

Institut für Visualisierung und Interaktive Systeme
Abteilung Mensch-Computer-Interaktion
Universität Stuttgart
Pfaffenwaldring 5a
D-70569 Stuttgart

Diplomarbeit Nr. 3437

Einfache Herstellung durch natürliche Interaktion mit Lasercuttern im Heimbereich

David Schmid

Studiengang: Softwaretechnik
Prüfer: Prof. Dr. Albrecht Schmidt
Betreuer: M. Sc. Stefan Schneegaß

begonnen am: 14. Januar 2013
beendet am: 28. Juni 2013

CR-Klassifikation: C.5.2, J.6

Kurzfassung

Die Einstiegshürde für die Prototypentwicklung mit elektronischen Komponenten ist mit der Einführung von Microsofts .NET Gadgeteer deutlich gesunken. Möchte man nun ein Gehäuse für einen Prototyp herstellen, so benötigt man jedoch entweder CAD-Kenntnisse oder handwerkliches Geschick. In dieser Arbeit wird ein Konzept vorgestellt, mit dem Gehäuse in wenigen Minuten entworfen werden können. Ein Lasercutter schneidet anschließend die Teile aus, die sich zu dem Gehäuse zusammensetzen lassen. Besonderes Augenmerk wird dabei auf die einfache Bedienung gelegt, so dass die Anwendung auch von Personen ohne technisches Hintergrundwissen genutzt werden kann. Für die Erstellung des Gehäusedesigns kommt Microsofts Touch-Tisch PixelSense zum Einsatz, der neben der Bedienung mit der Hand auch das Erkennen von Objekten ermöglicht. Damit können für externe Komponenten wie Displays und Buttons Aussparungen automatisch eingefügt werden. Platziert man eine externe Komponente auf dem Touch-Tisch, so wird diese erkannt und eine entsprechende Aussparung wird im Gehäuse eingefügt. Durch Verschieben und Drehen dieser externen, physikalischen Komponenten kann die virtuelle Position der Aussparung bestimmt werden. Die Implementierung zeigt eine erste Umsetzung des Konzepts. Durch eine explorative Benutzerstudie wurden die grundlegenden Ideen evaluiert und Ansätze für Verbesserungen untersucht. Alle Teilnehmer konnten in wenigen Minuten ein Gehäuse entwerfen und bestätigen das grundlegende Konzept.

Abstract

The entry barrier for prototype development with electronic components has decreased significantly with the introduction of Microsoft's .NET Gadgeteer. However, if one wants to make a casing for a prototype, it either requires CAD knowledge or manual skills. This paper presents an approach to design casings within minutes. After the design has been finished a laser cutter cuts out the parts that can be assembled into the casing. Particular attention is paid to the ease of use so that the application can also be used by people without technical background knowledge. For the creation of casing designs Microsoft's touch table PixelSense is used which, besides hand input control, detects objects as well. Thus, cut-outs for external components such as displays and buttons can be inserted automatically. Placing an external component on the touch table, it gets detected and according cut-outs are added to the casing. The virtual position of the cut-outs can be defined by moving and rotating these external, physical components. The implementation shows a first realization of the presented concept. Within an exploratory user study the basic ideas were evaluated and approaches for improvements are investigated. All participants were able to design a casing within a few minutes and hence confirm the basic concept.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Gliederung der Arbeit	10
2	Grundlagen	11
2.1	Die Rolle von CAD-Programmen	11
2.2	Personal Fabrication	13
2.2.1	Beispiele von Werkzeugmaschinen in der Personal Fabrication	19
2.3	Besonderheiten beim praktischen Arbeiten mit Lasercuttern	22
2.4	Microsoft .NET Gadgeteer	29
3	Verwandte Arbeiten	31
3.1	Wahl der Interaktionsart	31
3.1.1	Stift- und Touch-Eingabe	34
3.2	Besondere Einsatz- und Eingabemöglichkeiten für Lasercutter	36
3.3	Tangible User Interface	39
3.4	Material, Form und Ästhetik	40
4	Konzept	45
4.1	Szenario	48
5	Entwurf	49
5.1	Projekt-Aufbau	50
5.2	Natcut Application	51
5.3	ObjectDetection	55
5.4	SVG-Generator	55
5.4.1	CasingController	56
5.4.2	JointGenerator	57
5.4.3	SVG	57
5.5	Anwendungsfälle	57
5.5.1	Neue Komponente hinzufügen	57
5.5.2	Manipulation einer Komponente	58
5.5.3	Eigenschaften einer Grundfläche	59
6	Implementierung	61
6.1	Vorversuche und Entscheidungen	61
6.1.1	Verbindungen	61
6.1.2	Richtlinien der Implementierung	63

6.2	Natcut Application	63
6.3	ObjectDetection	66
6.4	SVG-Generator	68
6.4.1	CasingModel	68
6.4.2	CasingController	69
6.4.3	Fingerverbindung	70
6.4.4	Akkordeonverbindung	71
7	Explorative Benutzerstudie	73
7.1	Ziele	73
7.2	Versuchsaufbau	74
7.3	Teilnehmerfeld	75
7.4	Ablauf	75
7.5	Ergebnisse	77
8	Zusammenfassung und Ausblick	81
8.1	Zusammenfassung	81
8.2	Ausblick	82
A	Anhang	85
A.1	Objektmodell im Detail	85
A.2	XSD2Code	87
A.3	Komponentenübersicht	89
A.4	Fragebogen der Benutzerstudie	89
	Literaturverzeichnis	93

Abbildungsverzeichnis

2.1	Umsetzung vager Anforderungen	13
2.2	Funktionaler Test durch 3D Modelle	14
2.3	Design Test durch 3D Modelle	15
2.4	Günstige 3D-Drucker	16
2.5	Digital Fabrication als Lernmedium	18
2.6	Einfache Eingabe für 3D-Drucker	20
2.7	Mit 3D-Drucker hergestellte Mäuse	21
2.8	Rahmenverbindungen	24
2.9	Brettverbindungen	25
2.10	Eckverbindungen	26
2.11	T-Verbindungen	27
2.12	Formschlüssige Verbindungen	28
2.13	Akkordeonverbindung	29
2.14	Gadgeteer Komponenten	30
3.1	Ein- und Ausgabe bei Augmented Reality	32
3.2	Kombination aus Finger- und Stifteingabe	35
3.3	Lasercutter und hergestellte Objekte	36
3.4	LaserOrigami – 3D-Objekte mit Lasercutter ohne Montage	38
3.5	Iterative Gestaltung für Gehäuse	38
3.6	Tangible User Interfaces mit aktiven Tangibles	40
3.7	Personalisierung bei Schuhen	41
4.1	Natcut Ablauf	47
5.1	Schnittstellen zwischen der Anwendung und der realen Welt	49
5.2	Zusammenspiel der Komponenten	50
5.3	Hierarchie der ViewModels	54
6.1	Testreihe für Verbindungsarten – Acryl- und Schwalbenschwanzverbindung	62
6.2	Testreihe für Verbindungsarten – Schrauben- und Akkordeonverbindung	62
6.3	Benutzeroberfläche – Gehäuse-Wizard	64
6.4	Benutzeroberfläche – Hauptansicht	65
6.5	Benutzeroberfläche – Personalisierung und Feedback der Objekterkennung	67
6.6	Drei Flächen eines Quaders	69
6.7	Stabilität von Zapfen/Zapfen-Ecken	70
6.8	Generiertes Akkordeon-Modell	71

7.1	Gadeteer-Komponenten als Tangibles	74
7.2	Tangibles als Eingabe	76
7.3	Hergestellte Prototypen als Vorlage	76
7.4	Gehäuse von Teilnehmern	77
7.5	Ergebnisse der explorativen Studie	79
A.1	Klassendiagramm des Objektmodells – AArea	85
A.2	Klassendiagramm des Objektmodells – Casing	86
A.3	Klassendiagramm des Objektmodells – Verbindungsoptionen	87
A.4	XSD2Code	88
A.5	Übersicht der Komponenten	89

Tabellenverzeichnis

7.1	Entwurfsdauer eines Gehäuses mit einem CAD-Programm	78
7.2	Entwurfsdauer eines Gehäuses mit einem Grafikprogramm	78

Verzeichnis der Listings

6.1	Ausschnitt einer generierten SVG-Datei	68
6.2	Nut- und Zapfen-Algorithmus für Fingerverbindungen	72

1 Einleitung

Bis diese Arbeit in ihrer endgültigen Fassung vorlag, wurden verschiedene Zwischenstände ausgedruckt, um korrigiert oder besprochen zu werden. Das Drucken ist heute soweit in den Alltag integriert, dass es zur Selbstverständlichkeit geworden ist. Dabei ist die Verbreitung des Druckers im Heimbereich erst wenige Jahrzehnte her. Genau dieser Trend ist für Lasercutter und 3D-Drucker zu erwarten [SM10b]. In einigen Jahren wird demnach in vielen Haushalten ein solches Gerät stehen, mit dem wir dann nicht nur Prototypen fertigen, sondern Ersatzteile selbst herstellen und unseren persönlichen Geschmack ausdrücken können. Die Beliebtheit von individuellen Gegenständen ist heute in vielen Branchen zu beobachten. Die Anzahl der Modelle im Automobilssektor und ihre Ausstattungskombination stieg in den letzten Jahren enorm an¹. Durch neue Herstellungsverfahren und geschickte Kombination bestehender Plattformen, können Aussehen und Form einfach geändert werden, ohne das Auto komplett neu entwerfen zu müssen. Auch im Bereich der Mobiltelefone erfreuen sich Schutzhüllen immer größerer Beliebtheit. Dafür kommen unterschiedliche Materialien, Farben und Designs zum Einsatz. Längst können die Hüllen mit den letzten Urlaubsfotos, seiner Lieblingsband oder einer politischen Botschaft erstellt werden, um sein eigenes Statement abzugeben².

Der Frage, wann man sein eigenes Mobilgerät selbst herstellen kann gehen unter anderem Schmidt et al. nach [SDS11]. In ihrem Artikel skizzieren sie eine Vision, wie zukünftig auch komplexe Gegenstände hergestellt und persönlich gestaltet werden können. Es wird jedoch noch einige Zeit vergehen, bis diese Vision Realität wird. Um die Entwicklung voranzutreiben, wird mit dieser Arbeit ein Konzept entwickelt, auf dessen Basis zunächst die Herstellung von Prototyp-Gehäusen vereinfacht wird. Lasercutter eignen sich durch ihre hohen Schnittgeschwindigkeiten und die mittlerweile erschwinglichen Preise dafür besonders gut. Für einen professionellen Gehäuseentwurf werden häufig *Computer Added Design* (CAD)-Programme eingesetzt. Mit ihrer hohen Einstiegshürde durch Komplexität und Einarbeitungsdauer eignen sie sich im Heimbereich aber weniger. Darüber hinaus benötigt selbst ein erfahrener CAD-Benutzer einige Zeit um ein einfaches Gehäuse mit Verbindungselementen zu entwerfen. Diese Arbeit zeigt eine Möglichkeit, um diese Lücke zu schließen und das Entwerfen und Herstellen von einfachen Gehäusen zu beschleunigen. Dank kurzen Produktionszyklen kann das Ergebnis direkt evaluiert und verfeinert werden.

¹Artikel über die Vielfalt sind unter anderem beim Handelsblatt und der Zeit zu finden. <http://www.handelsblatt.com/auto/nachrichten/car-studie-modellvielfalt-bei-autoherstellern-explodiert/6630680.html>

<http://www.zeit.de/auto/2011-08/automarkt-typenvielfalt>

²<http://designskins.com/design-folien/handy-huelle/>

Das Konzept wurde dabei von mehreren Themengebieten beeinflusst, wovon die wichtigsten Holzverbindungen, der Lasercutter, der Multitouch-Tisch und die Entwicklung der *Personal Fabrication* sind. Lasercutter, die für Privatpersonen heute finanzierbar sind, haben zwei Freiheitsgrade und können damit Formen herstellen, die aus einer flachen Platte ausgeschnitten werden. Um einen Körper zu bilden, müssen die Teile anschließend zusammen gesetzt werden. Hier kommen die Holzverbindungen ins Spiel.

Bei der Entwicklung von Prototypen kommen häufig elektronische Bausätze wie dem *Microsoft .Net Gadgeteer*³ zum Einsatz. Mit solchen Bausätzen ist es möglich funktionsfähige Prototypen wie eine Digitalkamera innerhalb weniger Minuten zu realisieren. Das Konzept, das in dieser Arbeit entwickelt wird, erlaubt es dem Benutzer nun alle Bauteile in einem Gehäuse zu vereinen. Um den Designprozess weiter zu vereinfachen, werden die elektronischen Bauteile als Tangibles verwendet. Außerdem wird die Entwicklung der Personal Fabrication, mit dem die Personalisierung von Gegenständen forciert wird, berücksichtigt.

1.1 Gliederung der Arbeit

Die Arbeit gliedert sich in acht Teile, deren Inhalt im Folgenden kurz erläutert wird:

Kapitel 1 – Einleitung Die Motivation und Aufgabenstellung werden dargestellt.

Kapitel 2 – Grundlagen Die wichtigsten Ideen zur Personal Fabrication sowie die Rolle von CAD-Programmen werden erläutert. Weiterhin sind eine Einführung in Microsofts Plattform .Net Gadgeteer und Besonderheiten beim Arbeiten mit Lasercuttern zu finden. Zu Letzterem gehören verschiedene Verbindungsarten, insbesondere für Holz.

Kapitel 3 – Verwandte Arbeiten Das Kapitel beschreibt alle Themengebiete, die einen direkten Bezug zur Arbeit haben. Besonderes Augenmerk wurde auf die Interaktionsart, vor allem für Lasercutter, sowie auf Material, Form und Ästhetik gelegt.

Kapitel 4 – Konzept Dieses Kapitel befasst sich mit der grundlegenden Idee der Arbeit und der Abfolge, die eine einfache Herstellung von Gehäusen ermöglicht.

Kapitel 5 – Entwurf Die Einzelheiten über Architektur und deren Unterteilung in Komponenten werden hier vorgestellt.

Kapitel 6 – Implementierung Die Dokumentation der Implementierung beinhaltet Vorversuche zu Verbindungsarten, die verwendeten Richtlinien der Implementierung und Details über die Umsetzung der im Entwurf vorgestellten Komponenten.

Kapitel 7 – Explorative Benutzerstudie Um das Konzept zu evaluieren, wurde eine Studie zur Benutzerinteraktion durchgeführt, deren Details und Ergebnisse erläutert werden.

Kapitel 8 – Zusammenfassung Dieses Kapitel fasst die Arbeit zusammen und gibt einen Ausblick für zukünftige Entwicklungsmöglichkeiten.

³<http://www.netmf.com/gadgeteer/>

2 Grundlagen

Dieses Kapitel behandelt alle Grundlagen, die zum Verständnis der Arbeit und des folgenden Kapitels Verwandte Arbeiten notwendig sind. Zunächst wird die Rolle von CAD-Programmen aufgezeigt und auf die Entwicklung von Personal Fabrication eingegangen. Anschließend werden die Idee von Microsofts Gadgeteer und verschiedene Bauteile davon beschrieben, bevor im letzten Abschnitt eine Übersicht über Verbindungsarten für unterschiedliche Materialien vorgestellt wird. Mit den Verbindungsarten können mehrere Flächen mit einander verbunden und zu einem Gehäuse zusammengesetzt werden.

2.1 Die Rolle von CAD-Programmen

Um Gegenstände maßgenau herzustellen, ist es heute selbstverständlich, dass die Maße bekannt sind, die als Vorgabe dienen. Blickt man in der Geschichte zurück, so findet man bereits vor über 2000 Jahren technische Zeichnungen, die den Ägyptern halfen ihr Land mittels Seilkunst zu vermessen [Leo12]. Auch wenn in den folgenden Jahrhunderten viele Ideen als Zeichnung entstanden, so wurde die Herstellung von alltäglichen Gegenständen dennoch fast ausschließlich durch Handwerkskunst bestimmt. Nur für die wenige Gegenstände wurde eine Zeichnung angefertigt, da das Design implizit war und auf einem gemeinsamen Verständnis beruhte [Gro09]. Erst durch die Manufaktur wurde nach einem gemeinsamen Modell gefertigt. Durch den Einzug der industriellen Herstellung wurde das Vorgehen abgelöst und die Massenfertigung ermöglicht, für die eine einheitliche, allgemeingültige Vorlage unumgänglich war.

Der nächste Wandel wurde im 20. Jahrhundert durch die Entwicklung von Computern ausgelöst. Mit ihnen entstand Ende der 50er Jahre der Begriff des CAD. Der Benutzer konnte Maße, Körper, Flächen und geometrische Bedingungen festlegen, so dass die Zeichnungen nun leichter änderbar und kopierbar wurden. Durch diesen Schritt wurde erstmals der Designer und der Konstrukteur wiedervereinigt [SO06]. Mit CAD-Programmen stieg die Präzision der Modelle ebenso wie die technischen Möglichkeiten für neue Produkte. Das verlangt jedoch auch, dass sich der CAD-Designer mit den technischen Voraussetzungen auskennt, da die Auswirkungen des CAD-Entwurfs unmittelbaren Einfluss auf die Produktion haben [Oxm12].

Mit dem Fortschritt in Hardware und Software wurde die Leistungsfähigkeit der CAD-Programme kontinuierlich gesteigert, so dass heute ein Großteil der Schritte, von der Formulierung eines Problems bis zu ihrer Lösung, innerhalb eines Programms umgesetzt werden können. Die Eingabe erfolgt heute in der Regel mit Maus und Tastatur, während die Ausgabe

über einen Bildschirm erfolgt, auf dem ein 3D-Modell oder verschiedene Seitenansichten dargestellt werden. Ein modernes CAD-Programm kann über den kompletten Entwicklungsprozess eines Gegenstands unter anderem folgende Funktionen bieten: Freihand-Skizzieren eines ersten Entwurfs, dessen automatische Generierung in CAD-Modellen [CKMH12], die Verfeinerung dieser Modelle, das Simulieren von Bewegungsvorgängen und Testen von physikalischen Grenzen sowie die Generierung von Produktionsdaten.

Dass mit steigendem Funktionsumfang auch die Komplexität steigt, zeigen Ibrahim und Rahimian bei ihrem Vergleich zwischen Handskizzen mit Stift und Papier und einem CAD-Programm [IR10]. Auch die vielerorts angebotenen Schulungen für einen Einstieg in CAD-Software, die sowohl Teil einer Ausbildung als auch im industriellen Umfeld angesiedelt sind, zeugen von einer hohen Einstiegshürde. Neben der Weiterentwicklung von Eingabe- und Ausgabemöglichkeiten [DYZA05, IR10] wird besonders die Art und Weise untersucht, mit der der Designer mit dem CAD-Programm interagiert. Modi et al. schlagen die Architektur für ein intelligentes CAD-Programm vor, bei dem der Designer lediglich die Anforderungen spezifizieren muss [MTLZ11]. Basierend auf bereits getroffenen Entscheidungen, werden mit Hilfe eines Petrinetzes, das Fuzzy-Logik verwendet, neue Lösungen angeboten. Durch iteratives Suchen werden dabei, entsprechend den vorgegebenen Vorschriften, Möglichkeiten gesucht, bis das Limit der Designkosten erreicht ist. Beispielsweise wird ein Tisch durch eine Tischplatte, Beine und das verwendete Material beschrieben. Alle vom Designer getroffenen Entscheidungen werden vom CAD-Programm interpretiert, wobei der Designer entscheidet, welche Interpretation für darauffolgende Designentscheidungen verwendet werden sollen.

Auch Richter et al. legen den Fokus nicht auf die Vollautomatisierung der Designaufgabe, sondern auf die Zusammenarbeit zwischen Mensch und Computer, das heißt der Mensch soll aktiv im Designprozess teilnehmen [RWBB10]. Er unterscheidet dabei zwischen zwei Rollen, die mit dem CAD-Modell in Berührung kommen: dem Designer und dem Benutzer. Der Benutzer stellt dabei zum Beispiel einen Kunden dar, für den ein Architekturentwurf angefertigt wird und der über keine Kenntnisse des CAD-Programms verfügen muss. Er interessiert sich mehr für das Äußerliche oder gibt allgemeine Vorgaben, die der Designer umsetzen muss. Beispielsweise können die Bedingungen lauten möglichst einfache Laufwege zwischen Räumen zu haben und die Kaffeeküche in der Nähe des Treppenbereichs unterzubringen. Diese unkonkreten Anforderungen muss der Designer nun mit Hilfe von Regeln in der CAD-Software praktisch umsetzen (siehe Abbildung 2.1).

Neben alternativen Ansätzen, welche die Interaktionsart mit dem CAD-Programm betreffen, gibt es einige Usability-Prinzipien, deren Umsetzung auch bei herkömmlichen Programmen Verbesserungen erzielen können. Lee et al. haben diese untersucht und in Kategorien zusammengefasst, um schließlich Vorschläge für eine gute Implementierung geben zu können [LETH10]. Dennoch wird die Komplexität, die einem CAD-Programm mit so vielen Möglichkeiten immanent ist, nicht soweit gesenkt werden können, dass es für Personen ohne oder mit wenig technischem Verständnis angemessen ist.

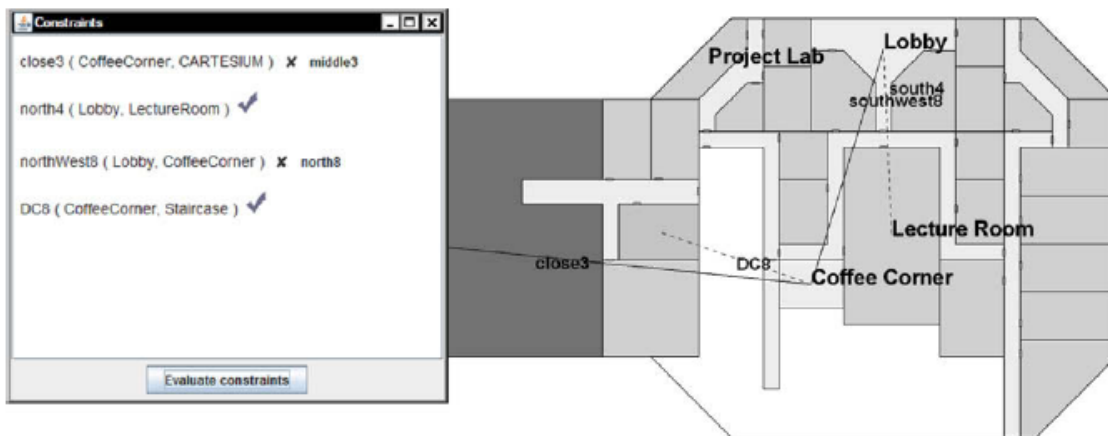


Abbildung 2.1: Die Umsetzung und Überprüfung der Anforderung (links) und die räumliche Darstellung (rechts). Dabei gibt es verschiedene Bedingungen wie relative Nähe, Anordnung und Berührung zu beachten. Quelle: [SXG10]

2.2 Personal Fabrication

In der *Personal Fabrication* geht es darum Gegenstände als Privatperson selbst herzustellen. Dazu werden Methoden der *Digital Fabrication* verwendet, die durch den Einsatz günstiger Werkzeugmaschinen umgesetzt werden. Verschiedene Aspekte und praktische Beispiele der Personal Fabrication werden im Folgenden hervorgehoben, wobei der nächste Abschnitt zuerst den Bezug zur *Digital Fabrication* herstellt. Beide Bereiche sind unabhängig voneinander, die Digital Fabrication hat die Personal Fabrication jedoch stark beeinflusst und vorangetrieben.

Prototyping und Digital Fabrication

Dass die historische Entwicklung der Herstellungsverfahren für die Massenfertigung nicht nur Vorteile mit sich bringt, macht sich besonders im Bereich der Einzelteillfertigung bemerkbar. Selbst bei hoher Flexibilität der Fertigungsmaschinen sind der Umsetzung einer Idee konstruktive oder finanzielle Grenzen gesetzt, da die Rüstzeit und die Vorbereitung des Modells nicht zu vernachlässigen sind.

Ein kurzer Ausflug in das geschäftliche Umfeld zeigt, dass dort das Entwickeln von komplexen Produkten durch *Rapid Prototyping*¹ bereits etabliert ist, um Entwicklungskosten zu sparen. Unter Rapid Prototyping versteht man eine Methode, in der der Entwicklungsprozess eines neuen Produktes beschleunigt wird. Dabei werden über mehrere Stufen hinweg Prototypen hergestellt und evaluiert. Während den verschiedenen Iterationen können unterschiedliche

¹Einen Einstieg und weitere Informationen können bei Jacobs gefunden werden [Jac95].

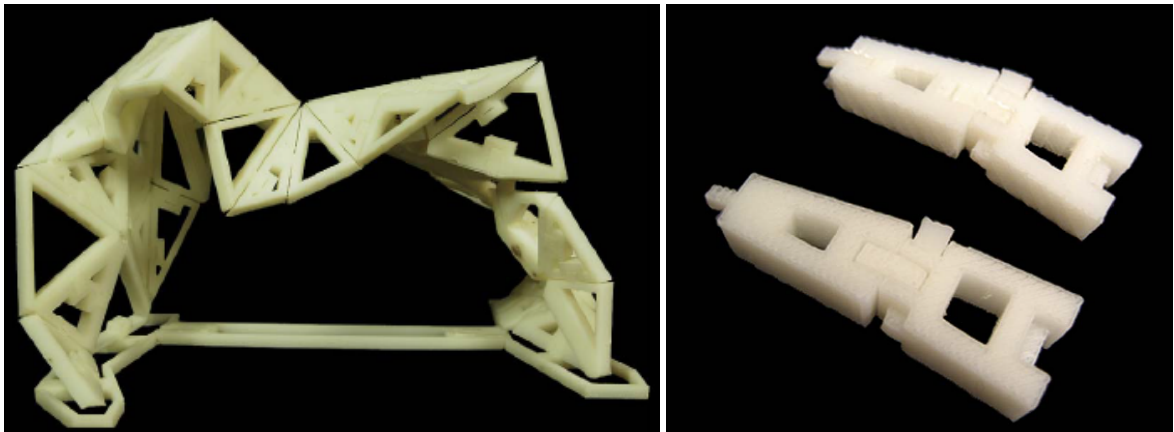


Abbildung 2.2: Durch die Erstellung von Modellen kann die Funktionalität bereits getestet werden. Links ist das Zusammenspiel von Form und Steckverbindungen zu sehen, rechts ein Verbindungselement aus Schaumstoff. Quelle: [SO06]

Technologien zum Einsatz kommen, die verschiedene Eigenschaften des Produkts hervorheben. Beispielsweise kann sich der Detailgrad von einem Modell zum nächsten erhöhen, funktionale Eigenschaften können getestet werden (siehe Abbildung 2.2), aber auch Design (siehe Abbildung 2.3) und Materialbeschaffenheit werden vor der Serienproduktion verifiziert [SO06].

Durch Digital Fabrication, einem Herstellungsprozess, der stark durch Computer gesteuert und geregelt wird, wird eine hohe Präzision des Modells ermöglicht, welche im Vergleich zu Handarbeit schneller und exakt reproduzierbar ist. Die Besonderheit der digitalen Herstellung stellt dabei der nahtlose Übergang zur Produktion dar. Durch gemeinsam verfügbare Dateien, die auf zentralen Datenservern gelagert sind, sind die Informationen zur Produktion aber auch zur Modifikation einheitlich und die Rüstzeit der Maschinen wird gesenkt. Produktionsmodelle, die nicht herstellbar, unrealistisch oder einer bestimmten Belastung nicht gewachsen sind, können dank physikalischer Berechnungsmodelle der CAD-Programme validiert und vor der Produktion aussortiert werden. Außerdem können komplexe Produkte in kleinere Modelle aufgeteilt werden, so dass jede Komponente einzeln betrachtet und ihre Funktion unabhängig getestet werden kann. Da die Digital Fabrication schon bei der Prototyp-Entwicklung eingesetzt wird, können hier bereits die Vorkenntnisse gesammelt werden, um sie für den späteren Produktionsprozess zu berücksichtigen. Synergien können auch durch das Wissen für die Personal Fabrication, die im Folgenden näher beschrieben wird, gefördert werden.

Von Digital Fabrication zu Personal Fabrication

Die Herstellung von Einzelstücken für den Heimbereich findet nach wie vor durch Handarbeit statt. Dazu muss zum einen ausreichend Erfahrung im Umgang mit den benötigten



Abbildung 2.3: Mehrere Designentwürfe können einfach hergestellt werden, um sie anhand ihres realen Modells zu bewerten. Quelle: [SO06]

Werkzeugen vorhanden sein, zum anderen müssen die Werkzeuge selbst verfügbar sein. Die Arbeitsdauer und die körperliche Leistung sind weitere Faktoren, die bei der Herstellung einfließen. Durch die immer günstiger werdenden Maschinen im CNC-, 3D-Druck- und Lasercutter-Bereich werden diese Voraussetzungen jedoch zunehmend aufgehoben. Der Übergang zwischen der professionellen Herstellung für Industriezwecke und dem persönlichen Bereich schwimmt zunehmend, so dass man mit genügend Grundkenntnissen von der Entwicklung mit CAD-Programmen und ein wenig Verständnis für den Herstellungsprozess bereits eigene Gegenstände realisieren kann. Durch die Preissenkung, mit der 3D-Drucker bereits ab 200 US\$² (siehe Abbildung 2.4) zu erhalten sind, und der Weiterentwicklung der Produkte, dank den aktiven *Do it yourself*-Gemeinden (DIY), sinkt die Einstiegshürde enorm. Mota [Mot11] behandelt unter anderem die Verbreitung von digitalen Modellen, die durch Online-Plattformen wie Thingiverse³ und ihrer aktiven OpenSource-Gemeinde weiter vorangetrieben wird, so dass bereits viele Vorlagen aus Bereichen wie Hobby, Spielzeug, Alltagsgegenstände aber auch Accessoires kostenlos heruntergeladen werden können.

²<http://makibox.com>

³<http://www.thingiverse.com>

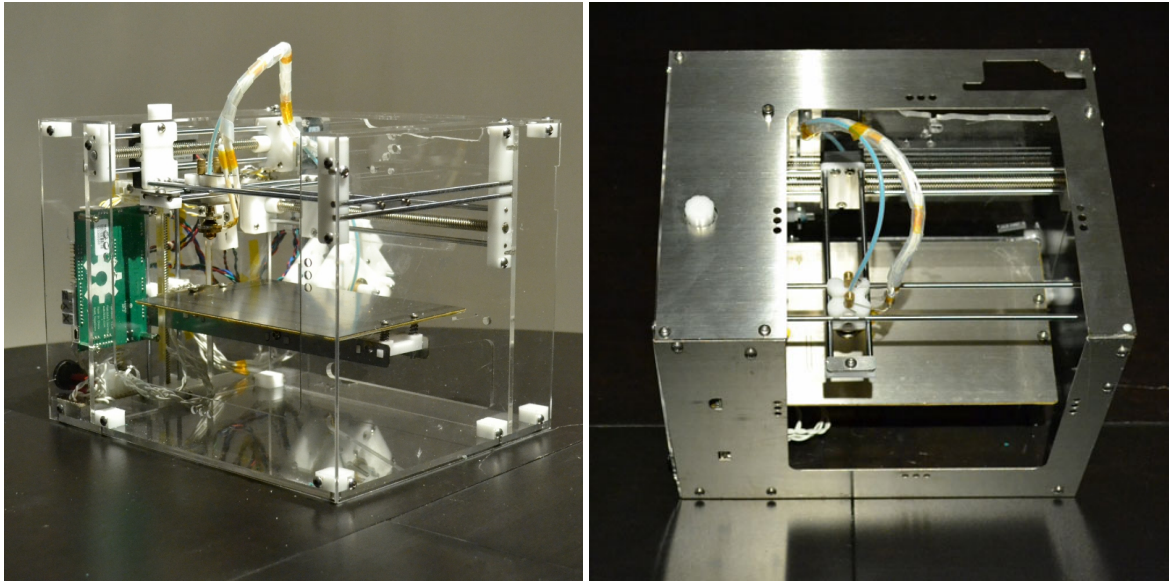


Abbildung 2.4: 3D-Drucker sind bereits ab 200 US\$ zu erhalten (linkes Bild). Die rechte Variante ist mit 350 US\$ zwar etwas teurer, die tragenden Elemente sind dafür aber aus Edelstahl gefertigt. Natürlich unterliegen die Drucker Einschränkungen wie den maximalen Maßen (150 x 110 x 90 mm) und der hohen Lieferzeit von mehreren Monaten. Dennoch sind die Produkte zu diesem Preis derzeit konkurrenzlos. Quelle: <http://makibox.com>

Da die Preise für 3D-Drucker und Lasercutter zwar deutlich gesunken, aber immer noch unrentabel für die Herstellung eines einzelnen Gegenstandes sind, wird von verschiedenen Firmen der Service angeboten, digitale Modelle auszudrucken beziehungsweise auszuschneiden und dann per Post zuzuschicken. Ponoko⁴ und Shapeways⁵ sind Beispiele solcher Firmen, die außerdem verschiedene Programme anbieten, um die Gegenstände zu modellieren. Ein weiterer Vorteil, den eine externe Firma mit sich bringt, ist das Wissen über die Produktion, so dass Kalibrierungswerte, Schnittgeschwindigkeiten oder das Entfernen von Kleberesten nicht in den Aufgabenbereich des Auftraggebers fallen. Auf der anderen Seite zeigt sich gerade bei prototypischer Herstellung, dass mehrere Iterationen nötig sind, bis das gewünschte Endprodukt erreicht ist. Möchte man die Gesamtentwicklungszeit reduzieren, ist ein Lasercutter beziehungsweise 3D-Drucker mit direktem Zugang unumgänglich. Kurze Zykluszeiten helfen außerdem direktes Feedback zu erhalten und einen Gedankengang zu Ende zu führen oder zu verfeinern, bevor er sich verflüchtigt.

Für die Herstellung im Heimbereich muss die Interaktion besonders auf die vorhandenen Vorkenntnisse des Anwenders abgestimmt werden. Diesem Punkt sollte hohe Beachtung geschenkt werden, um eine schnelle Marktdurchsetzung von Geräten der Personal Fabri-

⁴<https://www.ponoko.com>

⁵<http://www.shapeways.com>

ation zu ermöglichen. Zieht man die Parallele zu Druckern im Heimbereich, so kann man die einfache und standardisierte Verwendung als Vorbild nehmen. Mit einem simplen Bedienkonzept können die meisten Fälle abgedeckt werden, womit das Drucken einer breiten Bevölkerungsschicht zugänglich gemacht wird. Aus diesem Grund basiert das im vierten Kapitel vorgestellte Konzept nicht auf einem CAD-Programm, sondern bietet die Basisfunktionen zur Herstellung einfacher Gegenstände an. Nimmt man den Einsatzbereich des Druckers als Referenz, so zeigt sich auch heute noch, dass abhängig vom Grad der Professionalität, unterschiedliche Lösungen angeboten werden. Auch wenn ein Laserdrucker mit Duplexfunktion heutzutage finanziell erschwinglich ist, so werden Druckaufträge mit speziellen Anforderungen oder hohem qualitativen Anspruch nach wie vor in Copyshops aufgegeben. Ähnlich ist das Ziel von Personal Fabrication zu sehen. Es geht nicht darum Gegenstände höchster Qualität oder Komplexität herzustellen, sondern das Augenmerk auf Praktikabilität, sofortige Einsetzbarkeit und Alltagstauglichkeit zu legen. Die Umsetzung der eigenen Vorstellung und das Herstellen eines Unikats spielen dabei ebenfalls eine große Rolle. Durch die iterative Entwicklung ergeben sich auch pädagogisch wertvolle Möglichkeiten, wie die Neugierde von Schülern zu wecken und die räumliche Vorstellungskraft mit der realen Umsetzung eines Gegenstandes zu unterstützen. Je nach Alter kann der Schwerpunkt auf die Wahrnehmung und Erfahrung von Gegenständen oder ihre Abbilder auf einen zweidimensionalen Raum gelegt werden. Alternativ kann die Umsetzung ihrer Ideen beispielsweise mit der Gestaltung im Kunstunterricht oder der Herstellung im Werkunterricht erfolgen. Wu und Gross zeigen eine einfache Idee, wie auch die physikalischen Grundgesetze spielerisch erlernt werden können [WG10]. Dort werden Würfel modular aufgebaut, die jeweils genau eine Funktion beherbergen (siehe Abbildung 2.5). So kann aus einer Kombinationen von Batterie, Leuchtdiode und Fotozelle ein Bewegungsmelder mit Lichtanzeige erstellt werden. Für komplexere Ideen können aber auch Mikrocontroller verwendet werden, um das Einsatzspektrum zu erweitern und intelligente Lösungen zu ermöglichen. Mit dem Einsatz von Mikrocontrollern kann ein anschauliches Objekt für den Elektrotechnikunterricht entwickelt werden. Auch *Fab@School*⁶ möchte mit seinem Projekt die Integration von 3D-Druckern in Schulen zeigen und fördert die Begeisterung der Schüler für neue Technologien [SM10b].

