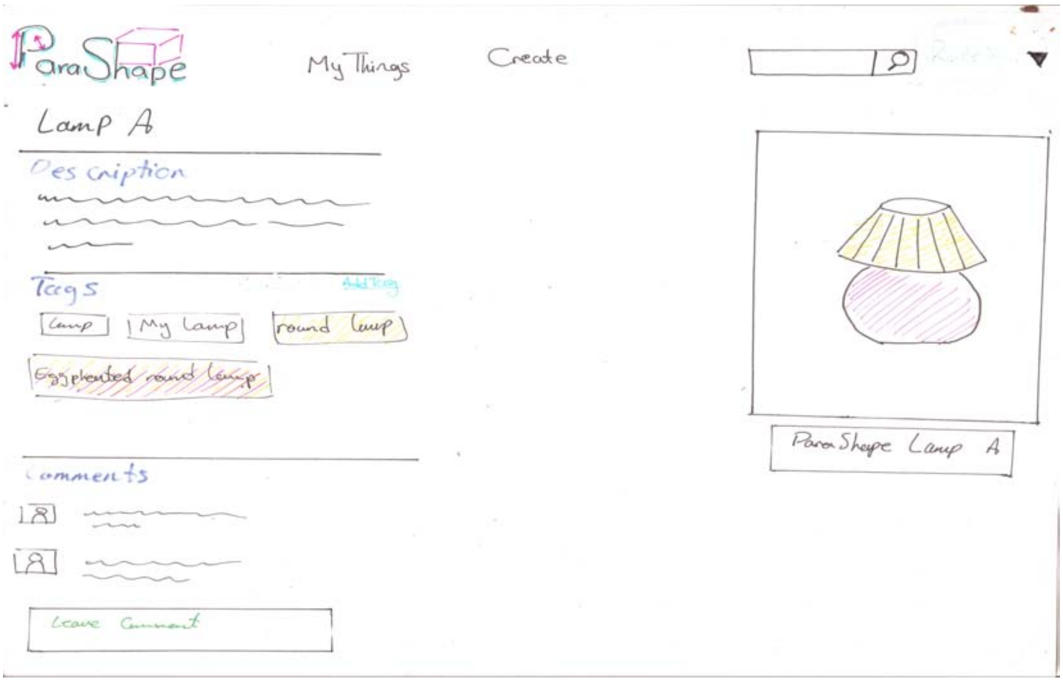


Figure C.3: Description page of a lamp

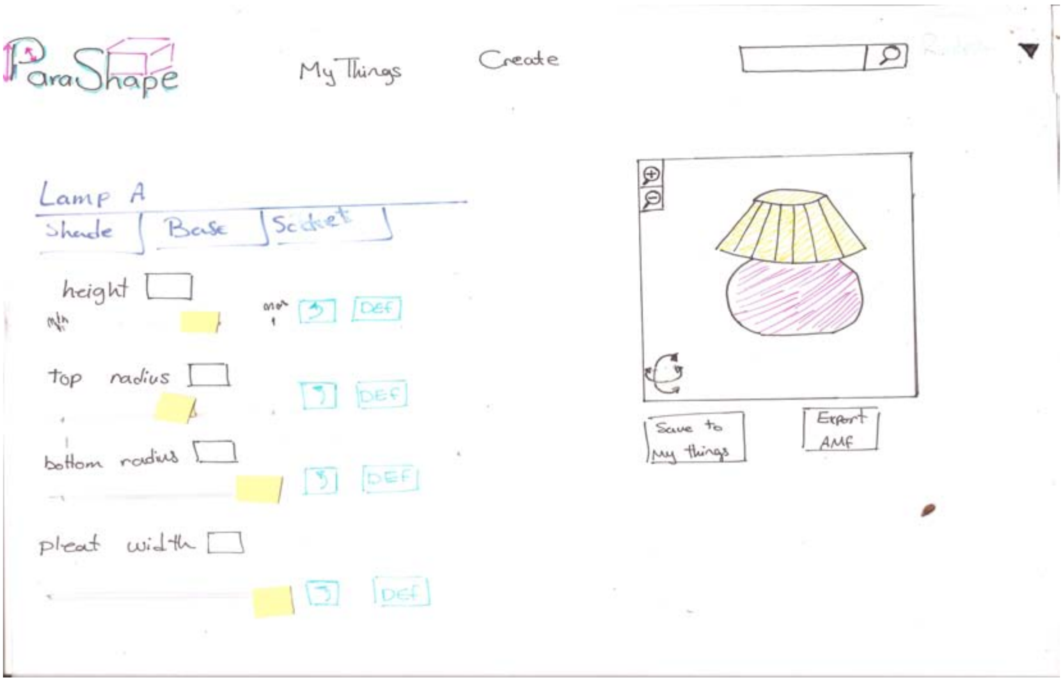


Figure C.4: ParaShape novice users' view

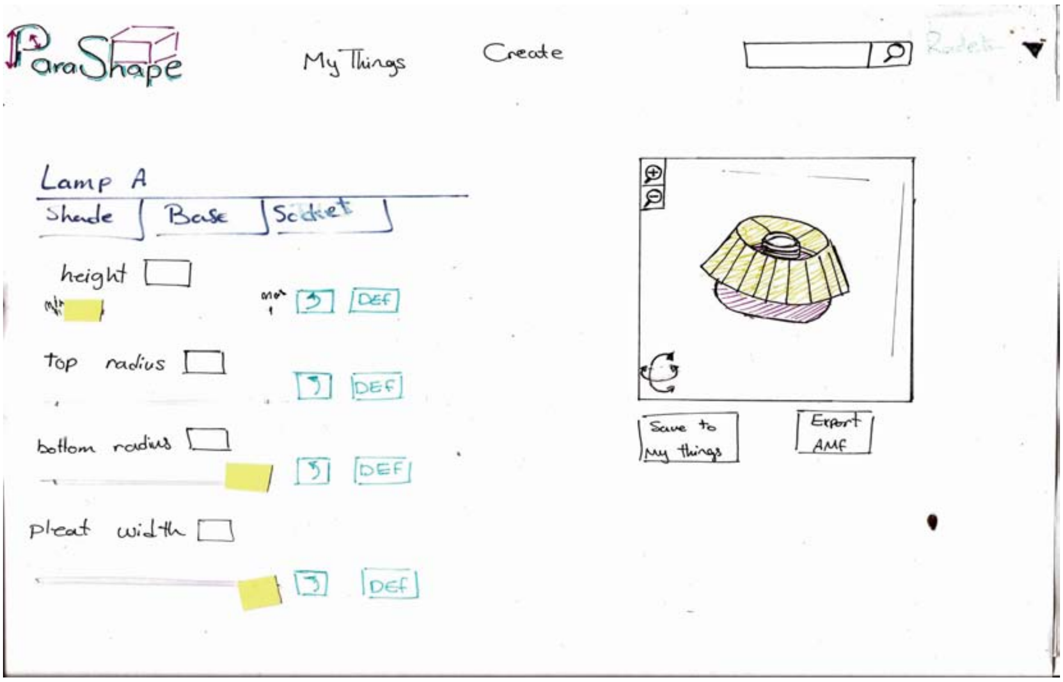


Figure C.5: ParaShape novice users' view-rotate model up