Chancen und Herausforderungen

Ein für das US-Amt für Wissenschaft- und Technologiepolitik erstellter Bericht [LK10] befasst sich mit dem aufstrebenden Markt, seinen Möglichkeiten und Herausforderungen. Lipson, einer der beiden Autoren, ist auch der Gründer des bereits erwähnten Fab@Home- beziehungsweise Fab@School-Projekts. Lipson und Kurman erwähnen, dass die Wirtschaft momentan noch nicht den Markt sehe, der tatsächlich vorhanden sei. Sie bescheinigen der Personal Fabrication-Branche Chancen, die dem Heimcomputer gleichkommen. Doch durch den zu niedrigen Stellenwert könne sich die Entwicklung nur langsam entfalten und es komme zu einer langsameren Neukundengewinnung, die den Markt wiederum antreiben könne. Daher werden in ihrem Bericht verschiedene Vorschläge zur Förderung gemacht, wie das

⁶<http://www.fabathome.org/?q=node/97>

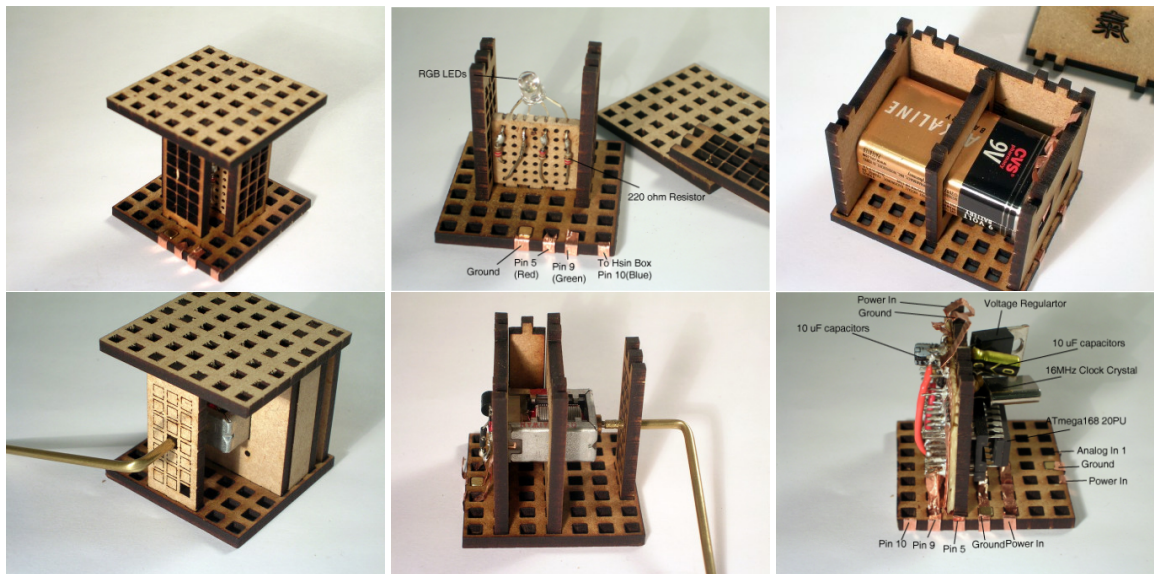


Abbildung 2.5: Vielseitige Würfel, die durch ihren modularen Aufbau einfach kombiniert werden können. Von der Herstellung über den Entwurf bis zu den physikalischen Erklärungen bieten sie viele Einsatzmöglichkeiten im Unterricht. Oben: LED-Würfel montiert (links) und offen (Mitte); der rechte Würfel liefert die Energie, mit der über die Kupferstreifen andere Würfel versorgt werden. Unten: Ein Motor montiert (links) und offen (Mitte) und ein Controller (rechts), mit dem sich komplexere Logik realisieren lässt. Quelle: <http://mtifall09.wordpress.com/author/dooflower/>

Einführen eines Lasercutters oder 3D-Druckers an Schulen sowie der darauf abgestimmte Unterricht oder sogar steuerliche Vergünstigungen für das Rohmaterial für Personal Fabrication. Potenzial der Personal Fabrication sehen die Autoren auch in der Marktforschung, da nicht wie bisher nur mit indirekten Mitteln gemessen werden kann, welches Produkt in der Gunst des Kunden liegt, sondern die bereits vom Kunden produzierten Prototypen verwendet werden können. Diese Prototypen können Firmen als Ausgangsbasis dienen, um sie zu verbessern und marktreif einzuführen. Dabei würde sich auch der Entwicklungszyklus verkürzen und der Bias durch die Beeinflussung bei Umfragen entfallen. *Crowdsourcing*⁷ könnte durch Personal Fabrication forciert werden, so dass beispielsweise ein Schuhhersteller seinem Kunden die Design-Datei zugänglich macht, dieser sie seinen Wünschen nach anpassen kann und vom Hersteller geliefert bekommt. Für verschiedene Prototypen kann der Kunde sich jeweils seinen Entwurf ausdrucken, um seine Vorstellung durch ein 3D-Modell verifizieren zu können. Er kann bei einer guten Idee, die der Hersteller übernehmen möchte, auf verschiede-

⁷ „Interaktive Form der Wertschöpfung unter Nutzung moderner IuK-Techniken (Web 2.0). Zusammengesetzt aus den Begriffen Crowd und Outsourcing. Einzelne Aufgaben, die bisher intern bearbeitet wurden, werden an eine Vielzahl von Nutzern oder Interessenten ausgelagert und häufig in Form eines Wettbewerbes ausgeführt.“
Quelle: <http://wirtschaftslexikon.gabler.de/Archiv/81583/crowdsourcing-v7.html>

ne Weisen vergütet werden, beispielsweise durch eine Reduktion des Kaufpreises oder durch ein Stellenangebot als freier Mitarbeiter.

Neben diesen Chancen sind mit der Verbreitung von Lasercuttern und 3D-Druckern auch Risiken verbunden. Dazu gehört die steigende Problematik das Urheberrecht zu schützen, da Produkte noch einfacher nachgebaut werden können, wenn die digitale Vorlage vorhanden ist. Dies ist vergleichbar mit der heutigen Situation der Musik- und Filmindustrie, die große Anstrengungen unternehmen, um Raubkopien zu verhindern. Auch die Firmen müssten sich umstellen und unter Umständen Teile ihrer Firmengeheimnisse preisgeben, wenn es um eine flexible Produktion geht, oder gar dem Kunden die Teile zuschicken, die dieser dann selbst zusammenbaut.

Eine zentrale Frage stellt sich auch bei Herstellung gefährlicher Gegenstände. 3D-Drucker sind weltweit verfügbar und bieten Potenzial Waffen einfach reproduzieren zu können. Zwar ist die Technologie momentan noch nicht weit genug fortgeschritten, um zu nennenswerten Auswirkungen beizutragen, aber um geeignet darauf reagieren zu können, sollte man die ethisch-gesellschaftlichen Fragen bis dahin geklärt haben. Möglicherweise könnte das Einfügen von *Machine Identification Codes* (MIC) bei der Rückverfolgung helfen. Besser wäre jedoch, ähnlich wie bei Verfahren einiger Hersteller zum Verhindern des Schwarzgelddrucks, die Herstellung von Waffen direkt zu unterbinden. Durch die flexible Gestaltungsmöglichkeit wird das Erkennen von Waffen jedoch ungleich schwerer realisierbar.

Eine Standardisierung, die Preisvergünstigungen und Vereinfachung ermöglichen könnte, wird durch die offene Gemeinde und die stetig wachsende Zahl an Produzenten erschwert. Außerdem sind die Risiken für eine Firma, die Geräte für Personal Fabrication anbietet, schwer abzuschätzen, da ihr Ruf vom guten Willen des Einsatzzwecks ihrer Kunden abhängt. Ebenso wird es bei der Produktion für eine Privatperson schwierig alle Patentrechte zu beachten, da nicht die nötigen Kenntnisse vorhanden sind. Dass trotz dieser Schwierigkeiten die Personalisierung eine immer größere Rolle beim Absatz von Produkten spielt, kann man bei der Wahl von Schuhen gut beobachten. Einige Hersteller bieten bereits die Möglichkeit, die Farbkombinationen individuell zu wählen, um seinen persönlichen Schuh zu gestalten (siehe auch Abbildung 3.7). Dem Faktor der Personalisierung widmet sich der letzte Abschnitt des Kapitels Verwandte Arbeiten, wobei die Geräte des Personal Fabrications nicht nur die Möglichkeit bieten die Farbwahl anzupassen, sondern auch die Form und das Material des Gegenstandes.

2.2.1 Beispiele von Werkzeugmaschinen in der Personal Fabrication

Im Folgenden wird der Einsatz einiger Werkzeugmaschinen beschrieben, um die Möglichkeiten von Personal Fabrication aufzuzeigen. Unterschieden wird dabei zwischen 3D-Druckern und Lasercuttern, da sie nicht nur unterschiedliche Herstellungsverfahren nutzen, sondern auch ihre Eingabe darauf abgestimmt ist.



Abbildung 2.6: Das Foto dient zur Orientierung und wird verwendet, um einen Teedeckel für die Kanne zu erstellen. Durch das Festlegen einer Rotationsachse ist es ausreichend die äußere Kontur (rechtes Bild) des Deckels nachzufahren. Quelle: [LSMI10]

3D-Drucker

Beispielhaft für die Entwicklung und Möglichkeiten von 3D-Druckern steht die Benutzerstudie von Lau et al. [LSMI10]. Die Aufgabe der Teilnehmer war es, einen Deckel für eine bereits vorhandene Teekanne zu kreieren. Weder das Aussehen noch der Gegenstand selbst sind besonders spektakulär, aber die individuelle Gestaltung und die direkte Umsetzung eines einsatzbereiten Alltagsgegenstandes zeigen neue Wege auf. Dabei wurde ein Foto der vorhandenen Teekanne als Grundlage genommen, um die Auflagefläche des Deckels zu bestimmen. Durch Gestenerkennung wird die Kannenöffnung, die im Foto zur Ellipse verschoben ist und vom Benutzer nachgefahren wird, erkannt und hinterher wieder zu einem Kreis im 3D-Modell transformiert. Da der Teedeckel eine rotationssymmetrische Achse hat, reicht es die Kontur zwischen Oberkante und Unterkante des Deckels mit einer Linie zu beschreiben. Auch wenn das System auf die Aufgabe zugeschnitten ist, ist die einfache Bedienung vorbildlich (siehe Abbildung 2.6).

Der 3D-Druck, der durch das schichtweise Drucken mehrerer Lagen die freie Gestaltungsform ermöglicht, kommt besonders bei Sketch Furniture⁸ zur Geltung. Hier wird aus Freihandgesten im Raum direkt das modellierte Möbelstück erstellt. Auch Mellis und Buechley zeigen, dass die Teilnehmer ihrer Studie die Formenfreiheit beim Erstellen unterschiedliche Computermäuse zu nutzen wissen [MB12]. Gegeben war eine Platine, die die zwei mechanischen Tasten und einen Infrarotsensor zum Abtasten des Untergrunds bereits einsatzbereit lieferte. Um die Kreativität nicht zu weit einzuschränken, wurde aber auch die Möglichkeit einer individuellen Ausgestaltung der Platine angeboten. Die Hauptaufgabe war es jedoch, die Form der Maus, aber auch die funktionellen Teile zum Montieren der Platine, zu gestalten. Die teilnehmende Gruppe bestand aus Designern, die Erfahrung mit dem Umgang von Rhino hatten, einer Software zum Erstellen von 3D-Modellen. Somit konnte die Erprobungsphase der Software ausgelassen und dennoch die Möglichkeiten, die ein 3D-Drucker bietet untersucht werden. Innerhalb von 24 Stunden wurden die Mäuse angefertigt, wobei eine davon statt Tasten Drucksensoren von Fingern als Eingabe verwendete (siehe Abbildung 2.7). Die Studie zeigt, dass schnelles Entwickeln und Kreativität mit 3D-Druckern gut umgesetzt werden

⁸<http://www.designfront.org/category.php?id=81&product=191>



Abbildung 2.7: Die beiden Mäuse (links und Mitte links) entstanden im Rahmen eines Workshops innerhalb weniger Stunden. Die Maus in der Abbildung Mitte links verwendet Drucksensoren der Finger, statt der üblichen Tasten. Die beiden rechten Bilder zeigen den prototypischen Entwurf und die vorgefertigte Platine, die als Vorlage dienten. Quelle: [MB12]

können und Potenzial für weitere Verbesserungen vorhanden ist. Weitere Informationen zur historischen Entwicklung, Einschränkungen und Vorteile von 3D-Druckern können bei Berman nachgelesen werden [Ber12].

Trotz der bereits vielseitig verwendbaren Technologie gibt es noch deutliche Einschränkungen, besonders im Bereich von einsetzbaren Materialien. Die Idee, sämtliche Komponenten, darunter auch Batterien, Motoren und Elektrocontroller drucken zu können, ist heutzutage noch nicht realisierbar. Auch wenn elektronische Leiterbahnen bereits gedruckt werden können [SZH12] und damit der nächste Schritt getan ist, wird die Herstellung kompletter elektronischer Geräte mit 3D-Druckern noch einige Zeit auf sich warten lassen. Bei der Verarbeitung von Metall sind momentan noch Schmelzpunkt und Viskosität entscheidende Knackpunkte [SM10b] und für natürliche Produkte wie Holz und Stein gibt es momentan noch keine Ansätze wie sie mit 3D-Druckern erstellt werden können.

Lasercutter

Die Lasercutter, die heute im Heimbereich eingesetzt werden, haben zwei Freiheitsgrade, mit denen der Laser horizontal bewegt wird. Damit können keine Freiformflächen realisiert werden. Die Gegenstände, die hergestellt werden, sind also beschränkt, da sie entweder flach sind, oder aus Flächen zusammen gesetzt werden. Eine runde Kugel aus einem Stück kann beispielsweise nicht hergestellt werden. Es gibt jedoch Ansätze⁹ Rundungen aus mehreren Teilen zusammen zu setzen. Im Gegensatz zu 3D-Druckern können Lasercutter jedoch verschiedene Materialien wie Holz, Papier und Acryl schneiden und ihre Genauigkeit ist bereits ausreichend.

In dem Workshop FabFM untersuchten Mellis und Buechley den Bau eines Radios mit Hilfe eines Lasercutters. Es waren nur wenige tragende Elemente vorgegeben, um die Kreativität

⁹<http://fablab.waag.org/archive/creating-3d-round-object-2d-laser-cutter>

möglichst wenig einzuschränken. Veränderungen wurden in einem breiten Bereich vorgenommen: freies Gestalten der Knöpfe zur Lautstärke und Senderwahl, Wahl anderer Materialien und Farbe, aber auch das Verhalten durch abgeänderte Logik wurde angepasst. Letzteres geschah durch Anpassen des Codes, bei dem einige Hilfestellung nötig war.

Daher haben Mellis und Buechley, im Gegensatz zum Workshop von FabFM, eine Möglichkeit gesucht, bei der Teilnehmer keine Vorkenntnisse benötigten [MB12]. Der Bau eines Lautsprechers konnte durch einfache Mittel realisiert werden, da die Verbindung der elektronischen Bauteile lediglich durch Löten realisiert wurde, aber keine Programmierkenntnisse notwendig waren. Die äußere Form wurde durch die Konstruktion und individuelle Gestaltung der Teilnehmer bestimmt.

Eine sehr interessante Idee kann man in SketchChair [SLMI11] sehen, bei dem Saul et al. eine Software entwickelt haben, mit der man Stühle entwerfen und herstellen kann. Im Vergleich zu herkömmlichen CAD-Programmen wurde die Funktionalität und damit auch die Komplexität reduziert und die Software dem Kontext angepasst. Das Entwerfen eines Stuhles erfolgt durch das Zeichnen einer zweidimensionalen Form, der Seitenansicht des Stuhls. Anschließend wird die dritte Dimension nach hinten projiziert, die in anpassbare Scheiben unterteilt wurde. Da der Stuhl mit einem Lasercutter hergestellt wurde, setzt sich die Struktur ausschließlich aus flachen Holzplatten zusammen. Um den Benutzer beim Entwerfen möglichst gut zu unterstützen, wurden verschiedene Hilfsmittel angeboten. Ein Messwerkzeug hat das reale Maß angegeben und mit einem speziellen Werkzeug für die Gestaltung der Beine, wurde das Ende der Füße automatisch parallel zum Boden ausgerichtet. Die Ergonomie des Stuhls konnte durch eine virtuelle Figur getestet werden, deren Alter und Größe anpassbar sind. Außerdem wurde die Statik überprüft, um das Umkippen des Stuhls zu verhindern.

Neben der Einschränkung von zwei Freiheitsgraden des Lasercutters, müssen die Stärke des Materials, aber auch Rauch und Geruchsentwicklung berücksichtigt werden. Um eine adäquate Absaugvorrichtung für den Heimbereich anzubieten, müssen die Hersteller ihre bisherigen Lösungen weiter verbessern. Mit der Geräuscentwicklung heutiger Lasercutter wird er auch zukünftig keinen Platz auf dem Schreibtisch im Privathaushalt finden, aber durch den immer kleiner werdenden Platzbedarf ist die Unterbringung im Keller oder Hobbyraum durchaus realistisch.

2.3 Besonderheiten beim praktischen Arbeiten mit Lasercuttern

Für die praktische Umsetzung von Modellen mit Lasercuttern gibt es bereits viele Informationen online zu finden. Hier wird ein kurzen Einstieg präsentiert und auf hilfreiche Quellen verwiesen. Alle eigenen Versuchsreihen und Ergebnisse, welche die Vorarbeit für dieses Projekt darstellen, können im Kapitel Implementierung gefunden werden.

Im Buch von Steen und Mazumder sind viele nützliche Informationen zu finden, die die physikalischen Grundlagen, den Aufbau eines Lasers und verschiedene Einsatzgebiete beschreiben [SM10a]. Zur Theorie des Schnittprozesses eines Lasercutters kann bei Cenna und Mathew nachgeschlagen werden [CM02].

Für einen schnellen Einstieg empfehlen sich die Seiten von SpikenzieLabs¹⁰ und Fabulous St. Pauli¹¹, die sowohl Tipps für die Arbeit bei der Erstellung des Grafikmodells geben als auch auf einige Besonderheiten hinweisen. Unter anderem muss man die Schnittkantenbreite des Lasers berücksichtigen, um die richtigen Nettomaße des Werkstücks zu erreichen oder darauf achten, welche Materialien zum Kleben von Acrylglas geeignet sind.

Entwerfen eines ersten Prototyps

Um die Vorlage der Prototypen zu erstellen kann ein beliebiger Vektorgrafikeditor, der SVG-Dateien als Ausgabeformat unterstützt, verwendet werden. Hilfreich sind Features für die Verwendung von Mustern, so dass beispielsweise eine Fingerverbindung beliebig oft an einem Pfad entlang eingefügt werden kann. Für die anschließende Übermittlung an den Lasercutter hat sich VisiCut¹² als komfortables Tool präsentiert. VisiCut speichert dabei die Schnittparameter des Lasers, abhängig von Material und Materialstärke. Praktisch ist die Möglichkeit, die importierten SVG-Grafiken auf dem Arbeitsbereich des Lasers verschieben zu können.

Geeignete Materialien

Eine Liste über geeignete und verbotene Materialien wird in der Regel vom Hersteller des Lasercutters mitgegeben. Darüber hinaus kann man verschiedene Erfahrungswerte bezüglich der Verarbeitung von Materialien finden. Als grundlegende Regel gilt, dass für heutige Lasercutter im Heimbereich keine Chlorverbindungen im Material enthalten sein dürfen, da keine ausreichende Filteranlagen für die Abluft integriert sind und die entstehenden Gase gesundheitsschädlich sein können. Materialien wie Glas und Metall können ebenfalls nicht geschnitten werden, da die Ausgangsleistung der für den Privatgebrauch erhältlichen Lasercutter zu gering ist. Die Parametrisierung muss für jeden Hersteller ermittelt werden, um einen sauberen Schnitt zu erhalten. Ist dies nicht der Fall, können sich Flammen bilden oder das Material ist nicht vollständig durchtrennt.

Aber auch innerhalb der Materialien, die bedenkenlos geschnitten werden können, gibt es erhebliche Unterschiede. Die Kunststoffe High Density Polyethylen (HDPE), Polystyren (PS), Polyethylen (PE), Polypropylen (PP) und Polyamid (PA) schmelzen stark und sind für die Verarbeitung weniger geeignet¹³. Empfehlenswert dagegen ist Polymethylmethacrylat (PMMA), umgangssprachlich auch als Plexiglas oder Acrylglas bezeichnet, das in den Varianten GS

¹⁰http://spikenzielabs.com/SpikenzieLabs/Laser_How_To.html

¹¹<http://www.fablab-hamburg.org/2011/12/27/howto-lasercut/>

¹²<http://visicut.org/>

¹³Verschiedene Autoren berichten über ihre Erfahrung mit dem Einsatz von Kunststoff für Lasercutter:

<http://www.fablab-hamburg.org/2011/12/27/howto-lasercut/>

<http://www.pololu.com/docs/0J24/3>

<http://wiki.fablab-muenchen.de/display/WIKI/Lasercutter+Wissen>

http://atxhackerspace.org/wiki/Laser_Cutter_Materials

und XT hergestellt wird. Im Artikel des Hamburger FabLabs¹³ kann man einen deutlichen Vergleich zwischen PMMA und PE sehen und wird auf die unterschiedlichen Herstellungsverfahren von PMMA hingewiesen. Da die Toleranz bei XT-PMMA Platten geringer ausfällt, sollte man sie verwenden, wenn exakte Abmaße eine Rolle spielen. Eine gute Übersicht über Materialien und ihre Erfahrungswerte findet man unter ATX Hackerspace¹³.

Verbindungsmöglichkeiten mehrerer Teile

Die Verbindung von Elementen kann auf unterschiedliche Arten erfolgen. Dabei hängt die Wahl der Verbindungsart vom Material und von den zu verbindenden Elementen ab. Gros zeigt 50 Holzverbindungen¹⁴, die in Rahmen-, Brett- und Gestellverbindungen aufgeteilt sind. Für jede Verbindung liefert er eine Zeichnung, hebt ihre Vorteile hervor und gibt ein mögliches Anwendungsbeispiel. Dabei können auch 3D-Daten verwendet werden, um die Verbindung direkt weiterverarbeiten zu können. Die folgenden Beispiele zeigen dabei die Vielfalt der Verbindung, die sich in Einsatzbereich, Funktion, aber auch in ihrer Optik unterscheiden.

Alle Beispiele von Gros sind für CNC-Fräsen ausgelegt. Da CNC-Fräsen in drei Freiheitsgraden bearbeiten, können auch Abstufungen, Gehrungen und Fugen zur Unterstützung der Verbindung verwendet werden (siehe Abbildung 2.8). Um Verbindungen mit Lasercuttern

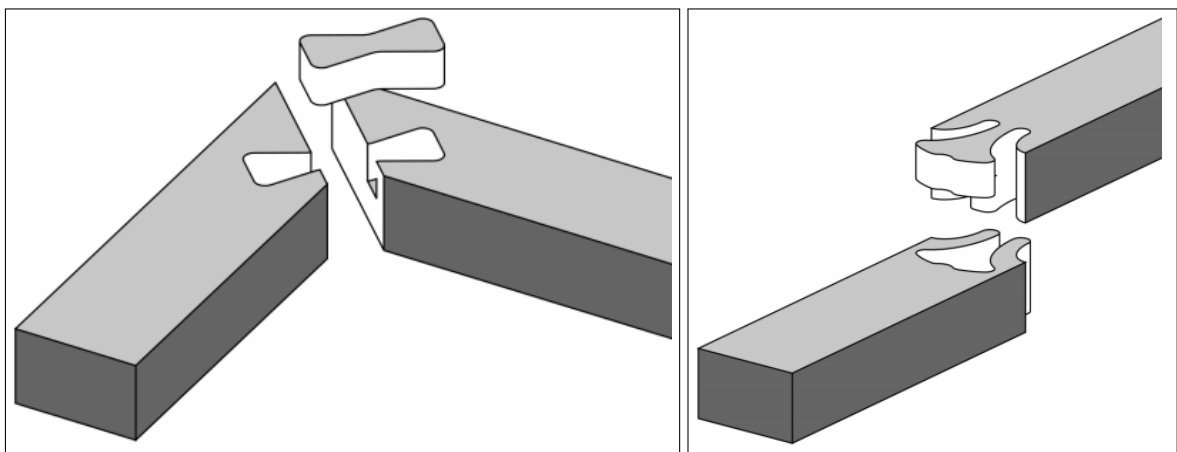


Abbildung 2.8: Zwei Rahmenverbindungen, links mit einer Feder („Eckgehrung mit Schwalbenschwanzriegel“), deren Spannkraft und Optik durch unterschiedliche Materialien verändert werden können. Der große Zapfen im rechten Bild ähnelt einem Ginkoblatt („Ginkoblatt mit Stummelzapfen“) und zeigt, dass bei gleichbleibender Funktion unterschiedliche Formen gewählt werden können. Quelle: <http://www.flexiblestream.org/Digital-Wood-Joints-001.php>

¹⁴<http://www.flexiblestream.org/Digital-Wood-Joints-001.php>

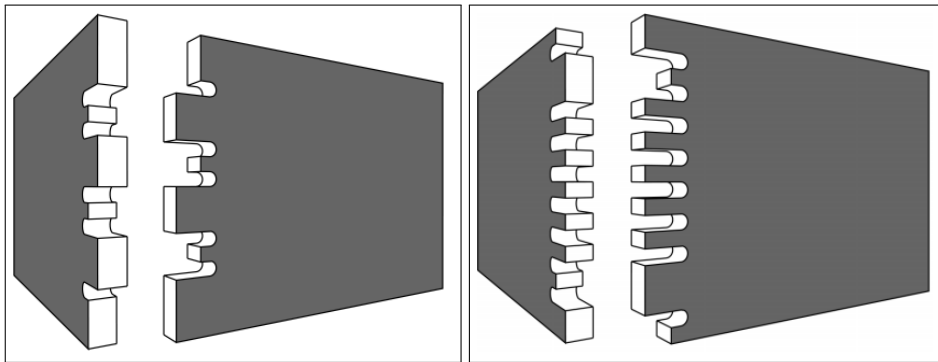


Abbildung 2.9: Hier sind zwei Verbundarten mit Verzinkung zu sehen. Die linke Variante („freigestellte Verzinkung“) ist etwas einfacher zu konstruieren, dafür bietet die rechte Variante („offene Fingerspitzen-Zinkung“) einen größeren Reibschluss, so dass eine stabilere Verbindung erreicht wird. Quelle: <http://www.flexiblestream.org/Digital-Wood-Joints-001.php>

herzustellen, müssen wir uns allerdings mit zwei Dimensionen begnügen¹⁵. Daher sind die Verbindungen in Abbildung 2.9 interessant. Da Lasercutter im Vergleich zu CNC-Fräsen sehr viel kleinere Kurveninnenradien schneiden können, kann die Nut fast¹⁶ mit 90° Innenwinkel angefertigt werden. Auf der anderen Seite sind die momentan für den Heimgebrauch erhältlichen Lasercutter in ihrer Ausgangsleistung stark beschränkt, so dass die maximale Materialstärke bei wenigen Millimetern liegt. Daher spielen Verbindungen mit Reibschluss eine große Rolle.

Auch Ragan behandelt das Thema der unterschiedlichen Radien und wählt einige Verbindungen aus, die sowohl für Lasercutter als auch für CNC-Fräsen Alternativen bieten¹⁷. Die praktischen Tipps bieten eine gute Ausgangsbasis, um sich mit den Möglichkeiten der Verbindungen vertraut zu machen. Unterteilt sind sie dabei je nach Anordnung der Verbindungselemente, so dass Kreuz-, T-, Eck- und Neigungs-Verbindungen beschrieben werden. Dabei werden verschiedene Eigenschaften aufgegriffen, die weitere Vorteile mit sich bringen. Verwendet man asymmetrische Verbindungen, wie in Abbildung 2.11 unten und in Abbildung 2.10 rechts, so wird bei der Montage eine Hilfestellung geboten: Entweder können die Verbindungen erst gar nicht falsch herum zusammengesteckt werden, oder sie fallen zumindest schnell auf, wenn ein Absatz im Kantenverlauf entsteht. Auch das praktische Prinzip der Rucksackschnalle hat Ragan verwendet, um einen Sicherungsmechanismus gegen selbständiges Öffnen einzubauen (siehe in Abbildung 2.11 rechts unten).

Für diese Arbeit sind besonders die Eckverbindungen (siehe Abbildung 2.10) interessant, da sie für einfache Körper wie Quader zum Einsatz kommen. T-Verbindungen (Abbildung 2.11)

¹⁵Über unterschiedliche Leistungswerte des Lasers kann die Tiefe zwar variiert werden, Genauigkeit und Geschwindigkeit sind dabei aber deutlich geringer.

¹⁶Da der Laserstrahl rund ist, kann keine perfekte Ecke erzielt werden.

¹⁷<http://blog.makezine.com/2012/04/13/cnc-panel-joinery-notebook/>

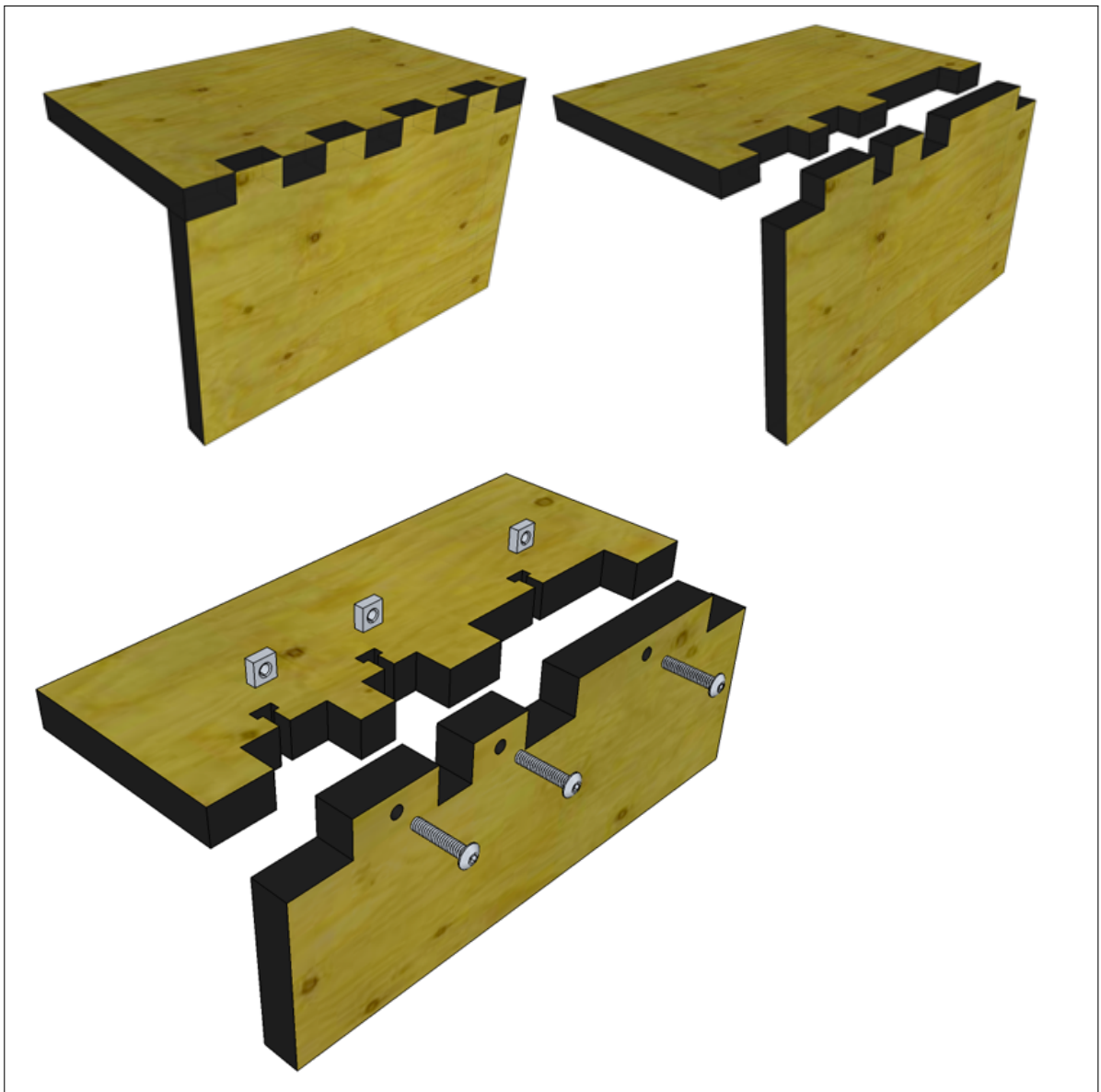


Abbildung 2.10: Eck-Verbindungen sind besonders für die Herstellung von Quadern geeignet. Die rechte Variante verwendete dabei asymmetrische Fingerverbindungen, womit verhindert wird, dass die Teile falsch herum montiert werden. Durch die Verwendung von Schrauben (unteres Bild) kann ein Formschluss erzielt werden, was die Stabilität der Verbindung erhöht. Dies ist jedoch erst ab einer gewissen Materialstärke realisierbar. Quelle: <http://blog.makezine.com/2012/04/13/cnc-panel-joinery-notebook/>

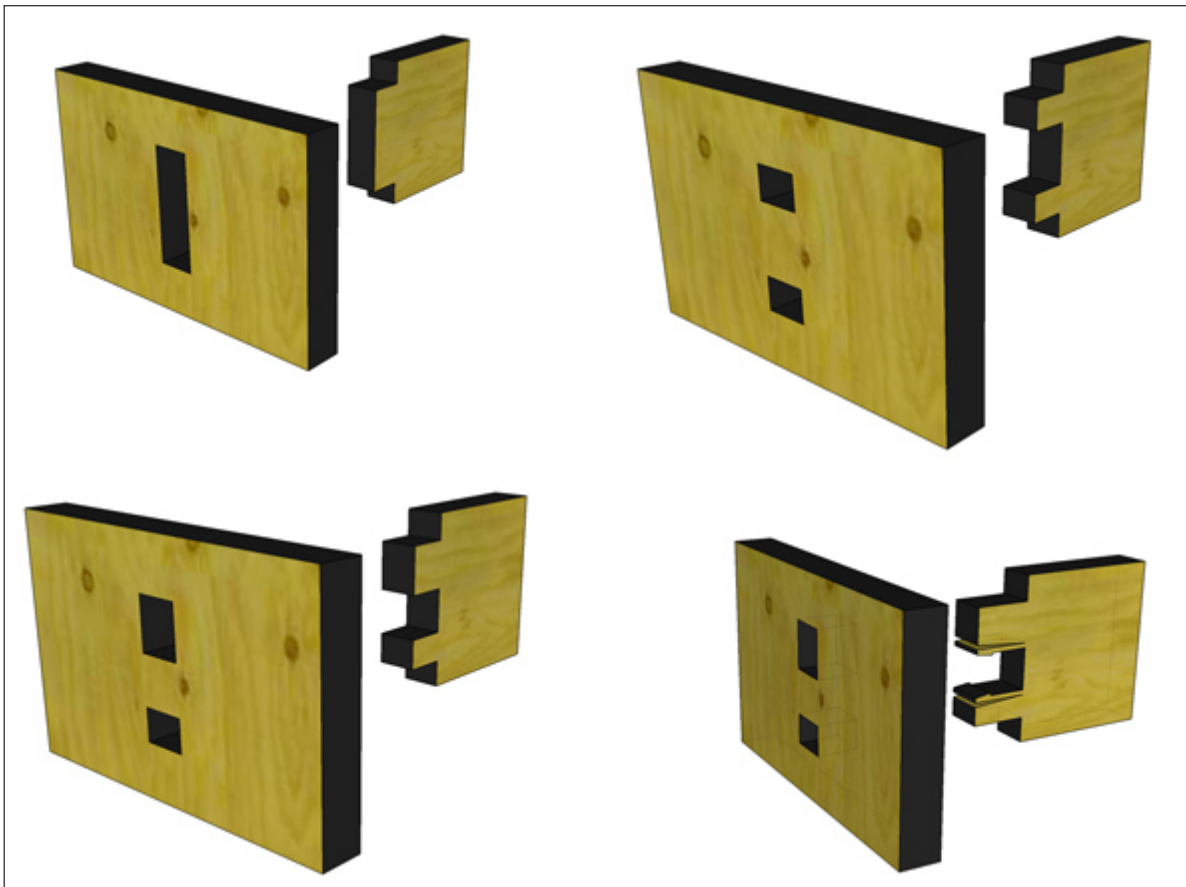


Abbildung 2.11: T-Verbindungen werden ebenfalls häufig eingesetzt und bieten die Möglichkeit Zwischenböden in Körper wie Quadern einzuziehen. Die beiden unteren Varianten sind wiederum asymmetrisch, die rechte davon hat einen Federmechanismus, der wie bei einer Rucksackschnalle funktioniert. Quelle: <http://blog.makezine.com/2012/04/13/cnc-panel-joinery-notebook/>

könnten zum Tragen kommen, wenn im Inneren eines Quaders eine weitere Unterteilung vorgenommen wird. Durch eine Kombination aus Fingerverbindung und Schrauben kann neben dem Reibschluss auch ein Formschluss erzielt werden, der die Verbindung deutlich verstärkt.

Der Autor von „How to Dove-Tail Joint a Laser Cut Enclosure + Lipo Charger“¹⁸ verwendet ebenfalls Schrauben um die Stabilität seines Gehäuses zu verbessern, allerdings verlaufen diese senkrecht durch das komplette Gehäuse und enden im Material der Bodenfläche. Verwendet man Schrauben, die ohne Vorbohren montiert werden können, so verjüngt sich diese an der Spitze, um das Eindringen in das Material zu erleichtern. Bei sehr dünnem

¹⁸<http://www.instructables.com/id/How-to-Dove-Tail-Joint-a-Laser-Cut-Enclosure-Lip/>

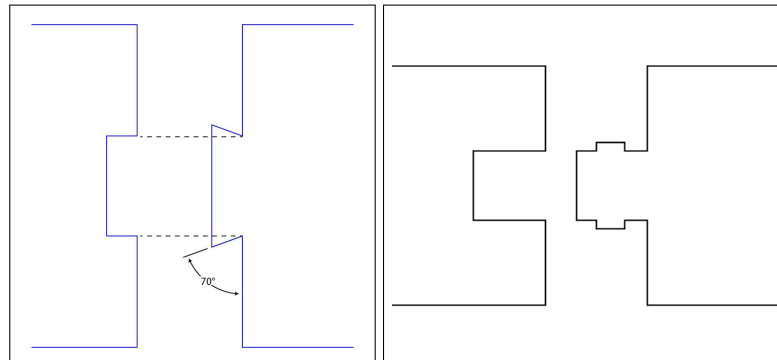


Abbildung 2.12: Um genügend Zusammenhalt einer Verbindung zu erreichen gibt es verschiedene Möglichkeiten. Der Schwalbenschwanz (links) bildet eine formschlüssige Verbindung. Durch Erhöhung des Reibschlusses wird der Zusammenhalt bei der rechten Variante erreicht. Quelle linkes Bild: <http://blog.ponoko.com/2010/06/17/how-to-make-snug-joints-in-acrylic/>

Material tritt aber der Fall ein, dass die Schraube noch nicht mit vollem Umfang in das Material eindringt, bevor sie droht auf der anderen Seite bereits wieder auszutreten. Der Formschluss ist also wesentlich geringer. Eine weitere Einschränkung dieser Art, ist die fixe Länge von Schrauben. Zwar gibt es Schrauben in vielen unterschiedlichen Längen, allerdings hängen die Maße des Gehäuses dann von Schraubenlänge und Materialstärke ab. Flexibler und stabiler scheint da der Ansatz mit Muttern zu arbeiten (siehe Abbildung 2.10).

Zwei Varianten der Fingerverbindung sind in Abbildung 2.12 zu sehen. Im linken Bild wird durch das Ändern des Winkels ein Formschluss erzielt, wodurch sich der Zusammenhalt der Verbindung verbessert. Eine andere Möglichkeit ist denkbar, abgeleitet von Emery¹⁹, indem die Haftreibung durch eine Nase am Zapfen erhöht wird.

Besonders für abgerundete Ecken ist eine weitere Verbindungsart interessant, die in Abbildung 2.13 und 3.3 auf Seite 36 (unten rechts) zu sehen ist. Hier wird Material quer zur Biegerichtung entfernt, um einen unterschiedlichen Innen- und Außenradius zu ermöglichen. Die ursprüngliche Idee dazu stammt von Snijlab²⁰, die mittlerweile in vielen Do-it-yourself-Projekten Anwendung findet. Diese Herstellung der Verbindung ist prädestiniert für den Lasercutter, da im Gegensatz zu CNC-Fräsen ein wesentlich kleinerer Abstand zweier Schnittfugen erzielt werden kann. Da kein einheitlicher Begriff dafür existiert, wird sie im Folgenden als *Akkordeonverbindung* bezeichnet. Gunn beschreibt in seinem Artikel²¹ das Vorgehen von einem einfachen Zickzack-Schnitt zu einer fortgeschrittenen, stabileren Version, wobei er auf auch auf die physikalischen Hintergründe eingeht. Kombiniert man nun die Akkordeon-Verbindung mit der Fingerverbindung, so ist die Konstruktion eines Zylinders möglich.

¹⁹<http://blog.ponoko.com/2010/06/17/how-to-make-snug-joints-in-acrylic/>

²⁰<https://www.snijlab.nl/nl>

²¹<http://blog.makezine.com/projects/creating-living-hinges/>



Abbildung 2.13: Durch das Entfernen von Material wird die Biegeeigenschaft ausgenutzt, um eine abgerundete Ecke zu ermöglichen. Quelle: <http://blog.makezine.com/projects/creating-living-hinges/>

2.4 Microsoft .NET Gadgeteer

Microsofts .NET Gadgeteer ist eine Plattform um verschiedene elektronische Geräte einfach zusammen zu bauen. Dabei gliedert sich die Idee in drei Teile. Zu erst wird die Hardware (siehe Abbildung 2.14), also Platinen, Sensoren, Aktoren etc. verbunden, dann kann die Logik in C# mit dem .NET-Framework angepasst werden. Der letzte Schritt ermöglicht es ein Gehäuse zu erstellen, in dem ein Plug-in für Solid Works, einem CAD-Programm, verwendet wird.

Es gibt bereits andere Plattformen für Bausätze elektronischer Komponenten wie beispielsweise Arduino, die auf dem Markt verbreitet sind. Sie sind günstiger, echtzeitfähig und können mit verschiedenen Betriebssystemen entwickelt werden. Als 2011 Microsoft die Gadgeteer-Plattform veröffentlichte, war die Grundlage zwar ähnlich, einige wichtige Unterschiede gibt es dennoch. Das Entwicklungsframework ist eine für Mikrocontroller angepasste Version des .NET-Frameworks (.NET Micro Framework). Damit ist das Betriebssystem auf Windows beschränkt, da keine Portierung auf Unix-Systeme existiert. Mit der Integration in die kostenlose Express-Version von Visual Studio ist aber eine Programmierumgebung geschaffen, die den Benutzer große Unterstützung bietet und viele Aufgaben selbstständig erledigt. Dazu gehören zum Beispiel die Verwaltung von Pins und Sockeln, das Entwerfen des Kabellayouts und das einfache Ansprechen der Hardware durch Abstrahierung in automatisch generierte Objekte und Eigenschaften.

Die Kosten sind zwar deutlich höher als bei einem vergleichbaren Arduino-Set, allerdings wird dieser Nachteil durch das einheitliche Steckersystem und die rasche Umsetzung eines Prototyps mehr als wett gemacht. Die Plattform eignet sich somit sowohl für den Einsatz in Schulen als auch für interessierte Hobby-Bastler. Außerdem gibt es viele Erweiterungsmög-

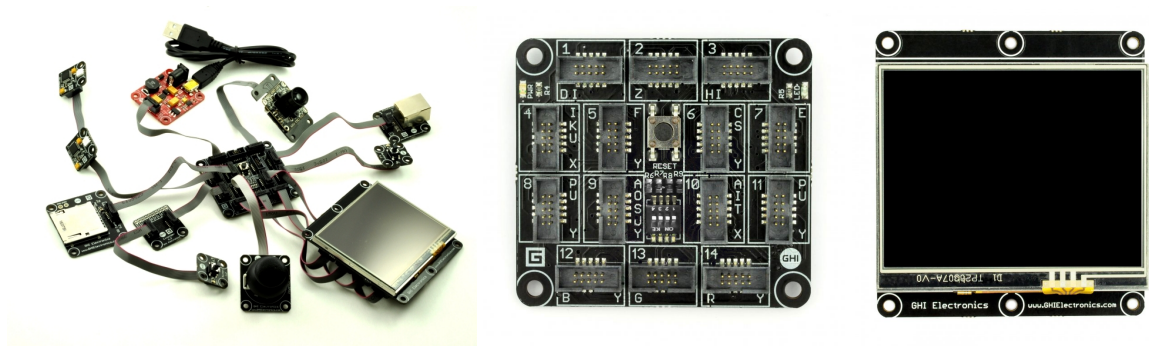


Abbildung 2.14: Das FEZ Spider Starter Kit (links) der Firma GHI Electronics liefert bereits einige Komponenten mit. Unter anderem ist das Mainboard (Mitte) mit einem 32 Bit, 72 MHz, ARM7 Prozessor und 16 MB Arbeitsspeicher ausgestattet. Beim Erstellen von Gehäusen sind besonders externe Geräte wie das Touchdisplay (rechts) zu berücksichtigen, damit die Aussparungen und die passenden Haltevorrichtungen ausgeschnitten werden. Quelle: <http://www.ghielectronics.com>

lichkeiten wie den Anschluss von USB-Ports, W-Lan Empfänger, Bluetooth, Infrarotsensoren, Kameras, Buttons und Touchdisplays, um die Verwendung vielseitig einsetzbar zu machen. Hinzukommt, dass sowohl das .NET Micro Framework als auch die Gadgeteer Bibliotheken quelloffenen Lizenzen unterliegen und bei Bedarf auf der Homepage von Microsoft eingesehen werden können.

Da die Umsetzung der Platinen mittels Steckverbindungen schnell abgeschlossen ist, fließt ein Großteil der aufgewendeten Zeit in die Programmierung. Durch eine verbesserte Hilfestellung bei der Suche nach Zugriff auf Hardware-Eigenschaften und der Entwicklung weiterer Standardfälle kann die Einstiegshürde diesbezüglich jedoch noch gesenkt werden. Schließlich bleibt die Aufgabe übrig, ein geeignetes Gehäuse für die Komponenten zu bauen. Microsoft setzt hier auf ein Plugin für Solid Works und kommt damit den ambitionierten Bastlern entgegen, in dem bereits alle Komponenten als 3D-Modell mitgeliefert werden. Auch ausgefallene und komplexe Gehäuse können so entworfen und mit einem 3D-Drucker oder einem Lasercutter hergestellt werden. Allerdings setzt dies Vorkenntnisse oder eine längere Einarbeitungszeit mit dem CAD-Programm voraus. Da die meisten Gadgeteer-Prototypen recht schnell entwickelt werden, liegt es nahe, auch die Erstellung eines Gehäuses in kurzer Zeit erstellen zu wollen. Oft sind einfache Formen ausreichend und erst bei mittelfristig verwendeten Prototypen lohnt sich die Verfeinerung der Ausgestaltung. Weitere Informationen zu Gadgeteer können bei [VSH⁺12] und unter <http://www.netmf.com/gadgeteer/> gefunden werden. Ein ausführlicher Vergleich zwischen Arduino und Gadgeteer ist auf Linders Blog²² nachzulesen.

²²<http://blog.timmlinder.com/2012/06/comparison-gadgeteer-netmf-arduino/>

3 Verwandte Arbeiten

Dieses Kapitel bietet eine Übersicht über verfasste Arbeiten, die thematisch angrenzen an die Idee der hier präsentierten Arbeit, zeigt aber auch die Abgrenzung zu ihnen auf. Das Kapitel gliedert sich dabei wie folgt: Zunächst wird die Wahl der Interaktionsart erläutert, anschließend werden besondere Einsatz- und Eingabemöglichkeiten bei der Bedienung eines Lasercutters vorgestellt. Bevor im letzten Abschnitt das Kapitel mit einem kurzen Ausflug über Material, Form und Ästhetik abgeschlossen wird, wird das Konzept des Tangible User Interfaces erläutert.

3.1 Wahl der Interaktionsart

Um auch Personen im Heimbereich ohne technischen Hintergrund angemessen ansprechen zu können, muss die Komplexität der Software so gering wie möglich gehalten werden. Die Komplexität hängt sowohl vom Funktionsumfang als auch von den Ein- und Ausgabemethoden ab. Auf Letzteres wird in diesem Abschnitt näher eingegangen, da abhängig von Eingabegeräten wie Maus und Tastatur, Ausgabegeräten wie einem Monitor, aber auch von der Entscheidung ob die Bedienung zweidimensional oder dreidimensional entwickelt wird, die Einstiegshürde beeinflusst wird. Die folgenden Aspekte legen die Entscheidungsgrundlage für die in diesem Projekt gewählten Interaktionsarten.

Alternative Eingabearten

Bei der computergestützten Entwicklung eines Modells kommen unterschiedliche Eingaben zum Einsatz. Wurde schon in den 50er Jahren die Technik eines Leuchstifts entdeckt und für die CAD-Anwendung Sketchpad in den 60ern verwendet, so ist heute die Kombination aus Tastatur und Maus am beliebtesten. Gerade durch die steigende Beliebtheit der Touchscreens ist aber die Stifteingabe wiederentdeckt worden [DYZA05]. Für Grafiker steht seit einigen Jahren die Möglichkeit per Stift ihre Zeichnungen und Skizzen automatisch zu digitalisieren, aber besonders mit der Einführung des Tablet-PCs wurden die Einsatzmöglichkeiten deutlich erweitert.

Dickinson et al. geben eine kurze Übersicht von Eingabegeräten im CAD-Bereich [DYZA05]. Unter anderem wird dabei eine Maus mit sechs Freiheitsgraden vorgestellt und auf das Potenzial der Spracheingabe eingegangen. Die Möglichkeiten hängen dabei nicht nur von der Eingabe, sondern auch vom Zusammenspiel der Ein- und Ausgabe ab. Wird beispielsweise

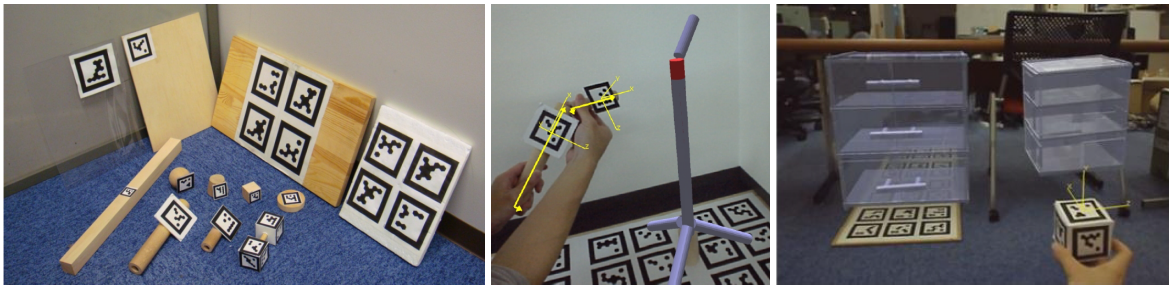


Abbildung 3.1: Die Werkzeuge, mit denen die Eingabe vorgenommen wird, sind selbst hergestellt, mit Materialien aus einem Baumarkt (linkes Bild). Die in allen drei Bildern zu sehenden Bodenplatten dienen zur räumlichen Orientierung bei der Eingabe. Durch die Verwendung zweier Holzstäbe kann beispielsweise der Winkel im Modell angepasst werden (mittleres Bild). Sowohl das mittlere als auch das rechte Bild zeigen die Ausgabe der Augmented Reality, wobei im rechten Bild die Rotation des Gegenstandes mit Hilfe des Würfels manipuliert werden kann. Quelle: [LHO⁺12]

ein Helm verwendet, mit dem stereoskopisches Sehen realisiert wird, so können die CAD-Objekte in allen drei Dimensionen abgebildet werden. Allerdings ist das System auf einen Benutzer beschränkt.

Ähnlich verhält es sich für Methoden der *Augmented Reality*. Auch wenn der Vorteil der bereits existierenden räumlichen Umgebung bei der Orientierung hilft, so bleibt es schwierig die Eingabe intuitiv und wenig ermüdend zu realisieren [LHO⁺12]. Die dort gestellte Bedingung, alle verwendeten Werkzeuge aus einem beliebigen Baumarkt erhalten beziehungsweise herstellen zu können, zeigt aber die Berücksichtigung der praxisnahen Umsetzung für den Privatanwender. Runde Holzstäbe, eckige Holzblöcke und Holzplatten unterschiedlicher Dicke und Größe werden mit Markern versehen und symbolisieren verschiedene Eingabemodi. Beispielsweise kann durch die Kombination von zwei runden Holzstäben der Winkel innerhalb eines grafischen Modells bestimmt werden (siehe Abbildung 3.1). Die mit einer Brille realisierte Augmented Reality projiziert das Modell direkt in den Raum. Durch das Ergreifen der Holzgegenstände wird ein haptisches Feedback über den gerade aktiven Modus geliefert. Das ausgefallene Eingabesystem bedarf aber einiger Übung und weist noch eine zu hohe Fehlerrate auf.

Ebenfalls durch physikalische Repräsentation versuchen Leduc-Mills und Eisenberg Schulkindern die Möglichkeit zu geben einfache geometrische Figuren zu definieren [LME11]. Das System basiert dabei auf einer Kiste, die mit 4 x 4 Löchern versehen ist. In ein Loch kann eine Lichtleiste gesteckt werden, so dass sich ein Koordinatensystem ergibt: Die Position in der Kiste bestimmt zwei Koordinaten, die Lichter, die in unterschiedlicher Höhe an der Leiste angebracht und angeschaltet werden können, bestimmen die dritte Koordinate. Auf einem Computer werden die Körper der so definierten Eckpunkte dargestellt und geben Aufschluss, ob sich die Vorstellung mit dem aktuell gebauten Körper deckt. Die Methode kann für spielerische Zwecke und zur Unterstützung von Vorstellung und Umsetzung dreidimensionaler

Körper in Schulen eingesetzt werden. Für einen Erwachsenen, der eine schnelle und etwas flexiblere Lösung sucht, ist sie im Alltag jedoch ungeeignet.

Auch die Freihandgesten als Eingabe, wie bei Willis et al. [WLMI10] und Sketch Furniture¹ verwendet, haben bei der präzisen Umsetzung von Figuren Probleme. Für kreative und künstlerisch freie Gestaltung haben sie zwar durchaus neue Möglichkeiten aufgezeigt, bei der Definition eines exakten, aber geometrisch einfachen Gegenstands, wie der eines Würfels, überwiegen aber das beschränkte Feedback, die auf Dauer ermüdende Bewegung und die benötigte Übung mit dem System. Außerdem erwähnen Willis et al., dass die Benutzer die zweidimensionale Eingabe bereits gewohnt sind und sich an die neu verfügbare Dimension erst gewöhnen müssen.

Neue Möglichkeiten zum Feedback bei einer Interaktion bietet die Idee das Smartphone als Touch-Instrument zu verwenden [SCRG10, SSRG12, WS07]. Durch Vibration, Ton oder visuellem Feedback auf dem Display können selten beachtete Ausgabekanäle verwendet werden. Allerdings sprechen die Genauigkeit, die Handlichkeit, und die Voraussetzung des Besitzes eines Smartphones gegen diese Verwendung in der hier vorgestellten Arbeit.

Eingabeart – weitere Faktoren der Komplexität

Um Komplexität bei der Eingabe von Modellen zu verringern, verfolgen Leduc-Mills und Eisenberg [LGHH08] und Saul et al. [SXG10] den Ansatz der angeleiteten Eingabe. Die bei Saul et al. vorgestellte Software soll ausgewogen sein, um einerseits die Einstiegshürde zu senken, andererseits die Kreativität nicht zu stark einzuschränken: Zur Erstellung von Papiergegenständen werden evolutionäre Algorithmen verwendet, die die Auswahl des Benutzers als Basis für weitere Evolutionen berücksichtigen. Der Designer wird so in einen interaktiven Prozess eingebunden, während gleichzeitig garantiert wird, dass die Modelle produzierbar und stabil sind, was bei Papier besonders heikel ist. Die angeleitete Eingabe wird im Konzeptkapitel (siehe Seite 45) nochmals aufgegriffen.

Ein weiterer Faktor, der die Komplexität wesentlich beeinflusst, ist die Entscheidung zwischen zwei- und dreidimensionalem Modell. Bei einer Abbildung eines dreidimensionalen Modells auf ein zweidimensionales Ausgabegerät, wie das typischerweise bei einem CAD-Programm der Fall ist, muss bereits die Wahl der Ansicht und deren Steuerung genau überlegt sein [SSB08]. Die eingangs erwähnte Maus mit sechs Freiheitsgraden, ermöglicht zwar das Rotieren eines Gegenstandes um die gewünschte Achse mit nur einer Bewegung, die Eingabe und das nötige Verständnis des Benutzers fordern allerdings längere Einarbeitungszeit. Das Vergrößern eines Objektes durch Ziehen mit dem Mauszeiger, verursacht bereits unterschiedliche kognitive Auslastung, je nachdem, ob es sich dabei um ein zwei- oder um ein dreidimensionales Objekt handelt [DYZA05].

Wie bei Willis et al. zu sehen, ist es bei Echtzeitsteuerung essentiell, dass die Eingabe schnell genug umgesetzt wird [WXW⁺11]. In ihrer Arbeit testen sie unter anderem das Steuern einer

¹<http://www.designfront.org/category.php?id=81&product=191>

CNC-Fräse mittels Handgestik. Die Handbewegungen, die auf einem Touchpad ausgeführt werden, werden dabei unmittelbar in Maschinenoperationen umgesetzt. Der Benutzer kann dabei das bearbeitete Objekt jederzeit vor sich sehen. Da die Schnittgeschwindigkeit der Fräse jedoch zu langsam für die Benutzereingabe ist, kommt es zu Verzögerungen, die das Ergebnis stark beeinflussen. Außerdem sind nur beschränkte Korrekturmaßnahmen (wie das Rückwärtsfahren) des Benutzers möglich, so dass die Materialverschwendung entsprechend ansteigt.

3.1.1 Stift- und Touch-Eingabe

Bevor die heute weit verbreitete Touch-Eingabe mittels Finger oder Stift genauer untersucht wird, sollte darauf hingewiesen werden, dass die Genauigkeit von der verwendeten Touch-Technologie abhängt [PWS88]. Einige Vergleiche von Touch-Eingabe mit Fingern und Mauseingaben kommen zu unterschiedlichen Ergebnissen: Die Präzision von Touch-Eingaben per Hand kommen nicht an die Eingaben per Maus heran [RLZK13]. Albert zeigt ebenfalls, dass Touch-Eingaben bei der Genauigkeit am schlechtesten abschneiden, aber dafür die schnellste Eingabe ermöglicht [Alb82]. Auch Muller kommt zu dem Ergebnis, dass Fingereingabe bezüglich der Geschwindigkeit besser sein kann als Mauseingaben, da durch die Verwendung beider Hände die jeweils Nähere am zu treffenden Objekt verwendet werden kann [Mul08].

Für die Eingabe per Hand kommen zwei weitere Aspekte hinzu, die die Eingabe erschweren. Es handelt sich dabei um das sogenannte *Fat Finger Problem*. Zum einen ist die Auflagefläche eines Fingers deutlich größer als der Cursor einer Maus oder die Auflagefläche eines Stifts. Somit ist es schwierig Bereiche auszuwählen, die schmalere sind als der dafür verwendete Finger [AZ03]. Zum anderen verdeckt der Finger den darunter liegenden Bildschirmbereich. Es gibt jedoch bereits mehrere Verbesserungsvorschläge mit denen die Genauigkeit deutlich erhöht werden kann. Vogel und Baudisch schlagen vor, für die Auswahl eines Objektes mit dem interagiert werden soll, einen Rahmen anzuzeigen, der etwas nach oben und links verschoben ist und nicht vom Finger verdeckt wird [VB07]. Eine andere Möglichkeit wird von Lee beschrieben [Lee10]. Er stellt fest, ob mehrere Objekte gleichzeitig getroffen wurden und vergrößert die getroffenen Objekte anschließend. Nun kann der Benutzer seine Eingabe präzisieren, da die Trefferfläche erhöht und die Unterscheidung erleichtert wird. Einen vielversprechenden Ansatz zeigen Su et al. mit ihrem *NailDisplay* [SCW⁺13]. Dabei wird der Daumen mit einem Ring ausgestattet, der über ein Display auf dem Fingernagel verfügt. Bedient der Benutzer eine Anwendung, so wird auf dem Nageldisplay die unter dem Daumen liegende, verdeckte Fläche erneut angezeigt. Damit bekommt der Benutzer unmittelbares Feedback über das momentan ausgewählte Element und seinen Eingabepunkt. Interessante Ergebnisse liefert auch die Arbeit von Holz und Baudisch, bei der das Fat-Finger-Problem näher beleuchtet wird [HB10]. Die Autoren stellen fest, dass die Eingabeungenauigkeit nicht direkt von dem zu dicken Finger abhängt, sondern der Punkt der Eingabe korrekt angepasst werden muss. In ihrem Experiment werden Winkel und Neigung des Fingers bei der Eingabe berücksichtigt und ein benutzerspezifischer Eingabepunkt ermittelt. Durch eine kurze Kalibrierungsphase kann damit die Genauigkeit nahezu verdoppelt werden.

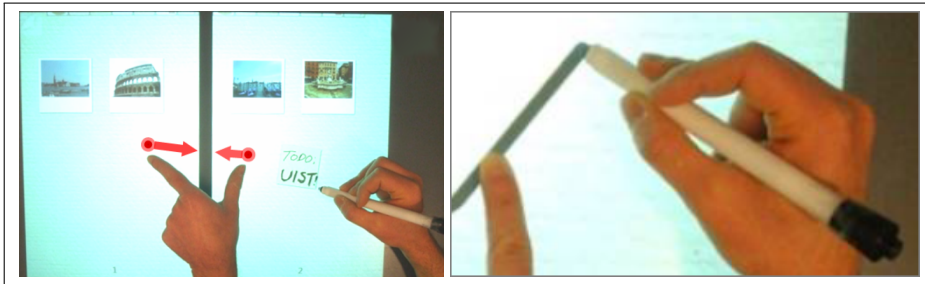


Abbildung 3.2: Durch die Kombination aus Finger und Stifteingabe werden neue Einsatzmöglichkeiten geschaffen. Links wird das verwendete Paradigma gezeigt: Fingereingabe entspricht Manipulation, Stifteingabe Schreiben und Zeichnen. Rechts wird durch die Kombination von beidem eine gerade Linie definiert. Wird der Finger weggenommen, wird das gezeichnete Teilstück durch Kurvenglättung dem bisher gezeichneten Verlauf angepasst. Quelle: [HYP⁺10]

Durch Multitouch-Gesten oder die Unterscheidung, mit welchem Teil des Fingers die Eingabe gemacht wird [HSH11], bietet die Eingabe per Hand trotz ihrer geringeren Präzision gute Eingabemöglichkeiten. Für Programme, bei denen die Konstruktion durch Zeichnen wichtig ist, können somit unterschiedliche Eingabearten realisiert werden, ohne explizit den Modus wechseln zu müssen.

Die Stifteingabe über Touch-Technologie ist im Gegensatz zur Fingereingabe bei Genauigkeit und die Geschwindigkeit mit der der Maus vergleichbar [ML89]. Da Menschen seit langer Zeit an die Verwendung von Stiften gewohnt sind, bietet diese Möglichkeit eine natürliche Eingabe bei präziser Steuerung und sehr geringer Verdeckung im Vergleich zur Fingereingabe. Davis et al. haben für die Prototypentwicklung speziell die Eigenschaften der Stifteingabe berücksichtigt [DSSL07], so dass eine schnelle Verifizierung möglich ist. Allerdings sind bei der Verwendung eines Stifts als Eingabe einige Nachteile zu berücksichtigen [HYP⁺10]. Um volle Bewegungsfreiheit zu ermöglichen, sind die Stifte in der Regel nicht durch eine Schnur mit dem Touchscreen verbunden. Damit besteht aber die Gefahr, die Eingabe durch Fallenlassen zu verzögern oder den Stift zu verlieren. Legt der Benutzer den Stift auf dem berührungssensitiven Bildschirm ab, so kann es dabei zu Fehleingaben kommen [ZBAK10].

Durch die Kombination von Stift und Finger als Touch-Eingabe sind neue Interaktionsmöglichkeiten geschaffen worden (siehe Abbildung 3.2). Hinckley et al. zeigen, dass unterschiedliche Eingaben, je nach Kombination von Finger(n) und Stift, Aktionen auslösen können ohne einen Modus explizit wechseln zu müssen [HYP⁺10]. Sie legen dabei das Paradigma zugrunde, dass der Stift für Schreib- und Zeicheneingaben, die Fingern für die Objektmanipulation verwendet werden.

3.2 Besondere Einsatz- und Eingabemöglichkeiten für Lasercutter

In diesem Abschnitt werden einige Alternativen zu CAD-Programmen vorgestellt, die speziell für Herstellung von Gegenständen mit einem Lasercutter entworfen wurden. Müller et al. zeigen einen Ansatz, bei dem der Benutzer mit einem Laserpointer direkt auf das Werkstück zeigt, was von einer Kamera erfasst, interpretiert und an den Lasercutter geschickt wird [MLB12] (siehe Abbildung 3.3). Durch die Idee, den Benutzer ohne Bildschirm interagieren zu lassen, kann er das Ergebnis sofort überprüfen und sein Gedankengang wird nicht unterbrochen. Da die Geschwindigkeit des Lasers jedoch noch gering ist und damit die Bearbeitungsdauer steigt, kann das Potenzial noch nicht ausgereizt werden. Durch die direkte Eingabe verliert man auch einige Vorteile des virtuellen Modells. Die Korrektur von Fehlern und das Betrachten des Gesamtbilds sind schwer beziehungsweise gar nicht möglich.

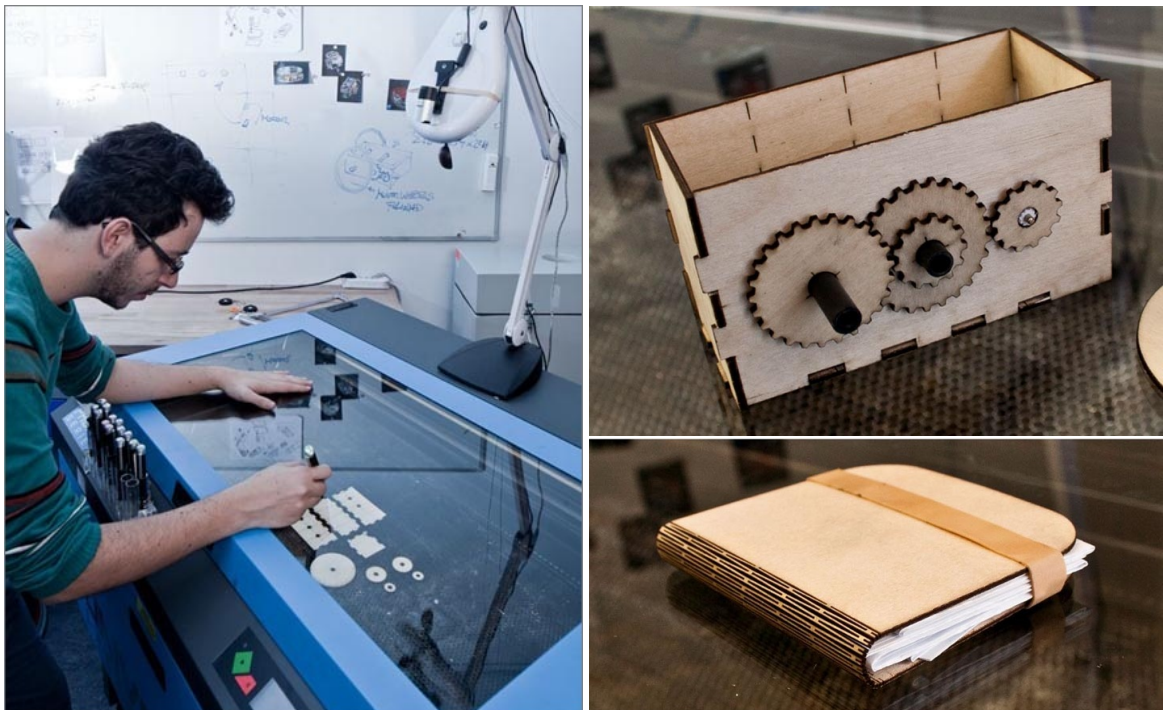


Abbildung 3.3: Links ist ein typischer Lasercutter zu sehen, dessen Arbeitsbereich durch eine Glasscheibe getrennt ist. Die Eingabe, die normalerweise durch das vorherige Bearbeiten eines Modells oder einer Grafik am PC erfolgt, wird hier direkt mit einem Laserpointer vorgenommen, dessen Punkt auf dem Material von einer Kamera (oben rechts in Bild) detektiert und als Eingabe für den Lasercutter verwendet wird. Auf der rechten Seite sind eine Box mit Zahnrädern sowie ein Einband für ein Buch zu sehen, die beide mit diesem Eingabeverfahren und dem abgebildeten Lasercutter hergestellt wurden. Quelle: [MLB12]

In einer weiteren Arbeit zeigen Müller et al. die unterschiedlichen Einsatzmöglichkeiten eines Lasercutters [MKB13]. Sie demonstrieren das Erstellen von dreidimensionalen Objekten, ohne nachträgliche Montage. Ermöglicht wird die dritte Dimension durch Biegen des Materials, was durch defokussieren des Lasers geschieht. Statt, wie beim Schneiden mit Lasercuttern üblich, den Fokus des Lasers auf die Materialoberfläche zu legen, wird dieser bewusst entfernt, um die Energie des Lasers auf eine größere Fläche zu verteilen. Durch mehrfache Wiederholungen wird das Material weich und lässt sich verformen. Das Werkstück, aus dem das Objekt erstellt wird, befindet sich auf Stelzen, so dass genügend Raum darunter ist, in den ein Teil durch die Schwerkraft hineinfallen kann (siehe Abbildung 3.4). Nach kurzer Zeit ist die durch Erhitzung gebogene Kante abgekühlt. Sie behält nun ihre Form und die Bearbeitung kann fortgesetzt werden. Sowohl das Schneiden als auch das Biegen können in einen gemeinsamen Prozess integriert werden, indem unterschiedliche Farbcodes verwendet werden. Die Erstellung des 3D-Objekts kann mit Microsoft Visio erfolgen oder durch eine Erweiterung der Laserpointer-Eingabe. Um mehr Flexibilität bei der Gestaltung von Kanten zu erhalten, wurde eine Spannvorrichtung mit einer Rotationsspindel integriert. Das Werkstück wird während der Bearbeitung gedreht, um einen Winkel größer 90° zu erreichen. Außerdem kann eine Halter-Konstruktion entworfen werden, deren Träger durch einer Kombination von Schnitt und Biegung entstehen.

Damit mit dieser Methode gearbeitet werden kann, ist es hilfreich einen Lasercutter zu haben, dessen Tisch sich computergesteuert in der Höhe verstellen lässt. Außerdem können mit der Methode von Müller et al. keine abgeschlossenen Formen erstellt werden, da das Verbinden von zwei getrennten Kanten nicht berücksichtigt wurde. Auch die vertikalen Verfahrenweite ist durch den Aufbau limitiert. Eine weitere Einschränkung besteht in der Materialwahl. Das Biegen kann nur mit solchen Materialien erreicht werden, die unter Einfluss von Energie verformbar werden (wie beispielsweise PMMA, siehe auch im Abschnitt 2.3 unter Materialien). Die herkömmliche Herstellung mit Lasercuttern, in der die Seitenteile durch Verbindungen miteinander verknüpft werden, setzt zwar nachträgliche Montage voraus, hat aber den Vorteil, dass Seitenteile ausgetauscht werden können, ohne das Werkstück komplett neu erstellen zu müssen.