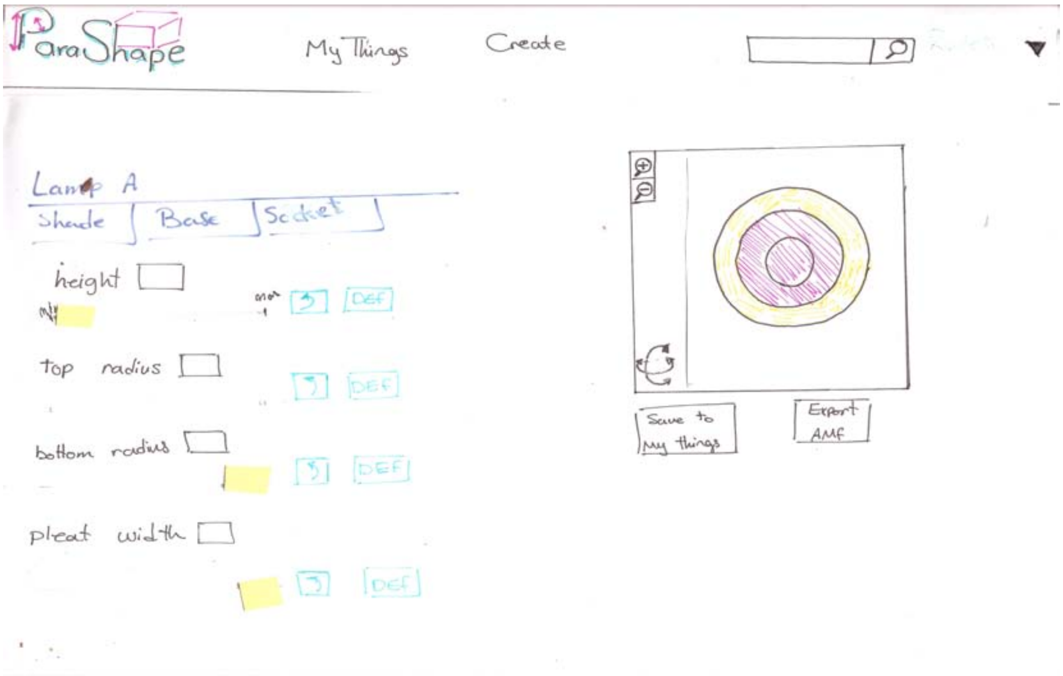


Figure C.6: ParaShape novice users' view- bottom view of model

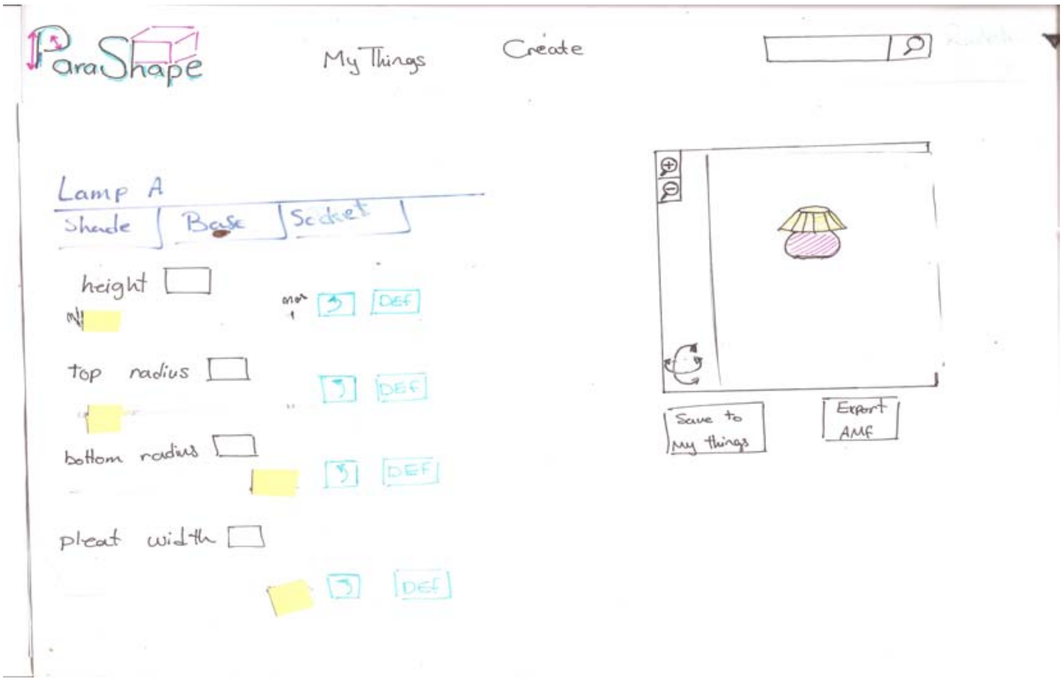


Figure C.7: ParaShape novice users' view-zoom model out

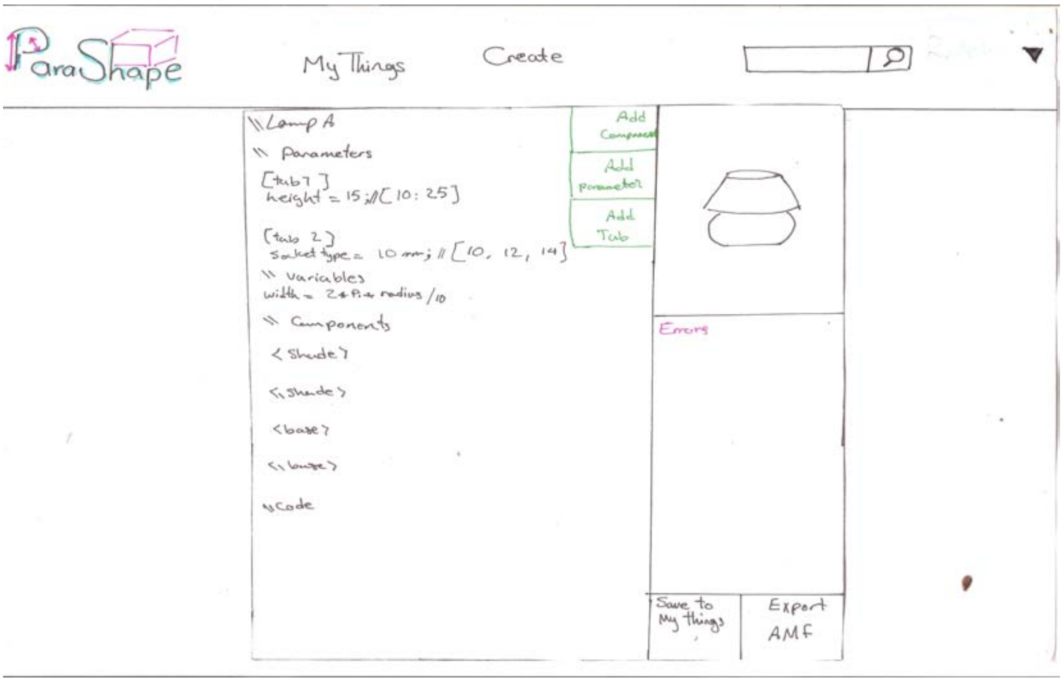


Figure C.8: ParaShape expert users' view

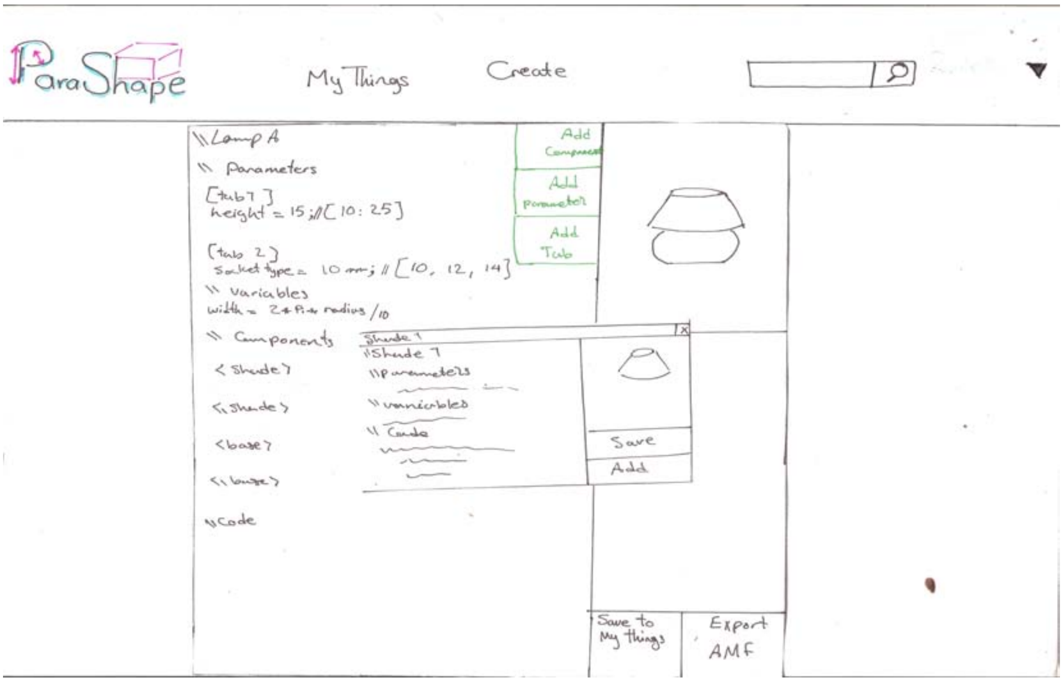


Figure C.9: ParaShape expert users' view- add component

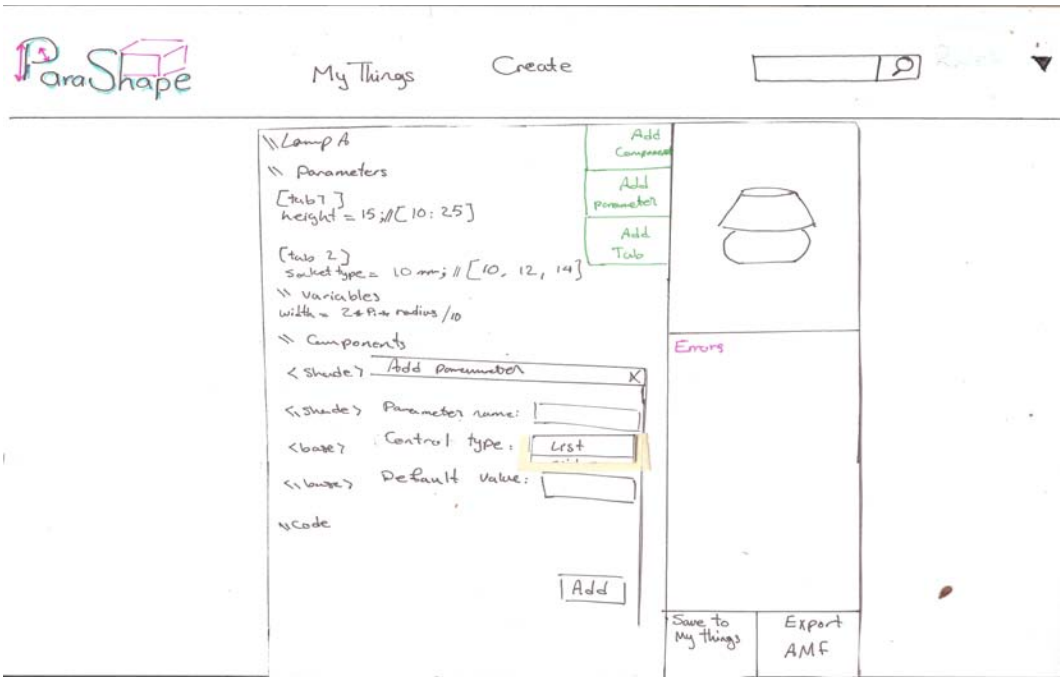


Figure C.10: ParaShape expert users' view-add param

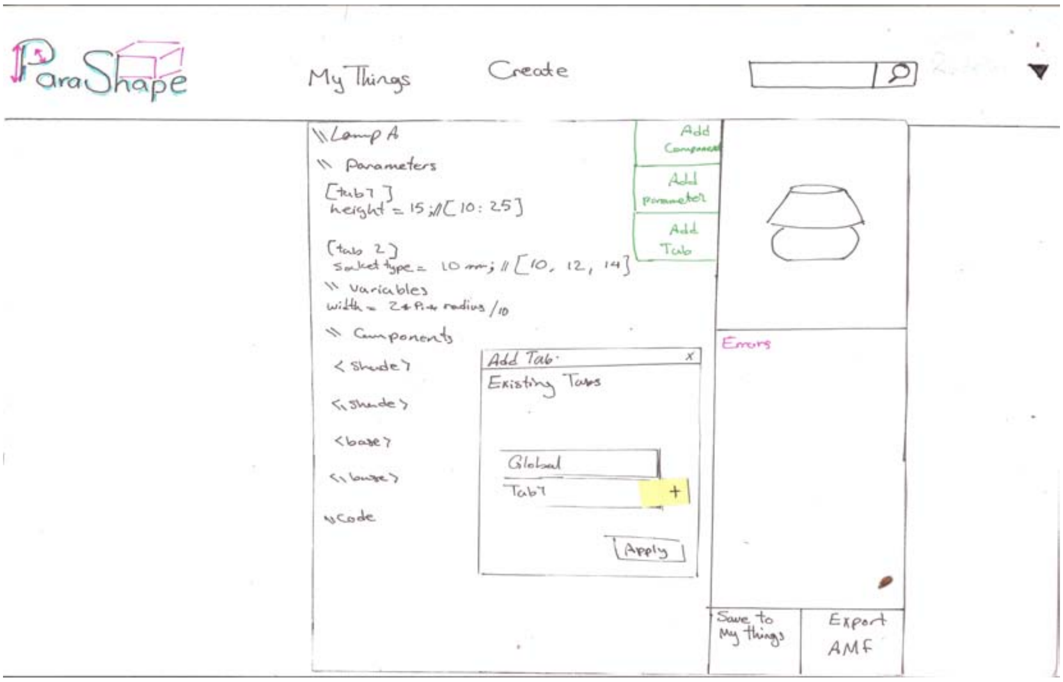
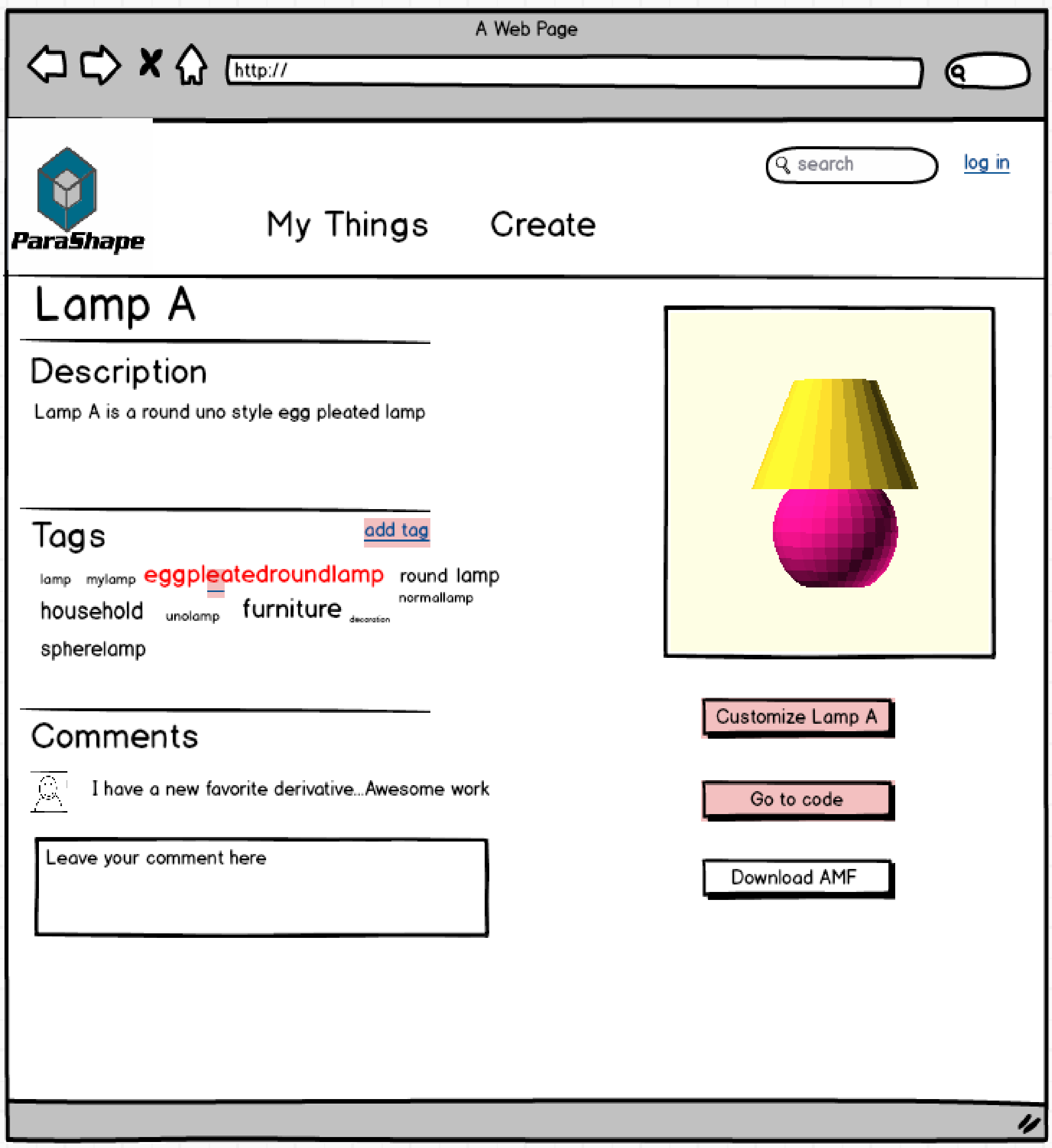