Weichel et al. zeigen eine Alternative zu hoch komplexen CAD Programmen auf. Beispielhaft für die Erstellung von Gehäusen für ein Gadgeteer-Set (siehe Abschnitt 2.4) haben sie ein Tool entwickelt, das den Designprozess für Prototypen beschleunigen soll. Dreidimensionale Formen werden per Maus bearbeitet – wie bei herkömmlichen CAD-Programmen. Allerdings wird nach dem Konfigurieren und Verbinden des Prototyps die Liste der verwendeten Komponenten als Ausgangsbasis für den Entwurf des Gehäuses verwendet. Dabei wird unterschieden, ob eine Komponente intern oder extern platziert werden soll. Alle externen Komponenten stehen zur Auswahl und können auf einer Gehäuseseite platziert werden. Die Form des Gehäuses kann dabei iterativ verfeinert werden (siehe Abbildung 3.5). Die Größe des Gehäuses wird dabei automatisch angepasst, wenn eine Komponente größer ist als die Fläche auf der sie platziert wurde. Die Wahl der Joints wird ebenfalls automatisch gewählt. Je nach Länge der Kante kommt eine Kombination aus Finger- und Schraubenverbindung zum Einsatz. Im Vergleich zum Entwurf mit einem CAD-Programm ist die Anzahl der auszuführenden Schritte in etwa um den Faktor 10 verringert worden.

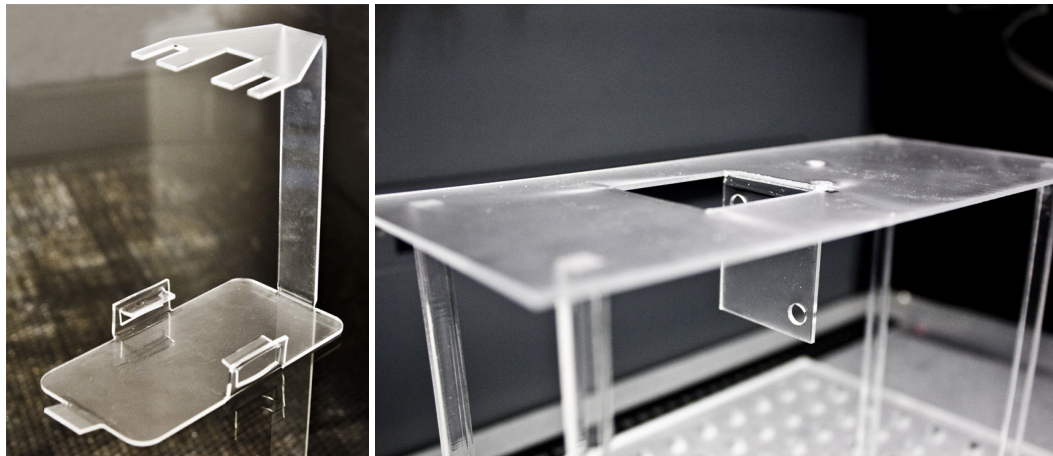


Abbildung 3.4: Müller et al. zeigen dreidimensionale Objekte, die mit einem Lasercutter hergestellt wurden, aber keine Montage benötigen (linkes Bild). Durch Defokussieren des Lasers können Flächen an Kanten gebogen werden. Durch den Raum unter dem Werkstück kann mit Hilfe der Schwerkraft die dritte Dimension ermöglicht werden (rechtes Bild). Quelle: <http://stefaniemueller.org/>

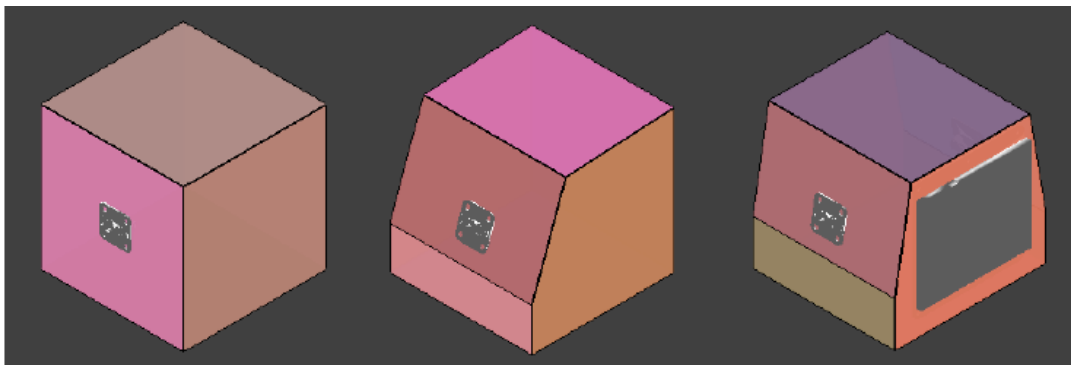


Abbildung 3.5: Iterative Gestaltung des Gehäuses für einen Prototypen Quelle: [WLG13]

Durch die Verwendung von Rotationen können Winkel zwischen den Außenseiten angepasst und das Gehäuse individuell gestaltet werden (siehe Abbildung 3.5). Auch wenn die Komplexität gegenüber CAD-Programmen reduziert ist, erfordern diese Möglichkeiten ein gewisses technisches Verständnis. Das Konzept, das im folgenden Kapitel vorgestellt wird, verzichtet bewusst auf diese Freiheit zu Gunsten einer niedrigeren Einstiegshürde. Außerdem kann durch den Einsatz von Akkordeonverbindungen (siehe Abschnitt 2.3 auf Seite 28) eine ästhetische Alternative zur Verbindung zweier Seiten angeboten werden.

3.3 Tangible User Interface

Viele Aufgaben, die früher mit physikalischen Gegenständen durchgeführt wurden, wie das Schreiben auf Papier oder das Rechnen mit einem Abakus, wurden mit der Einführung des PCs abgelöst. Durch die Vielseitigkeit ist er in vielen Dingen überlegen. Dennoch gingen bei der Verwendung des PCs mit Maus und Tastatur einige Eigenschaften wie die Haptik und physikalische Repräsentation verloren. Ishii und Ullmer führten 1997 den Begriff *Tangible User Interface* (TUI) ein, mit dem Ziel die Vorteile dieser verlorenen Eigenschaften zurückzugewinnen [IU97]. Tangibles sind physikalische Gegenstände, die als Eingabe dienen und digitale Informationen repräsentieren können. Beispielsweise demonstrieren Ishii und Ullmer den Einsatz von Tangibles bei einem interaktiven Lageplan. Mit einem durchsichtigen Zylinder kann die Ansicht gedreht werden und eine *aktive Linse* ermöglicht eine 3D-Ansicht der abgebildeten Gebäude. Neben den Elementen, die im direkten Fokus des Benutzers liegen, schlagen sie vor auch die Umgebung mit einzubeziehen. Beispielsweise könnte ein Modellauto, das als Tangible fungiert, in die Nähe eines Lautsprechers gehalten werden, um auditives Feedback über eine bestimmte Eigenschaft zu erhalten.

Die Vorteile von Tangibles werden besonders durch die Wahrnehmung mit allen Sinnen verdeutlicht. Die Tangibles können sich dabei in Form, Material und Volumen unterscheiden und bieten in jedem Fall haptisches und optisches Feedback. Durch die Reduktion des *kognitiven Workloads* [NLH⁺13] kann sich der Benutzer besser auf seine eigentliche Ziele konzentrieren, ohne die Abstraktion zwischen physikalischem und virtuellem Modell leisten zu müssen. Besonders durch die Kombination von TUIs mit Multitouch-Tischen können für die Interaktion alltägliche Gegenstände zur Eingabe für die virtuelle Steuerung verwendet werden [NLH⁺13].

Jordà et al. [JJG10] ordnen die Entwicklung von Tangibles in einen historischen Kontext ein, der bei der *direkten Eingabe* und der *Desktop Metapher* von Shneiderman beginnt. Sie sehen Tangibles ebenso wie Jacob et al. [JGH⁺08] als Teil der *Post-Window, Icon, Menu, Pointing Device*-Ära. Jacob et al. fassen dabei verschiedene Bereiche zur *Reality Based Interaction* zusammen, bei welcher der Fokus auf den engen Bezug der Benutzerschnittstelle zur realen Welt gelegt wird. Die Eigenschaften, die aus der Realität übertragen werden sollen, sind dabei in vier Bereiche unterteilt. Unter der *naïven Physik* versteht man das allgemeine Verständnis über die physikalische Welt. *Körperbewusstsein* ist das Bewusstsein und die Kontrolle über den eigenen Körper. *Umgebungsbewusstsein* ist die Wahrnehmung der Umgebung und die Manipulation von ihr sowie die Navigation und Verhandlung in ihr. Die letzte Eigenschaft ist das *soziale Bewusstsein*, welche die Wahrnehmung von Anderen und Interaktion mit ihnen berücksichtigt.

Neben der Untersuchung von passiven Tangibles gibt es einige Arbeiten, die Erweiterungen für aktive Tangibles aufweisen. Weiss et al. [WSJB10] und Nowacka et al. [NLH⁺13] zeigen unterschiedliche Ansätze dafür (siehe Abbildung 3.6). Bei Weiss et al. basiert die Bewegung auf der Änderung elektromagnetischer Felder. Unter dem Multitouch-Tisch sind viele kleine Magnete angebracht, mit denen die Tangibles verschoben und durch Induktion mit Strom versorgt werden können. Nowacka et al. hingegen verwenden aktive Elemente der Tangibles, die durch Vibrationsmotoren angetrieben und mit optischen Signalen gesteuert werden. Sie

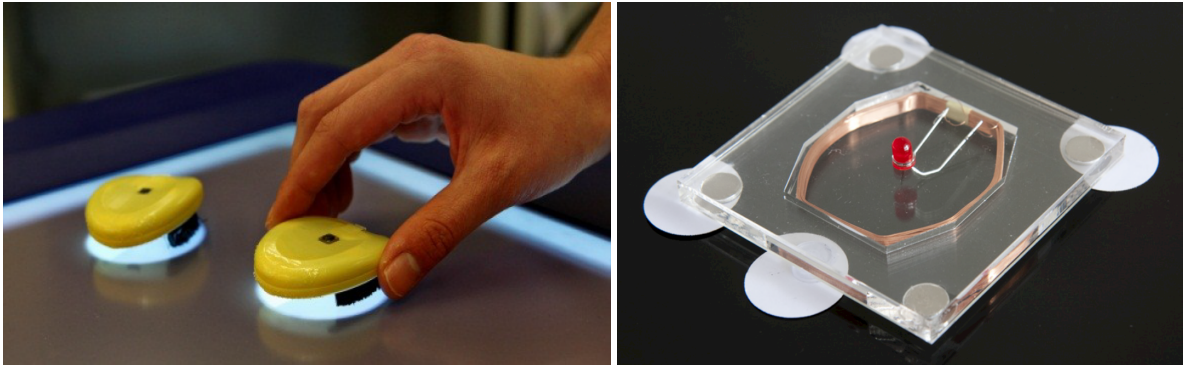


Abbildung 3.6: Links sind Tangibles zu sehen, die mit Vibrationsmotoren bewegt und durch Lichtsignale gesteuert werden können. Per Induktion kann die LED des Tangibles rechts mit Strom versorgt werden. Unter dem Touch-Tisch befinden sich dazu mehrere Magnete, die das Tangible auch bewegen können. Quellen: [NLH⁺13] und [WSJB10]

erreichen eine weichere Bewegung benötigen jedoch eine Batterie. Mithilfe eines Beschleunigungssensors kann bei der Bewegung eine Feinjustierung vorgenommen werden. Mit aktiven Tangibles ist auch eine direkte Interaktionen möglich, so dass sie auf Schütteln oder Drücken reagieren können.

3.4 Material, Form und Ästhetik

Durch das große Bedürfnis und der stetig wachsenden Möglichkeiten von personalisierten Produkten [Fut07, LK10] (siehe Abbildung 3.7) wird auch die Wahl des Materials immer wichtiger. Eine Möglichkeit zur Materialauswahl besteht darin, die verschiedenen Materialien hinsichtlich ihrer technischen Eigenschaften zu vergleichen und durch eine definierte Gewichtung jeder Eigenschaft eine Rangliste für das geeignetste Material zu ermitteln [CC12]. Auch durch Expertenwissen kann die Materialentscheidung getroffen werden, wie Karana et al. untersuchten [KHK08]. Sie befragten professionelle Designer nach ihren Kriterien und gruppieren diese zu relevanten Eigenschaften. Dazu gehören die sensorischen Eigenschaften wie Geruch, Geschmack, Aussehen und Haptik oder immaterielle Eigenschaften wie die wahrgenommenen Werte, Assoziationen, Emotionen und kulturelle Bedeutung. Ebenso beeinflusst wird die Auswahl von den technischen Eigenschaften, dem Herstellungsprozess, dem Produktionsvolumen, der Strapazierfähigkeit, bereits existierenden Herstellungsverfahren, Designbeschränkungen, Verfügbarkeit und vor allem von vorhandenem Fachwissen zu einem Material. Rognoli et al. beschäftigen sich ebenfalls mit einiger dieser Eigenschaften im Hinblick auf Materialwahrnehmung und ihrer Akzeptanz [RKP11].

Um die emotionale Verbindung zwischen Material und Benutzer zu untersuchen, wurde ein Werkzeug entwickelt, das Umfragen von Kunden zu einer bestimmten Eigenschaft ver-

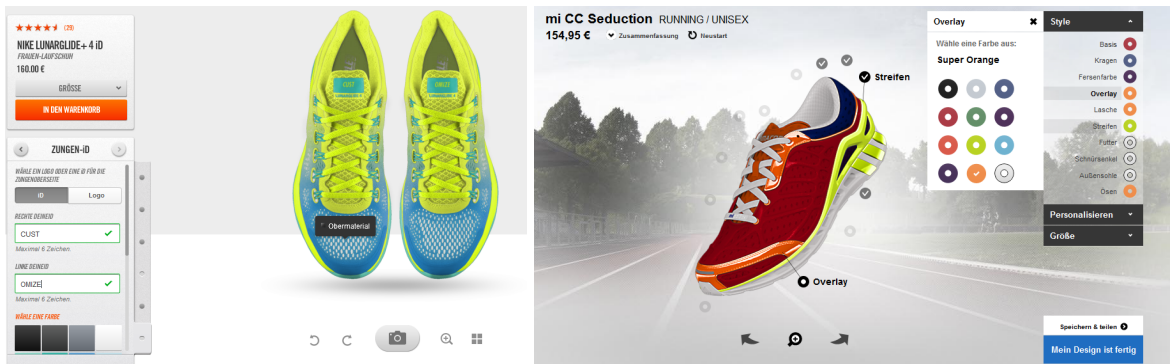


Abbildung 3.7: Viele Schuhhersteller (links Nike, rechts Adidas) bieten bereits Möglichkeiten zur individuellen Gestaltung ihrer Schuhe. Hier können Farben, Logos und Schriftzüge angepasst werden. Quellen: http://www.nike.com/de/de_de/c/nikeid und <http://www.adidas.de/personalisieren>

waltet, aber auch die Auswertung und die Erstellung solcher Umfragen anbietet [KHK10]. Dabei wird die Umfrage durch einen Designer erstellt, die sowohl materielle als auch immaterielle Eigenschaften beinhaltet. Um die Wahl für den Designer durch eine Erkennung von Entscheidungsmustern zu vereinfachen, werden Verknüpfungen von Werten und einer Begründung zur Auswahl abgespeichert. Beispielsweise sind dies Wortpaare wie hart-weich, matt-glänzend oder stark-schwach.

Noch tiefer steigen Nagai und Georgiev ein, die durch Tiefeneindrücke die immaterielle Wahrnehmung bei taktiler Interaktion untersuchen [NG11]. Für die Wörter, die der Teilnehmer zu einem Material äußert, werden Stimuluswörter über ein assoziatives Konzeptnetzwerk ermittelt. In diesem Netzwerk werden Knoten mit vielen Eingängen hervorgehoben, um die wichtigen Eigenschaften beziehungsweise Assoziationen mit einem Material herauszufiltern. Ein einfaches Beispiel wäre über das geäußerte Wort Winter das Wort kalt zu assoziieren. Die Teilnehmer sollten Materialien beurteilen, unter denen auch ein unbekanntes, selbst hergestelltes Material war. Damit lässt sich ein Vergleich zu Holz oder Glas ziehen, zu denen Menschen über die Jahrhunderte hinweg ein Gefühl entwickelt haben. Außerdem muss bei der Bewertung beachtet werden, dass bestimmte Materialien mit einem konkreten Produkt verbunden werden, wie beispielsweise Holz mit einem Küchenbrett assoziiert wird.

Dass neben der Materialwahl auch die Form eine große Auswirkung hat, zeigt sich bei Jung und Stolterman [JS11]. Sie verfolgen einen Ansatz, der im Gegensatz zum materialbasierten Vorgehen auf der formgetriebenen Entwicklung basiert. Jung und Stolterman beschäftigen sich mit der Frage wie Form und Funktion zusammenhängen und welchen Einfluss die Erkenntnisse auf die Digital Fabrication haben. Im herkömmlichen Entwicklungsprozess wird nach dem Prinzip *Form follows function* (Form folgt Funktion) vorgegangen. Damit wird über die Form erst zu einem späten Zeitpunkt entschieden. Jung und Stolterman hingegen schlagen vor, die Entscheidung bereits vorher zu berücksichtigen, um ihrer großen Auswirkung gerecht zu werden. Sie untersuchen interaktive Produkte auf Eigenschaften wie die kognitive

Form (das Erkennen und Verstehen von Funktionen) und die Ausdrucksmöglichkeit, wie Personalisierungs- und Anpassungsmöglichkeiten. Jung und Stolterman kommen zu dem Schluss, dass durch formgetriebene Entwicklung der Ästhetik mehr Gewicht verliehen wird. Mit ihr kann die gleiche Funktionalität jedoch mit unterschiedlichen Designs erreicht werden. Das heißt, dass im Gegensatz zum Form-follows-function-Prinzip auch ohne konkrete Benutzeranforderungen eine Form entwickelt werden kann, mit der neue Trends gesetzt werden. Die Digital Fabrication war bei ihrer Analyse besonders hilfreich, da sie kürzere Entwicklungszyklen ermöglichte, um Produktvariationen wissenschaftlich zu untersuchen.

Ästhetik

Im Folgenden wird der Zusammenhang zwischen Ästhetik und Usability vertieft. Reeves und Nass zeigen, dass Interaktionen mit Geräten wie PCs fundamental sozial und natürlich sind, so dass sie der Interaktion mit Menschen ähneln [RN96]. Damit wird unterstrichen, dass Ästhetik im Bereich der Produktentwicklung, insbesondere bei der Interaktion, eine ernst zu nehmende Rolle spielt.

Kurosu und Kashimura haben zunächst die Verbindung zwischen sichtbarer und inhärenter Usability untersucht [KK95]. In ihrem Experiment verwenden sie Benutzeroberflächen von Geldautomaten, die sie von Teilnehmern bewerten lassen. Das Ergebnis ihrer Studie belegt, dass es eine Abhängigkeit zwischen Usability und Ästhetik gibt, der Zusammenhang zwischen sichtbarer und inhärenter Usability jedoch schwach ist. Tractinsky greift das Experiment auf und ändert die auf die japanische Bevölkerung ausgelegten Eingabemasken ab, um sie für die israelischen Teilnehmer anzupassen [Tra97]. Tractinskys Ergebnisse weisen ebenfalls die Abhängigkeit von Usability und Ästhetik auf und bestätigen damit die Resultate von Kurosu und Kashimura. Zusammenfassend lässt sich sagen, dass trotz der Abhängigkeit der ästhetischen Wahrnehmung vom kulturellen Hintergrund [Fer95, Maq88], ihr Einfluss auf die Usability über verschiedene Kulturen hinweg gezeigt werden kann [KK95, Tra97].

Tractinsky et al. haben eine weitere Studie durchgeführt, um nicht nur die Verbindung zwischen Ästhetik und Usability *vor*, sondern auch *nach* dem Benutzen der Testgegenstände zu zeigen [TKI00]. Für den Versuch wurden die bereits in [Tra97] verwendeten Masken für Bankautomaten wiederverwendet. Um die Auswirkung der Usability zu untersuchen, wurden in einigen Fällen die Antwortzeit der Interaktion erhöht, einzelne Eingaben beim ersten Mal ignoriert und mögliche Abkürzungen entfernt. Das Ergebnis dieser Studie legt nahe, dass die Ästhetik Auswirkung auf die Wahrnehmung von Usability hat und dieser Effekt auch nach dem Benutzen (also dem Verifizieren der Usability) bestehen bleibt. Dies ist besonders erstaunlich, da Usability und Ästhetik als Gegensatz [But96] angesehen werden. Denn Usability bezieht sich auf die funktionale Ebene, wohingegen Ästhetik durch Subjektivität bestimmt wird.

Hassenzahl bezweifelt jedoch diese Abhängigkeit, da die Fehlerrate allgemein sehr gering gewesen und die Zeit zur Aufgabenerfüllung zwar die Usability beeinflusst, jedoch, bei geringen Unterschieden, nicht genügend aussagekräftig sei [Has08]. Im Gegensatz zu Tractinsky zeigen die Studien von Hassenzahl keine Abhängigkeit zwischen Usability und Ästhetik.

Hassenzahl verwendete Benutzeroberflächen von Mp3-Spielern und untersuchte Usability, Güte, Schönheit und Hedonik². Verschiedene Aspekte eines Produkts können durch die Kombination von Usability und Hedonik beschrieben werden. So kann ein Produkt pragmatisch, zur Erfüllung von Aufgaben gut geeignet, aber auch hedonisch sein, wenn durch die Emotionen, die mit dem Produkt verbunden werden, die Persönlichkeit ausgedrückt werden kann. In diesem Zusammenhang könnten auch die Geldautomaten, die sowohl bei Tractinsky als auch bei Kurosu und Kashimura verwendet wurden, als ungeeigneter Testgegenstand gelten, da sie nie persönlich besessen werden und man sich dadurch nie mit ihnen identifizieren wird. Generell stellt sich dann die Frage, ob das Einsatzfeld eines Produktes Einfluss auf das Ergebnis hat, wenn man zu Gegenständen, die im geschäftlichen Umfeld genutzt werden, andere Beziehungen hat, wie zu welchen im Freizeitbereich.

Da sich abschließend noch keine eindeutige Beurteilung über den Zusammenhang zwischen Usability und Ästhetik ergibt, bleiben weitere Studien abzuwarten. Wie eingangs bereits erwähnt, ist die Rolle, die Ästhetik bei der Produktentwicklung spielt, zweifellos wichtig. Roberts et al. ließen von Benutzern beispielsweise die Schönheit bewerten [RRS⁺03]. Damit im Personal Fabrication die Ästhetik weiter an Gewicht gewinnen kann, ist es wichtig, die technischen Voraussetzungen bei der Entwicklung zu verringern und zu standardisieren. Mit der Möglichkeit unterschiedliche Materialien, Farben und Formen zu verwenden, wird das Spektrum zur Ausgestaltung der Produkte immer größer und der Wunsch sein individuelles Design realisieren zu können rückt in greifbare Nähe.

²hier: Freude, durch den Testgegenstand hervorgerufen

4 Konzept

Das Ziel dieser Arbeit ist es, ein System zu entwickeln, mit dem Gehäuse einfach entworfen und hergestellt werden können. Dafür ist zunächst die Designidee notwendig. Baskinger weist auf die Bedeutung des Zeichnens von Hand hin, da die Entwicklung von Ideen dadurch wesentlich verbessert werden kann [Bas08]. Die vorgestellten Methoden ermutigen auch weniger zeichnerisch begabte Personen und fordern ein strukturiertes Vorgehen. Nachdem eine Vorstellung über das Design entwickelt wurde, werden die folgenden Punkte bis zur Fertigstellung unterstützt:

1. Konstruktion des virtuellen, grafischen Modells.
2. Herstellen des physischen Modells.
3. Zusammenbau des Gehäuses und Einbau bereits existierender Komponenten.

Ein wesentlicher Bestandteil des Konzepts ist die Eingabe mit Hilfe von Tangibles. Da die elektronischen Bauteile des zu erstellenden Prototyps bereits vorliegen, können sie als Tangibles (siehe Abschnitt 3.3) verwendet werden, um die Erstellung des virtuellen Gehäusemodells zu vereinfachen. Durch eine Objekterkennung, die mittels Touch-Technologie realisiert wird, können die Maße des Bauteils erfasst und dem virtuellen Modell ein entsprechendes Objekt hinzugefügt werden. Der Benutzer kann das Bauteil auf dem Touch-Tisch positionieren und so Lage und Winkel der Aussparungen des Bauteils im Gehäuse bestimmen. Werden mehrere Komponenten gleichzeitig aufgelegt, so werden die Beziehung zu einander und physikalische Beschränkungen offensichtlich. Durch die physikalische Repräsentation kommen die Vorteile von Haptik und direkter Steuerung durch den Gegenstand zur Geltung und erlauben eine natürliche Eingabe.

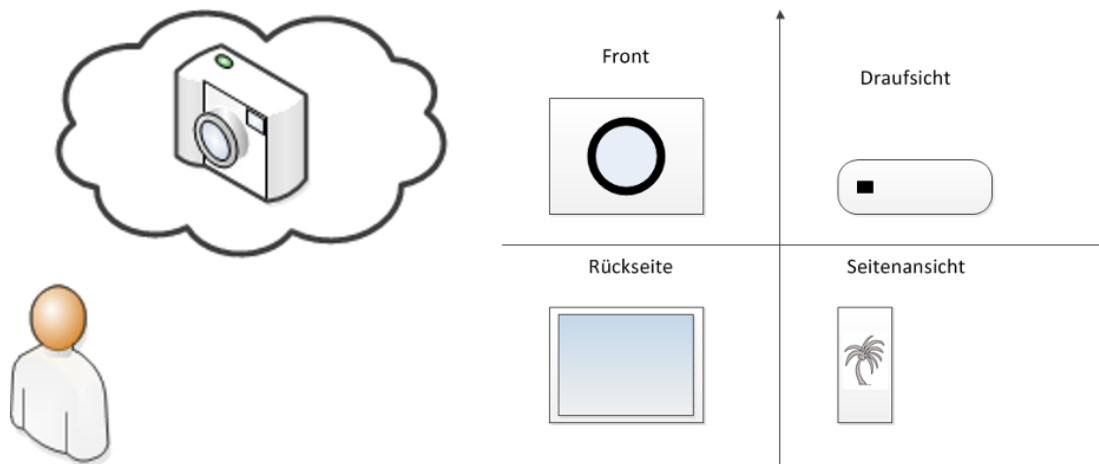
Für dieses Projekt kommen insbesondere Gadgeteer-Komponenten (siehe Abschnitt 2.4) zum Einsatz, deren Aussparungen entsprechend ihrer Platzierung eingefügt werden. Um für einen beliebigen Gegenstand auf einfachem und schnellem Weg eine Hülle anzufertigen, ist es denkbar alle Maße durch die Touch-Erkennung zu erfassen.

Im Gegensatz zu Müller et al. (siehe Abschnitt 2.2.1), die mit direkter Eingabe per Laserpointer arbeiten, ist es für dieses Projekt wichtiger, dass Aktionen einfach rückgängig gemacht oder der Entwurf weitergegeben werden kann. Deswegen erfolgt die Aufteilung in ein grafisches und ein physisches Modell. Mit den Möglichkeiten, die ein grafisches Modell bietet, bekommt der Benutzer direkte, visuelle Rückmeldung bei seiner Eingabe. Des weiteren spielt Präzision eine entscheidende Rolle, die mit direkter Eingabe per Laserpointer ohne Korrekturmöglichkeit nur schwer zu erreichen ist.

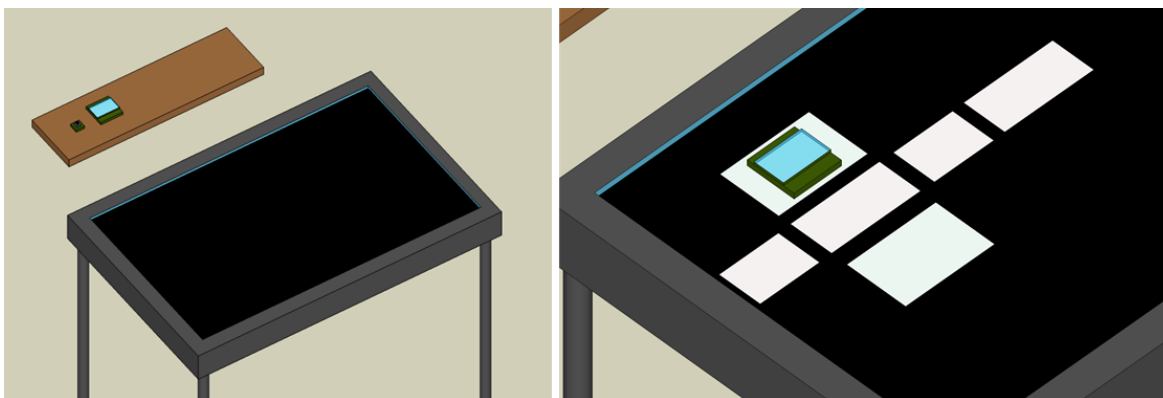
Das System, das in diesem Projekt entwickelt wird, unterstützt den Konstruktionsprozess und ermöglicht das Erstellen des virtuellen Modells für ein Gehäuse (siehe Abbildung 4.1). Bei der Entwicklung muss besonders das Hintergrundwissen des Anwenders beachtet werden, da Personen im Heimbereich ohne technische Vorkenntnisse angesprochen werden sollen. Aus diesem Grund wird kein CAD-ähnliches Programm entwickelt, damit die Komplexität und somit auch die Einstiegshürde angemessen sind (siehe Abschnitt 2.1 auf Seite 11). Es werden grundlegende Funktionen wie das Hinzufügen einfacher Formen angeboten, wobei sie möglichst so gestaltet sind, dass auf bekannte Eingabemodi zurückgegriffen wird. Der Benutzer kann sich somit schnell mit dem System vertraut machen. Die Interaktion wird möglichst natürlich gestaltet, so dass sie einfach zu erlernen ist. Die Basiselemente der Eingabe bestehen aus Gesten, Zieh- und Klickevents. Durch die weite Verbreitung von Touchdisplays sind viele Menschen bereits den Umgang mit dieser Eingabeart gewohnt. Möchte man die Präzision erhöhen und dennoch eine direkte Eingabeart verwenden, so bietet sich die Eingabe per Stift an (siehe auch Abschnitt 3.1). Neben der höheren Genauigkeit ist die Umsetzung einer Idee mit Stift und Papier seit vielen Jahren integraler Bestandteil des Designprozesses. Die Aktionen, die mit dem Stift ausgeführt werden können, müssen jedoch immer redundant sein, so dass auch ohne Stift die volle Funktionalität zur Verfügung steht und die eventuell ungewohnte Stifteingabe nicht verwendet werden muss.

Ist das Bearbeiten des virtuellen Modells abgeschlossen, so tritt Schritt zwei, das Herstellen des physischen Modells, in den Vordergrund. Körper, die aus geraden Flächen hergestellt werden, stellen besondere Ansprüche an die Verbindungsstelle (siehe Abschnitt 2.3). Diese Verbindungen werden von der Software automatisch generiert und ermöglichen das Zusammenfügen der Elemente. Das virtuelle Modell wird als Grafikdatei direkt an den Lasercutter geschickt. Dort kann der Arbeitsvorgang gestartet werden, so dass in wenigen Minuten alle Einzelteile zur Verfügung stehen. Je nach Material und Ausführung der Verbindung (beispielsweise Presspassung), muss das Gehäuse anschließend zusammengesteckt oder geklebt werden. Alle vorgesehenen Komponenten, zum Beispiel das Display von Gadgeteer, werden eingesetzt und befestigt, so dass in kurzer Zeit das Gehäuse einsatzbereit ist.

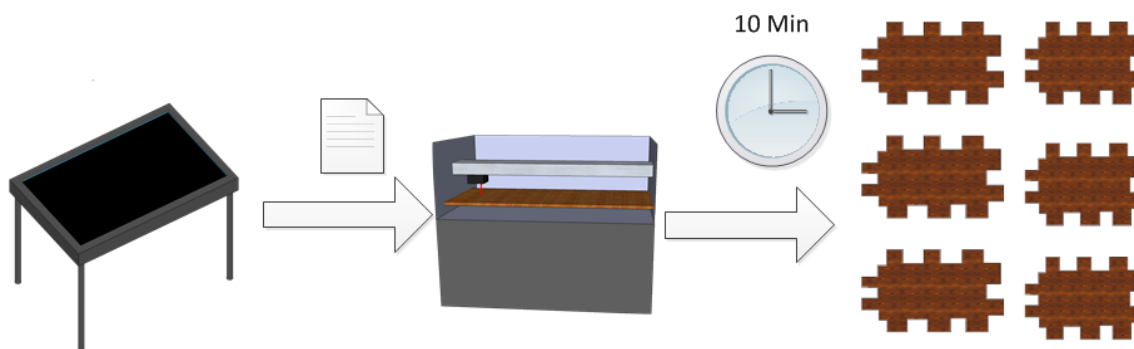
Savage et al. demonstrieren einen nahtlosen Übergang zwischen verschiedenen Arbeitsschritten [SZH12]. Sie bieten Unterstützung vom Verlegen der Leiterbahnen der Touch-Sensoren bis zum Konfigurieren der Anweisungen, die bei einem Touch-Event ausgelöst werden. Diese Idee wird auf das hier präsentierte Projekt übertragen, so dass der Benutzer von Entwurf bis zum Zusammenbau des Gehäuses durchgängig unterstützt wird. Auch die Arbeit von Lau et al. [LSMI10] besticht durch die einfache Eingabe. Die Unterstützung des Benutzers motiviert die Herstellung von Gehäusen mit der Beschränkung auf zweidimensionales Zeichnen weiter zu vereinfachen.



(a) Die Idee – Ein Kameragehäuse aus verschiedenen Winkeln



(b) Die Ausgangssituation und die Eingabe über Tangibles



(c) Der Herstellungsprozess – Übertragung der Daten und Ausschneiden durch den Lasercutter

Abbildung 4.1: Zur Herstellung eines Gehäuses muss zunächst die Idee entwickelt werden, wie das Gehäuse aussehen soll. Anschließend kann die Eingabe zum Beispiel über Tangibles erfolgen, die die Aussparungen für Komponenten erleichtert. Im nächsten Schritt werden die Daten an den Lasercutter übertragen, der die gewünschte Form ausschneidet. Zum Schluss kann das Gehäuse montiert werden (ohne Abbildung).

4.1 Szenario

Um bei der Entwicklung ein durchgängiges Beispiel zu haben, das die Hauptanwendungsfälle abdeckt, wird das folgende Szenario beschrieben:

Andrea hat die Vorgabe ein Gadgeteer-Prototyp zu entwickeln, bei dem ein Display zur Ausgabe verwendet wird. Sie möchte einen Prototypen für eine Kamera entwerfen. Nachdem Andrea die Gadgeteer-Software konfiguriert und angepasst hat, muss nun ein Gehäuse gestaltet werden, an das das Display, die Kamera und ein Button zum Auslösen angebracht werden. Diese Komponenten sollen möglichst gut in das Gehäuse integriert werden, so dass beispielsweise von der Kamera nur das Objektiv und die Befestigung von außen sichtbar sind. Um einen ersten Entwurf zu entwickeln, entscheidet sich Andrea für einen Quader als Ausgangsbasis ihres Gehäuses. Sie startet die Software auf dem Touch-Tisch und wählt aus den Grundformen den Quader aus. Nun legt sie Breite, Höhe und Tiefe fest (150 x 100 x 50 mm) und geht zum nächsten Schritt. Hier wählt sie als Material Pappel und gibt dessen Stärke (4 mm) an. Die Verbindungsart belässt sie auf der Standardauswahl Fingerverbindung. Nun fängt die eigentliche Bearbeitung an. Sie startet die Objekterkennung und legt das Display, die Kamera und den Button auf den Touch-Tisch. Nachdem sie das Display auf der Vorderseite, die Kamera auf die Rückseite und den Button auf die obere Seite platziert hat, beendet sie die Objekterkennung und überprüft die hinzugefügten Aussparungen. Um der Kamera eine persönliche Note zu verleihen wählt sie eine Seitenfläche aus und fügt ihr letztes Urlaubsbild ein, das auf die Kamera graviert werden soll. Sie speichert diese erste Version ab und lässt sich die Vorlage für den Lasercutter anzeigen. Sie ist mit ihrem Entwurf zufrieden und schickt den Auftrag an den Lasercutter, um ihn in wenigen Minuten in den Händen zu halten.