Figure C.11: ParaShape expert users' view- add tab

Appendix D

Balsamic Mockup

D.1 Mockup

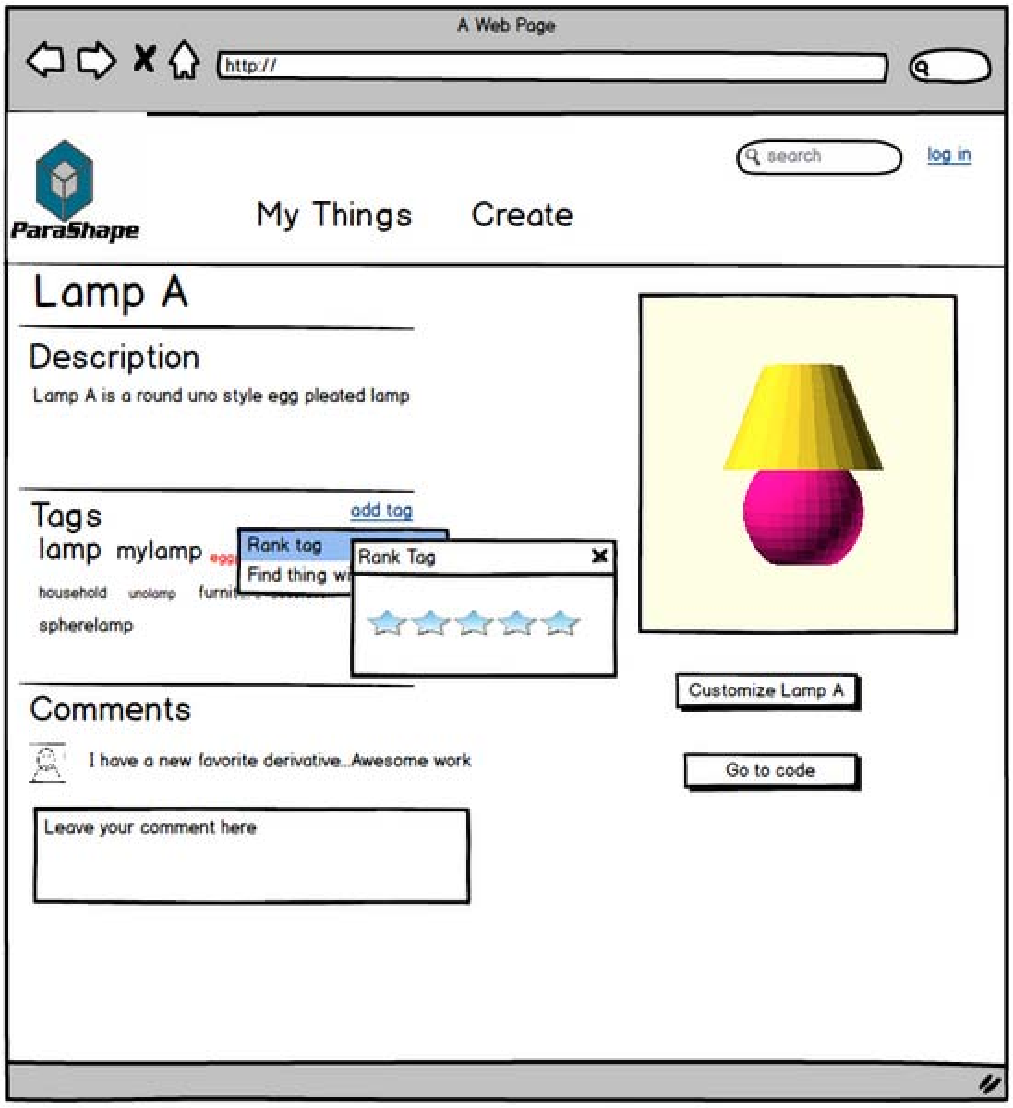


Lamp A is a round uno style egg pleated lamp

I have a new favorite derivative...Awesome work

Leave your comment here

Figure D.1: ParaShape description page of lamp A



Lamp A

Description

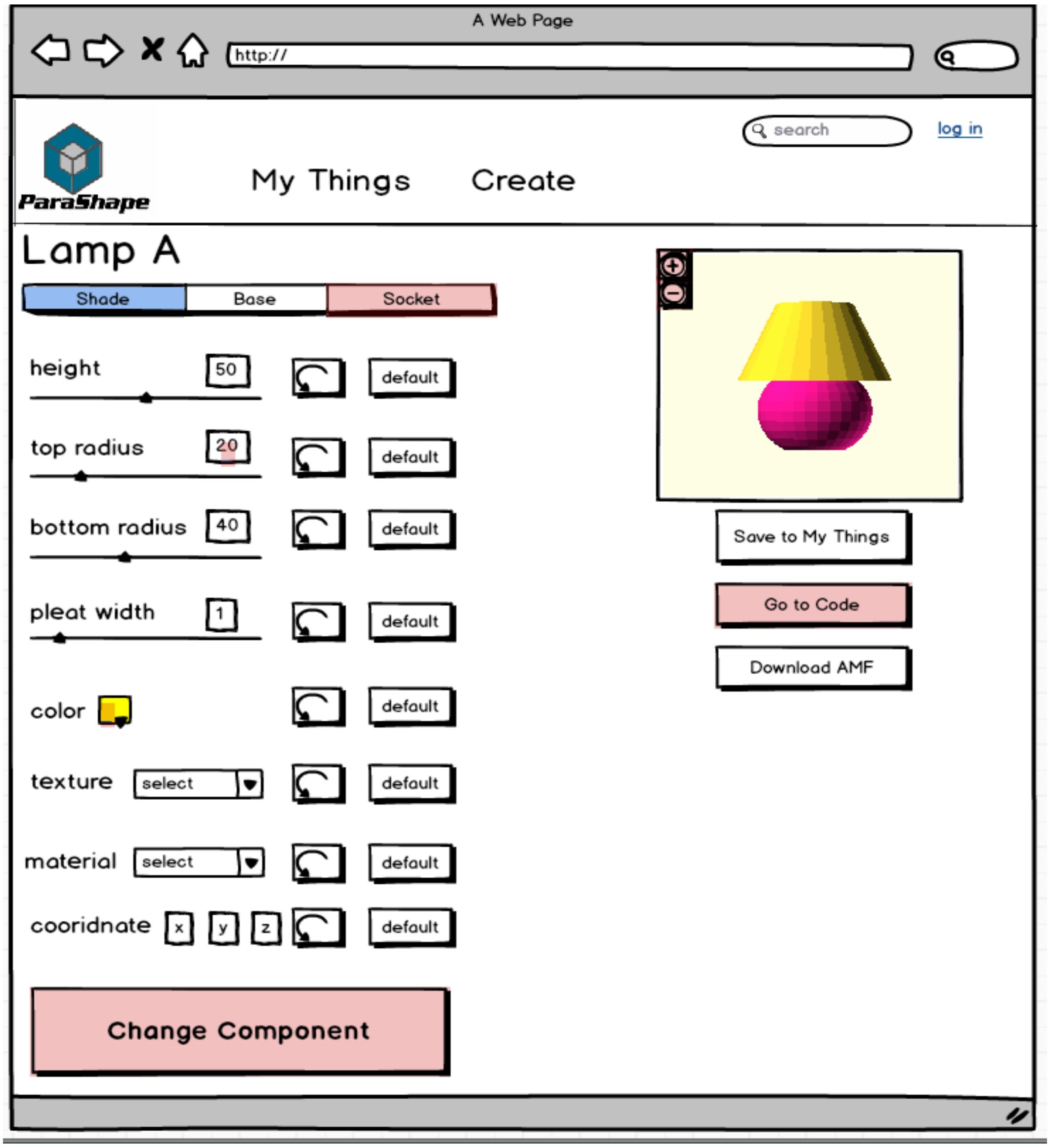
Lamp A is a round uno style egg pleated lamp