5 Entwurf

Dieses Kapitel gibt zunächst einen Überblick über das Projekt. Anschließend wird die Architektur des zugrundeliegenden Systems beschrieben. Im letzten Teil erläutern Anwendungsfälle den Ablauf aus Benutzersicht. Beim Erstellen des Entwurfs spielte die Erweiterbarkeit eine große Rolle, so dass Module einfach eingefügt werden können. Um die Komponenten möglichst einfach wiederverwenden zu können, wurde das System in drei Teile aufgliedert: Natcut, ObjectDetection und SVG-Generator, die in den letzten Abschnitten näher beschrieben werden. Abbildung 5.1 zeigt die Schnittstellen zwischen der realen Welt und des entwickelten Systems, während in Abbildung 5.2 das Zusammenspiel der Komponenten erläutert wird. Im Anhang ist eine detailliertere Darstellung der Komponenten zu finden (Abbildung A.5).

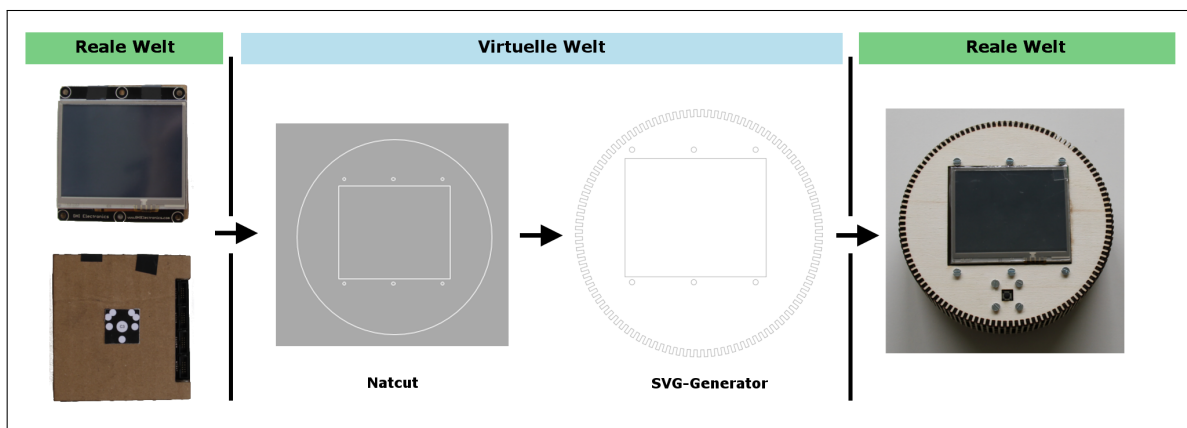


Abbildung 5.1: Das bereits vorliegende Display wird als Eingabe verwendet. Nach dem das Display mit Hilfe des Markers erkannt wurde, fügt Natcut die Aussparungen ein. Der SVG-Generator realisiert die Verbindungselemente und bereitet die Eingabedatei für den Lasercutter vor. Sind die Teile durch den Lasercutter ausgeschnitten, können sie zusammengesetzt und die externen Komponenten eingefügt werden.

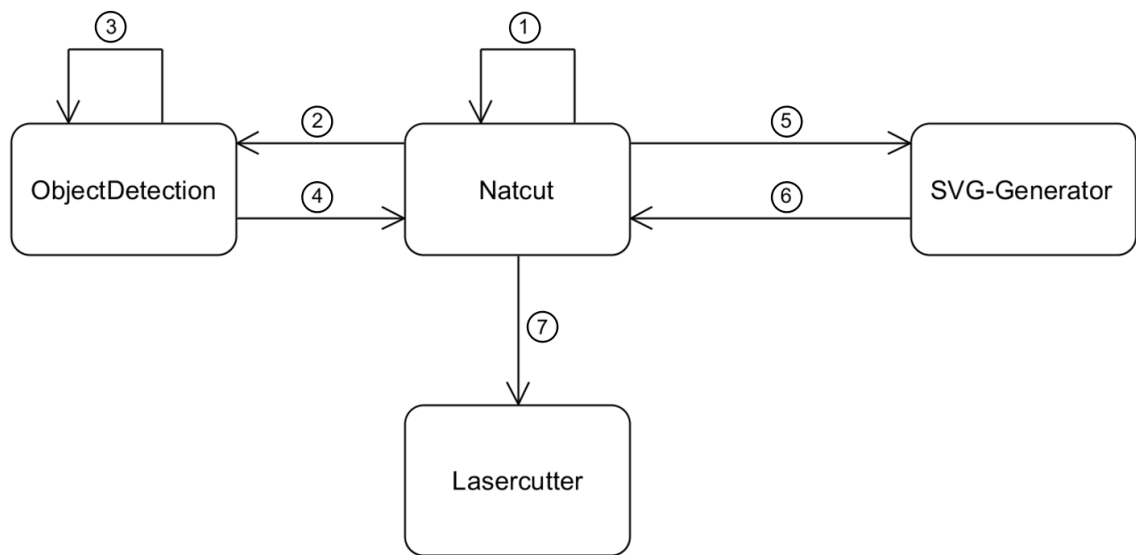


Abbildung 5.2: Die Anwendung besteht aus drei Komponenten: ObjectDetection, Natcut und SVG-Generator. Die Natcut-Komponente ist dabei für die Verbindung zum Lasercutter zuständig. Der Entwurfsprozess des Gehäuses mit Hilfe von Gadgeteer-Komponenten erfolgt in 7 Schritten: Schritt 1 – Auswählen einer Basisform für das Gehäuse (Quader oder Zylinder); Schritt 2 – Starten der Objekterkennung; Schritt 3 – Positionieren der Hardware auf dem Display; Schritt 4 – Beenden der erkannten Komponenten und Übertragung der Informationen an Natcut, wo die entsprechenden Aussparungen eingefügt werden; Schritt 5 – Der Auftrag zum Generieren des Gehäuses mit allen Verbindungen wird an den SVG-Generator übertragen; Schritt 6 – Ein Verweis auf die erstellte SVG-Datei wird der Natcut-Komponente geschickt, die dann das Gehäuse an den Lasercutter überträgt (Schritt 7).

5.1 Projekt-Aufbau

Ein kurzer Überblick über den Aufbau des Projekts beinhaltet die wichtigsten Schritte, die der Benutzer während des Herstellungsprozesses ausführt. Die grundlegenden Ideen werden im folgenden beschrieben. Der Ablauf innerhalb der Software zur Erstellung eines neuen virtuellen Modells, das als Eingabe für den Lasercutter dient, ist in drei Schritte unterteilt:

1. Grundform auswählen
2. Allgemeine Einstellungen zur Form vornehmen
3. Form bearbeiten

Diese Aufteilung erfolgt aus zwei Gründen: Zum einen wird die Komplexität verringert, wenn nur die Funktionen angeboten werden, die der Benutzer im Moment tatsächlich ausführen

kann. Zum anderen ist das Ergebnis eines Arbeitsschritts abhängig von der vorherigen Entscheidung. Die Parameter für die Dimensionen eines Gehäuses hängen von der zuvor gewählten Grundform ab und haben Einfluss auf die Auswahl an Materialien. Nicht alle Materialien sind für alle Verbindungsarten gleich gut geeignet. Die Maße der bearbeitbaren Fläche der Form hängen wiederum von den Einstellungen ab, da je nach Materialstärke unterschiedlich viel Platz für die Verbindungen reserviert werden muss. Je größer die Stärke, umso mehr Platz benötigt die Verbindung, um so geringer wird die zu bearbeitende Fläche.

5.2 Natcut Application

Diese Komponente ist die Hauptanwendung, die den Ausgangspunkt für alle anderen Komponenten darstellt. Sie beinhaltet die Benutzeroberfläche und greift auf die Komponenten der Objekterkennung und des SVG-Generators zu. Da die Implementierung mit der *Windows Presentation Foundation*¹ (WPF) realisiert wurde, bietet sich das *Model-View-ViewModel*²-Entwurfsmuster (MVVM) an. Bei diesem Entwurfsmuster wird die Benutzeroberfläche (View) von der Logik getrennt, ähnlich wie bei beim *Model-View-Controller*-Entwurfsmuster (MVC) [MHR79]. Allerdings findet zwischen View und Model keine Kommunikation statt, diese erfolgt ausschließlich über das ViewModel. Die Definition der Benutzeroberfläche erfolgt hauptsächlich in XAML³ und hat damit bereits eine Trennung in Syntax und Bearbeitungsweise im Vergleich zur Logik, die mit C# realisiert wurde. Im Folgenden werden die wichtigsten Teile der Hauptanwendung beschrieben.

Benutzeroberfläche

Alle XAML-Definitionen der Benutzeroberfläche werden in dieser Komponente zusammengefasst. Nur bei einigen dynamisch erzeugten Elementen wird auf die Implementierung mit C# zurückgegriffen. Hierarchisch ist die Hauptansicht in mehrere Reiter unterteilt, wobei jedem Reiter genau ein Gehäuse zugeordnet ist. Alle Eingaben, die vom Benutzer vorgenommen werden, werden gemäß dem Entwurfsmuster MVVM an die ActionController-Komponente weitergeleitet. Beispiele der konkret implementierten Oberfläche sind im Kapitel Implementierung zu finden.

ActionController

Der ActionController kapselt alle Aktionen gegenüber der Benutzeroberfläche und überwacht sie, um dem Benutzer gegebenenfalls die Möglichkeit zu bieten sie rückgängig zu machen oder wiederherzustellen. Dies wird erleichtert durch die Verwendung des *Command-Patterns*

¹<http://msdn.microsoft.com/de-de/library/vstudio/ms754130.aspx>

²<http://msdn.microsoft.com/de-de/magazine/dd419663.aspx>

³<http://msdn.microsoft.com/en-us/library/ms752059.aspx>

[GHJV95] an Stelle des weit verbreiteten Event-Handlings. Ist eine Aktion umkehrbar, wird das Command-Pattern durch eine Schnittstelle realisiert, die auf dem *Undo-Command-Pattern*⁴ basiert. Für nicht umkehrbare Aktionen (beispielsweise das Exportieren des Gehäuses in das SVG-Format) kommt die *ICommand*-Schnittstelle⁵ zum Einsatz. Durch den *Command Manager*⁶ wird der Zustand einer Aktion (ausführbar oder nicht) direkt an angebundene Buttons übertragen, die sich entsprechend (de-)aktivieren.

Die wichtigsten Aktionen, die umkehrbar sind, betreffen die Bearbeitung einzelner Formen. Dazu gehören im Detail:

- Größe ändern
- Verschieben
- Drehen
- Löschen
- Neue Komponente erkennen
- Neue Komponente hinzufügen
- Grafik hinzufügen
- Leinwand hinzufügen

Weitere Details sind in den Anwendungsfällen (Abschnitt 5.5) beschrieben. Die zugehörige Benutzeroberfläche ist im Kapitel Implementierung zu finden.

Settings

Alle Einstellungen, die von der Hauptkomponente Natcut verwendet werden, werden hier verwaltet. Unterschieden wird dabei in Eigenschaften, die zur Bedienung des Programms dienen und denen, die die Eigenschaften von Materialien und Lasercuttern beschreiben. Letztere werden in einer externen XML-Datei gespeichert, um sie einfach austauschen zu können. Insbesondere wird dort definiert, welche Materialien für welche Verbindungsarten geeignet und welche Parameter welchem Lasercutter zu geordnet sind.

⁴http://www.codeproject.com/Articles/33384/Multilevel-Undo-and-Redo-Implementation-in-C-Part#_rating

⁵<http://msdn.microsoft.com/de-de/library/system.windows.input.icommand.aspx>

⁶<http://msdn.microsoft.com/de-de/library/system.windows.input.commandmanager.aspx>

LaserModel

Die Aufgabe der LaserModel-Komponente ist es, die Vorbereitung und Ausführung eines Gehäuseauftrags zu kontrollieren. Dazu muss der Benutzer zunächst den Lasercutter auswählen, mit dem er das Gehäuse erstellen möchte. Anschließend wird überprüft, ob alle notwendigen Parameter für den Lasercutter mit dem ausgewählten Material vorhanden sind. Im nächsten und letzten Schritt wird das Gehäuse als SVG-Datei exportiert (siehe Abschnitt ShapeAPI), um diese zum Lasercutter zu schicken. Nun kann der Benutzer den Auftrag am Lasercutter starten.

ShapeAPI

Die Hauptaufgabe besteht darin, eine Schnittstelle zwischen Hauptanwendung und dem externen SVG-Generator zu bilden. ShapeAPI kapselt dabei den Zugriff innerhalb der Hauptanwendung auf Instanzen des Objektmodells und bietet Operationen an, die die Bearbeitung des Objektmodells betreffen. InstanceManager, ein Teil der ShapeAPI-Komponente, hält eine Liste der Instanzen des Objektmodells, die jeweils genau einem Reiter in der Benutzeroberfläche zugeordnet sind. Weitere wichtige Funktionen sind das Laden eines gespeicherten Gehäuses (Deserialisieren und Aufbereiten des Objektmodells) und das Exportieren von Gehäusen als SVG-Dateien.

DetectorAPI

Die DetectorAPI ist die Schnittstelle zwischen ObjectDetection und Natcut. Alle Identifikationsnummern der Marker, die von der Komponente ObjectDetection erkannt werden sollen, werden in dieser Komponente definiert. Außerdem wird die grafische Repräsentation einer Komponente festgelegt, während die ObjectDetection im Erkennungsmodus (also aktiv) ist. Ist das Erkennen abgeschlossen, werden die Informationen über Marker, wie Orientierung und Position, für das Gehäusedesign in Natcut aufbereitet. Dafür werden die Aussparungen verwendet, die für die Identifikationsnummer eines Markers in der DetectorAPI-Komponente hinterlegt sind.

ViewModel

Alle ViewModels, die in Natcut verwendet werden, sind hier zusammengefasst. Zu den wichtigsten zählen MainWindowViewModel, CasingViewModel und ShapeViewModel (siehe Abbildung 5.3). MainWindowViewModel, wie der Name vermuten lässt, bietet alle Informationen an, die in der Hauptansicht dargestellt werden. Für die meisten Aktionen, die durch Buttons in der Hauptansicht ausgelöst werden können, sind hier die Commands (siehe Abschnitt ActionController) gespeichert. Das MainWindowViewModel hält weitere ViewModels für Dialoge, wie beispielsweise zur Erstellung eines neuen Gehäuses vor, hat aber auch eine Liste von CasingViewModels für alle Gehäuse.

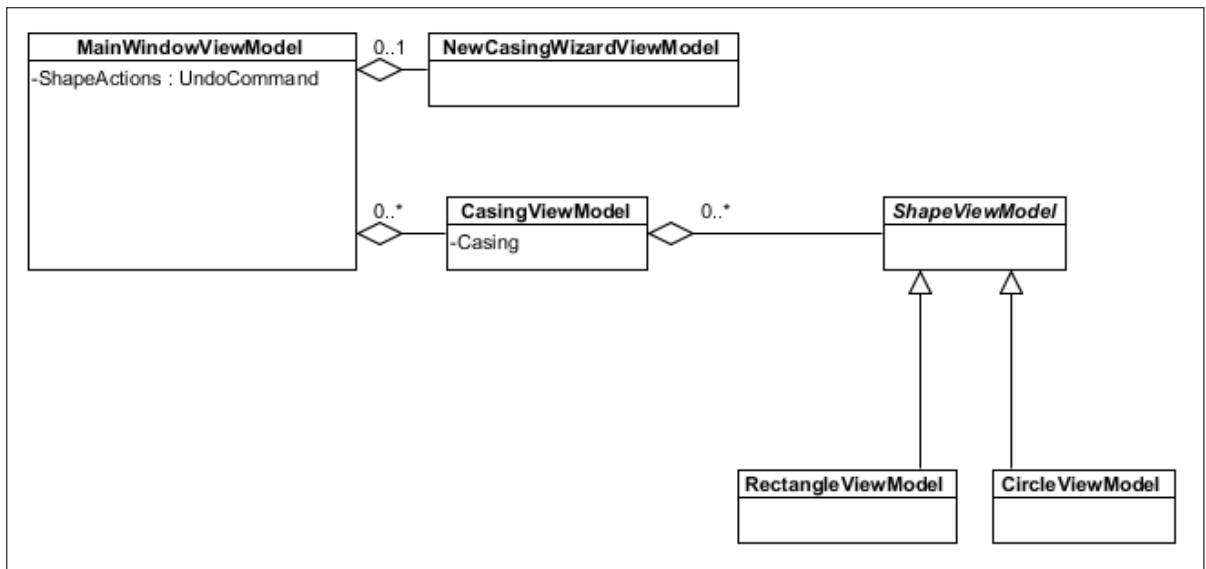


Abbildung 5.3: Das MainWindowViewModel ist zentraler Ausgangspunkt für viele weitere ViewModels. Zu ihnen gehören alle ViewModels für Dialoge (hier beispielhaft das NewCasingWizardViewModel). Das MainWindowViewModel bietet die meisten Commands an, die beispielsweise für die Bearbeitung von Formen verwendet werden (ShapeActions). Für jedes geöffnete Gehäuse wird ein CasingViewModel bereitgehalten, das wiederum ShapeViewModels besitzt, die die zugehörigen Flächen repräsentieren. Ein CasingViewModel hat außerdem eine Referenz auf die Objektmodell-Instanz (Casing), die die geometrischen Eigenschaften speichert. Die ShapeViewModel-Klasse ist abstrakt, ihre abgeleiteten Klassen stellen Rectangle- und CircleViewModel dar.

Ein CasingViewModel bietet alle Eigenschaften an, die für die Bearbeitung eines Gehäuses notwendig sind. Es besitzt eine Referenz auf die zugehörige Instanz des Objektmodells und hat eine Liste von ViewModels für alle Flächen, die Teil des Gehäuses sind. CasingViewModel ist für das Aktualisieren dieser ViewModels zuständig, um sie in einem konsistenten Zustand zu halten. Beispielsweise müssen beim Ändern der Dimension des Gehäuses alle betroffenen Flächen angepasst werden. Diese ViewModels der Flächen sind von der Basisklasse ShapeViewModel abgeleitet, die die Eigenschaften für die Rotation, das Verschieben oder das Vergrößern anbietet. Je nach Form (Rechteck oder Kreis) gibt es eine konkrete Implementierung, welche die Konvertierung von Pixeln zu Realeinheiten (Millimeter oder Inch) vornehmen, in denen das Objektmodell gespeichert ist.

5.3 ObjectDetection

Die externe Komponente ObjectDetection ist für das Erkennen von Gegenständen zuständig, die auf den Touch-Tisch gelegt werden. Für eine schnelle Umsetzung eines Software-Prototyps erfolgt die Objekterkennung mit Hilfe von Markern (vergleiche Abbildung 7.1 auf Seite 74). Eine Alternative, die Erkennung von Komponenten durch Bilderkennungsverfahren, wird außerdem durch die Auflösung der Touch-Erkennung (32 PPI beziehungsweise 0,79 mm Pixelabstand) beschränkt. Die wichtigsten Teile der ObjectDetection sind:

- DetectorControl
- DetectorWindow
- ADeviceDetector
- DetectorTagViewModel
- DetectedTagModel

DetectorControl realisiert die Logik für das Erkennen und Aktualisieren von Position und Orientierung von Gegenständen. Es ist in ein UserControl ausgelagert, so dass es sowohl als Teil einer Anwendung verwendet werden kann (wie das bei Natcut der Fall ist) als auch eigenständig gestartet werden kann, wenn es über das DetectorWindow aufgerufen wird. DetectorWindow stellt dabei lediglich die Umgebung zur Verfügung, innerhalb der die Erkennung gestartet werden kann. ADeviceDetector ist eine abstrakte Klasse, die eine Eigenschaft DetectedDevices vorschreibt und die als Schnittstelle für externen Zugriff fungiert. Diese Eigenschaft ist eine Liste vom Typ DetectedTagModel, mit der die erkannten Gegenstände abgerufen werden können, wenn die Objekterkennung abgeschlossen ist. DetectedTagModel besitzt Eigenschaften wie Orientierung, Position und Identifikationsnummer des Markers. DetectorTagViewModel ist das ViewModel von DetectorControl und ist von ADeviceDetector abgeleitet. Es bietet neben den bereits erkannten Gegenständen auch eine Liste an, in der die Eigenschaften der Marker definiert werden, die als Objekte erkannt werden sollen. Durch die Verwendung der abstrakten Klasse ADeviceDetector können weitere Erkennungsarten einfach hinzugefügt werden. Denkbar wäre beispielsweise das Erkennen durch Bildverarbeitungsalgorithmen. DetectedTagModel müsste dann um geometrische Eigenschaften der erkannten Formen erweitert werden. Für das Zusammenspiel zwischen ObjectDetection und Natcut siehe auch Abschnitt DetectorAPI.

5.4 SVG-Generator

Diese externe Komponente erzeugt zu einem gegebenen Casing-Objekt die SVG-Repräsentation. Die Komponente wurde ausgelagert, um die Wiederverwendung für andere Projekte zu erleichtern. Die Funktionalität des SVG-Generators kann sowohl über das Einbinden der dll-Bibliothek als auch über Konsolenaufruf erfolgen. In beiden Fällen werden die Parameter,

die in der übergebenen Instanz des Objektmodells gespeichert sind, als Ausgangsbasis verwendet. Hinzu kommen globale Einstellungen, wie die Breite des Laserstrahls, die bei der Generierung beachtet werden müssen. Die Klassen im SVG-Generator sind ausschließlich für die Generierung der SVG-Grafiken zuständig und bieten verschiedene Verbindungsarten an.

CasingModel

Die Casing-Komponente ist die zentrale Datenkomponente, die alle geometrischen Objekte vorhält und ihre Objektstruktur definiert. Casing steht dabei für das dreidimensionale Objekt, das gefertigt werden soll (beispielsweise ein Würfel).

Der wichtigste Teil der CasingModel-Komponente ist das Objektmodell. Es gibt die Datenstruktur für alle Parameter vor, die zur Beschreibung eines Gehäuses notwendig sind. Das Objektmodell wird sowohl von der Hauptanwendung als auch vom SVG-Generator benutzt und beschreibt das Austauschformat zwischen diesen. Durch die Abstrahierung des Objektmodells sind Benutzeroberfläche, die interne Verarbeitung und die Weiterverwendung durch den SVG-Generator unabhängig voneinander. Die Komponenten können daher leichter ausgetauscht oder wiederverwendet werden. Das Objektmodell ist ein Metamodell, das in einem XML-Schema beschrieben ist. Es definiert die Hierarchie der Elemente, legt jedoch noch keine Logik fest. Erst durch die Implementierung der Elemente als Klassen, wird beispielsweise festgelegt, dass ein Quader sechs Seiten besitzt.

Der Unterschied zwischen einer gespeicherten Objektmodell-Instanz und der daraus generierten SVG-Datei liegt im Verwendungszweck. Die Objektmodell-Instanz wird im XML-Format gespeichert und hält sämtliche Informationen des Gehäuses vor. Die SVG-Datei, die daraus generiert wird, beschreibt die geometrischen Elemente des Schnittauftrags, mit dem der Lasercutter ein Gehäuse herstellen kann. Aus einer Objektmodell-Instanz kann mit Hilfe von Lasercutter-Eigenschaften eine SVG-Datei generiert werden, jedoch nicht anders herum. Auch wenn das SVG-Format auf XML basiert, wurden die generierten SVG-Daten aus Gründen der Wartbarkeit ausgelagert. Bei der Wahl beider Formate war die Offenheit und die Plattformunabhängigkeit wichtig, um für zukünftige Wiederverwendung keine Einschränkungen einzuführen. Weitere Details sowie ein ausführliches Klassendiagramm sind im Anhang A.1 zu finden.

5.4.1 CasingController

Der CasingManager, der Teil des CasingControllers ist, nimmt das Casing-Objekt entgegen, und bietet Funktionen zur SVG-Generierung sowie zum Speichern der Datei an. Er kapselt die konkreten Abläufe, die zur Erstellung von Quader, abgerundetem Quader oder Zylinder durchlaufen werden. Außerdem besitzt er eine Liste für alle generierten SVG-Objekte, die entsprechend der Außenseiten des Gehäuses gruppiert werden. Für jede (Gehäuse-)Form gibt es einen eigenen Controller, der seine Außenseiten in Linien aufteilen kann und die entsprechenden Verbindungsgeneratoren aufruft. Der Controller ist somit auch verantwortlich für die

konkrete Umsetzung der Verbindungen. Das heißt beispielsweise bei der Erstellung eines Quaders, dass bei einer Ecke, in der alle drei Seiten aufeinander treffen, der Controller entscheidet welche Seite den inneren, den mittleren und den äußeren Zapfen bekommt. Ebenso fällt es in seinen Aufgabenbereich die Stabilität der äußersten Zapfen zu garantieren oder die Koordination von Akkordeonverbindung zu ihrer Seitenfläche zu übernehmen. Die entsprechenden Algorithmen dazu, befinden sich im Abschnitt 6.4 des Implementierungskapitels.

5.4.2 JointGenerator

Der Verbindungsgenerator erstellt für gegebene Linien oder Flächen eine passende Verbindung (Finger-, Schwalbenschwanz-, Schrauben- und Akkordeonverbindungen). Er ist verantwortlich, dass die Kanten zweier Außenflächen nach der Herstellung miteinander verbunden werden können. Ein wichtiger Parameter stellt dabei die Materialstärke dar, der für die Mindestbreite für Zapfen von Fingerverbindungen verwendet wird. Je nach Verbindungsart können jedoch weitere Kriterien hinzukommen. Beispielsweise müssen bei Schraubverbindungen die Maße der Mutter und der Schraube vorliegen, um die entsprechenden Aussparungen umzusetzen.

5.4.3 SVG

Die SVG-Komponente trennt SVG-Repräsentanten von den restlichen Komponenten des SVG-Generators. Die SVG-Komponente stellt den SVG-Serializer zur Verfügung, der alle SVG-Repräsentanten in die Datei schreibt. Zu den SVG-Repräsentanten gehören alle abstrahierten Klassen der verwendeten SVG-Basisformen wie Circle, Line und Polyline. Um die Ausgabe in der SVG-Datei übersichtlich zu gestalten, ermöglicht es das SVGAreaModel die erstellten SVG-Objekte pro Außenfläche zu gruppieren.

5.5 Anwendungsfälle

Aus dem im Szenario (Abschnitt 4.1 auf Seite 48) beschriebenen Beispiel ergeben sich einige Anwendungsfälle, die im Folgenden genau spezifiziert werden. Hier wird ausschließlich der Ablauf der Bearbeitung beschrieben. Deren Realisierung und weitere Programmfunktionen folgen im Kapitel der Implementierung (siehe Seite 61).

5.5.1 Neue Komponente hinzufügen

Um eine neue virtuelle Komponente hinzuzufügen, gibt es zwei Möglichkeiten. Zum einen können bereits vorhandene, physikalische Komponenten des Prototyps als Eingabe für den Touch-Tisch verwendet werden. Zum anderen kann eine eigene Form eingefügt werden.

Einfügen durch Objekterkennung

Wird die Objekterkennung für bereits existierende Komponenten verwendet, kann mit folgenden Schritten eine virtuelle Komponente hinzugefügt werden:

1. Objekterkennung aktivieren.
2. Komponenten auf dem Touch-Tisch platzieren und in gewünschte Position und Orientierung bringen.
3. Objekterkennung beenden. Nun werden alle für die Komponente hinterlegten Aussparungen als virtuelle Komponente hinzugefügt.

Manuelle Eingabe

Ist keine physikalische Vorlage vorhanden, können Basisformen hinzugefügt werden. Zur Auswahl stehen Rechteck und Kreis. Durch folgende Schritte wird eine Komponente manuell hinzugefügt:

1. Die Grundfläche, auf der die Komponente eingefügt werden soll, wird per Klick ausgewählt.
2. Über die Buttons im Menü kann Kreis oder Rechteck ausgewählt werden.
3. Nun berührt man den Touchscreen um die linke obere Ecke der Form zu bestimmen (auch beim Erstellen eines Kreises). Die Größe der Form passt sich automatisch an, bis der Finger vom Touchscreen genommen wird.

5.5.2 Manipulation einer Komponente

Es gibt drei Möglichkeiten eine Komponente zu verändern: Verschieben, Drehen und Ändern der Größe. Für alle Vorgänge gilt das gleiche Vorgehen:

1. Komponente per Klick auswählen.
2. Manipulation durch die Buttons im Menü auswählen.
3. Drag and Drop um die Manipulation durchzuführen.

Die *Drag and Drop*-Aktion wird auf unterschiedliche Elemente angewendet, das Verschieben kann an eine beliebige Stelle innerhalb der Komponente gestartet werden. Für das Ändern der Größe oder um eine Komponente zu drehen werden unterschiedliche Anfasser (visuelle Elemente) am Rahmen der Komponente angeboten. Jede Manipulation kann auch als absoluter Wert durch Texteingabe gesetzt werden.

Das Verschieben der Komponente ist die Standardaktion, nach dem sie ausgewählt wurde. Hier kann also der zweite Schritt ausgelassen werden. Außerdem kann jede Komponente über den entsprechenden Button gelöscht werden, nachdem sie ausgewählt wurde.

Nicht für alle Komponenten stehen alle Manipulationen zur Verfügung, da beispielsweise Komponenten, die mittels Objekterkennung eingefügt wurden, schon die richtige Größe besitzen. Sind Komponenten gruppiert, wird die ausgeführte Aktion für alle Gruppenmitglieder analog ausgeführt.

5.5.3 Eigenschaften einer Grundfläche

Eine Grundfläche kann drei verschiedene Zustände haben. Standardmäßig besitzt sie keinen Hintergrund und es können beliebige Komponenten hinzugefügt werden. Eine weitere Option ist das Personalisieren, das es erlaubt einer Grundfläche ein Bild hinzuzufügen. Die dritte Option macht die Grundfläche zu einer Leinwand. Hier werden automatisch die Elemente für die Rahmen hinzugefügt, wobei die Rahmenbreite gleich der Dicke des Materials gesetzt wird. Nach der Herstellung der Gehäuseteile durch den Lasercutter, kann ein Stück Stoff zwischen die beiden Rahmen gespannt werden.

6 Implementierung

Konkrete Entscheidungen, die während der Umsetzung des Entwurfs getroffen wurden, werden in diesem Kapitel beschrieben. Zunächst werden die Vorversuche zur Gehäuseherstellung erläutert, anschließend folgen die Richtlinien der Implementierung, die in der Benutzeroberfläche berücksichtigt wurden. Alle weiteren Abschnitte behandeln Besonderheiten bei der Implementierung selbst. Dazu gehören beispielsweise Ausschnitte der Benutzeroberfläche oder Algorithmen zur Generierung von Verbindungen.

6.1 Vorversuche und Entscheidungen

Bevor die Implementierung realisiert werden konnte, wurden einige Versuche durchgeführt, um praxistaugliche Parameter für Presspassungen je nach Material zu erhalten. Außerdem wurden Richtlinien für die Implementierung festgelegt. Die folgenden beiden Abschnitte erläutern die Vorüberlegungen und stellen die wichtigsten Parameter für die Erstellung von Presspassungen vor.

6.1.1 Verbindungen

Bis zur endgültigen Implementierung, wurden zunächst einige Versuchsreihen durchgeführt, um passende Parameter für Verbindungen zu erhalten. Alle hier genannten Werte sind experimentell ermittelt und haben sich in der Praxis bewährt, sind jedoch nicht allgemeingültig, da sie stark von Lasercutter und Material abhängen. Um eine Presspassung mit Holz (das Testobjekt besteht aus 4 mm dicken Pappelholz) zu erzeugen, hat sich ein Unterschied von 0,07 mm bei einer Laserbreite von 0,1 mm als praktikabel erwiesen. Bei der verwendeten Schwalbenschwanzverbindung ist der Zapfen am schmalsten Teil gleich breit wie die gegenüberliegende Nut (vergleiche Abbildung 2.12). Eine stabile Verbindung von Holzelementen kann optimal mit einem 80°-Winkel erreicht werden (Abbildung 6.1, rechts). Je nach Materialart sind bestimmte Verbindungen weniger geeignet. Beispielsweise neigen Schwalbenschwanzverbindungen für Acrylelemente dazu zu brechen (Abbildung 6.1, links).

Schraubenverbindungen können trotz relativ dünnem Material (4 mm) realisiert werden (Abbildung 6.2, links). Durch die Mutter ist der Halt zwischen beiden Elementen deutlich größer, als bei reinen Steckverbindungen, wie Finger- und Schwalbenschwanzverbindungen. Dass Materialien auch kombiniert werden können sieht man in Abbildung 6.2 rechts, bei dem sowohl Fingerverbindungen als auch eine Akkordeonverbindung umgesetzt sind.

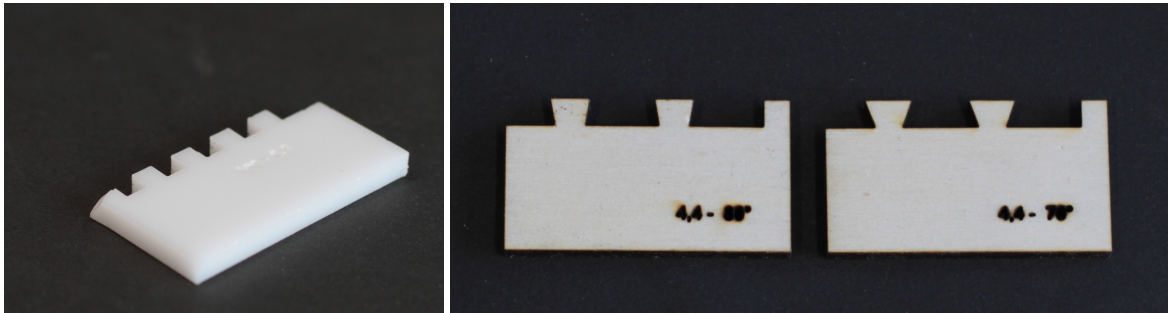


Abbildung 6.1: Acryl neigt bei mehrfacher (De-)Montage dazu zu brechen, wie im linken Bild am ersten Zapfen zu sehen ist. Rechts: Zwei Schwalbenschwanz-Prüflinge aus Holz, mit unterschiedlich geneigtem Winkel.

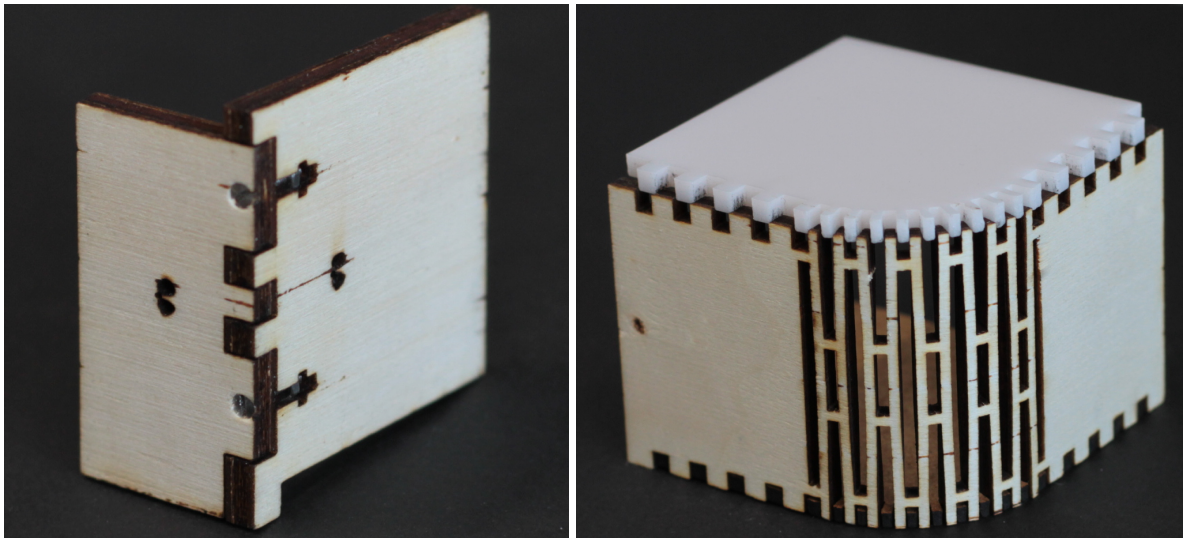


Abbildung 6.2: Die Schraubverbindung (links) wird durch eine Mutter fixiert. Für die 4 mm starke Holzplatte wurden 10 mm Senkkopfschrauben(DIN 963) M1,6 und die passende Mutter (DIN 934) verwendet. Rechts zu sehen ist sowohl die Akkordeonverbindung, welche die abgerundete Ecke ermöglicht, als auch eine Fingerverbindung, die Boden und Decke mit dem Mantel verbindet.

6.1.2 Richtlinien der Implementierung

Die Entwicklung der Software wird durch verschiedene Ergebnisse beeinflusst. Die Usability-Richtlinien, die von Lee et al. [LETH10] erörtert werden, dienen als Basis um die Anwendung benutzerfreundlich zu gestalten. Außerdem ist eine geringe Komplexität in der Anwendung essenziell, um die Einstiegshürde so gering wie möglich zu halten. Des Weiteren wurde darauf geachtet, dass die Benutzbarkeit eine hohe Priorität erhielt. Dies zeigt sich beispielsweise an der Größe von Touch-Objekten. Sears und Shneiderman sowie Hall et al. haben eine deutlich erhöhte Genauigkeit bei der Touch-Eingabe festgestellt, wenn die Objekte mindestens 26 mm bzw. 32 Pixel groß sind. Daher wird beim Erstellen neuer Formen darauf geachtet, dass diese Mindestgröße gegeben ist. Im Konflikt steht dies jedoch zu dem Wunsch die Befestigungslöcher für Schrauben der Gadgeteer-Komponenten zu integrieren, da diese nur einen Durchmesser von 3 mm haben. Die Eingabe für vorhandene Komponenten wird von der Software unterstützt, so dass die Löcher von Schrauben nicht manuell hinzugefügt werden müssen.

Auch bei der Entscheidung, ob der Eingabemodus häufiger implizit oder explizit gewählt werden sollte, kommt es zu Konflikten. Einerseits wird durch die implizite Eingabe das Bearbeiten komfortabler und schneller. Andererseits ist der Modus nicht selbsterklärend und der Benutzer bekommt weniger Feedback, welcher Modus gerade ausgeführt wird. Besonders für Neueinsteiger des Systems überwiegt jedoch das Feedback. Daher werden alle Aktionen explizit ausgeführt. Ein Ausnahme stellt das Auswählen von Elementen dar, was implizit durch einen einfachen Klick realisiert wurde. Um dem Benutzer weiteren Komfort zu bieten, wird das Verschieben automatisch ausgewählt, nachdem eine Komponente selektiert wurde.

Die Platzierung der Symbolleisten beeinflusst die Benutzbarkeit ebenfalls. Wurde sie anfangs auf der rechten Seite, unter den Textboxen platziert, so wurde sie am Ende nach unten verschoben. Besonders durch das große Display muss man den Fokus nun weniger weit verschieben um die Symbolleiste zu bedienen. Außerdem wurde das Auswählen der weiter entfernten Elemente auf Dauer ermüdend.

Beim Erstellen neuer Formen wurde auf Konsistenz geachtet. Beim Aufziehen des Rechtecks definiert der Berührungspunkt die obere linke Ecke. Für den Kreis erfolgt das Aufziehen auch von der linken oberen Ecke, anstatt beispielsweise aus der Mitte.

6.2 Natcut Application

Für die Implementierung der Hauptanwendung Natcut wurde C# gewählt. Da als Touch-Tisch Microsofts *Pixelsense*¹ (ehemals Surface 2.0) verwendet wird, bot es sich an, die Benutzeroberfläche mit der *Windows Presentation Foundation (WPF)* zu realisieren. Damit wird die Definition der Benutzeroberfläche mit XAML und die Realisierung der Logik in C# getrennt (vergleiche

¹<http://www.microsoft.com/en-us/pixelsense/default.aspx>

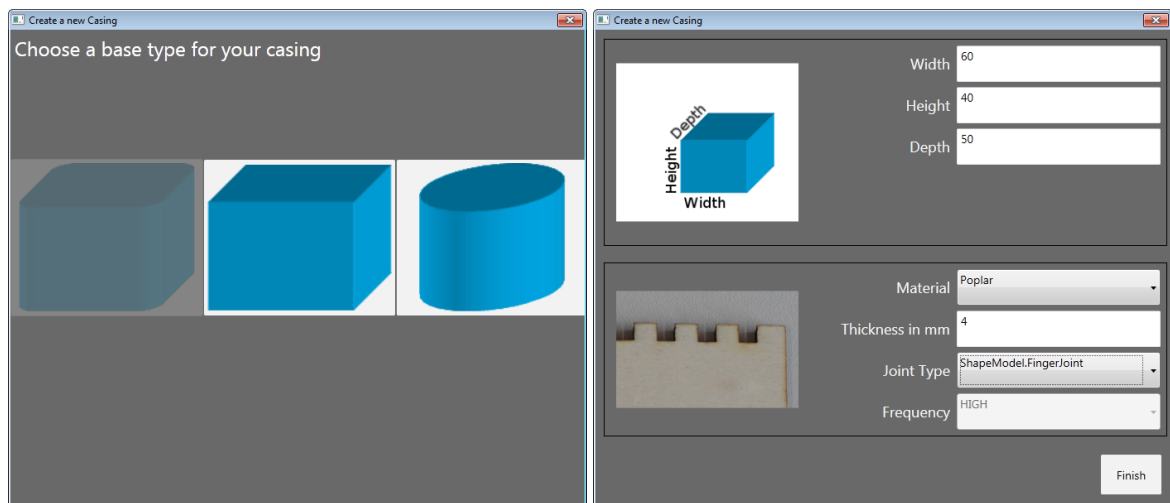


Abbildung 6.3: Der Wizard leitet die Eingabe des Benutzers. Zunächst wird die Grundform ausgewählt (links), anschließend werden alle Einstellungen vorgenommen (rechts). Nur bei gültigen Eingaben kann das Gehäuse erzeugt werden.

Abschnitt 5.2). Der Projekttyp *SurfaceApplication* (WPF) bietet eine gute Ausgangsbasis, um auf die Touch-Tisch-spezifischen Eigenschaften zuzugreifen. Mit diesem Typ werden zum Beispiel Buttons angeboten, die bereits auf Touch-Eingabe reagieren. Außerdem gibt die Benutzeroberfläche Feedback über Zieh-Bewegungen (leichter Schattenwurf), so dass es für den Benutzer einfacher wird, seine Eingabe und dessen genauer Ort nachzuvollziehen. Auch Fehleingaben können vom Benutzer besser verstanden werden. Um die Fehleingaben so gering wie möglich zu halten, werden alle Events von der Anwendung ignoriert, die keine Fingerberührung, Mausklick oder Markereingabe darstellen.

Der Benutzer wird mit einem Wizard angeleitet (siehe Abbildung 6.3), um ein neues Gehäuse zu erstellen. Zunächst wählt er die Grundform (linkes Bild). Anschließend legt er die Außenmaße, die Dicke des Materials, das Material selbst und die Verbindungsart fest. Für jede Verbindungsart wird ein Bild eingeblendet (rechtes Bild, links unten), das dem Benutzer die Unterschiede verdeutlicht. Die Frequenz ist eine Erweiterungsmöglichkeit, die es zukünftig erlauben soll, die Anzahl der Verbindungen pro Kante zu bestimmen. Sind alle Eingaben gültig, kann der Benutzer das Gehäuse hinzufügen.

Eine Übersicht der Benutzeroberfläche ist in Abbildung 6.4 zu sehen. Im Zentrum befindet sich der Bearbeitungsbereich der Gehäuse, der in Reiter gegliedert ist. Wenn eine Außenfläche nicht veränderbar ist (hier der Mantels des Zylinders), so wird diese schraffiert dargestellt. Dieser Fläche können weder Elemente hinzugefügt werden noch kann sie zur Personalisierung oder als Leinwand dienen. Über dem Bearbeitungsbereich befindet sich die Symbolleiste, die von links nach rechts, die folgenden Aktionen ermöglicht:

- Hinzufügen neuer Gehäuse
- Öffnen bestehender Gehäuse

- Aktion rückgängig
- Aktion wieder herstellen
- Gehäuse speichern
- Gehäuse herstellen (drucken)
- Gehäuse als SVG-Datei exportieren
- Eigenschaften-Dialog des Gehäuses öffnen

Beim Hinzufügen eines neuen Gehäuses öffnet sich der Wizard, der in Abbildung 6.3 zu sehen ist. Der Eigenschaften-Dialog des Gehäuses, ermöglicht es die Einstellungen, die beim Erstellen angegeben wurden, nachträglich anzupassen. Soll das Gehäuse hergestellt werden, so wird es zunächst exportiert und anschließend mit dem Standardprogramm geöffnet. Damit kann eine Vorschau, beispielsweise im Browser, geöffnet werden, bevor die Datei über den Standarddialog des Betriebssystems an den Lasercutter gesendet werden kann.

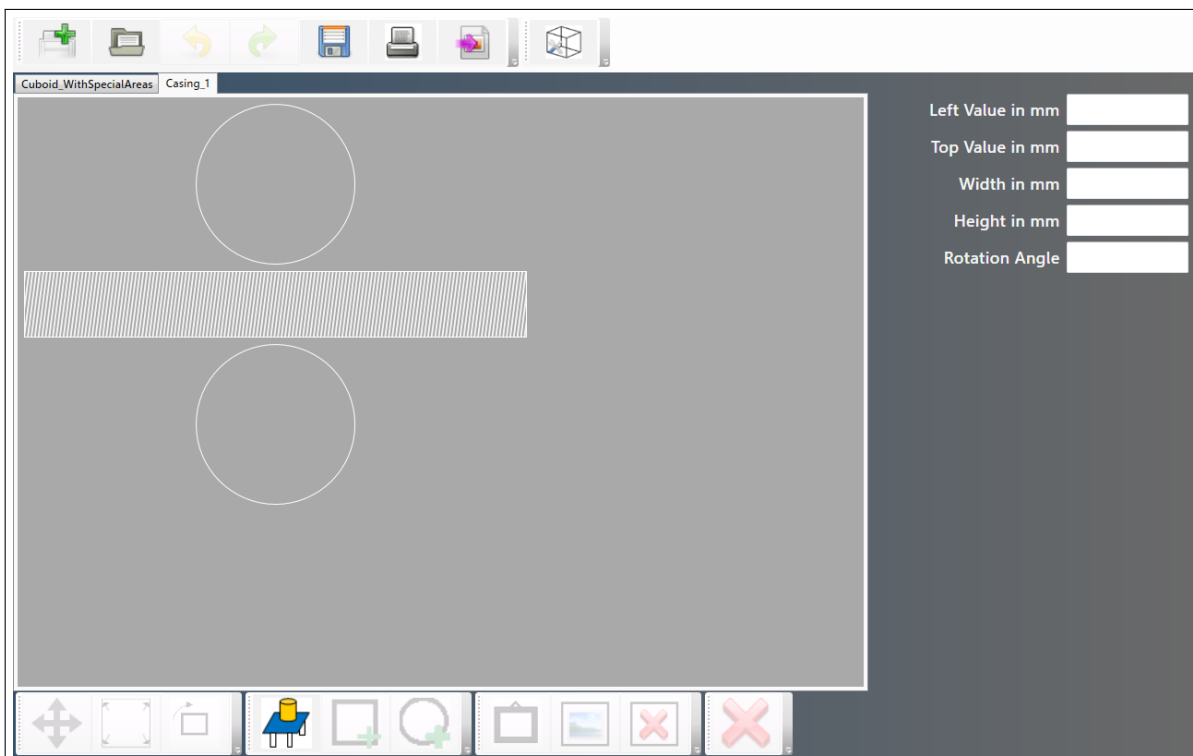


Abbildung 6.4: Die Hauptansicht der Benutzeroberfläche besteht aus zwei Symbolleisten oben und unten, dem Bearbeitungsbereich, der in Reiter gegliedert ist und den Eigenschaften einer Fläche, die als Textfelder auf der rechten Seite zu sehen sind.

Die untere Symbolleiste bezieht sich auf den Bearbeitungsbereich. Hier stehen folgende Funktionen zur Verfügung (von links nach rechts):

- Verschieben einer Fläche
- Ändern der Größe einer Fläche
- Rotieren einer Fläche
- Starten/Beenden der Objekterkennung
- Hinzufügen eines Rechtecks
- Hinzufügen eines Kreises
- Außenfläche als Leinwand
- Außenfläche als Bild
- Löschen der Elemente innerhalb einer Fläche (Bild oder Leinwand werden ebenfalls entfernt)
- Löschen einer Fläche

Sollen genaue Maßvorgaben erfüllt werden, so kann die Eingabe für Eigenschaften einer Fläche über die Textfelder (siehe rechte Seite der Abbildung 6.4) erfolgen.

In Abbildung 6.5 sind im unteren, linken Bild die drei Optionen für eine Außenfläche zu sehen: in der Mitte ist ein Bild (Urlaubsbild am Strand) eingefügt zur Personalisierung des Gehäuses, rechts daneben die Möglichkeit die Außenfläche als Leinwand zu verwenden (hellgrau hinterlegte Fläche) und die Standardvariante der übrigen Flächen, denen weitere Elemente hinzugefügt werden können (wie an der oberen Fläche zu sehen). Die Aussparungen für das Display der oberen Außenfläche wurden dabei mit Gadgeteer-Komponenten eingefügt, die von der Objekterkennung erfasst wurden. Das rechte Bild zeigt die aktivierte Objekterkennung (Bearbeitungsbereich ist blau hinterlegt) und den Umriss der erkannten Komponente, der als Feedback dient. Dieser Screenshot wurde mit Hilfe des *Input Simulators*² erzeugt, um das Feedback zu verdeutlichen. Normalerweise wäre nur der Umriss zu erkennen, da der Rest durch die physikalische Komponente bedeckt wird. Ein Beispiel mit aufgelegtem Gadgeteer-Bauteil ist im Abschnitt 7 abgebildet.

6.3 ObjectDetection

Die ObjectDetection-Komponente wurde ebenfalls auf Basis einer SurfaceApplication (WPF) mit C# implementiert. Da während der Implementierung keine Besonderheiten auftraten, wird hier auf den entsprechenden Abschnitt im Entwurfskapitel verwiesen, in dem die wesentlichen Entscheidungen getroffen wurden.

²Der Input Simulator ist im Microsoft Surface 2.0 SDK enthalten. Weitere Informationen sind unter <http://msdn.microsoft.com/en-us/library/ff727911.aspx> zu finden.

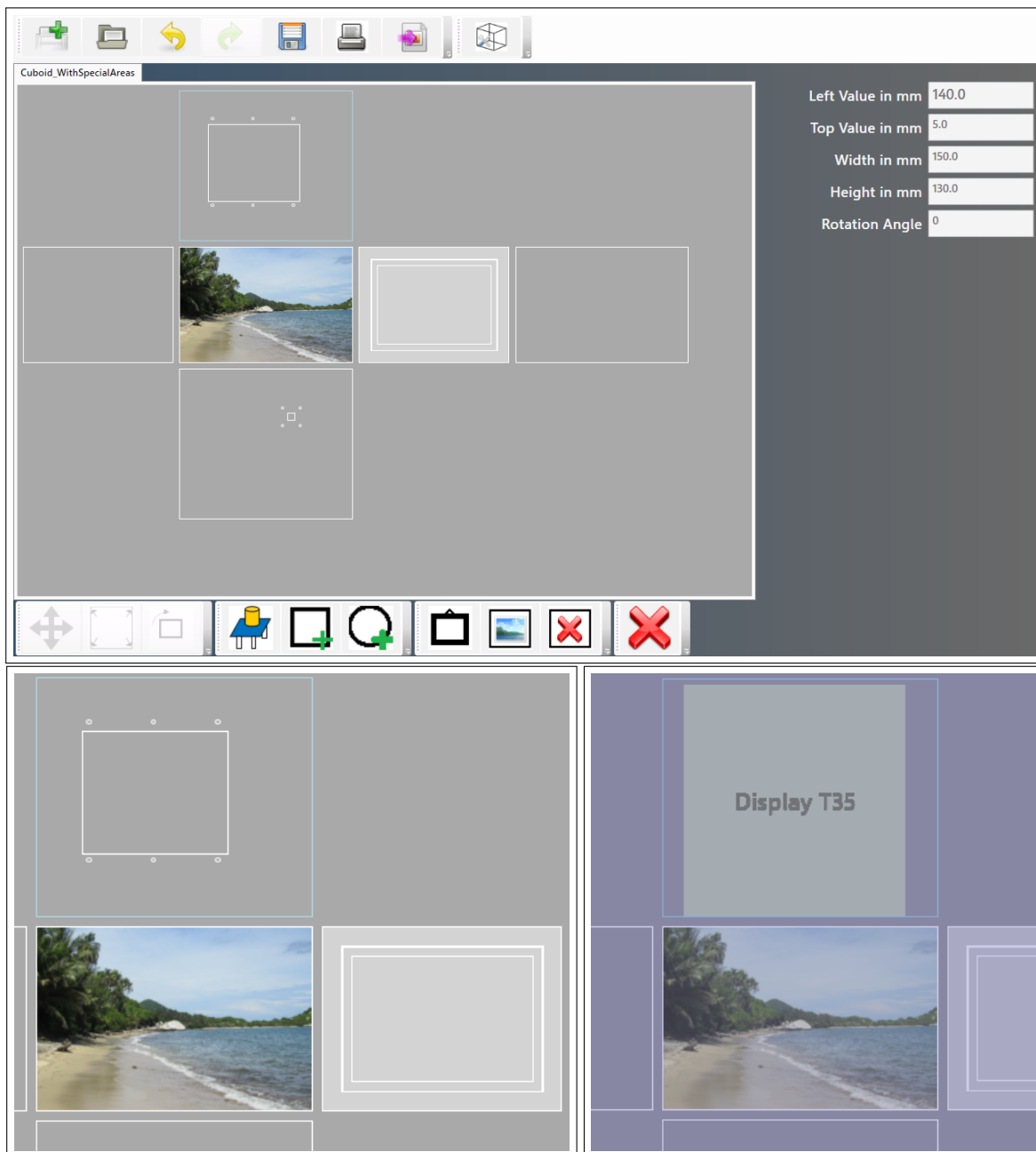


Abbildung 6.5: Oben: Die Hauptansicht im Überblick; Unten: Im linken Bild sind die drei Möglichkeiten für eine Außenfläche zu sehen: die Personalisierung per Bild (Urlaubsfoto), eine Leinwand (Fläche rechts daneben) und eine leere Fläche, der Komponenten hinzugefügt werden können. Das rechte Bild zeigt die Oberfläche im Objekterkennungsmodus, bei dem gerade ein Display aufgelegt wurde (hier ist nur der Umriss zu sehen, welchen das System als Feedback zeigt, nicht aber die aufgelegte Komponente).

6.4 SVG-Generator

Der SVG-Generator wurde als C#-Konsolenanwendung implementiert, die als ausführbare Datei und als dll-Bibliothek zur Verfügung steht. Um die Parameterverarbeitung der Konsole zuverlässig und einfach umzusetzen wurde die NDesk-Bibliothek³ verwendet. Mit einfachen Regeln können die Parameter, die via Konsole übergeben werden, überprüft und eine Anzeige aller verfügbaren Parameter erstellt werden. Ein kurzer Ausschnitt aus einer generierten Beispieldatei zeigt die Details:

Listing 6.1 Ausschnitt einer generierten SVG-Datei, welche die geometrischen Flächen eines Gehäuses beschreibt. Durch die Viewbox wird die Grafik auf Realgröße skaliert. Alle Kind-Elemente unter dem g-Element werden gruppiert.

```
<svg xmlns="http://www.w3.org/2000/svg" Version="1.1" viewBox="0 0 200.2 80.2"
  width="200.2mm" height="80.2mm">
  <g id="LEFT">
    <polyline points="35.1,8.9 ..." style="fill:none;stroke:black;stroke-width:0.1" />
  </g>
</svg>
```

Durch die Attribute `viewBox`, `Width` und `Height` transformiert der SVG-Generator die Koordinaten von Pixel, dem Standardwert in SVG-Grafiken, zu der im Objektmodell spezifizierten Maßeinheit. Das `g`-Element gruppiert alle Kind-Elemente. Hier ist exemplarisch nur eines vorhanden, für eine rechteckige Fläche wären dies entsprechend drei weitere Linien. Der Bezeichner `Id` hilft der Orientierung und Zuordnung der Parameter eines Shape-Objekts zur generierten SVG-Datei. Das `Polyline`-Element beschreibt eine zusammenhängende Linie, die durch Aufzählung von Punkten definiert wird.

6.4.1 CasingModel

Das Objektmodell (siehe auch in Abschnitt 5.4 und Anhang A.1) spielt eine zentrale Rolle und wird sowohl vom SVG-Generator als auch von der Hauptanwendung verwendet. Es ist im SVG-Generator definiert und bietet der Hauptanwendung den Zugriff darauf an. Da das Objektmodell durch eine XSD-Datei beschrieben ist, bietet sich die automatische Code-Generierung an. Das Tool `XSD2Code`⁴ bietet diese Möglichkeit und lässt sich nahtlos in Visual Studio integrieren. Damit werden alle C# Klassen generiert, welche die Eigenschaften eines Gehäuses vorhalten. `XSD2Code` unterliegt einigen Einschränkungen bei der Parametrierung, um einen reibungslosen Ablauf zu erreichen. Alle Eigenschaften dazu können im Anhang A.4 gefunden werden.

³<http://ndesk.org>

⁴<http://xsd2code.codeplex.com/>

Um das Arbeiten mit den generierten Klassen zu erleichtern, wurden einige Konstruktoren hinzugefügt, die jedoch in einer separaten Klasse ausgelagert sind. Dort erfolgt die Definition und Initialisierung von Gehäuseaußenseiten und die Zuordnung zu ihren Listenpositionen.

6.4.2 CasingController

Um Klarheit bei der folgenden Idee zu schaffen, wird zuerst definiert, was unter einer Nut-Linie (Slotline) und was unter einer Zapfen-Linie (Keyline) verstanden wird. Die Funktionsweise ist bekannt aus dem *Nut und Feder-Prinzip*. Da die Feder aber nicht durchgängig ist und mehrmals vorkommt, wird hier der Begriff Zapfen verwendet.

Nut-Linie Eine Nut-Linie weist mehr Nuten auf, als Zapfen. Die äußersten Elemente werden dabei nicht mitgezählt.

Zapfen-Linie Eine Zapfen-Linie weist mehr Zapfen auf, als Nuten. Die äußersten Elemente werden dabei nicht mitgezählt.

Auch wenn es zunächst irritierend sein mag, dass die äußeren Elemente ignoriert werden und so eine vermeintliche Zapfen-Linie zur Nut-Linie wird, so leuchtet diese Definition ein, wenn man Linien mit minimalen Nuten beziehungsweise Zapfen betrachtet. Eine Linie mit einer Nut benötigt zwei äußere Elemente, die sie umgeben. Eine Nut besteht also nur dann, wenn die zwei Elemente, die sie umgeben, erhaben sind. Analog gilt für den Zapfen, dass seine zwei ihn umgebenden Elemente niedriger sein müssen.

Bei der Erstellung eines Quaders werden drei verschiedene Seiten jeweils zwei Mal benötigt (siehe Abbildung 6.6). Diese drei Seiten ergeben sich aus Kombinationen von Zapfen- und Nut-Linien.

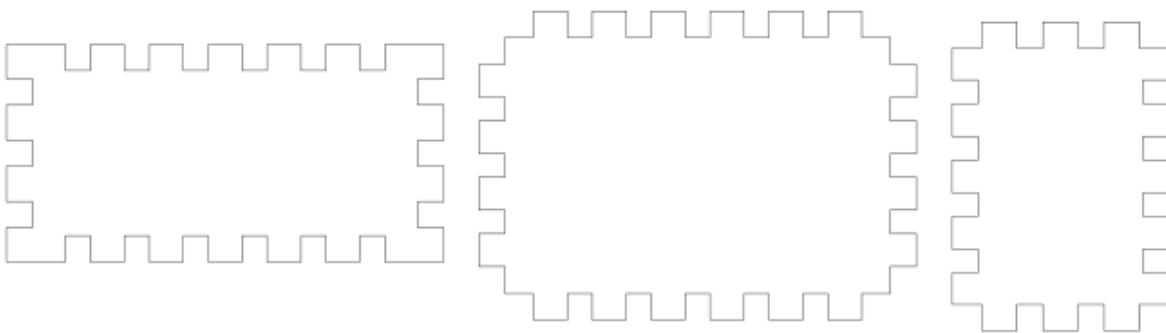


Abbildung 6.6: Die drei verschiedenen Flächen eines Quaders werden aus Kombination von Zapfen- und Nut-Linien generiert. Der CasingController bestimmt, welche Fläche durch welche Kombination realisiert wird.

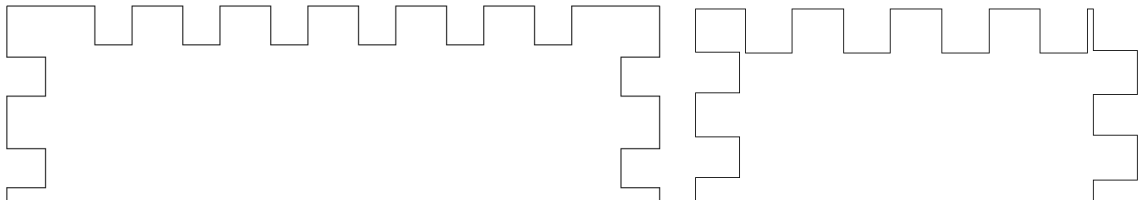


Abbildung 6.7: Links: die äußeren Zapfen der oberen Linie sind stabil; Rechts: Zwar ist der äußerste linke Zapfen der oberen Linie an sich breit genug, die Verbindung des Zapfens zur Gesamtlänge ist jedoch nicht ausreichend. Der rechte äußere Zapfen ist auf Grund seiner geringen Breite instabil. Der linke Fall wird durch den ShapeController verhindert, der rechte durch die Implementierung des Quaders, siehe Abschnitt 6.4.3.

Eine Besonderheit tritt bei der Kombination von Ecken auf, die aus zwei Nut-Linien gebildet wird. Wie in Abbildung 6.7 zu sehen, ist die Stabilität der äußersten Elemente nicht garantiert, wenn die Breite des Zapfens der Materialstärke entspricht. Um dem entgegenzuwirken wird die Stabilität von äußeren Zapfen (Nut/Nut-Linien) erhöht, indem die Zapfen mindestens die zweifache Materialstärke aufweisen. Somit bleibt die einfache Materialstärke, um den Zapfen zur Fläche zu verbinden. Für innere Zapfen tritt dieses Phänomen nicht auf, da die Verbindung zwischen Zapfen und Fläche durch keine andere Linie beeinflusst wird.

6.4.3 Fingerverbindung

Bei der Generierung eines Fingerverbunds wird aus Stabilitätsgründen darauf geachtet, dass kein Zapfen schmaler als die Materialstärke ist (als Gegenbeispiel siehe Abbildung 6.7 rechtes Bild, rechter äußerer Zapfen). Eine Möglichkeit ist die Breite eines Zapfens durch einen ganzzahligen Abstand in Millimeter zu wählen und einen Ausgleichsteil in der Mitte zu verwenden. Bei dieser Möglichkeit ist die Breite der Zapfen und Nuten äquidistant, mit Ausnahme des Ausgleichteils. Alternativ kann man mit komplett äquidistanten Abständen arbeiten, was aber zu krummen Zahlen als Abstand führen kann. Da Letzteres optisch einheitlicher aussieht und für die Herstellung ein Laser verwendet wird, der sehr exakt arbeitet, bietet es keinen nennenswerten Vorteil, nur ganzzahlige Abstände zu erlauben.

Die notwendigen Berechnungen und Definitionen zur Bestimmung der Länge von Nut und Zapfen sind in Listing 6.2 zu finden. Der Unterschied zwischen maximaler Nuten- und Zapfenzahl `tabNumberMax` (Zeile 9) und der Tatsächlichen `tabNumberReal` (Zeile 10) kommt daher, dass die Summe von Nuten und Zapfen eine ungerade Zahlen sein muss. Damit ist jede Linie, die sich aus Nuten und Zapfen zusammensetzt, achsensymmetrisch. Die Anzahl der Zapfen und Nuten soll in Zukunft über die Frequenz beeinflusst werden können (siehe Abschnitt 6.2). Die Frequenz stellt einen Faktor dar, der abhängig von der maximalen Nuten- und Zapfenzahl, beispielsweise 50% weniger Zapfen und Nuten (und damit doppelt so breite Elemente) erlaubt.

Um eine Presspassung zu erzeugen, wird nun die Breite der Zapfen erhöht, die der Nuten verringert (Zeile 32 und 46). Dieser Wert ist materialabhängig und kann in den Einstellungen (siehe Abschnitt 5.2) festgelegt werden.

6.4.4 Akkordeonverbindung

Bei Akkordeonelementen steht die Biegsamkeit der Stabilität gegenüber. Je geringer die Breite eines Elementes gewählt wird, um so geringer kann der Kurvenradius ausfallen. Dies geschieht jedoch auf Kosten der Stabilität. Für dieses Projekt wurde der Abstand der Elemente in der Akkordeonverbindung gleich der halben Materialstärke gewählt. Momentan gibt es zwei Verbindungen pro Element (siehe blaue Markierung Abbildung 6.8). Soll zukünftig eine flexible Anzahl erlaubt werden, muss die Implementierung angepasst werden. Das Objektmodell hingegen erlaubt bereits eine flexible Anzahl festzulegen. Eine weitere Variation ist die Form eines Verbindungselements. Die implementierte Lösung verwendet gerade Übergänge mit 90° -Winkeln, denkbar ist jedoch auch eine abgerundete Version. Auch hier ist das Objektmodell auf Änderungen bereits vorbereitet.

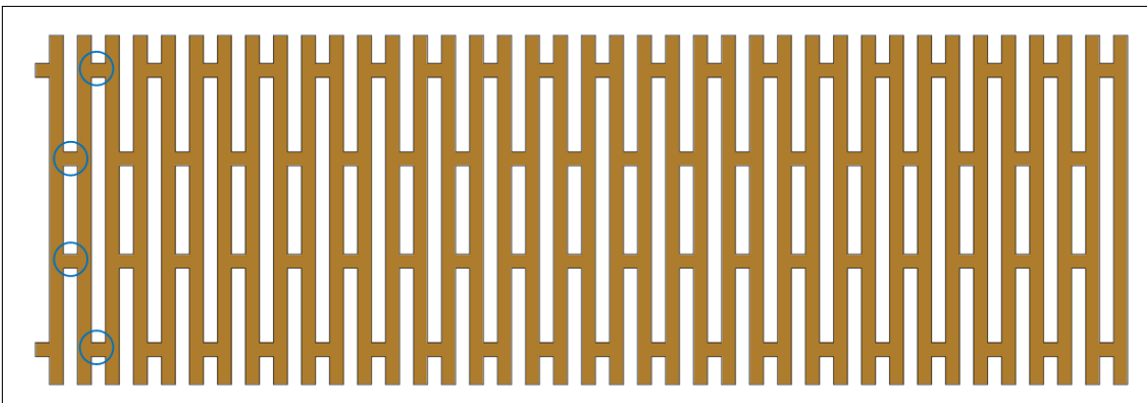


Abbildung 6.8: Ein Akkordeon-Modell, das als Mantel eines Zylinders verwendet wird. Die Markierungen zeigen, dass jedes Element genau zwei Verbindungen zu seinem Nachbarn hat.

Listing 6.2 Algorithmus um die Länge von Nut und Zapfen einer Fingerverbindung zu bestimmen.

```
1 //
2 // Berechnet die durchschnittliche Länge eines Nut- und Zapfenelements.
3 //
4 // sidelength - Seitenlänge, über die sich Nut und Zapfen erstrecken
5 // thickness - Materialstärke des Gehäuses
6 private double GetTabWidth(double sidelength, double thickness)
7 {
8     double keyMinLength = thickness;
9     int tabNumberMax = (int)Math.Floor(sidelength / keyMinLength);
10    int tabNumberReal = tabNumberMax;
11
12    if (tabNumberReal % 2 == 0)
13    {
14        tabNumberReal = tabNumberReal - 1;
15    }
16
17    double tabLength = sidelength / tabNumberReal;
18
19    return tabLength;
20 }
21
22 //
23 // Berechnet die Breite eines Nutelements unter Berücksichtigung
24 // der Laserstrahlbreite.
25 //
26 // sidelength - Seitenlänge, über die sich Nut und Zapfen erstrecken
27 // thickness - Materialstärke des Gehäuses
28 // interference - Maß mit dem die Presspassung erzeugt wird (in mm)
29 // laserOffset - Radius des Laserstrahls
30 public double GetSlotWidth(double sidelength, double thickness, double interference, double
    laserOffset)
31 {
32     double slotWidth = GetTabWidth(sidelength, thickness) - interference - 2 * laserOffset;
33     return slotWidth;
34 }
35
36 //
37 // Berechnet die Breite eines Zapfenelements unter Berücksichtigung
38 // der Laserstrahlbreite.
39 //
40 // sidelength - Seitenlänge, über die sich Nut und Zapfen erstrecken
41 // thickness - Materialstärke des Gehäuses
42 // interference - Maß mit dem die Presspassung erzeugt wird (in mm)
43 // laserOffset - Radius des Laserstrahls
44 public double GetKeyWidth(double sidelength, double thickness, double interference, double
    laserOffset)
45 {
46     double keyWidth = GetTabWidth(sidelength, thickness) + interference + 2 * laserOffset;
47     return keyWidth;
48 }
```

7 Explorative Benutzerstudie

Um das Konzept, das im entsprechenden Kapitel vorgestellt wurde, zu verfeinern, wurde eine Benutzerstudie durchgeführt. Diese legt den Fokus auf den Schritt das Gehäuse zu entwerfen, verzichtet aber auf die physikalische Herstellung und den Zusammenbau durch die Studienteilnehmer. Die Software, die das Entwerfen ermöglicht, ist in Kapitel Entwurf und Implementierung beschrieben. Im Folgenden werden die Ziele der Benutzerstudie, der Versuchsaufbau, das Teilnehmerfeld, ihr Ablauf und die erzielten Ergebnisse näher beschrieben.

7.1 Ziele

Da sich das Projekt noch in einem frühen Entwicklungsstatus befindet, wurde eine explorative Studie gewählt, um qualitative Aussagen über die Ideen, die dem Konzept zu Grunde liegen, zu evaluieren. Dabei wurden folgende Ziele untersucht:

1. Umgang mit Tangibles zur einfachen Erstellung mit bereits existierenden Gegenständen.
2. Generierung der SVG-Grafiken der entworfenen Gehäuse.
3. Erwartungen an den Prototyp durch Vorkenntnisse in CAD oder Grafikprogrammen.

Um Aussparungen für vorhandene Gegenstände in einem Gehäuse vorzunehmen, wurden diese Gegenstände als Tangibles angeboten. Dabei wurde ihre Handhabung, das Positionieren und Interagieren mit dem Prototyp und die allgemeine Akzeptanz dieser Methode untersucht. Am Ende der Entwurfsphase wurde aus Informationen über Gehäusedimension, Position von Gegenständen und Verbindungstypen die SVG-Grafik generiert, die zur Herstellung des physikalischen Gehäuses dient. Der Generator wurde damit unter verschiedenen Bedingungen in einem realitätsnahen Szenario geprüft.

Jeder Benutzer bringt inhärente Erwartungen mit, was die Gestaltung des Prototyps und sein Verhalten betrifft. Besonders markant sind hierbei die Bereiche CAD und Grafikdesign, die ausgeprägte Arbeitsmethoden aufweisen und deren Ablauf dem Konzept dieser Arbeit thematisch ähnlich sind.

7.2 Versuchsaufbau

Dem Teilnehmer standen während der Benutzerstudie einige Gadgeteer-Komponenten zur Interaktion zur Verfügung (siehe Abbildung 7.1). Alle Komponenten waren an der Unterseite mit einem Marker ausgestattet, um die Erkennung der Objekte durch den Touch-Tisch zu erleichtern. Zur Auswahl standen:

- Display T35
- Joystick (2x)
- Button
- LED
- Kamera
- Drehregler

Als Multitouch-Tisch, auf dem die Software ausgeführt wurde, kam Microsofts PixelSense (ehemals Surface 2.0) zum Einsatz¹. Die Auflösung des Displays beträgt 1920 x 1080, die Auflösung der Infrarot-LEDs, die für die Touch-Eingabe und Marker-Erkennung zuständig sind, beträgt 1024 x 768 Pixel.