Comments

I have a new favorite derivative...Awesome work

Leave your comment here

Figure D.2: ParaShape rank tag



Lamp A

height 50 default

top radius 20 default

bottom radius 40 default

pleat width 1 default

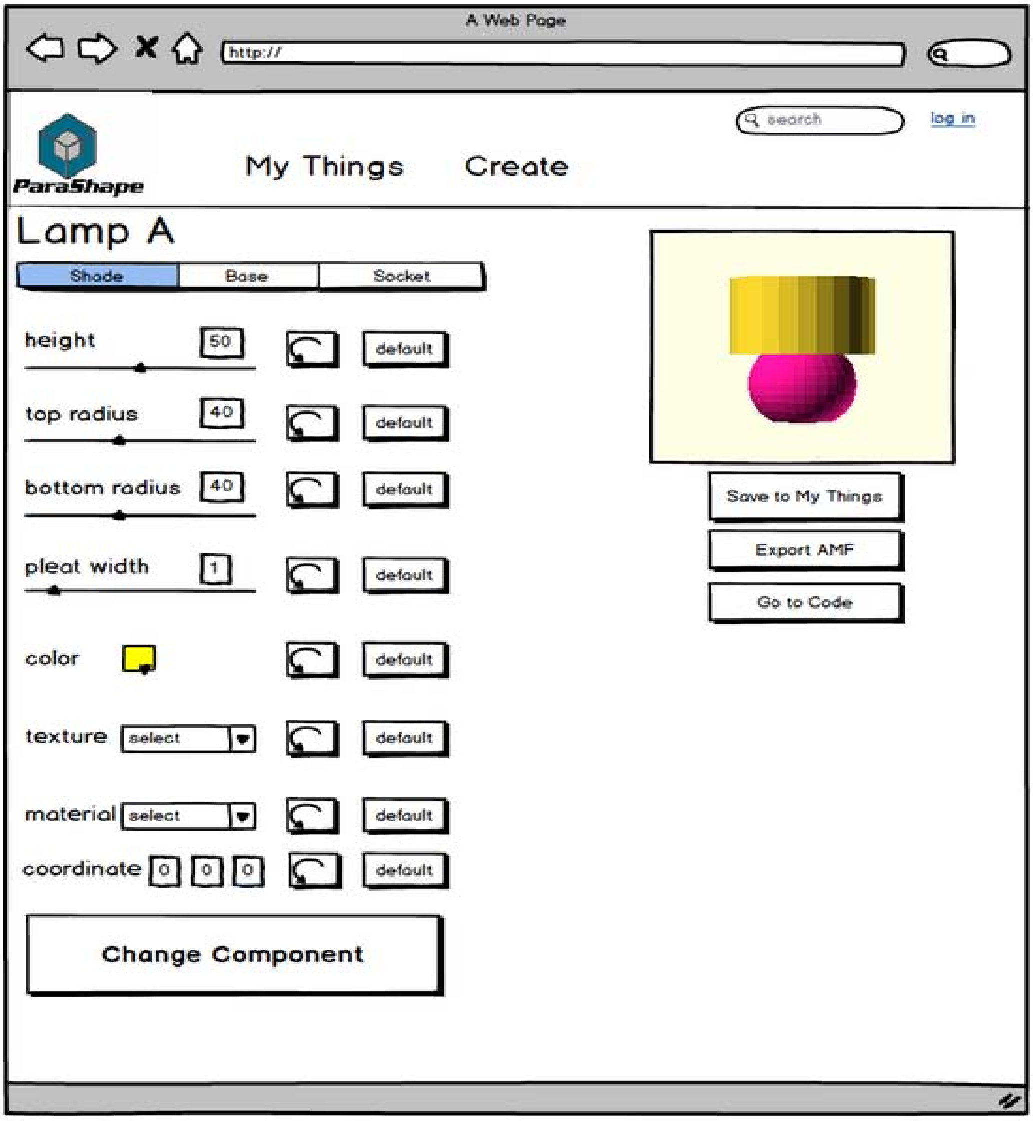
color default

texture select default

material select default

cooridnate x y z default

Figure D.3: ParaShape customizer



height 50

top radius 40

bottom radius 40

pleat width 1

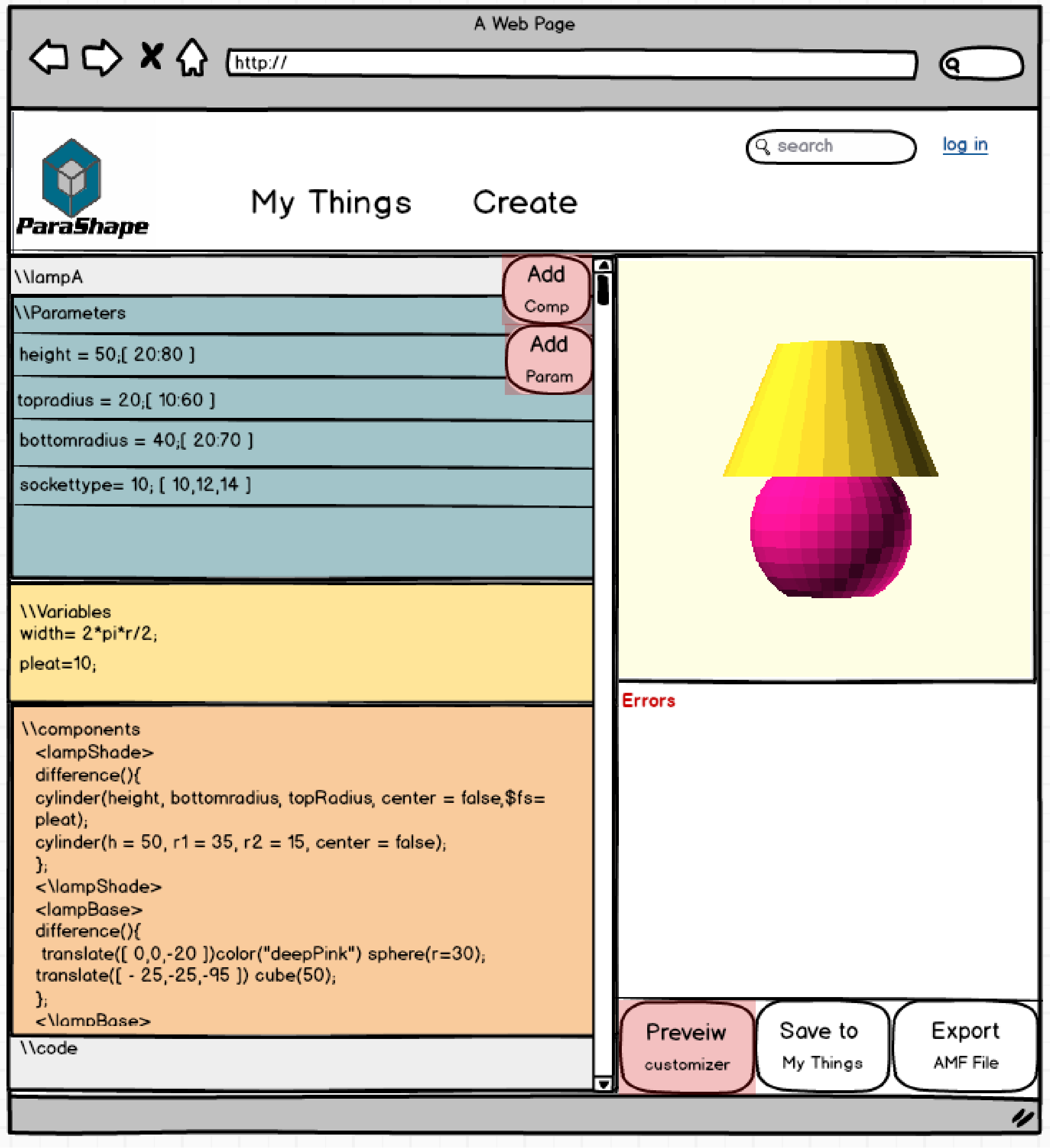
color

texture select

material select

coordinate 0 0 0

Figure D.4: ParaShape customizer- parameter value changed

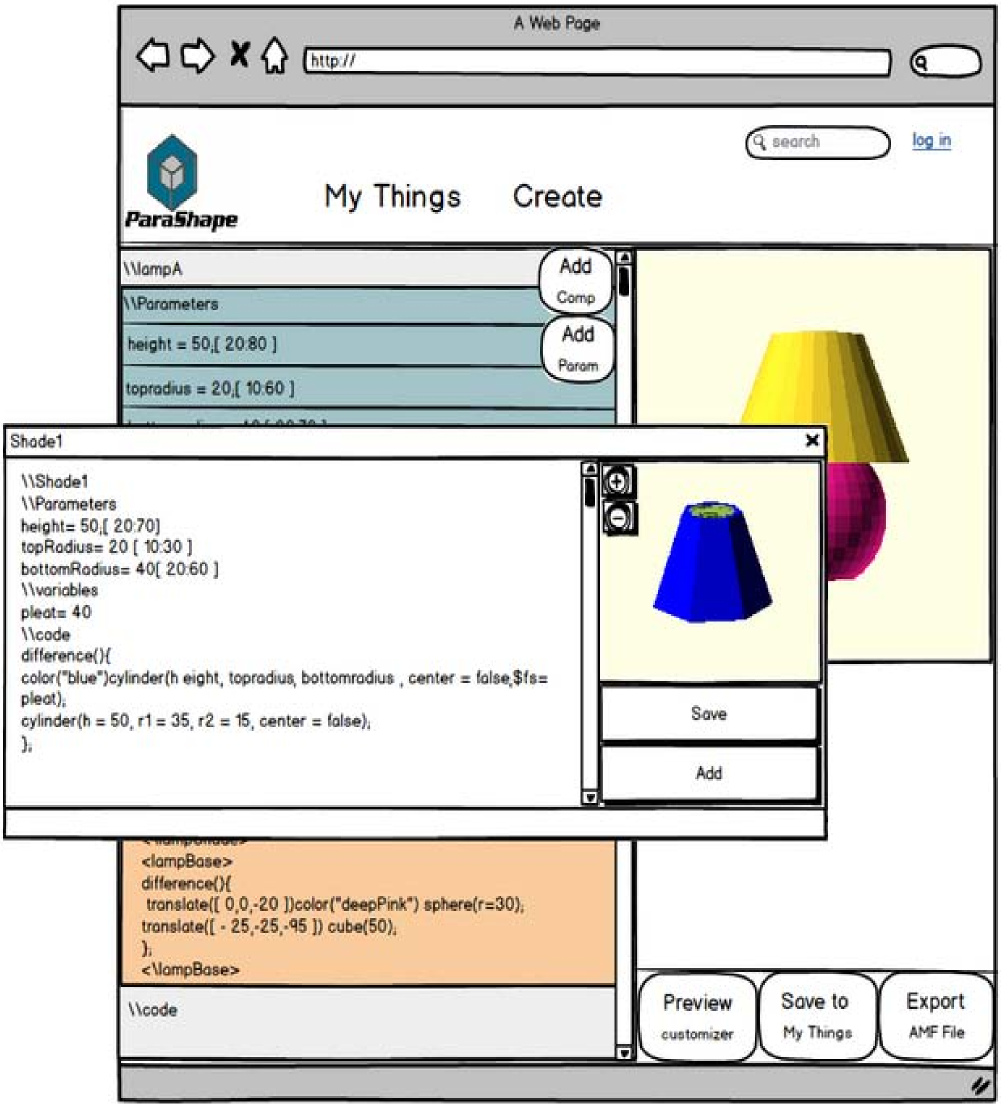


```
\\lampA
\\Parameters
height = 50;[ 20:80 ]
topradius = 20;[ 10:60 ]
bottomradius = 40;[ 20:70 ]
sockettype= 10; [ 10,12,14 ]
```

```
\\Variables
width= 2*pi*r/2;
pleat=10;
```

```
\\components
  <lampShade>
  difference(){
  cylinder(height, bottomradius, topRadius, center = false,$fs=
  pleat);
  cylinder(h = 50, r1 = 35, r2 = 15, center = false);
  };
  <\lampShade>
  <lampBase>
  difference(){
   translate([ 0,0,-20 ])color("deepPink") sphere(r=30);
  translate([ - 25,-25,-95 ]) cube(50);
  };
  <\lampBase>
\\code
```

Figure D.5: ParaShape coding environment



```
\\lampA
\\Parameters
height = 50,[ 20:80 ]
topradius = 20,[ 10:60 ]
```

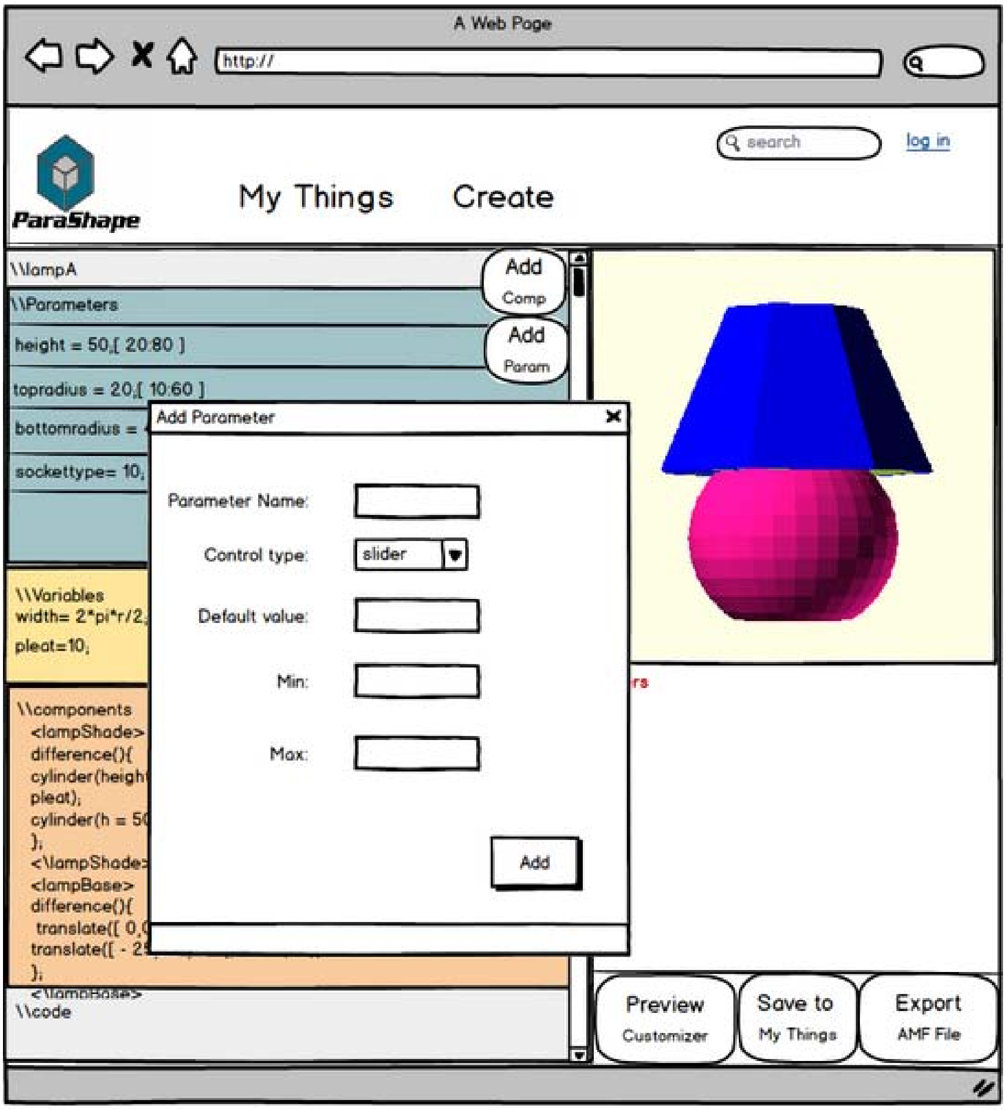
Shade1

```
\\Shade1
\\Parameters
height= 50,[ 20:70]
topRadius= 20 [ 10:30 ]
bottomRadius= 40[ 20:60 ]
\\variables
pleat= 40
\\code
difference(){
color("blue")cylinder(h eight, topradius, bottomradius , center = false,$fs=
pleat);
cylinder(h = 50, r1 = 35, r2 = 15, center = false);
};
```

```
<lampBase>
difference(){
 translate([ 0,0,-20 ])color("deepPink") sphere(r=30);
translate([ - 25,-25,-95 ]) cube(50);
};
<\lampBase>
```

```
\\code
```

Figure D.6: ParaShape coding environment-add component



```
\\lampA
\\Parameters
height = 50,[ 20:80 ]
topradius = 20,[ 10:60 ]
bottomradius = 
sockettype= 10,

\\Variables
width= 2*pi*r/2,
pleat=10;

\\components
  <lampShade>
  difference(){
  cylinder(height
  pleat);
  cylinder(h = 50
  };
  <\lampShade>
  <lampBase>
  difference(){
   translate([ 0,0
  translate([ - 25
  };
  <\lampBase>
\\code
```

Figure D.7: ParaShape coding environment-add parameter

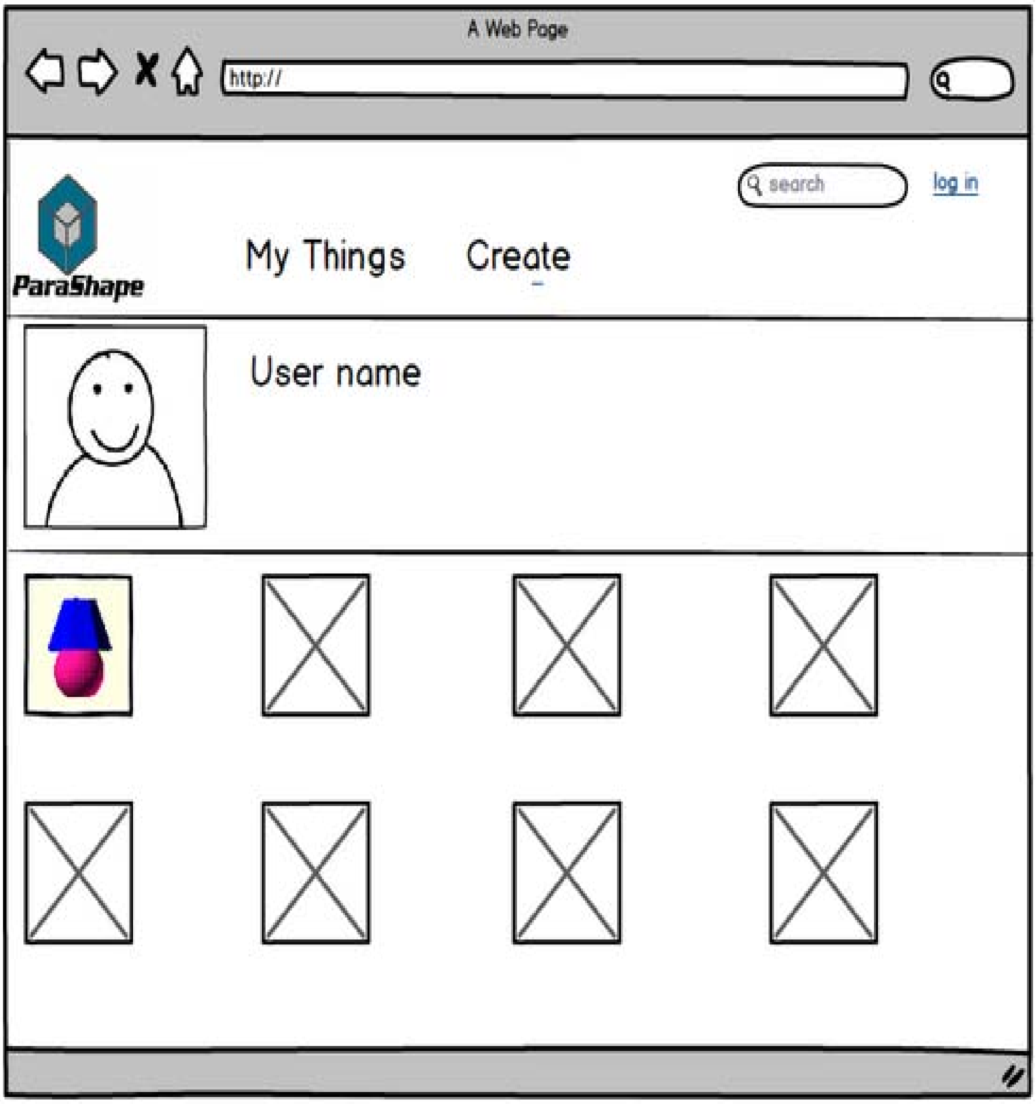


Figure D.8: ParaShape my things page

D.2 Post Study System Usability Questionnaire (Version 3)

Age:
Gender:
Occupation:
The questionnaire gives you an opportunity to tell us your reactions to the system you used. Your response will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you. To as great an extent as possible, think about all the tasks that you have done with the system while you answer these questions. Please read each statement and indicate how strongly you agree or disagree with the statement. If a statement does not apply to you, choose NA. Please write comments to elaborate your answers.

After you have completed this questionnaire, I will go over your answers with you to make sure I understand all of your responses. Thank you!

	Statement	Strongly agree						Strongly disagree
1	Overall, I am satisfied with how easy it is to use this system.	□	□	□	□	□	□	□
2	It was simple to use this system.	□	□	□	□	□	□	□
3	I was able to complete the tasks and scenarios quickly using this system.	□	□	□	□	□	□	□
4	I felt comfortable using this system.	□	□	□	□	□	□	□
5	It was easy to learn to use this system.	□	□	□	□	□	□	□
6	I believe I could become productive quickly using this system.	□	□	□	□	□	□	□
7	The system gave error messages that clearly told me how to fix problems.	□	□	□	□	□	□	□
8	Whenever I made a mistake using the system, I could recover easily and quickly.	□	□	□	□	□	□	□
9	The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear.	□	□	□	□	□	□	□
10	It was easy to find the information I needed.	□	□	□	□	□	□	□
11	The information was effective in helping me complete the tasks and scenarios.	□	□	□	□	□	□	□
12	The organization of information on the system screens was clear.	□	□	□	□	□	□	□
13	The interface of this system was pleasant.	□	□	□	□	□	□	□
14	I liked using the interface of this system.	□	□	□	□	□	□	□
15	This system has all the functions and capabilities I expect it to have.	□	□	□	□	□	□	□
16	Overall, I am satisfied with this system.	□	□	□	□	□	□	□

Table D.1: PSSUQ questionnaire version 3

[**?**] []

Bibliography

[Ayr90] R. Ayres. *Computer Integrated Manufacturing: Volume 1 : Revolution in Progress*. IIASA computer integrated manufacturing series v. 2. Chapman & Hall, 1990. http://books.google.de/books?id=NgKcFcJYxw8C

[Bla12] J. Blain. *The Complete Guide to Blender Graphics: Computer Modeling and Animation*. Taylor & Francis, 2012. http://books.google.de/books?id=abLlz82BZvYC

[Bro96] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189:194, 1996.

[Chu11] J. L. Chulilla. The Cambrian Explosion of Popular 3D Printing. *IJIMAI* 1(4):30–32, 2011. http://dblp.uni-trier.de/db/journals/ijimai/ijimai1.html#Chulilla11

[CKGK11] S. Chaudhuri, E. Kalogerakis, L. Guibas, V. Koltun. Probabilistic Reasoning for Assembly-Based 3D Modeling. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30(4), 2011.

[CLL10] C. Chua, K. Leong, C. Lim. *Rapid Prototyping: Principles and Applications*. World Scientific, 2010. http://books.google.de/books?id=4OYcyiDUpsQC

[CS01] P. Constantopoulos, I. Solvberg. *Research and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001, Darmstadt, Germany, September 4-9, 2001. Proceedings*. Lecture Notes in Computer Science Bd. 5. Springer, 2001. http://books.google.de/books?id=OrJsIULAH7MC

[FKS$^+$04] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, D. Dobkin. Modeling by example. *ACM Trans. Graph.* 23(3):652–663, August 2004. doi:10.1145/1015706.1015775 http://doi.acm.org/10.1145/1015706.1015775

[Fla10] L. Flavell. *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design*. Apress, Berkely, CA, USA, 1st edition, 2010.

[Fol96] J. Foley. *Computer Graphics: Principles and Practice, Second Edition in C*. Addison-Wesley systems programming series. Addison Wesley Professional, 1996. http://http://books.google.de/books?id=-4ngT05gmAQC

[Fra12] M. Frauenfelder. *Make: Technology on Your Time*. Make : Technology on Your Time. O'Reilly Vlg. GmbH & Company, 2012. http://books.google.de/books?id=LHMwuQR1D_sC

[GA09] K. Grote, E. Antonsson. *Springer Handbook of Mechanical Engineering*. Springer Handbook of Mechanical Engineering Bd. 10. Springer, 2009. http://books.google.de/books?id=9T5kd-ewRE8C

[Geb03] A. Gebhardt. *Rapid prototyping*. Hanser Gardner Publications, 2003. http://books.google.de/books?id=9V_HoiF4gHsC

[Ger05] N. Gershenfeld. *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. Basic Books, 2005. http://books.google.de/books?id=Oil3bH6fKBkC

[GH05] S. Golder, B. A. Huberman. The structure of collaborative tagging systems. *arXiv preprint cs/0508082*, 2005.

[GS10] J. Groza, J. Shackelford. *Materials Processing Handbook*. Taylor & Francis, 2010. http://books.google.de/books?id=BNjw-a7_uKAC

[Ham10] M. Hamad. *AutoCAD? 2010 3D Modeling Essentials*. Jones & Bartlett Learning, 2010. http://books.google.de/books?id=naL7oslVt-oC

[HL09] J. Hiller, H. Lipson. STL 2.0: A Proposal for a Universal Multi-Material Additive Manufacturing File Format. In *Solid Freeform Fabrication Symposium (SFF'09)*. 2009.

[Jur10] J. Jursa. *UX Storytellers - Connecting the Dots*. Jan Jursa, 2010. http://books.google.fr/books?id=wWrqzNdZ5o8C

[KGS00] S. Krar, A. Gill, P. Smid. *Cnc Simplified, Lab Manual*. Industrial Press, 2000. http://books.google.de/books?id=9fK9CjkpIHcC

[KJS07] V. Kreavoy, D. Julius, A. Sheffer. Model Composition from Interchangeable Components. In *Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on*. Pp. 129–138. 2007. doi:10.1109/PG.2007.40

[McC99] T. McCarthy. *Autocad Express Nt: Covering Release 14*. Springer, 1999. http://books.google.de/books?id=MeALWlOkNtgC

[Mil09] A. B. P. K. J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies* 4(3):114–123, 2009.

[MM06] G. Macgregor, E. McCulloch. Collaborative tagging as a knowledge organisation and resource discovery tool. *Library review* 55(5):291–300, 2006.

[Mot11] C. Mota. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition*. C&#38;C '11, pp. 279–288. ACM, New York, NY, USA, 2011. doi:10.1145/2069618.2069665 http://doi.acm.org/10.1145/2069618.2069665

[Nor02] D. A. Norman. *The design of everyday things*. Basic Books, [New York], 1. basic paperback ed., [nachdr.] edition, 2002.

[PFS13] B. Pettis, A. France, J. Shergill. *Getting Started with MakerBot*. Oreilly and Associate Series. O'Reilly Vlg. GmbH & Company, 2013. http://books.google.de/books?id=-iaTMXZJQ9cC

[Ros78] D. T. Ross. Origins of the APT language for automatically programmed tools. *SIGPLAN Not.* 13(8):61–99, August 1978. doi:10.1145/960118.808374 http://doi.acm.org/10.1145/960118.808374

[SB98] Shneiderman, Ben. *Designing the User Interface – Strategies for Effective Human-Computer-Interaction*. Addison-Wesley Longman Inc., 3 edition, 1998.

[SGMB08] A. Shepitsen, J. Gemmell, B. Mobasher, R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*. RecSys '08, pp. 259–266. ACM, New York, NY, USA, 2008. doi:10.1145/1454008.1454048 http://doi.acm.org/10.1145/1454008.1454048

[Sin10] S. Singh. *Beginning Google Sketchup for 3D Printing*. Apresspod Series. Apress, 2010. http://books.google.de/books?id=yIZsd5vVZtYC

[SL12] J. Sauro, J. Lewis. *Quantifying the User Experience: Practical Statistics for User Research*. Morgan Kaufmann. Elsevier/-Morgan Kaufmann, 2012. http://http://books.google.de/books?id=QncH02-bvjoC

[SLMI11] G. Saul, M. Lau, J. Mitani, T. Igarashi. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. TEI '11, pp. 73–80. ACM, New York, NY, USA, 2011. doi:10.1145/1935701.1935717 http://http://doi.acm.org/10.1145/1935701.1935717

[SQF11] Y. Song, B. Qiu, U. Farooq. Hierarchical tag visualization and application for tag recommendations. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. CIKM '11, pp. 1331–1340. ACM, New York, NY, USA, 2011. doi:10.1145/2063576.2063769 http://doi.acm.org/10.1145/2063576.2063769

[SSS05] R. Shimonski, R. Steiner, S. Sheedy. *Network Cabling Illuminated*. Jones and Bartlett Illuminated Series. Jones and Bartlett, 2005. http://books.google.de/books?id=rVnMZTCEfl8C

[Tad09] M. Tadros. *Real World Google SketchUp 7*. Real World. Peachpit Press, 2009. http://books.google.de/books?id=gOUevtJtt5wC

[Tur10] F. Turner. *From Counterculture to Cyberculture: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. University of Chicago Press, 2010. http://books.google.de/books?id=2SNFpgX_WigC

[Ved11] B. Vedin. *The Design-Inspired Innovation Workbook*. World Scientific, 2011. http://books.google.de/books?id=j-_lDwEHsboC

[VGAS12] A. Valero-Gomez, J. Gonzalez-Gomez, M. Almagro, M. Salichs. Boosting mechanical design with the C++ OOML and open source 3D printers. In *Global Engineering Education Conference (EDUCON), 2012 IEEE*. Pp. 1 –7. april 2012. doi:10.1109/EDUCON.2012.6201114

[WM94] W. Wilson, D. M. Martinez. Milling machine. May 3 1994. US Patent 5,308,201.

[YLYW06] T. Yu, T. Liu, S. Yang, W. Wang. Research on system theory of integrated numerical control. In *System Theory, 2006. SSST '06. Proceeding of the Thirty-Eighth Southeastern Symposium on*. Pp. 6 –10. march 2006. doi:10.1109/SSST.2006.1619045

Typeset September 16, 2013