Die Studie wurde unter Berücksichtigung der Lichtverhältnisse für den PixelSense durchgeführt. Der Raum war abgedunkelt, um Störungen bei der Eingabeerkennung durch direkte Lichtquellen zu vermeiden. Lediglich eine gedimmte, indirekte Beleuchtung erhellte die Umgebung ohne dabei die Erkennungsrate des Touch-Tischs zu beeinflussen.

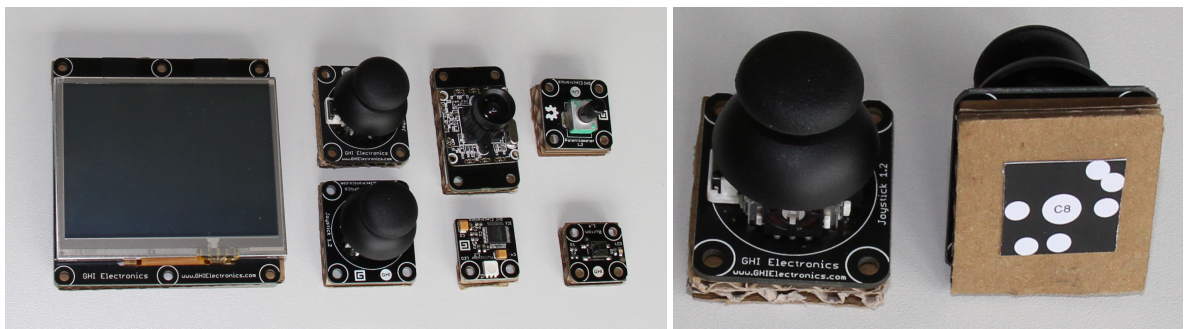


Abbildung 7.1: Alle Gadgeteer-Komponenten, die dem Teilnehmer zur Verfügung gestellt wurden, sind auf dem linken Bild zu sehen. Daneben ist ein angebrachter Marker abgebildet, mit dessen Hilfe die Erkennung zuverlässiger gestaltet wurde.

¹<http://www.microsoft.com/en-us/pixelsense/default.aspx>

7.3 Teilnehmerfeld

Die Benutzerstudie umfasste acht Teilnehmer, von denen einer weiblich und sieben männlich waren. Die Teilnehmer waren zwischen 19 und 27 Jahre alt (im Durchschnitt 24,4 Jahre mit einer Standardabweichung von 2,7 Jahren). Sieben von acht sind in technischen Studiengängen oder Berufen tätig, die achte Person in der Werbebranche. Ebenfalls gaben sieben von acht an, Touchdisplays mehrmals täglich zu benutzen, so dass von einem vertrauten Umgang mit Touch-Geräten ausgegangen werden kann. Die Hälfte der Teilnehmer hatte Kenntnisse im CAD Bereich und sechs von acht besaßen Kenntnisse im Umgang mit Grafikprogrammen wie Photoshop, Corel oder Gimp.

7.4 Ablauf

Die Dauer der Studie pro Teilnehmer betrug zwischen 30 und 45 Minuten, wobei Unterstützung angeboten wurde, wenn Schwierigkeiten auftraten. Nach der Begrüßung der Teilnehmer wurde ihnen der Zweck der Studie und die allgemeine Aufgabe, Gehäuse für einen Prototypen zu entwerfen, erläutert. Im Anschluss wurden sie darüber aufgeklärt, dass sie die Studie zu jedem Zeitpunkt abbrechen oder eine Pause einlegen können. Die Teilnehmer unterschrieben eine Einverständniserklärung, dass die Daten, die während der Studie gesammelt wurden, zur Auswertung verwendet werden durften. Anschließend wurden die Teilnehmer in die wichtigsten Möglichkeiten der Software eingeführt. Die Demonstration beinhaltete die Wahl des Gehäusetyps, die nachträgliche Anpassung der Gehäusedimension, das Erkennen von Gadgeteer-Komponenten und die Möglichkeit Flächen durch Auswahl von Bildern zu personalisieren.

Um Aussparungen für Gadgeteer-Komponenten einzufügen, muss zunächst der Modus für die Komponentenerkennung aktiviert werden. Anschließend können die Gadgeteer-Komponenten auf dem Touch-Tisch platziert werden. Ist die gewünschte Position und Ausrichtung erreicht, wird der Modus für die Komponentenerkennung durch Knopfdruck beendet und entsprechend der aktuellen Lage werden die Komponenten eingefügt (siehe Abbildung 7.2). Jede Komponente kann pro Gehäuse nur einmal verwendet werden. Dafür können jederzeit die Komponenten, die bereits eingefügt wurden, verschoben oder gedreht werden, in dem man sie erneut auf dem Touch-Tisch platziert. Die vorherigen Aussparungen werden gelöscht und an der aktuellen Position wieder eingefügt. Als Anschauungsobjekt war eine Kamera vorhanden, stellvertretend für die Zylinderform und eine Spielekonsole als Repräsentant für eine Quaderform (siehe Abbildung 7.3).

Die erste Aufgabe der Teilnehmer bestand darin die Kamera nachzubauen. Dazu wurde ihnen der Prototyp ausgehändigt und die nötigen Dimensionen über Durchmesser und Höhe sowie die Dicke des Materials mitgeteilt. Das Gehäuse der Kamera besaß Aussparungen für ein Display, eine LED, eine Kamera und einen Button, die auf Vorder- und Rückseite verteilt waren. Die zweite Aufgabe bestand darin eine Spielekonsole zu entwerfen. Dabei wurden keine weiteren Vorgaben gemacht, um die Kreativität der Teilnehmer zu fordern. Alle angebotenen Gadgeteer-Komponenten konnten dabei eingesetzt werden. Ob die Komponenten

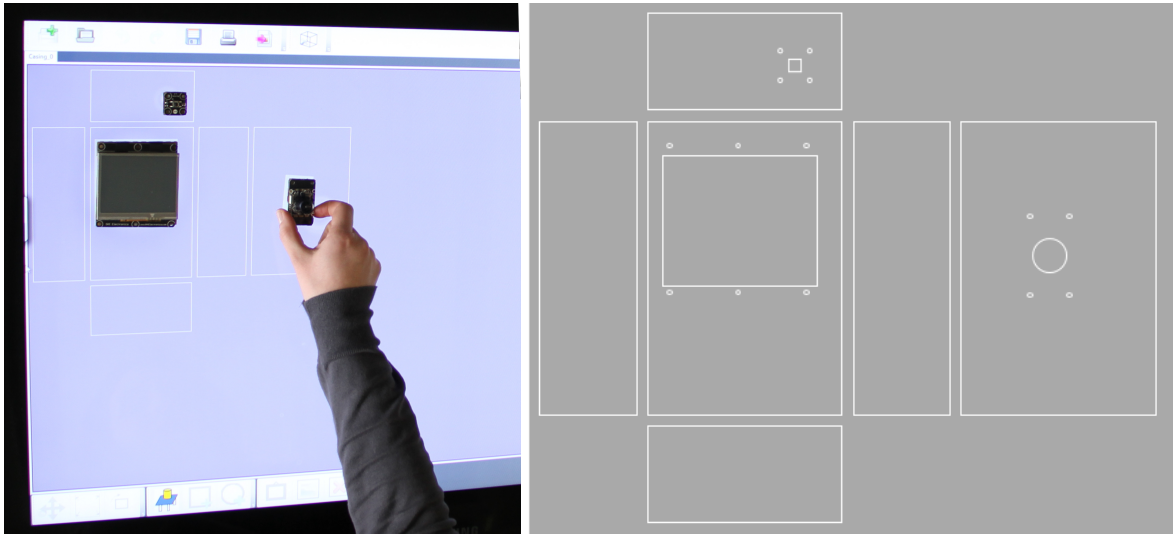


Abbildung 7.2: Im Objekterkennungs-Modus (blau hinterlegt) können Gadgeteer Komponenten als Vorlage verwendet werden (linkes Bild). Nach dem die Objekterkennung abgeschlossen ist, werden die Aussparungen an die passende Position eingefügt werden (rechtes Bild).

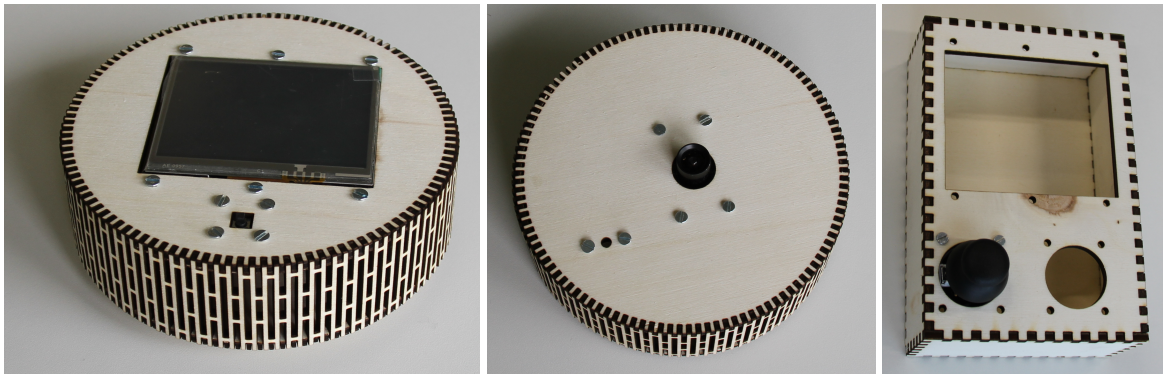


Abbildung 7.3: Eine Kamera als Beispiel für eine Zylinderform mit eingebautem Display und Button (links). Daneben ist die Rückseite zu sehen, an der Kamera und LED angebracht sind. Auf der rechten Seite ist ein Nachbau einer Spielekonsole als Beispiel für einen Quader mit Fingerverbindungen.

nacheinander oder mehrere auf einmal aufgelegt wurden, war dem Teilnehmer überlassen. Auch das manuelle Hinzufügen von Formen war erlaubt. Abschließend füllten die Teilnehmer einen Fragebogen (siehe Anhang A.4) aus und wurden in einem strukturiertem Gespräch über Verbesserungen und Eindrücke befragt.

7.5 Ergebnisse

Vier der acht Teilnehmer verfügten über CAD-Kenntnisse. Sie sollten einschätzen, wie lange sie zur Erstellung des Gehäuses, das sie entworfen hatten, mit einem CAD-Programm benötigen würden. Als Vorlage diente dabei die SVG-Grafik, die Finger- (siehe Abbildung 7.4) und gegebenenfalls Akkordeonverbindungen beinhalten. Die Spanne der Kamera lag zwischen 3 und 90 Minuten, die der Spielekonsole zwischen 5 und 75 (siehe Tabelle 7.1). Für die Erstellung mit einem Grafikprogramm, gaben die Teilnehmer an zwischen 5 und 120 Minuten für die Spielekonsole und zwischen 5 und 150 Minuten für die Kamera zu benötigen (Tabelle 7.2). Im Vergleich hat die Erstellung der Gehäuse im Experiment nicht länger als 10 Minuten gedauert. Damit zeigt sich, dass das Ziel, ein Gehäuse innerhalb von wenigen Minuten zu erstellen, gelungen ist. Außerdem hat eine kurze Erklärung ausgereicht, um die Funktionen des Prototyps zu erklären, wohingegen die Einarbeitungszeit für CAD-Programme und Grafik-Programme deutlich länger ausfällt.

Die Ergebnisse der Studienteilnehmer belegt auch, dass das Konzept gut angenommen wird (siehe Abbildung 7.5). Sieben von acht Teilnehmern stufen die Eingabe mit elektronischen Bauteilen als hilfreich oder sehr hilfreich ein. Nur ein Teilnehmer hat sie als wenig hilfreich empfunden, was auf eine zu geringe Zuverlässigkeit bei der Erkennung der Komponenten

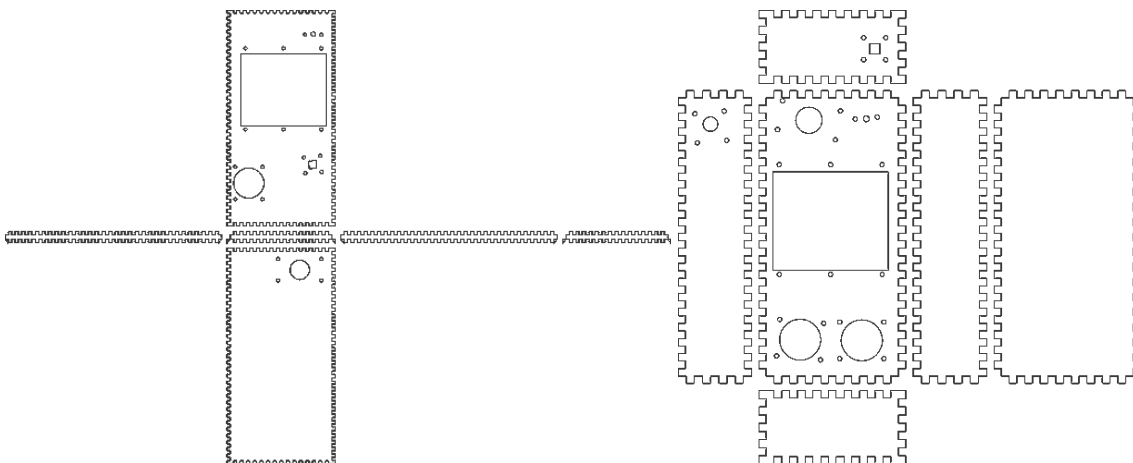


Abbildung 7.4: Die Grafiken zeigen Gehäuse von Spielekonsolen, die von Teilnehmern während der Benutzerstudie entworfen wurden. Dabei wird auch der Unterschied deutlich, welche Fläche Teilnehmer als Front verwenden (obere Fläche im linken Bild, mittlere Fläche im Rechten).

Teilnehmer	Spielekonsole [min]	Kamera [min]
2	5	3
3	5	5
4	15	90
5	75	45

Tabelle 7.1: Geschätzte Dauer der Erstellung eines Gehäuses inklusive Verbindungen mit einem CAD-Programm.

Teilnehmer	Spielekonsole [min]	Kamera [min]
1	5	5
3	5	10
4	15	60
6	60	120
7	60 – 120	60 – 120
8	120	150

Tabelle 7.2: Geschätzte Dauer der Erstellung eines Gehäuses inklusive Verbindungen mit einem Grafikprogramm.

zurückzuführen ist. Die Teilnehmer haben insbesondere Vorteile beim Abschätzen der Dimensionen gesehen. Durch die physikalische Vorlage konnten Kollisionen direkt erkannt werden. Außerdem gaben alle Teilnehmer an, dass die Vorlage bei der Vorstellung des Gehäuseentwurfs geholfen hat. Auch die Möglichkeit die Gehäuse durch Bilder zu personalisieren, kam gut bei den Teilnehmern an. Über die Hälfte bewerteten das Personalisieren mit gut oder ausgezeichnet (rechte, obere Grafik in Abbildung 7.5). Diese Möglichkeit kann noch verbessert werden, wenn ein fixes Seitenverhältnis oder die Auswahl eines Bildausschnittes angeboten wird.

Der Gesamteindruck des Systems wurde überwiegend positiv aufgefasst (siehe Abbildung 7.5, unten). Eine niedrige Erkennungsrate des Touch-Tischs hat jedoch eine bessere Beurteilung verhindert. Da die Marker auf einem Karton angebracht wurden, lag die Oberfläche nicht immer plan auf dem Touch-Tisch. Für weitere Studien kann die Fehlerrate verringert werden, wenn ein stabileres Material wie Holz zum Einsatz kommt. Dennoch bleibt die Erkennungsrate des Touch-Tischs auch damit beschränkt. Dass sich viele Benutzer Komfort-Features wie das Ausrichten von Flächen relativ zur Grundfläche oder das Einrasten an einem Raster wünschen, zeigt sich in der rechten, unteren Grafik der Abbildung 7.5. Für die weitere Entwicklung des Prototyps werden diese Features schnellst möglich implementiert. Weitere Details über Verbesserungsmöglichkeiten sind im Ausblick zu finden.

Bei dem Einsatz von Microsofts PixelSense traten neben dem Erkennen von Markern weitere Schwierigkeiten auf. Zunächst mussten die Lichtverhältnisse angepasst werden (siehe auch Abschnitt 7.2). Auch die direkte Eingabe, beispielsweise das Aktivieren von Buttons,

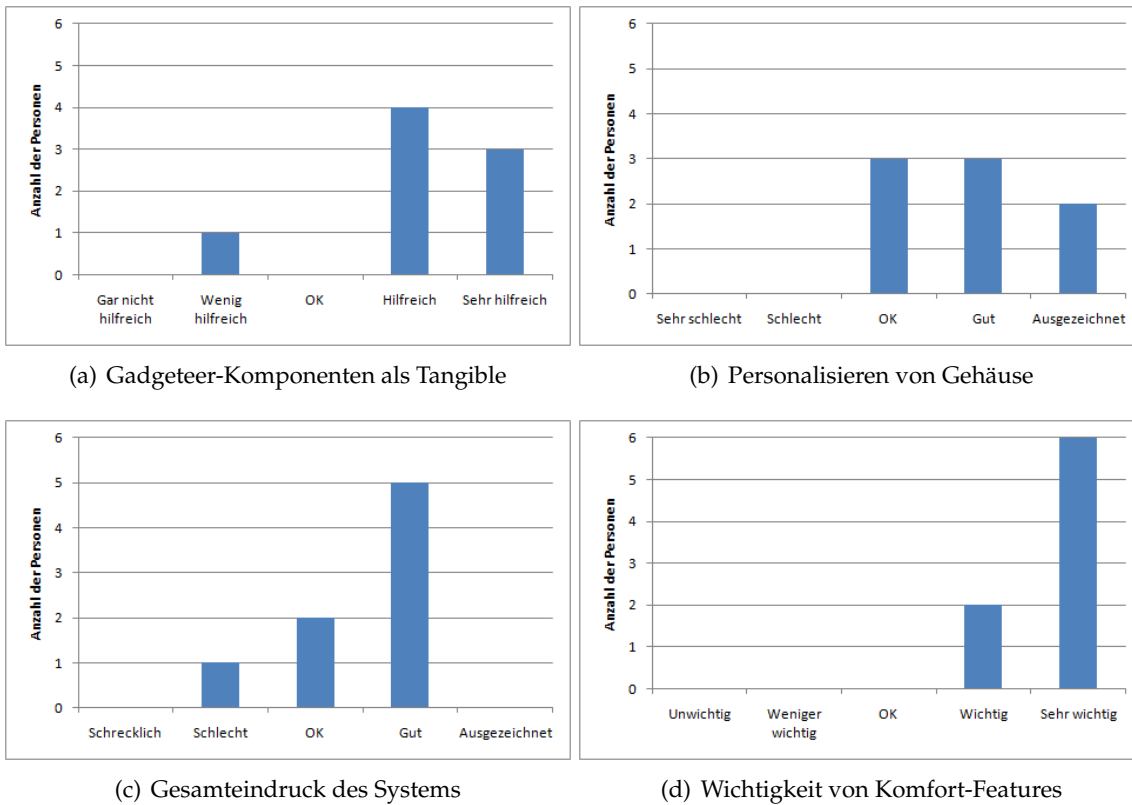


Abbildung 7.5: Feedback der Teilnehmer während der Benutzerstudie. (a): 7 von 8 Teilnehmern befanden die Eingabe mit Tangibles und die physikalische Repräsentation als hilfreich oder sehr hilfreich. (b): Ebenso wurde die Möglichkeit zur Personalisierung von Gehäusen mit Hilfe von Bildern, die eingraviert werden können, von allen Teilnehmern positiv oder zumindest neutral aufgefasst. (c): Der Gesamteindruck des Systems wurde überwiegend positiv aufgefasst, wobei die Erkennungsrate bei Touch-Eingaben eine bessere Bewertung verhindert hat. (d): Alle Teilnehmer hingegen schätzen weitere Features wie das Ausrichten an einem Raster oder in Relation zu anderen Komponenten als wichtig, 6 von 8 sogar als sehr wichtig ein.

benötigte in einigen Fällen mehrere Versuche. Außerdem wurden bei der Interaktion mit dem Touch-Tisch einige Fehleingaben produziert. PixelSense interpretierte dabei bereits Finger als Eingabe, die nahe an der Oberfläche waren, den Tisch aber noch nicht berührten.

Abschließend lässt sich festhalten, dass die Erwartungen und die Vorgehensweise der Teilnehmer durch ihrer Vorkenntnisse beeinflusst wurden. Personen, die über CAD-Kenntnisse verfügten, wünschten sich als zukünftige Features das Extrudieren oder die Subtraktion von Formen. Ebenso wurde beim Erstellen von Formen oder bei ihrer Bemaßung ein ähnliches Verhalten wie bei CAD-Programmen erwartet. Im Gegensatz dazu lag der Fokus einer Person, die Experte im Umgang von Grafikprogrammen war, in einem anderen Bereich. Beim Einfügen der Komponenten im Erkennungsmodus erwartete sie eine Option zum Bestätigen oder Verwerfen der Komponente, wie sie zum Beispiel bei Photoshop eingesetzt wird. Dieses Spannungsfeld der unterschiedlichen Erwartungen bietet interessante Ansätze für zukünftig Forschungsfragen.

8 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ziele der Arbeit zusammengefasst und das vorgestellte Konzept kurz erläutert. Durch eine Studie wurden die grundlegenden Ideen des Konzepts evaluiert. Darüber hinaus boten die Ergebnisse der Studie Inspiration für Erweiterungen, die im Ausblick zu finden sind.

8.1 Zusammenfassung

Personal Fabrication ist dank sinkender Preise für Lasercutter und 3D-Drucker immer weiter auf dem Vormarsch. Die Herstellung von Gegenständen im Heimbereich wird vereinfacht und lässt bereits heute die Grenze zwischen industrieller und privater Herstellung verschwimmen. Das Einsatzgebiet von Personal Fabrication ist interdisziplinär und spricht Personen aus Fachbereichen von Elektrotechnik, Informatik und Design gleichermaßen an. Da die Anwendung aber neben den technisch versierten Gruppen auch Personen im Heimbereich bedienen sollen, können Kenntnisse insbesondere im CAD-Bereich nicht vorausgesetzt werden. Deshalb wurde in dieser Arbeit ein Konzept entwickelt und evaluiert, das eine schnellere Produktion von Gehäusen für Privatpersonen ermöglicht. CAD-Programme benötigen eine längere Einarbeitungszeit und sind durch ihren Funktionsumfang sehr komplex. Das hier vorgestellte Konzept hat hingegen ein einfaches Bedienkonzept mit beschränkten Gestaltungsmöglichkeiten. Für die Herstellung eines Gehäuses, das einen prototypischen Bausatz beinhaltet, steht jedoch die geringe Produktionsdauer im Vordergrund. Durch den Einsatz eines Lasercutters sind die Gehäuse in weniger als einer Stunde verfügbar. Die einzelnen Teile, die mit dem Lasercutter hergestellt werden, können durch Verbindungen zu einem Körper zusammengefügt werden. Durch Presspassung entfällt das Kleben oder Schrauben. Bei der Entwicklung des Konzepts wurden insbesondere elektronische Bauteile wie Displays und Buttons berücksichtigt, die Aussparungen an der Oberfläche des Gehäuses benötigen. Ein Touch-Tisch hat die elektronischen Komponenten erfasst, so dass die Aussparungen im Gehäuse durch die Position der Komponente auf dem Tisch bestimmt werden.

Um ein Gehäuse zu entwerfen, muss zunächst eine Grundform ausgewählt werden. In der entwickelten Anwendung stehen Zylinder und Quader zur Verfügung. Anschließend können individuelle Aussparungen eingefügt oder die Objekterkennung für elektronische Bauteile gestartet werden. Eine Gehäuseaußenfläche kann außerdem drei verschiedene Ausführungen haben. Die erste Ausführung erlaubt Aussparungen für weitere Komponenten hinzuzufügen, die zweite Ausführung erlaubt die Personalisierung des Gehäuses, indem ein Bild eingraviert wird. Damit wird dem steigenden Trend Rechnung getragen, Produkte individuell gestalten zu können. Die letzte Ausführung fügt eine Leinwand hinzu, die als Projektionsfläche genutzt

werden, zur Gewichtsersparnis dienen oder als schalldurchlässige Wand agieren kann. Dafür wird Stoff zwischen der Seitenfläche und einem weiteren Rahmen gespannt.

Bei der Umsetzung des Konzepts wurde die Anwendung in die drei Komponenten ObjectDetection, Natcut und SVG-Generator aufgeteilt. Microsofts PixelSense diente als Eingabepattform, deren Dienste besonders von ObjectDetection und Natcut genutzt wurden. ObjectDetection ist ein eigenständiges Projekt, das für die Erkennung von Gegenständen verantwortlich ist. Natcut ist die Hauptanwendung, welche die mit WPF realisierte Oberfläche zur Verfügung stellt. Außerdem regelt sie den Zugriff auf die beiden anderen Komponenten, sowie die Verbindung zum Lasercutter. Der SVG-Generator ist ebenfalls ein eigenständiges Projekt und bietet die Konvertierung von geometrischen Beschreibungen der Formen in eine Grafik an, die dem Lasercutter als Eingabe dient. Dabei sind die Eingangsdaten in XML definiert, die Ausgabedatei verwendet das SVG-Format. Der SVG-Generator stellt Algorithmen zur Erstellung von Akkordeon- und Fingerverbindungen bereit, mit denen Seitenflächen zusammen gefügt werden. Er ist dafür verantwortlich, dass die Eigenschaften des Lasers und des Materials berücksichtigt werden, um beispielsweise eine Presspassung zu erzielen. Um valide Parameter für die Verbindungen zu ermitteln und weitere Verbindungsarten zu testen, wurden vor der Implementierung einige Vorversuche durchgeführt.

In einer explorativen Studie belegen die Teilnehmer, dass innerhalb weniger Minuten das Design eines Gehäuses erstellt werden kann. Sie bestätigen weiterhin, dass die Eingabe mit elektronischen Bauteilen als hilfreich empfunden wird und die Abschätzung der Dimensionen vereinfacht.

8.2 Ausblick

Bei der implementierten Software handelt es sich um einen Prototyp. Daher unterliegt sie einigen Einschränkungen, die vor einem Einsatz als Produktivsystem angepasst werden müssen. Im Folgenden werden Optimierungen und Erweiterungsmöglichkeiten beschrieben, mit denen weitere Einsatzmöglichkeiten erschlossen werden.

Mit der Integration des Herstellungsprozesses in die Hauptanwendung, wie im Konzept bereits vorgesehen, kann eine durchgängige Bearbeitung des Gehäuses erreicht werden. Standardwerte des Lasercutters für Materialien und Materialstärken werden gespeichert und können wiederverwendet werden. Um eine bessere Ausnutzung des Materials zu erzielen, bietet sich ein Algorithmus an, der das *Strip Packing*-Problem löst. Eine Verwendung zeigen Weichel et al. [WLG13].

In der Prototyp-Implementierung wurde eine Möglichkeit zur Personalisierung beispielhaft gezeigt. Mit Freihandskizzen, Bézierkurven, Linien, Text oder bestimmten Schnittmustern gibt es jedoch weitere Ideen, um das Angebot zu bereichern. Je nach Fokus des Prototyps können auch verschiedene Gehäusedesigns miteinander verglichen werden. Durch die Verwendung weiterer Materialien, wie farbiger Acrylplatten oder einer Kombination aus verschiedenen Hölzern, werden neue Gestaltungsmöglichkeiten eröffnet.

Auch die Validierung der Benutzereingaben kann verfeinert werden. Um ausschließlich stabile und realisierbare Quader zu erstellen, muss eine Seite eine Mindestbreite aufweisen. Anderenfalls kann es vorkommen, dass die Fläche durch die Aussparungen der Fingerverbindungen instabil wird. Mit Hilfe von visuellem Feedback, wie schraffierten Linien, kann dem Benutzer erklärt werden, in welchem Bereich die Bearbeitung möglich ist. Um einen Überblick des Gehäuses im zusammengebauten Zustand zu gewinnen, kann eine 3D-Ansicht helfen.

Wie die Teilnehmer der Studie bereits bestätigten, haben Features wie das Ausrichten von Elementen eine hohe Priorität. Ebenso könnte die Einführung eines Rasters das Platzieren von Elementen erleichtern. Möchte man den Abstand zwischen zwei Elementen bestimmen, so wäre ein Messwerkzeug hilfreich. Weitere Möglichkeiten bieten das Ziehen eines Auswahlrahmens, das manuelle Gruppieren und das Duplizieren von Elementen. Um den Konflikt zwischen zu kleinen Elementen für Touch-Auswahl und der Umsetzung von Schraubenlöchern als Befestigung zu lösen, könnte das Vergrößern und Verkleinern angeboten werden. Bei der Eingabe mit physikalisch vorhandenen Komponenten sollte jedoch der Maßstab immer im Verhältnis 1:1 sein.

Für den Prototyp gibt es weitere Einsatzmöglichkeiten: Verzichtet man auf die Bedienung per Touch-Eingabe, so bietet die Anwendung auch im Desktop-Einsatz den Vorteil Gehäuse schnell entwerfen und herstellen zu können. Durch *Drag and Drop* könnten dann die elektronischen Komponenten platziert werden. Möchte man die Touch-Eingabe beibehalten, aber die Genauigkeit erhöhen so bietet sich der Einsatz per Stift an. Da die wesentlichen Dateiformate des Prototyps auf SVG und XML basieren, ist auch eine Portierung in eine Cloud denkbar. Entwürfe können im Sinne des *Collaborative Engineering*¹ bearbeitet und nach deren Herstellung beurteilt werden. Schließlich könnte es eine Schnittstelle zu CAD-Programmen geben, welche die Flächen eines Modells zur Verfügung stellen. Vorhandene Entwürfe können dann weiter bearbeitet und mit einem Lasercutter hergestellt werden.

Bisher werden elektronische Komponenten mit Hilfe von Markern erkannt, die auf der Unterseite angebracht sind. Für eine flexiblere Erkennung könnten die Komponenten mit Bildverarbeitungsverfahren erfasst werden. Dazu kann ein Touch-Tisch wie der PixelSense verwendet werden, wenn die Auflösung ausreichend ist. Alternativ könnte eine externe Kamera die Position der Komponente sowie ihre Größe bestimmen.

Der vorgestellte Prototyp bietet bisher zwei Grundformen an, aus denen ausgewählt werden kann. Durch weitere Formen kann der Benutzer sein Gehäuse besser seinen Vorstellungen anpassen. Einige Benutzer wünschen sich zum Beispiel einen abgerundeten Quader, N-Ecke, Pyramiden, trapezförmige Körper oder eine Kombination daraus. Bei der Befragung der Teilnehmer zeichnete sich jedoch ab, dass die wichtigsten zwei Formen bereits vorhanden sind. Außer den Grundformen können auch weitere Verbindungsarten das Angebot bereichern (siehe auch Abbildung 6.2 und 6.1).

¹<http://wirtschaftslexikon.gabler.de/Archiv/82844/collaborative-engineering-v7.html>

A Anhang

A.1 Objektmodell im Detail

Das Objektmodell wird durch ein XSD-Schema beschrieben und beinhaltet alle Parameter für die Erstellung des Gehäuses. Das Klassendiagramm veranschaulicht den Aufbau der generierten Klassen aus dem XSD-Schema. Die zentralen Klassen, die im folgenden kurz vorgestellt werden, sind Casing, AArea und JointSettings.

Die abstrakte Klasse AArea hält alle Daten einer Fläche vor (Abbildung A.1). Dazu wird die Grundform und ihr Name angegeben. Je nachdem ob die konkrete Instanz von Area oder RestrictedArea ist, können weitere AArea-Objekte enthalten sein, oder nicht. RestrictedAreas steht dabei für eine Fläche, die der Benutzer nicht weiter bearbeiten kann, wie beispielsweise die Fläche für Akkordeonverbindungen. Die Enumerationen AreaNames bzw. RestrictedAreaNames werden bei der SVG-Generierung für die Flächennamen verwendet und dienen als Orientierung (siehe Abschnitt 6.1).

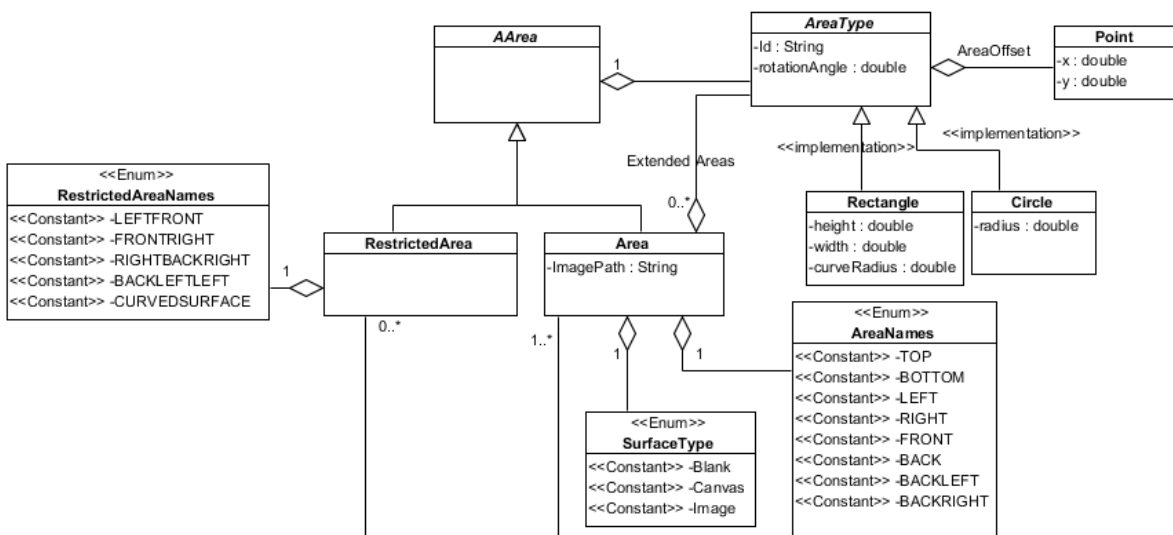


Abbildung A.1: Die Klasse AArea ist die zentrale, abstrakte Klasse, die einen AreaType besitzt und als Container für SVG-Objekte dient.

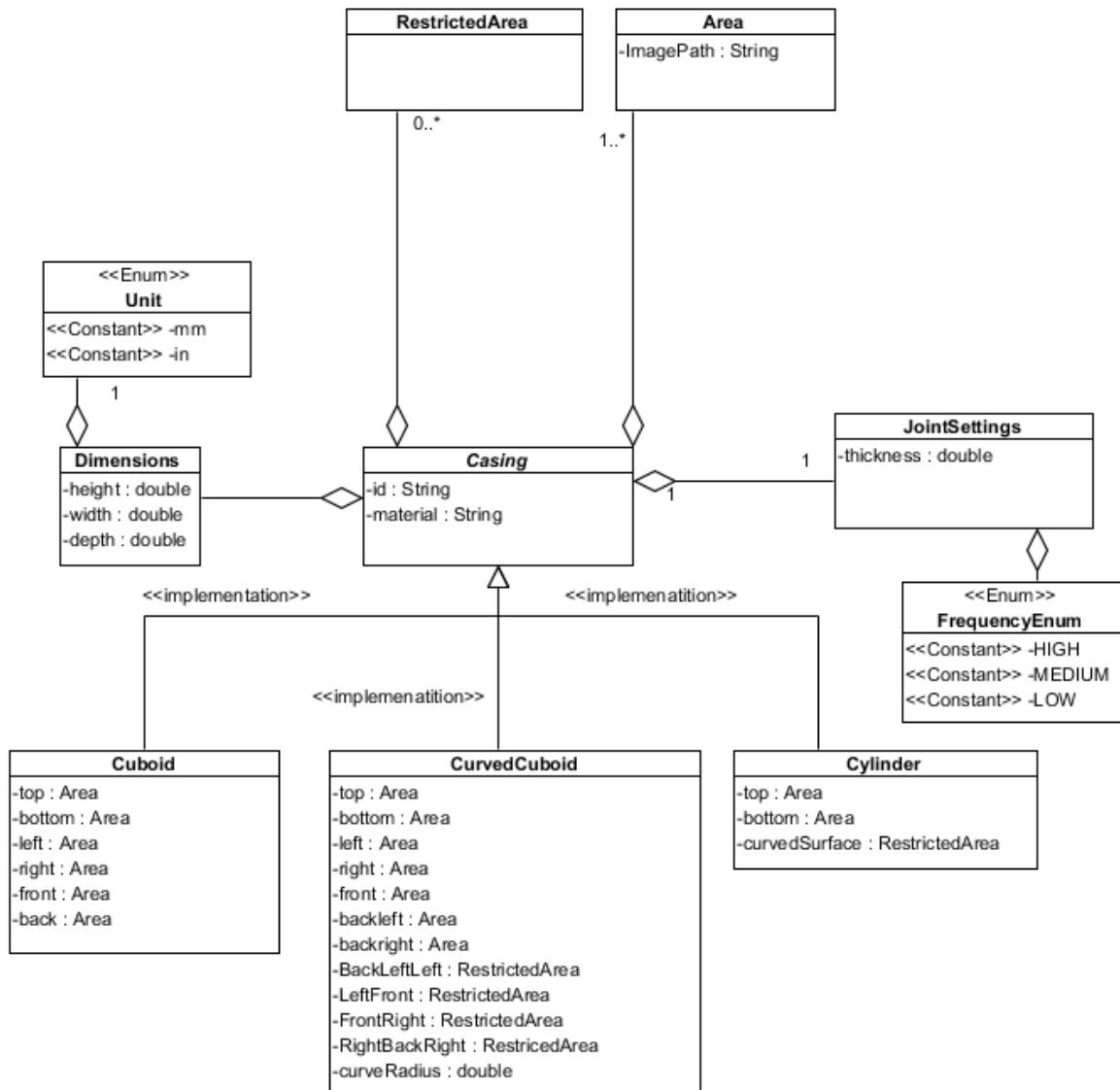


Abbildung A.2: Die Casing-Klasse ist eine zentrale Komponente, von der aus auf alle anderen zugegriffen werden kann. Ein von Casing abgeleitetes Objekt, beinhaltet alle Parameter eines Gehäuses. Dieses Objekt definiert sowohl alle verwendeten Seiten, als auch die Verbindungsoptionen.

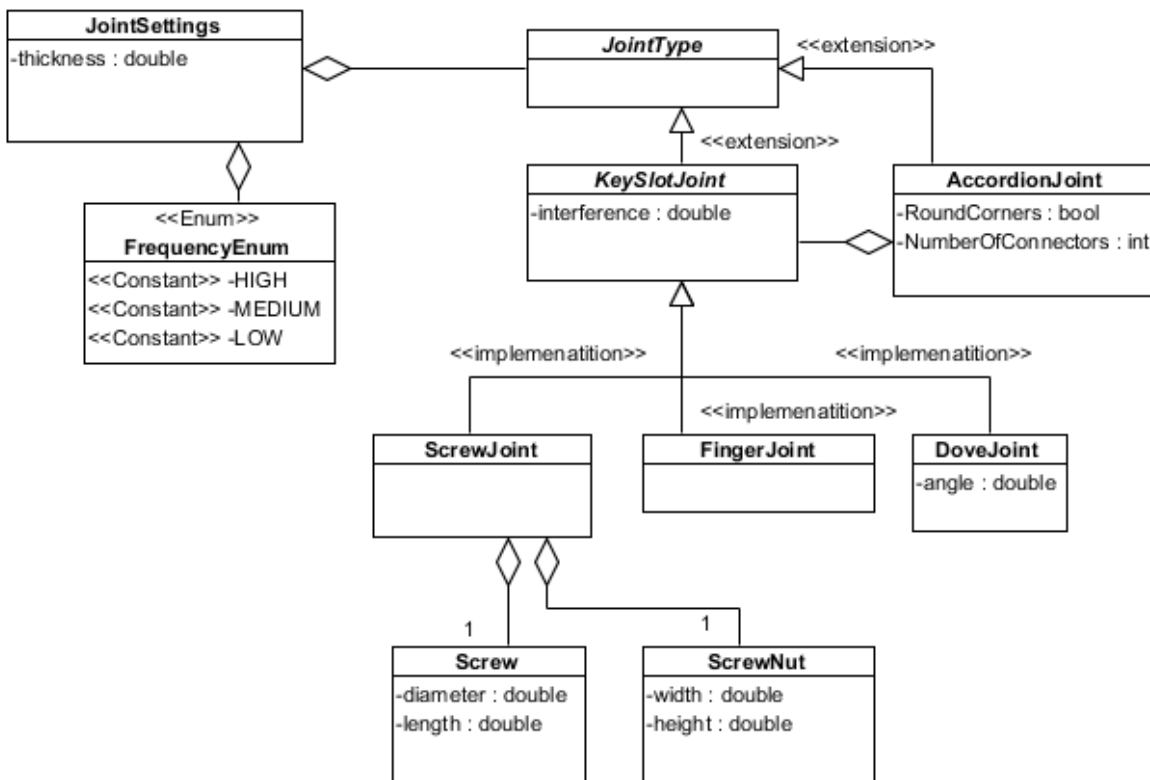


Abbildung A.3: Die Verbindungsoptionen (JointSettings) beschreiben, über welchen Verbindungstyp die Flächen verbunden sind. Interference beschreibt dabei, um wie viel ein Zapfen eine Nut überragt, um eine Presspassung zu erzeugen. Die Frequenz ist ein Parameter der die Anzahl von Zapfen beeinflusst. Die Zahl der Verbindungen der Akkordeonverbindung beschreibt wie viele Verbindungen jedes Segment zum nächsten hat. RoundCorners bestimmt, ob die Ecken der Segmente abgerundet sind.

A.2 XSD2Code

XSD2Code ist ein Werkzeug, das aus einer XML Datei C#-Klassen generiert. Hier werden die Parameter, mit denen XSD2Code konfiguriert wurde, beleuchtet. Mit den folgenden Eigenschaften funktioniert die Code-Generierung reibungslos:

- AutomaticProperties = False (da für abstrakte Attribute das zugehörige Feld bei Setter/-Getter Methoden nicht generiert wird)

- CollectionObjectType = Array, da beim Serialisieren sonst Probleme auftreten¹. Damit werden keine Konstruktoren erzeugt -> manuelles hinzufügen nötig, wo Objekte standardmäßig mit neuen Unterobjekten angelegt werden sollen, beispielsweise Casing.
- Serialization.Enabled = True, um die notwendigen Methoden zur Serialisierung nicht selbst generieren zu müssen.

Code	
CustomUsings	(Auflistung)
Language	CSharp
NameSpace	ShapeModel
TargetFramework	Net40
Collection	
CollectionBase	
CollectionObjectType	Array
Verhalten	
EnableDataBinding	False
EnableInitializeFields	True
GenerateCloneMethod	False
GenerateDataContracts	False
GenericBaseClass	(Use generic base class=True)
BaseClassName	EntityBase
Enabled	True
GenerateBaseClass	True
Miscellaneous	
DisableDebug	False
EnableSummaryComment	False
ExcludeIncludedTypes	False
HidePrivateFieldInIde	True
PropertyParams	
AutomaticProperties	False
EnableLazyLoading	False
EnableVirtualProperties	False
GeneratePropertyNameSpecified	Default
GenerateShouldSerializeProperty	False
Serialization	
DefaultEncoder	UTF8
DeserializeMethodName	Deserialize
Enabled	True
EnableEncoding	False
GenerateXmlAttributes	True
LoadFromFileMethodName	LoadFromFile
SaveToFileMethodName	SaveToFile
SerializeMethodName	Serialize

Abbildung A.4: Die Eigenschaften des Werkzeugs *XSD2Code*. Es wird nahtlos in Visual Studio integriert und erlaubt die Generierung von Code aus XSD-Dateien mit einigen Parametern.

¹siehe auch <https://resources.riskfocusinc.com/portal/display/PUBLIC/CSharp+Code+Generation+for+FpML+with+Xsd2Code>

A.3 Komponentenübersicht

In der folgenden Abbildung ist die detaillierte Architektur in einem Komponentendiagramm zu sehen.

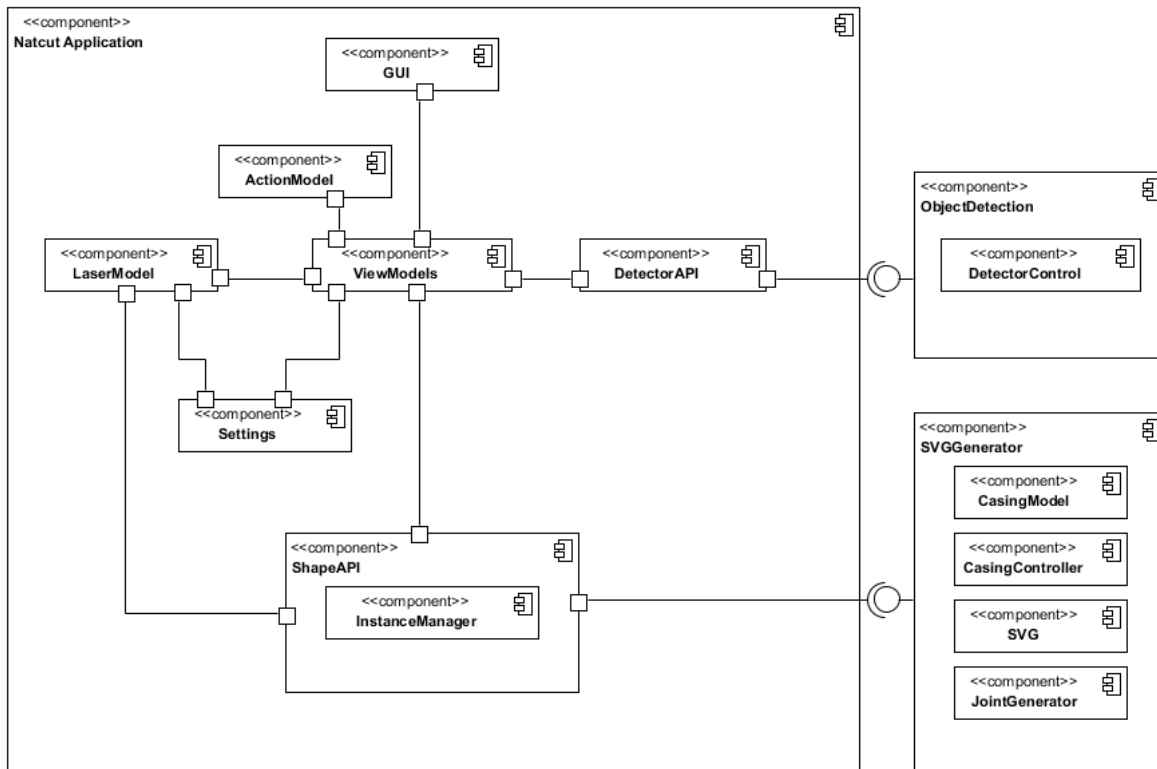


Abbildung A.5: Hier sind alle im Projekt verwendeten Komponenten zu sehen. Die Hauptkomponente Natcut gliedert sich in weitere Unterkomponenten. Die externe Komponente SVG-Generator erstellt aus einem grafischen Modell die Eingabe für den Lasercutter. ObjectDetection ist dafür verantwortlich elektronische Bauteile zu erkennen, die auf dem Touch-Tisch positioniert werden.

A.4 Fragebogen der Benutzerstudie

Der folgende Fragebogen wurde nach der Durchführung der Studie von den Teilnehmern ausgefüllt. Anschließend wurden offene Fragen gestellt, um neue Einsatzmöglichkeiten und Verbesserungen zu untersuchen.

Fragebogen – Explorative Studie Natcut

1. Alter _____
2. Geschlecht
 - Weiblich
 - Männlich
3. Bitte geben Sie ihren aktuellen Studiengang bzw. Ihren Beruf an

4. Bitte schätzen Sie ihre Kenntnisse über den Umgang von CAD Programmen ein
 - Keine
 - Grundkenntnisse
 - Fortgeschritten
 - Experte
5. Falls Sie über CAD-Kenntnisse verfügen, schätzen Sie bitte ein, wie lange sie für die Erstellung des Designs („Zickzack“-Muster) benötigen würden
 - Gameboy _____
 - Kamera _____
6. Bitte schätzen Sie ihre Kenntnisse über den Umgang von Grafikprogrammen wie Photoshop, Gimp, Corel Designer o.Ä. ein
 - Keine
 - Grundkenntnisse
 - Fortgeschritten
 - Experte
7. Falls Sie über Grafikprogramm-Kenntnisse verfügen, schätzen Sie bitte ein, wie lange sie für die Erstellung des Designs („Zickzack“-Muster) benötigen würden
 - Gameboy _____
 - Kamera _____
8. Wie oft benutzen Sie Touch-Displays (Smartphones, Tablets etc.)?
 - Nie
 - Mehrmals täglich
 - Ein Mal pro Tag
 - Ein Mal pro Woche

9. Wie bewerten Sie den Gesamteindruck des Systems?
- Schrecklich
 - Schlecht
 - OK
 - Gut
 - Ausgezeichnet
10. Bitte bewerten Sie die Personalisierung von Gehäusen mit Hilfe von Bildern
- Sehr schlecht
 - Schlecht
 - OK
 - Gut
 - Ausgezeichnet
11. Für wie wichtig halten Sie Komfort-Features wie Gruppieren von Elementen, nach gleicher Höhe ausrichten etc.?
- Unwichtig
 - Weniger wichtig
 - Ok
 - Wichtig
 - Sehr wichtig
12. Wie hilfreich ist die physikalische Repräsentation der Komponenten zur Eingabe?
- Gar nicht hilfreich
 - Wenig hilfreich
 - Ok
 - Hilfreich
 - Sehr hilfreich

Literaturverzeichnis

- [Alb82] A. E. Albert. The Effect of Graphic Input Devices on Performance in a Cursor Positioning Task. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting October*, Band 26, S. 54 – 58. 1982. (Zitiert auf Seite 34)
- [AZ03] P.-A. Albinsson, S. Zhai. High precision touch screen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, S. 105–112. ACM, New York, NY, USA, 2003. doi:10.1145/642611.642631. URL <http://doi.acm.org/10.1145/642611.642631>. (Zitiert auf Seite 34)
- [Bas08] M. Baskinger. Pencils before pixels: a primer in hand-generated sketching. *interactions*, 15(2):28–36, 2008. doi:10.1145/1340961.1340969. URL <http://doi.acm.org/10.1145/1340961.1340969>. (Zitiert auf Seite 45)
- [Ber12] B. Berman. 3-D printing: The new industrial revolution. *Business Horizons*, 55(2):155 – 162, 2012. doi:10.1016/j.bushor.2011.11.003. URL <http://www.sciencedirect.com/science/article/pii/S0007681311001790>. (Zitiert auf Seite 21)
- [But96] K. A. Butler. Usability engineering turns 10. *interactions*, 3(1):58–75, 1996. doi:10.1145/223500.223513. URL <http://doi.acm.org/10.1145/223500.223513>. (Zitiert auf Seite 42)
- [CC12] P. Chatterjee, S. Chakraborty. Material selection using preferential ranking methods. *Materials & Design*, 35(0):384 – 393, 2012. doi:10.1016/j.matdes.2011.09.027. URL <http://www.sciencedirect.com/science/article/pii/S0261306911006418>. (Zitiert auf Seite 40)
- [CKMH12] S.-U. Cheon, B. C. Kim, D. Mun, S. Han. A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems. *Comput. Aided Des.*, 44(2):123–131, 2012. doi:10.1016/j.cad.2011.10.003. URL <http://dx.doi.org/10.1016/j.cad.2011.10.003>. (Zitiert auf Seite 12)
- [CM02] A. Cenna, P. Mathew. Analysis and prediction of laser cutting parameters of fibre reinforced plastics (FRP) composite materials. *International Journal of Machine Tools and Manufacture*, 42(1):105 – 113, 2002. doi:10.1016/S0890-6955(01)00090-6. URL <http://www.sciencedirect.com/science/article/pii/S0890695501000906>. (Zitiert auf Seite 22)

- [DSSL07] R. C. Davis, T. S. Saponas, M. Shilman, J. A. Landay. SketchWizard: Wizard of Oz prototyping of pen-based user interfaces. In *Proceedings of the 20th annual ACM symposium on User interface software and technology, UIST '07*, S. 119–128. ACM, New York, NY, USA, 2007. doi:10.1145/1294211.1294233. URL <http://doi.acm.org/10.1145/1294211.1294233>. (Zitiert auf Seite 35)
- [DYZA05] J. K. Dickinson, Z. Yu, Y. Zeng, H. Antunes. Pen-tablet as a CAD interface alternative. *Robotics and Computer-Integrated Manufacturing*, 21(4-5):465–474, 2005. doi:10.1016/j.rcim.2004.11.007. URL <http://www.sciencedirect.com/science/article/pii/S0736584504001280>. (Zitiert auf den Seiten 12, 31 und 33)
- [Fer95] T. Fernandes. *Global interface design: a guide to designing international user interfaces*. Academic Press Professional, Inc., San Diego, CA, USA, 1995. (Zitiert auf Seite 42)
- [Fut07] I. for the Future. Manufacturing: Do it yourself? 2007. URL http://www.iftf.org/uploads/media/SR-1064_TYF07_05_Manufacturing.pdf. (Zitiert auf Seite 40)
- [GHJV95] E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995. (Zitiert auf Seite 52)
- [Gro09] M. D. Gross. Now More Than Ever : Computational thinking and a science of design. *Special issue of Japanese Society for Science of Design*, 16(2):50–54, 2009. URL <http://ci.nii.ac.jp/naid/110007123573/en/>. (Zitiert auf Seite 11)
- [Has08] M. Hassenzahl. The interplay of beauty, goodness, and usability in interactive products. *Hum.-Comput. Interact.*, 19(4):319–349, 2008. doi:10.1207/s15327051hci1904_2. URL http://dx.doi.org/10.1207/s15327051hci1904_2. (Zitiert auf Seite 42)
- [HB10] C. Holz, P. Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, S. 581–590. ACM, New York, NY, USA, 2010. doi:10.1145/1753326.1753413. URL <http://doi.acm.org/10.1145/1753326.1753413>. (Zitiert auf Seite 34)
- [HSH11] C. Harrison, J. Schwarz, S. E. Hudson. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology, UIST '11*, S. 627–636. ACM, New York, NY, USA, 2011. doi:10.1145/2047196.2047279. URL <http://doi.acm.org/10.1145/2047196.2047279>. (Zitiert auf Seite 35)
- [HYP⁺10] K. Hinckley, K. Yatani, M. Pahud, N. Coddington, J. Rodenhouse, A. Wilson, H. Benko, B. Buxton. Pen + touch = new tools. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology, UIST '10*, S. 27–36. ACM, New York, NY, USA, 2010. doi:10.1145/1866029.1866036. URL <http://doi.acm.org/10.1145/1866029.1866036>. (Zitiert auf Seite 35)

- [IR10] R. Ibrahim, F. P. Rahimian. Comparison of CAD and manual sketching tools for teaching architectural design. *Automation in Construction*, 19(8):978 – 987, 2010. doi:10.1016/j.autcon.2010.09.003. URL <http://www.sciencedirect.com/science/article/pii/S0926580510001329>. (Zitiert auf Seite 12)
- [IU97] H. Ishii, B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, CHI '97, S. 234–241. ACM, New York, NY, USA, 1997. doi:10.1145/258549.258715. URL <http://doi.acm.org/10.1145/258549.258715>. (Zitiert auf Seite 39)
- [Jac95] P. F. Jacobs. *StereoLithography and Other RP&M Technologies: From Rapid Prototyping to Rapid Tooling*. Society of Manufacturing Engineers, 1995. (Zitiert auf Seite 13)
- [JGH⁺08] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, J. Zigelbaum. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, S. 201–210. ACM, New York, NY, USA, 2008. doi:10.1145/1357054.1357089. URL <http://doi.acm.org/10.1145/1357054.1357089>. (Zitiert auf Seite 39)
- [JJG10] S. Jordà, C. F. Julià, D. Gallardo. Interactive surfaces and tangibles. *XRDS*, 16(4):21–28, 2010. doi:10.1145/1764848.1764855. URL <http://doi.acm.org/10.1145/1764848.1764855>. (Zitiert auf Seite 39)
- [JS11] H. Jung, E. Stolterman. Form and materiality in interaction design: a new approach to HCI. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, S. 399–408. ACM, New York, NY, USA, 2011. doi:10.1145/1979742.1979619. URL <http://doi.acm.org/10.1145/1979742.1979619>. (Zitiert auf Seite 41)
- [KHK08] E. Karana, P. Hekkert, P. Kandachar. Material considerations in product design: A survey on crucial material aspects used by product designers. *Materials & Design*, 29(6):1081 – 1089, 2008. doi:10.1016/j.matdes.2007.06.002. URL <http://www.sciencedirect.com/science/article/pii/S0261306907001458>. (Zitiert auf Seite 40)
- [KHK10] E. Karana, P. Hekkert, P. Kandachar. A tool for meaning driven materials selection. *Materials & Design*, 31(6):2932 – 2941, 2010. doi:10.1016/j.matdes.2009.12.021. URL <http://www.sciencedirect.com/science/article/pii/S0261306909007110>. (Zitiert auf Seite 41)
- [KK95] M. Kurosu, K. Kashimura. Apparent usability vs. inherent usability: experimental analysis on the determinants of the apparent usability. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, S. 292–293. ACM, New York, NY, USA, 1995. doi:10.1145/223355.223680. URL <http://doi.acm.org/10.1145/223355.223680>. (Zitiert auf Seite 42)

- [Lee10] Y.-L. Lee. Comparison of the conventional point-based and a proposed finger probe-based touch screen interaction techniques in a target selection task. *International Journal of Industrial Ergonomics*, 40(6):655 – 662, 2010. doi:10.1016/j.ergon.2010.09.001. URL <http://www.sciencedirect.com/science/article/pii/S0169814110000867>. (Zitiert auf Seite 34)
- [Leo12] C. Leopold. *Geometrische Grundlagen der Architekturdarstellung*, Band 4. Vieweg+Teubner Verlag, 2012. (Zitiert auf Seite 11)
- [LETH10] G. Lee, C. M. Eastman, T. Taunk, C.-H. Ho. Usability principles and best practices for the user interface design of complex 3D architectural design and engineering tools. *Int. J. Hum.-Comput. Stud.*, 68(1-2):90–104, 2010. doi:10.1016/j.ijhcs.2009.10.001. URL <http://dx.doi.org/10.1016/j.ijhcs.2009.10.001>. (Zitiert auf den Seiten 12 und 63)
- [LGHH08] C. Liao, F. Guimbretière, K. Hinckley, J. Hollan. Papiercraft: A gesture-based command system for interactive paper. *ACM Trans. Comput.-Hum. Interact.*, 14(4):18:1–18:27, 2008. doi:10.1145/1314683.1314686. URL <http://doi.acm.org/10.1145/1314683.1314686>. (Zitiert auf Seite 33)
- [LHO⁺12] M. Lau, M. Hirose, A. Ohgawara, J. Mitani, T. Igarashi. Situated modeling: a shape-stamping interface with tangible primitives. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, S. 275–282. ACM, New York, NY, USA, 2012. doi:10.1145/2148131.2148190. URL <http://doi.acm.org/10.1145/2148131.2148190>. (Zitiert auf Seite 32)
- [LK10] H. Lipson, M. Kurman. *Factory@Home: The emerging economy of personal manufacturing*. Technischer Bericht, 2010. (Zitiert auf den Seiten 17 und 40)
- [LME11] B. Leduc-Mills, M. Eisenberg. The UCube: a child-friendly device for introductory three-dimensional design. In *Proceedings of the 10th International Conference on Interaction Design and Children*, IDC '11, S. 72–80. ACM, New York, NY, USA, 2011. doi:10.1145/1999030.1999039. URL <http://doi.acm.org/10.1145/1999030.1999039>. (Zitiert auf Seite 32)
- [LSMI10] M. Lau, G. Saul, J. Mitani, T. Igarashi. Modeling-in-context: user design of complementary objects with a single photo. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, S. 17–24. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2010. URL <http://dl.acm.org/citation.cfm?id=1923363.1923367>. (Zitiert auf den Seiten 20 und 46)
- [Maq88] J. Maquet. *The Aesthetic Experience: An Anthropologist Looks at the Visual Arts*. Yale University Press, 1988. (Zitiert auf Seite 42)
- [MB12] D. A. Mellis, L. Buechley. Case studies in the personal fabrication of electronic products. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, S. 268–277. ACM, New York, NY, USA, 2012. doi:10.1145/2317956.2317998. URL

- <http://doi.acm.org/10.1145/2317956.2317998>. (Zitiert auf den Seiten 20, 21 und 22)
- [MHR79] T. M. H. Reenskaug. Thing-Model-View-Editor – an Example from a planning system, 1979. URL <http://heim.ifi.uio.no/trygver/1979/mvc-1/1979-05-MVC.pdf>. (Zitiert auf Seite 51)
- [MKB13] S. Mueller, B. Kruck, P. Baudisch. LaserOrigami: laser-cutting 3D objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, S. 2585–2592. ACM, New York, NY, USA, 2013. doi:10.1145/2470654.2481358. URL <http://doi.acm.org/10.1145/2470654.2481358>. (Zitiert auf Seite 37)
- [ML89] R. Mack, K. Lang. A Benchmark Comparison of Mouse and Touch Interface Techniques for an Intelligent Workstation Windowing Environment. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting October*, Band 33, S. 325 – 329. 1989. (Zitiert auf Seite 35)
- [MLB12] S. Mueller, P. Lopes, P. Baudisch. Interactive construction: interactive fabrication of functional mechanical devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, S. 599–606. ACM, New York, NY, USA, 2012. doi:10.1145/2380116.2380191. URL <http://doi.acm.org/10.1145/2380116.2380191>. (Zitiert auf Seite 36)
- [Mot11] C. Mota. The rise of personal fabrication. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, S. 279–288. ACM, New York, NY, USA, 2011. doi:10.1145/2069618.2069665. URL <http://doi.acm.org/10.1145/2069618.2069665>. (Zitiert auf Seite 15)
- [MTLZ11] S. Modi, M. Tiwari, Y. Lin, W. Zhang. On the architecture of a human-centered CAD agent system. *Computer-Aided Design*, 43(2):170 – 179, 2011. doi:10.1016/j.cad.2010.10.006. URL <http://www.sciencedirect.com/science/article/pii/S0010448510002009>. (Zitiert auf Seite 12)
- [Mul08] Muller. *Multi-touch displays: design, applications and performance evaluation*. Diplomarbeit, University of Amsterdam, 2008. (Zitiert auf Seite 34)
- [NG11] Y. Nagai, G. V. Georgiev. The role of impressions on users' tactile interaction with product materials: An analysis of associative concept networks. *Materials & Design*, 32(1):291 – 302, 2011. doi:10.1016/j.matdes.2010.05.040. URL <http://www.sciencedirect.com/science/article/pii/S026130691000333X>. (Zitiert auf Seite 41)
- [NLH⁺13] D. Nowacka, K. Ladha, N. Y. Hammerla, D. Jackson, C. Ladha, E. Rukzio, P. Olivier. Touchbugs: actuated tangibles on multi-touch tables. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, S. 759–762. ACM, New York, NY, USA, 2013. doi:10.1145/2470654.2470761. URL <http://doi.acm.org/10.1145/2470654.2470761>. (Zitiert auf den Seiten 39 und 40)

- [Oxm12] R. Oxman. Informed tectonics in material-based design. *Design Studies*, 33(5):427–455, 2012. doi:10.1016/j.destud.2012.05.005. URL <http://www.sciencedirect.com/science/article/pii/S0142694X12000373>. (Zitiert auf Seite 11)
- [PWS88] R. L. Potter, L. J. Weldon, B. Shneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, S. 27–32. ACM, New York, NY, USA, 1988. doi:10.1145/57167.57171. URL <http://doi.acm.org/10.1145/57167.57171>. (Zitiert auf Seite 34)
- [RKP11] V. Rognoli, E. Karana, O. Pedgley. Natural fibre composites in product design: an investigation into material perception and acceptance. In *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*, DPPI '11, S. 36:1–36:4. ACM, New York, NY, USA, 2011. doi:10.1145/2347504.2347543. URL <http://doi.acm.org/10.1145/2347504.2347543>. (Zitiert auf Seite 40)
- [RLZK13] S. Radhakrishnan, Y. Lin, I. Zeid, S. Kamarthi. Finger-based multitouch interface for performing 3D CAD operations. *International Journal of Human-Computer Studies*, 71(3):261–275, 2013. doi:10.1016/j.ijhcs.2012.07.004. URL <http://www.sciencedirect.com/science/article/pii/S1071581912001279>. (Zitiert auf Seite 34)
- [RN96] B. Reeves, C. Nass. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. CSLI Publications, 1996. (Zitiert auf Seite 42)
- [RRS⁺03] L. Roberts, L. Rankin, E. Silver, D. Moore, S. Plunkett, D. Washburn, B. Wilch-Ringen. Looks good to me. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, S. 818–819. ACM, New York, NY, USA, 2003. doi:10.1145/765891.766010. URL <http://doi.acm.org/10.1145/765891.766010>. (Zitiert auf Seite 43)
- [RWBB10] K.-F. Richter, B. Weber, B. Bojduj, S. Bertel. Supporting the designer's and the user's perspectives in computer-aided architectural design. *Adv. Eng. Inform.*, 24(2):180–187, 2010. doi:10.1016/j.aei.2009.08.012. URL <http://dx.doi.org/10.1016/j.aei.2009.08.012>. (Zitiert auf Seite 12)
- [SCRG10] D. Schmidt, F. Chehimi, E. Rukzio, H. Gellersen. PhoneTouch: a technique for direct phone interaction on surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, S. 13–16. ACM, New York, NY, USA, 2010. doi:10.1145/1866029.1866034. URL <http://doi.acm.org/10.1145/1866029.1866034>. (Zitiert auf Seite 33)
- [SCW⁺13] C.-H. Su, L. Chan, C.-T. Weng, R.-H. Liang, K.-Y. Cheng, B.-Y. Chen. NailDisplay: bringing an always available visual display to fingertips. In *Proceedings of the 2013 ACM annual conference on Human factors in computing systems*, CHI '13, S. 1461–1464. ACM, New York, NY, USA, 2013. doi:10.1145/2466110.2466193. URL <http://doi.acm.org/10.1145/2466110.2466193>. (Zitiert auf Seite 34)

- [SDS11] A. Schmidt, T. Doring, A. Sylvester. Changing How We Make and Deliver Smart Devices: When Can I Print Out My New Phone? *IEEE Pervasive Computing*, 10(4):6–9, 2011. doi:10.1109/MPRV.2011.68. URL <http://dx.doi.org/10.1109/MPRV.2011.68>. (Zitiert auf Seite 9)
- [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, S. 73–80. ACM, New York, NY, USA, 2011. doi:10.1145/1935701.1935717. URL <http://doi.acm.org/10.1145/1935701.1935717>. (Zitiert auf Seite 22)
- [SM10a] W. M. Steen, J. Mazumder. *Laser Material Processing*. Springer Verlag, 4 Auflage, 2010. (Zitiert auf Seite 22)
- [SM10b] G. Stemp-Morlock. Personal fabrication. *Commun. ACM*, 53(10):14–15, 2010. doi:10.1145/1831407.1831414. URL <http://doi.acm.org/10.1145/1831407.1831414>. (Zitiert auf den Seiten 9, 17 und 21)
- [SO06] L. Sass, R. Oxman. Materializing design: the implications of rapid prototyping in digital design. *Design Studies*, 27(3):325 – 355, 2006. doi:10.1016/j.destud.2005.11.009. URL <http://www.sciencedirect.com/science/article/pii/S0142694X05000864>. (Zitiert auf den Seiten 11, 14 und 15)
- [SSB08] R. Schmidt, K. Singh, R. Balakrishnan. Sketching and Composing Widgets for 3D Manipulation. *Computer Graphics Forum*, 27(2):301–310, 2008. URL <http://www.dgp.toronto.edu/~rms/pubs/SketchWidgetsEG08.html>. Proceedings of Eurographics 2008. (Zitiert auf Seite 33)
- [SSRG12] D. Schmidt, J. Seifert, E. Rukzio, H. Gellersen. A cross-device interaction style for mobiles and surfaces. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, S. 318–327. ACM, New York, NY, USA, 2012. doi:10.1145/2317956.2318005. URL <http://doi.acm.org/10.1145/2317956.2318005>. (Zitiert auf Seite 33)
- [SXG10] G. Saul, C. Xu, M. D. Gross. Interactive paper devices: end-user design & fabrication. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, S. 205–212. ACM, New York, NY, USA, 2010. doi:10.1145/1709886.1709924. URL <http://doi.acm.org/10.1145/1709886.1709924>. (Zitiert auf den Seiten 13 und 33)
- [SZH12] V. Savage, X. Zhang, B. Hartmann. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, S. 579–588. ACM, New York, NY, USA, 2012. doi:10.1145/2380116.2380189. URL <http://doi.acm.org/10.1145/2380116.2380189>. (Zitiert auf den Seiten 21 und 46)
- [TKI00] N. Tractinsky, A. Katz, D. Ikar. What is beautiful is usable. *Interacting with Computers*, 13(2):127 – 145, 2000. doi:10.1016/S0953-5438(00)00031-X. URL <http://>

- [//www.sciencedirect.com/science/article/pii/S095354380000031X](http://www.sciencedirect.com/science/article/pii/S095354380000031X). (Zitiert auf Seite 42)
- [Tra97] N. Tractinsky. Aesthetics and apparent usability: empirically assessing cultural and methodological issues. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems, CHI '97*, S. 115–122. ACM, New York, NY, USA, 1997. doi:10.1145/258549.258626. URL <http://doi.acm.org/10.1145/258549.258626>. (Zitiert auf Seite 42)
- [VB07] D. Vogel, P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, S. 657–666. ACM, New York, NY, USA, 2007. doi:10.1145/1240624.1240727. URL <http://doi.acm.org/10.1145/1240624.1240727>. (Zitiert auf Seite 34)
- [VSH⁺12] N. Villar, J. Scott, S. Hodges, K. Hammil, C. Miller. .NET gadgeteer: a platform for custom devices. In *Proceedings of the 10th international conference on Pervasive Computing, Pervasive'12*, S. 216–233. Springer-Verlag, Berlin, Heidelberg, 2012. doi:10.1007/978-3-642-31205-2_14. URL http://dx.doi.org/10.1007/978-3-642-31205-2_14. (Zitiert auf Seite 30)
- [WG10] K.-J. Wu, M. D. Gross. TOPAOKO: interactive construction kit. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10*, S. 3619–3624. ACM, New York, NY, USA, 2010. doi:10.1145/1753846.1754028. URL <http://doi.acm.org/10.1145/1753846.1754028>. (Zitiert auf Seite 17)
- [WLG13] C. Weichel, M. Lau, H. Gellersen. Enclosed: a component-centric interface for designing prototype enclosures. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, TEI '13*, S. 215–218. ACM, New York, NY, USA, 2013. doi:10.1145/2460625.2460659. URL <http://doi.acm.org/10.1145/2460625.2460659>. (Zitiert auf den Seiten 38 und 82)
- [WLMII10] K. D. Willis, J. Lin, J. Mitani, T. Igarashi. Spatial sketch: bridging between movement & fabrication. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction, TEI '10*, S. 5–12. ACM, New York, NY, USA, 2010. doi:10.1145/1709886.1709890. URL <http://doi.acm.org/10.1145/1709886.1709890>. (Zitiert auf Seite 33)
- [WS07] A. D. Wilson, R. Sarin. BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proceedings of Graphics Interface 2007, GI '07*, S. 119–125. ACM, New York, NY, USA, 2007. doi:10.1145/1268517.1268539. URL <http://doi.acm.org/10.1145/1268517.1268539>. (Zitiert auf Seite 33)
- [WSJB10] M. Weiss, F. Schwarz, S. Jakubowski, J. Borchers. Madgets: actuating widgets on interactive tabletops. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology, UIST '10*, S. 293–302. ACM, New York, NY,

USA, 2010. doi:10.1145/1866029.1866075. URL <http://doi.acm.org/10.1145/1866029.1866075>. (Zitiert auf den Seiten 39 und 40)

[WXW⁺11] K. D. Willis, C. Xu, K.-J. Wu, G. Levin, M. D. Gross. Interactive fabrication: new interfaces for digital fabrication. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, S. 69–72. ACM, New York, NY, USA, 2011. doi:10.1145/1935701.1935716. URL <http://doi.acm.org/10.1145/1935701.1935716>. (Zitiert auf Seite 33)

[ZBAK10] R. Zeleznik, A. Bragdon, F. Adeptura, H.-S. Ko. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, S. 17–26. ACM, New York, NY, USA, 2010. doi:10.1145/1866029.1866035. URL <http://doi.acm.org/10.1145/1866029.1866035>. (Zitiert auf Seite 35)

Alle URLs wurden zuletzt am 19.06.2013 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift