



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Masterarbeit

Hauke Wittern

Konzeption, Entwicklung und Evaluierung einer  
Geschäftsprozessmodellierungsanwendung für  
Multitouch Geräte

Hauke Wittern  
Konzeption, Entwicklung und Evaluierung einer  
Geschäftsprozessmodellierungsanwendung für  
Multitouch Geräte

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Studiengang Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Olaf Zukunft  
Zweitgutachter : Prof. Dr. Stefan Sarstedt

Abgegeben am 20. April 2011

**Hauke Wittern**

**Thema der Arbeit**

Konzeption, Entwicklung und Evaluierung einer Geschäftsprozessmodellierungsanwendung für Multitouch Geräte

**Stichworte**

Multitouch, Surface, Geschäftsprozessmodellierung, Usability-Test

**Kurzzusammenfassung**

Im Rahmen dieser Masterarbeit wird eine Anwendung konzipiert, entwickelt und evaluiert, mit der mehrere Personen gemeinsam an einem Multitouch-Tisch oder Multitouch-Bildschirm Geschäftsprozesse modellieren können. Mit Hilfe eines Usability-Tests wird belegt, dass Multitouch-Tische besser für die kollaborative Geschäftsprozessmodellierung geeignet sind, als klassische Computer, Multitouch-Bildschirme und Tafeln.

**Hauke Wittern**

**Title of the paper**

Design, development and evaluation of a business process modeling application for multitouch-devices

**Keywords**

Multitouch, Surface, Business Process Modeling, Usability-Test

**Abstract**

Within this master thesis a business process modeling application is designed, implemented and evaluated. This application allows a small number of users to model business processes collaboratively. Based on a usability-test with this application it is proven that multitouch-tables are more appropriate for collaborative business process modeling than classic computers, multitouch-screens and whiteboards.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>7</b>
1.1. Inhaltlicher Aufbau der Arbeit . . . . .	8
<b>2. Grundlagen</b>	<b>10</b>
2.1. Grundlagen der Geschäftsprozessmodellierung . . . . .	10
2.1.1. Definition: Geschäftsprozess . . . . .	10
2.1.2. Modellierung von Geschäftsprozessen . . . . .	11
2.1.3. Ziele der Geschäftsprozessmodellierung . . . . .	13
2.1.4. Geschäftsprozessmodellierungssprachen . . . . .	13
2.2. Business Process Model and Notation (BPMN) . . . . .	15
2.2.1. Die Elemente der Notation . . . . .	16
2.3. Multitouch-Benutzungsschnittstellen . . . . .	19
2.3.1. Direkte Eingabe . . . . .	19
2.3.2. Interaktion mit Gesten . . . . .	20
2.3.3. Präzision von Toucheingabe . . . . .	21
2.3.4. Texteingabe bei touchbasierten Benutzungsoberflächen . . . . .	21
2.4. Benutzungsschnittstellen von Tabletop-Computern . . . . .	22
2.4.1. Ausrichtung von Elementen der Benutzungsoberfläche . . . . .	22
2.4.2. Physische Objekte . . . . .	23
2.4.3. Zusammenhang zwischen Tischgröße, Anzahl der Benutzer und der Arbeitsweise der Benutzer . . . . .	25
2.5. Kollaboratives Arbeiten . . . . .	25
2.5.1. Kollaboratives Modellieren . . . . .	26
2.5.2. Auswirkungen der Arbeitsplatz-Konfiguration auf das kollaborative Arbeiten . . . . .	27
2.5.3. Die Bedeutung von Multitouch für kollaboratives Arbeiten . . . . .	27
2.5.4. Indikatoren für Awareness . . . . .	27
2.6. Model-View-ViewModel Architekturmuster . . . . .	29
<b>3. Analyse</b>	<b>31</b>
3.1. Auswahl der Modellierungssprache . . . . .	31
3.1.1. Anforderungen an die Modellierungssprache . . . . .	31
3.1.2. Auswahl der Modellierungssprache für Geschäftsprozesse . . . . .	33

---

3.2. Anforderungen . . . . .	35
<b>4. Entwurf und Implementierung</b>	<b>37</b>
4.1. Modi der Benutzungsoberfläche . . . . .	37
4.1.1. Modellierungsmodus . . . . .	38
4.1.2. Simulationsmodus . . . . .	39
4.1.3. Aktivierung des Modellierungs- und Simulationsmodus . . . . .	48
4.2. Architektur . . . . .	50
4.2.1. Fachliche Architektur . . . . .	50
4.2.2. Technische Architektur und Implementierung . . . . .	59
4.3. Entwurf der Benutzungsoberfläche . . . . .	64
4.3.1. Entwurfsmetaphern Werkzeug, Material und Arbeitsfläche . . . . .	64
4.3.2. Bereiche der Benutzungsoberfläche . . . . .	64
4.3.3. Benutzungsoberfläche für Computer mit vertikalen Multitouch- Bildschirmen . . . . .	66
4.3.4. Benutzungsoberfläche für Tabletop-Computer . . . . .	68
<b>5. Konzeption und Durchführung der Usability-Studie</b>	<b>77</b>
5.1. Testziel . . . . .	77
5.2. Testablauf und Szenario . . . . .	77
5.2.1. Szenario . . . . .	78
5.3. Probanden . . . . .	79
5.4. Testumgebungen . . . . .	79
5.4.1. Multitouch-Bildschirm . . . . .	80
5.4.2. Multitouch-Tisch (Microsoft Surface) . . . . .	81
5.4.3. Notebook und Beamer . . . . .	82
5.4.4. Whiteboard . . . . .	83
5.4.5. Reihenfolge der Testumgebungen bei den Testläufen . . . . .	83
<b>6. Testauswertung</b>	<b>85</b>
6.1. Charakterisierung der Modellierungssitzungen nach Testumgebung . . . . .	85
6.1.1. Notebook und Beamer . . . . .	86
6.1.2. Multitouch-Bildschirm . . . . .	87
6.1.3. Multitouch-Tisch (Microsoft Surface) . . . . .	89
6.1.4. Whiteboard . . . . .	90
6.1.5. Probandenspezifische Beobachtungen . . . . .	90
6.2. Auswertung der Awareness der Probanden . . . . .	91
6.2.1. Positive Awareness-Indikatoren . . . . .	91
6.2.2. Negative Awareness-Indikatoren . . . . .	97
6.2.3. Indikatoren für Awareness Work . . . . .	98
6.2.4. Bewertung . . . . .	100

---

6.3. Gesprächsablauf abhängig vom Blickkontakt . . . . .	101
6.3.1. Ablauf von Gesprächen mit Blickkontakt . . . . .	101
6.3.2. Ablauf von Gesprächen ohne Blickkontakt . . . . .	102
6.3.3. Schlussfolgerung . . . . .	103
6.3.4. Blickkontakt bei den einzelnen Testumgebungen . . . . .	104
6.4. Statistische Auswertung . . . . .	104
6.4.1. Produktivität und Anteile von Gesprächsphasen sowie Modellierung .	105
6.4.2. Anteil der Probanden am Gespräch . . . . .	110
6.4.3. Bewertung . . . . .	113
6.5. Feedback der Probanden . . . . .	114
6.5.1. Nützlichkeit der Simulation . . . . .	114
6.5.2. Bewertung der Zusammenarbeit . . . . .	114
6.5.3. Ergonomie und Sitzposition . . . . .	115
6.5.4. Lesbarkeit von Text und Modellen . . . . .	115
6.5.5. Bearbeitung von Modellen . . . . .	116
6.5.6. Texteingabe . . . . .	117
6.5.7. Verwendung physischer Objekte beim Surface . . . . .	117
6.5.8. Eignung der Geräte für die Geschäftsprozessmodellierung . . . . .	118
<b>7. Bewertung</b>	<b>119</b>
7.1. Argumente für Multitouch-Tische . . . . .	119
7.2. Argumente gegen Multitouch-Tische . . . . .	121
7.3. Schlussfolgerung . . . . .	123
<b>8. Zusammenfassung und Ausblick</b>	<b>125</b>
8.1. Ausblick . . . . .	126
<b>A. Usability Test</b>	<b>127</b>
A.1. Aufgabenstellung für die Fachexpertenrolle . . . . .	127
A.2. Aufgabenstellung für die Entwicklerrolle . . . . .	128
<b>Tabellenverzeichnis</b>	<b>130</b>
<b>Abbildungsverzeichnis</b>	<b>131</b>
<b>Glossar</b>	<b>133</b>
<b>Literaturverzeichnis</b>	<b>135</b>

# 1. Einleitung

Die Geschäftsprozesse, welche in einem Unternehmen ablaufen, sind am besten den Personen bekannt, die beim Ablauf der Geschäftsprozesse eine aktive Rolle spielen. Deshalb beziehen die Personen, die Geschäftsprozesse modellieren, solche Fachexperten bei der Modellierung mit ein. Die Fachexperten wiederum müssen keine Experten in der Geschäftsprozessmodellierung sein. Die unterschiedlichen Kenntnisse können jedoch zu folgenschweren Missverständnissen führen, denn die Fachexperten könnten wegen der ihnen nicht vertrauten Modellierungssprache ein Geschäftsprozessmodell anders interpretieren als die Modellierungsexperten. Infolgedessen könnten die Personen unbemerkt ein Geschäftsprozessmodell anfertigen, das die Realität nicht korrekt abbildet. Das Risiko von Missverständnissen kann eine Modellierungssprache reduzieren, welche Geschäftsprozesse anschaulich darstellt und nicht so komplex ist, dass sie die Fachexperten überfordert. Missverständnisse können jedoch nur Gespräche aufdecken, in denen die an der Modellierung teilnehmenden Personen über Geschäftsprozesse und deren Modelle diskutieren. Die Arbeitsumgebung, in der modelliert wird, muss deshalb allen Personen ermöglichen, sich an der Modellierung zu beteiligen, und darf Diskussionen nicht unterbinden.

Die an einer Modellierungssitzung teilnehmenden Personen geben die Geschäftsprozessmodelle üblicherweise entweder in einen Computer ein oder zeichnen sie auf eine Tafel. Ein Vorteil von Computern ist, die Modelle in digitaler Form vorliegen zu haben und somit nach einer Modellierungssitzung weiterverarbeiten zu können. Wenn ein klassischer Computer mit Maus und Tastatur verwendet wird, ist eine Person dafür bestimmt, gemäß der Anleitung der anderen Personen das Geschäftsprozessmodell am Computer zu modellieren, da nur eine Person zurzeit die Maus und Tastatur verwenden kann.

In den letzten rund fünf Jahren hat sich neben der Maus und der Tastatur die Multitouch-Eingabe etabliert. Diese ermöglicht, dass mehrere Benutzer gleichzeitig einen Computer mit ihren Fingern bedienen. Eine Übergabe der Kontrolle über den Computer ist dabei nicht nötig. Deshalb wird in der Literatur häufig propagiert, Multitouch-Computer beim kollaborativen Arbeiten einzusetzen. Insbesondere sogenannte Tabletop-Computer beziehungsweise Multitouch-Tische, bei denen die Tischplatte ein Multitouch-Bildschirm ist, versprechen eine besondere Eignung für das gemeinsame Arbeiten, weil mehrere Personen um den Tisch herum sitzen können. Diese Eigenschaften von Multitouch-Geräten und vor allem von Multitouch-Tischen scheinen diese Geräte besser für die Geschäftsprozessmodellierung zu

qualifizieren als klassische Computer. Denn diese Eigenschaften passen zu den oben genannten Anforderungen an eine Arbeitsumgebung, die das Aufdecken von Missverständnissen zwischen Fachexperten und Modellierungsexperten fördert.

Zahlreiche Studien belegen einzelne Stärken und Schwächen von Tabletop-Computern und Multitouch-Bildschirmen. Allerdings gibt es trotz der langjährigen Forschung kaum Anhaltspunkte, für welche kollaborativen Tätigkeiten Tabletop-Computer oder Multitouch-Bildschirme in der Praxis besser geeignet sind als herkömmliche Computer oder eine Arbeitsumgebung ohne Computer. In den Studien wurden bisher eher einfache und praxisferne Szenarien untersucht (z. B. Sortieren von Fotos). Untersuchungsergebnisse zu komplexen Szenarien, die auch in Unternehmen vorkommen, gibt es bislang nicht. Somit gibt es auch noch keinen Nachweis für die Eignung von Multitouch-Geräten für die Geschäftsprozessmodellierung.

Das Ziel dieser Arbeit ist zu beantworten, ob Multitouch-Tische bei der Geschäftsprozessmodellierung mit unterschiedlich qualifizierten Fachexperten und Modellierungsexperten tatsächlich besser geeignet sind als andere (klassische oder multitouch) Computer und Arbeitsumgebungen ohne Computer.

Um diese Frage zu beantworten, wird in dieser Arbeit eine Geschäftsprozessmodellierungsanwendung konzipiert, entwickelt und in einem Usability-Test evaluiert. Bei dem Usability-Test kommt die Geschäftsprozessmodellierungsanwendung mit einem Multitouch-Tisch, einem Multitouch-Bildschirm und einem Notebook, welches an einen Beamer angeschlossen ist, zum Einsatz. Außerdem werden Modellierungssitzungen mit einem Whiteboard durchgeführt. Mit diesen vier Geräten soll die Geschäftsprozessmodellierung in vier verschiedenen Arbeitsumgebungen verglichen werden. Mit Hilfe der Ergebnisse des Usability-Tests wird die Eignung der vier verwendeten Geräte für die Geschäftsprozessmodellierung bewertet und die oben genannte primäre Fragestellung dieser Arbeit beantwortet.

## 1.1. Inhaltlicher Aufbau der Arbeit

Nach dieser Einleitung bespricht das zweite Kapitel die wichtigsten Grundlagen, auf der diese Arbeit aufbaut. Dazu gehören die Geschäftsprozessmodellierung und die Modellierungssprache BPMN 2.0. Weitere Grundlagen sind die Forschungsergebnisse zu Multitouch-Benutzungsschnittstellen im allgemeinen und spezielle Eigenschaften von Benutzungsschnittstellen bei Multitouch-Tischen.

Im dritten Kapitel werden Anforderungen an eine Geschäftsprozessmodellierungssprache aufgestellt, die für die Geschäftsprozessmodellierung mit Multitouch-Geräten geeignet ist. Anhand dieser wird anschließend eine Modellierungssprache ausgewählt. Zudem werden



die Anforderungen genannt, welche die Modellierungsanwendung erfüllen muss, um für die kollaborative Geschäftsprozessmodellierung mit Multitouch-Geräten geeignet zu sein.

Das vierte Kapitel beschreibt den Entwurf der Geschäftsprozessmodellierungsanwendung. Dabei liegt der Fokus auf der Benutzungsschnittstelle, da diese entscheidend für die Erfüllung der Anforderungen ist. Das vierte Kapitel beschreibt außerdem die fachliche und technische Architektur, mit der die Geschäftsprozessmodellierungsanwendung realisiert wurde.

Das fünfte Kapitel erläutert, wie die Geschäftsprozessmodellierungsanwendung in einem Usability-Test evaluiert wird. Unter anderem werden das Szenario und die vier verwendeten Testumgebungen beschrieben. Anschließend präsentiert das sechste Kapitel die Ergebnisse des Usability-Tests. Auf diese stützt sich die Argumentation im siebten Kapitel. Dort wird argumentiert, welches Gerät für die Geschäftsprozessmodellierung am geeignetsten ist.

Zum Schluss fasst das achte Kapitel die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf die zu ziehenden Konsequenzen für die Geschäftsprozessmodellierung und den Einsatz von Multitouch-Geräten.

## 2. Grundlagen

Dieses Kapitel erläutert die Grundlagen, auf denen diese Arbeit aufbaut. Abschnitt [2.1](#) definiert zunächst, was ein Geschäftsprozess ist. Anschließend wird erläutert, worum es bei der Geschäftsprozessmodellierung geht, und es wird ein Überblick über vier Modellierungssprachen gegeben. In Abschnitt [2.2](#) wird die Modellierungssprache BPMN, auf welche diese Arbeit im weiteren Verlauf Bezug nimmt, genauer beschrieben. Die Abschnitte [2.3](#) und [2.4](#) nennen die in dieser Arbeit relevanten Forschungsergebnisse zu Multitouch-Benutzungsschnittstellen beziehungsweise Benutzungsschnittstellen von Tabletop-Computern. Abschnitt [2.5](#) erläutert, was man unter kollaborativem Arbeiten versteht, wie das kollaborative Arbeiten beim Modellieren aussieht und welche Auswirkungen die Arbeitsplatz-Konfiguration dabei haben kann. Zum Schluss dieses Kapitels stellt Abschnitt [2.6](#) das Model-View-ViewModel Architekturmuster vor, welches beim Entwurf der Architektur der Geschäftsprozessmodellierungsanwendung eine Rolle spielt.

### 2.1. Grundlagen der Geschäftsprozessmodellierung

#### 2.1.1. Definition: Geschäftsprozess

Allgemein ist ein Prozess eine inhaltlich abgeschlossene Folge von Aktivitäten. Eine Aktivität ist eine Handlung, die ein Akteur (eine Person oder ein Computer) durchführt. Der Anfang und das Ende eines Prozesses sind genau definiert. Durch die Aktivitäten wird eine Eingabe (Input) in eine Ausgabe (Output) umgewandelt (siehe [Abbildung 2.1](#)). Die Literatur definiert einen Geschäftsprozess als einen Prozess mit speziellen Eigenschaften. So ist ein Geschäftsprozess ein Vorgang, der in einem Unternehmen wiederholt abläuft und aus wertschöpfenden Aktivitäten besteht [[Schmelzer und Sesselmann \(2010\)](#), [Vossen und Becker \(1996\)](#)].

Die Geschäftsprozesse, die in einem Unternehmen ablaufen, erzeugen die von den Kunden des Unternehmens erwarteten Leistungen. Die Eingabe eines Geschäftsprozesses sind die Anforderungen der Kunden. Diese werden durch ein betriebswirtschaftlich relevantes Objekt, wie zum Beispiel einen Vertrag, verkörpert. Die Aktivitäten des Geschäftsprozesses wirken wertschöpfend auf die Eingabe und erzeugen das Ergebnis. Zum Beispiel könnte

ein Produkt gemäß einem Kaufvertrag hergestellt werden. Aktivitäten, die nicht zur Wertschöpfung beitragen, werden nicht zu einem Geschäftsprozess gezählt. Das Ergebnis des Geschäftsprozesses sind die von dem Kunden erwarteten Leistungen (siehe Abbildung 2.1). Diese können Produkte oder Dienstleistungen sein. [Schmelzer und Sesselmann (2010), Vossen und Becker (1996), Riekhof (1997)]

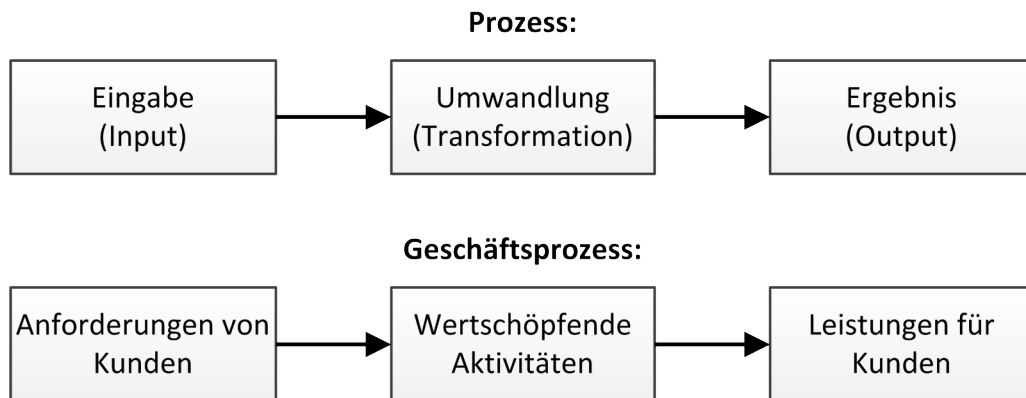


Abbildung 2.1.: Input-Output-Beziehung in Prozessen und Anforderungs-Leistungs-Beziehung in Geschäftsprozessen (nach [Schmelzer und Sesselmann (2010)]).

Die Aufbauorganisation eines Unternehmens schränkt die Reichweite eines Geschäftsprozesses nicht ein. Ein Geschäftsprozess kann funktions- und abteilungsübergreifende Aktivitäten beinhalten. Ebenso können Aktivitäten zu einem Geschäftsprozess gehören, die Kunden, Zulieferer oder Partner eines Unternehmens durchführen [Schmelzer und Sesselmann (2010)].

In einem Unternehmen laufen üblicherweise mehrere Geschäftsprozesse ab. Jeder Geschäftsprozess kann in Teilprozesse aufgegliedert werden. Ein einzelner Geschäftsprozess erfüllt direkt (z. B. Produktion) oder indirekt (z. B. Marketing) ein Teilziel des Unternehmens. Alle Geschäftsprozesse zusammen erfüllen das gesamte Unternehmensziel.

### 2.1.2. Modellierung von Geschäftsprozessen

Die Modelle der Geschäftsprozesse eines Unternehmens sind fachliche Modelle, die das Unternehmen aus einer betriebswirtschaftlichen Sicht beschreiben. Ein Modell ist eine abstrakte Repräsentation einer realen Welt. Die abstrakte Sichtweise hilft, Sachverhalte zu verstehen und reale Systeme zu gestalten. Dementsprechend helfen Geschäftsprozessmodelle, die Abläufe von Geschäftsprozessen zu verstehen und zu verändern [Vossen und Becker (1996)].

In ein Modell werden nur die Eigenschaften der realen Welt übernommen, die für den Zweck des Modells dienlich sind (Verkürzungsmerkmal). Bei Geschäftsprozessen werden die Aktivitäten und die Ablaufstrukturen modelliert. Die Ablaufstruktur beschreibt, in welcher Reihenfolge die Aktivitäten ausgeführt werden. Die wichtigsten Eigenschaften der Ablaufstruktur sind Sequenzen, Entscheidungen und Nebenläufigkeit. Diese können zum Beispiel als alternative Abläufe, parallele Abläufe oder Wiederholungen auftreten. Weitere Eigenschaften von Geschäftsprozessen, die häufig modelliert werden, sind:

- verwendete Ressourcen,
- ausgetauschte Datenobjekte,
- ausgetauschte Nachrichten und Kommunikationsstrukturen,
- Zeitaspekte,
- Geschäftsregeln,
- Rollen der Teilnehmer am Geschäftsprozess

Abbildung 2.2 zeigt ein Beispiel für ein Geschäftsprozessmodell mit Aktivitäten, Sequenzen, Entscheidungen und Parallelität.

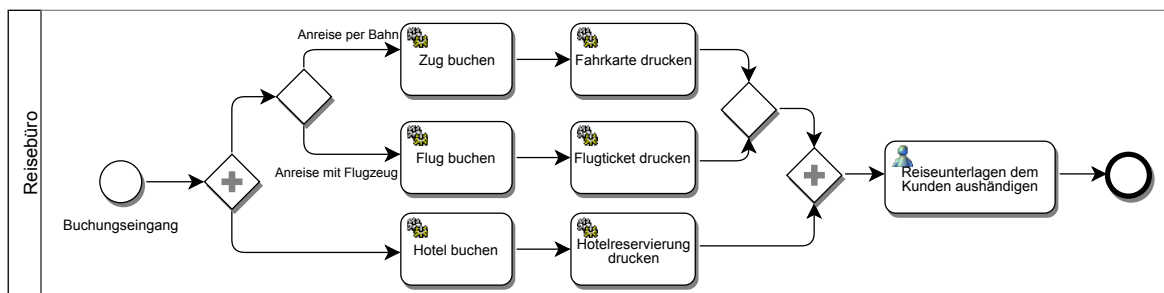


Abbildung 2.2.: Beispiel für einen mit der BPMN erstellten Geschäftsprozess

Geschäftsprozesse werden immer von Fachexperten modelliert, welche die Geschäftsprozesse verstehen. Solche Personen können Systemanalytiker, Sachbearbeiter und externe Berater sein. Die Geschäftsprozessmodelle können für die gleiche Personengruppe oder für Softwareentwickler oder das Management erstellt werden. An einer Modellierungssitzung können neben den modellierenden Personen auch die Personen teilnehmen, für die modelliert wird. Durch die Teilnahme der Modell-Konsumenten können Missverständnisse zwischen den Personen vermieden werden.

### 2.1.3. Ziele der Geschäftsprozessmodellierung

Ein Unternehmen strebt stets danach, seine unternehmerischen Ziele zu erreichen. Um diese Ziele zu erreichen, ergreift ein Unternehmen Maßnahmen, deren Grundlage Geschäftsprozessmodelle sein können. Geschäftsprozessmodelle sind wegen ihrer fachlichen Nähe als Grundlage für Maßnahmen, welche die betrieblichen Abläufe betreffen, besonders geeignet. Solche Maßnahmen können zum einen rein betriebswirtschaftlicher Natur sein, beispielsweise wenn das Management den Ablauf eines Geschäftsprozesses optimiert. Zum anderen kann ein System zur elektronischen Datenverarbeitung eingeführt werden, um einzelne Aktivitäten oder den gesamten Ablauf eines Geschäftsprozesses zu unterstützen. In diesem Zusammenhang spielt die Geschäftsprozessmodellierung bei der Informatik eine Rolle. Bei folgenden Szenarien werden Geschäftsprozessmodelle häufig verwendet:

- Dokumentation von Geschäftsprozessen.
- Als Diskussionsgrundlage zwischen verschiedenen Personengruppen (z.B. Management, Softwareentwickler).
- Betrieb von Workflow-Management-Systemen.
- Entwicklung von Standardsoftware.
- Simulation von Geschäftsprozessen.
- Analyse von Geschäftsprozessen.
- Überwachung von Geschäftsprozessen.
- Tiefgreifende und radikale Umgestaltung von Geschäftsprozessen durch Business Process Reengineering.
- Optimierung von Geschäftsprozessen durch kontinuierlich Umgestaltung in kleinen Schritten beim (Kontinuierlicher Verbesserungsprozess).

### 2.1.4. Geschäftsprozessmodellierungssprachen

Dieser Abschnitt gibt einen Überblick über vier Modellierungssprachen, mit denen Geschäftsprozesse modelliert werden können: BPMN, EPK, Petri Netze und UML-Aktivitätsdiagramme. Diese Modellierungssprachen spielen in der Praxis oder Theorie eine große Rolle [Ko u. a. (2009), Becker u. a. (2010)].

#### 2.1.4.1. Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN, früher: Business Process Modeling Notation) ist ein von der Object Management Group (OMG) standardisiertes Metamodell und eine zugehörige Notation für Geschäftsprozesse. Das vorrangige Ziel der BPMN ist, eine für alle Anwender einfach verständliche Notation bereitzustellen [OMG (2011)]. Damit soll die Kluft zwischen dem Entwurf und der Implementierung eines Geschäftsprozesses überbrückt werden.

Ein BPMN Diagramm stellt einen Geschäftsprozess als einen gerichteten Graphen dar. Die Aktivitäten, die während eines Geschäftsprozesses ausgeführt werden, werden durch Knoten repräsentiert und den Teilnehmern an dem Geschäftsprozess zugeordnet, die sie ausführen. Die Reihenfolge der Aktivitäten eines Teilnehmers werden durch den Kontrollfluss bestimmt. Zwischen den Teilnehmern werden die Aktivitäten durch einen Nachrichtenaustausch koordiniert. Der Kontrollfluss und Nachrichtenaustausch werden durch gerichtete Kanten dargestellt.

#### 2.1.4.2. Ereignisgesteuerte Prozesskette (EPK)

Die Ereignisgesteuerte Prozesskette (EPK) ist eine semiformale und grafische Modellierungssprache für Geschäftsprozesse. Sie ist ein zentrales Konzept der Architektur Integrierter Informationssysteme (ARIS) von August-Wilhelm Scheer [Scheer (1998)].

Die EPK stellt einen Geschäftsprozess als einen Graph dar, dessen Knoten Ereignisse, Funktionen oder Verknüpfungsoperatoren sind. Die Ausführung einer Funktion folgt immer auf ein Ereignis und ist wiederum der Auslöser eines oder mehrerer Ereignisse. Mit gerichteten Kanten werden die inhaltlichen und zeitlichen Abhängigkeiten zwischen den Ereignissen und Funktionen beschrieben.

#### 2.1.4.3. Petri-Netze

Carl-Adam Petri hat in seiner Dissertation ein Netz aus Bedingungen und Ereignissen definiert, welches als formales Modell für nebenläufige und verteilte Systeme dienen kann [Petri (1962)]. Ein solches Netz wurde später nach Carl-Adam Petri benannt und als Petri-Netz bezeichnet.

Petri-Netze werden mathematisch mit mehreren Mengen beschrieben, die einen gerichteten Graphen definieren. Ein solcher Graph veranschaulicht die kausalen Zusammenhänge in dem modellierten System. Die Knoten des Graphens sind Stellen und Transitionen, welche

mit gerichteten Kanten verbunden sind. Ein bestimmter Zustand des modellierten Systems wird durch Markieren der Stellen mit Tokens beschrieben.

Petri-Netze sind besonders für die Modellierung nebenläufiger beziehungsweise verteilter Systeme geeignet, da sie keinen globalen Synchronisationsmechanismus benötigen. Aufgrund ihrer sehr umfangreich erforschten Theorie, sind Petri-Netze auch für eine Analyse und Simulation solcher Systeme sehr gut geeignet. Mit Petri-Netzen können aber auch Geschäftsprozesse modelliert werden. Allerdings kann ein Petri-Netz nicht immer alle relevanten Eigenschaften eines Geschäftsprozess abbilden. Beispielsweise können Attribute von Datenobjekten mit der ursprünglichen Form von Petri-Netzen nicht modelliert werden. Aus diesem Grund gibt es zahlreiche Erweiterungen, wie zum Beispiel gefärbte Petri-Netze [Jensen (1997)].

#### 2.1.4.4. UML 2 Aktivitätsdiagramme

Aktivitätsdiagramme sind eine Diagrammart der Unified Modeling Language (UML) zur Modellierung von Abläufen innerhalb einer Aktivität. Die UML ist eine grafische Modellierungssprache, die aus mehreren objektorientierten Methoden zur Softwareentwicklung hervorgegangen ist. Sie wird deshalb primär für die Analyse, den Entwurf und die Implementierung objektorientierter Software eingesetzt. Sie ist aber für die Softwareentwicklung im Allgemeinen geeignet. Ebenso kann sie auch außerhalb der Softwareentwicklung für die fachliche Modellierung eingesetzt werden, wie zum Beispiel der Modellierung von Geschäftsprozessen [OMG (2010)]. Aktivitätsdiagramme werden bei der Softwareentwicklung zur Modellierung des Verhaltens eines Systems verwendet.

Ein Aktivitätsdiagramm beschreibt eine bestimmte Aktivität. Eine Aktivität kann beispielsweise bei der objektorientierten Softwareentwicklung eine Methode einer Klasse repräsentieren oder bei der Geschäftsprozessmodellierung einen Geschäftsprozess. Ein Aktivitätsdiagramm spezifiziert, welche Aktionen während einer Aktivität ausgeführt werden und wie die Ausführung dieser Aktionen mit Kontroll- und Datenflüssen koordiniert wird [OMG (2010)]. Die Aktionen werden als Knoten eines Graphen dargestellt und die Kontroll- und Datenflüssen als gerichtete Kanten, welche Aktionen und Kontrollpunkte miteinander verbinden.

## 2.2. Business Process Model and Notation (BPMN)

Wie zuvor skizziert ist Business Process Model and Notation<sup>1</sup> (BPMN) ein Metamodell und eine Notation für Geschäftsprozesse [OMG (2011)]. Abbildung 2.2 zeigt ein Beispiel für einen

<sup>1</sup>Seit der Version 2.0 steht das Akronym BPMN für Business Process Modeling Notation. Vorher stand es für Business Process Modeling Notation

mit der BPMN beschriebenen Geschäftsprozess. Der aktuelle BPMN 2.0 Standard spezifiziert neben den Elementen des Metamodells und deren Darstellung in einem Diagramm auch ein XML-basiertes Dokumentenaustauschformat. Dieses Kapitel gibt einen Überblick über die Elemente, die in dieser Arbeit verwendet wurden.

### 2.2.1. Die Elemente der Notation

Die BPMN umfasst zahlreiche Elemente zur Beschreibung von Geschäftsprozessen. Eine kleine Untermenge der Elemente reicht aus, um den Ablauf eines Prozesses und seine wichtigsten Eigenschaften zu beschreiben. Viele Elemente werden nur benötigt, um die Semantik eines Prozesses präzise zu beschreiben. Wenn zum Beispiel ein Geschäftsprozess ausgeführt oder sein Ablauf analysiert werden soll, muss sein Modell eine präzise Semantik haben. Wenn das Modell eines Geschäftsprozesses aber nur zum Verständnis und Dokumentationszwecken dient, reicht es häufig aus, nicht alle Details zu modellieren. [Silver \(2009\)](#) teilt die Elemente der BPMN deshalb in drei Stufen ein: descriptive BPMN (Level 1), analytical BPMN (Level 2) und executable BPMN (Level 3). Die "Level 1 Modeling Palette" ist die kleinste Untermenge und reicht für die Beschreibung von Prozessen zu Dokumentationszwecken aus. Diese Arbeit orientiert sich an der Level 1 Modeling Palette, da die konzipierte Geschäftsprozessmodellierungsanwendung den Entwicklern vor allem in frühen Projektphasen helfen soll, die Geschäftsprozesse der Kunden zu verstehen und zu modellieren. Im Folgenden werden die verwendeten Notationselemente gemäß der BPMN Spezifikation ([\[OMG \(2011\)\]](#)) beschrieben.

#### 2.2.1.1. Process und Pool

Bei BPMN werden die Aktivitäten eines Geschäftsprozesses den Teilnehmern (Participants) zugeordnet. Pools ([Abbildung 2.3](#)) repräsentieren graphisch die Teilnehmer an einem Geschäftsprozess. Die Aktivitäten, die ein Teilnehmer während eines Geschäftsprozesses durchführt, kapselt die BPMN in einem "Process". Ein Pool dient als Behälter für den Process. Die BPMN modelliert einen Geschäftsprozess als das Zusammenspiel (Collaboration) mehrerer Processes. Ein Process ist nicht äquivalent mit einem Geschäftsprozess.

Ein BPMN Diagramm stellt einen Process als einen aus Knoten (Flow Nodes) und gerichteten Kanten (Sequence Flows) bestehenden Graphen dar. Aktivitäten (Activities), Ereignisse (Events) und Kontrollpunkte (Gateways) der Ablaufstruktur eines Geschäftsprozesses werden durch Knoten repräsentiert. Sequence Flows verbinden die Knoten und bestimmen zusammen mit den Gateways die Abfolge der Aktivitäten.





Abbildung 2.3.: Ein Pool

Das Verhalten eines Geschäftsprozesses kann bei BPMN mit dem Konzept eines **Tokens**, welches den Graphen traversiert, veranschaulicht werden. Dieses Konzept ähnelt Petri-Netzen und soll das Verständnis erleichtern. Die BPMN 2.0 Spezifikation weist aber deutlich darauf hin, dass dies bloß ein theoretisches Konzept ist und BPMN Werkzeuge es nicht implementieren müssen.

### 2.2.1.2. Activities

Die BPMN klassifiziert verschiedene Arten von Aktivitäten (Abbildung 2.4). Tasks sind atomare Aktivitäten. Je nachdem, ob ein Task von einer Person oder automatisiert von einem Computersystem durchgeführt wird, unterscheidet die BPMN zwischen User Tasks und Service Tasks. In einem Prozess kann ein anderer Prozess als zusammengesetzte Aktivität verwendet werden, was die BPMN als Sub-Process bezeichnet.



Abbildung 2.4.: Die Aktivitäten Service Task, und User Task und Sub-Process

### 2.2.1.3. Events

Ereignisse, die in einem Prozess auftreten, bezeichnet die BPMN als Events. Events können Start Events, Intermediate Events oder End Events sein.

Ein Start Event (Abbildung 2.5) definiert den Anfang eines Prozesses. Es gibt verschiedene Typen von Start Events, mit denen der Auslöser (Trigger) des Ereignisses angegeben werden kann. Mögliche Auslöser sind zum Beispiel der Empfang einer Nachricht (Message) oder



Abbildung 2.5.: Start Events (None, Message, Timer)

ein bestimmter Zeitpunkt (Timer). Der Auslöser kann auch offen gelassen werden (None). Üblicherweise hat ein Prozess ein Start Event.

Ein End Event (Abbildung 2.6) definiert das Ende eines Prozesses. Der Typ eines End Events bestimmt das Resultat des Ereignisses. Das Resultat kann unter anderem der Versand einer Nachricht oder der Abbruch aller in dem Prozess noch laufenden Aktivitäten sein. Das Resultat kann auch undefiniert sein (None). Ein Prozess kann über mehrere End Events verfügen.



Abbildung 2.6.: End Events (None, Message, Terminate)

Intermediate Events (Abbildung 2.7) signalisieren Ereignisse, die während eines Prozesses auftreten. Mit dem Typ des Intermediate Events kann der Auslöser oder das Resultat des Ereignisses angegeben werden.

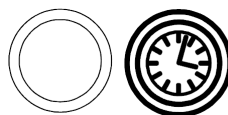


Abbildung 2.7.: Intermediate Events (None, Timer)

#### 2.2.1.4. Gateways

Neben den Sequence Flows definieren Gateways die Ablaufstruktur eines Prozesses. Die wichtigsten Gateways sind das Exclusive Gateway und das Parallel Gateway.

Exclusive Gateways mit *einem* eingehenden Sequence Flow und *mehreren* ausgehenden Sequence Flows repräsentieren eine Entscheidung über mehrere Alternativen für den weiteren Ablauf (exclusive Decision). Anhand der Bedingungen der ausgehenden Sequence

Flows wird genau ein ausgehender Sequence Flow ausgewählt, der weiterverfolgt wird. An Exclusive Gateways mit *mehreren* eingehenden Sequence Flows und *einem* ausgehenden Sequence Flow, laufen alternative Abläufe wieder zusammen.

Parallel Gateways mit einem eingehenden Sequence Flow und mehreren ausgehenden Sequence Flows spalten einen Ablauf in mehrere parallele Abläufe auf (parallel Split). Jeder ausgehende Sequence Flow wird weiterverfolgt. An Parallel Gateways mit mehreren eingehenden Sequence Flows werden parallele Abläufe synchronisiert wieder zusammengeführt (parallel Join). Erst wenn alle eingehende Sequence Flows abgeschlossen sind, wird mit dem ausgehenden Sequence Flow fortgefahren.

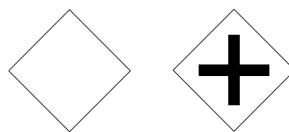


Abbildung 2.8.: Exclusive Gateway (links) und Parallel Gateway (rechts)

## 2.3. Multitouch-Benutzungsschnittstellen

Die Multitouch-Eingabe hat sich in den vergangenen Jahren als Eingabe-Methode neben der Maus- und Tastatur-Eingabe etabliert. Ein wesentlicher Faktor für die Beliebtheit von Multitouch-Eingabe ist das Multitouch-fähige iPhone, das Apple im Jahr 2007 auf den Markt gebracht hat. Im Folgenden werden die wesentlichen Eigenschaften von Multitouch-Benutzungsschnittstellen erläutert, die in dieser Arbeit eine Rolle spielen.

### 2.3.1. Direkte Eingabe

Die Touch-Eingabe ist im Gegensatz zur Maus- und Tastatur-Eingabe eine direkte Eingabemethode. Wegen der direkten Eingabe kann die Benutzbarkeit einer Benutzungsschnittstelle mit Touch-Eingabe besser sein als mit Maus und Tastatur.

Mit Eingabegeräten wie einer Maus oder einer Tastatur steuert ein Benutzer seine Eingaben indirekt. Der Mauscursor und der Tastaturcursor zeigt ihm die Position, an der er die Eingaben vornimmt. Durch Bewegen der Maus und durch Drücken der Tasten der Maus und der Tastatur teilt der Benutzer dem System seine Absichten mit. Bei einem Touchdisplay kann der Benutzer hingegen durch berühren des Displays direkt auf die Benutzungsoberfläche einwirken. Der Benutzer benötigt keinen Cursor als virtuelles Hilfsmittel. Der Berührungspunkt eines Fingers oder eines Gegenstands verkörpert den Zeiger. Seine Absichten teilt

der Benutzer dem System durch Bewegen und Entfernen des Fingers beziehungsweise Gegenstandes von der Bildschirmfläche mit.

Die direkte Eingabe kann die Bedienung einer Benutzungsschnittstelle erleichtern, denn sie ist eine natürlichere und intuitivere Eingabemethode als die indirekte und erfordert eine geringere Denkleistung vom Benutzer [Jacob u. a. (2008), Mohs u. a. (2007), Burmester und Koller (2010)]. Menschen lernen frühzeitig in ihrer Entwicklung, Dinge anzufassen und mit ihren Händen und Werkzeugen zu bearbeiten. Ihr Denkmodell ist dementsprechend geprägt. Die Fähigkeit, Computer mit Maus und Tastatur zu bedienen, lernen Menschen erst später. Die indirekte Eingabemethode passt aber nicht in ihr zuvor erlerntes Denkmodell und erfordert deshalb eine zusätzliche Denkleistung.

Ein weiterer Vorteil von direkter Touch-Eingabe ist die höhere Effizienz. Die Benutzer können touchbasierte Benutzungsschnittstellen in der Regel schneller bedienen als mausbasierte, weil sie die Elemente der Benutzungsschnittstelle schneller auswählen können [Kin u. a. (2009)]. Die Fehlerrate bei der Elementauswahl ist außerdem deutlich geringer, vorausgesetzt die Elemente sind ausreichend groß (ab ca. 1cm pro Seite) [Sears und Shneiderman (1991)].

### 2.3.2. Interaktion mit Gesten

Bei Multitouch-Benutzungsoberflächen können die Benutzer bestimmte Handbewegungen durchführen, um dem System ihre Absichten mitzuteilen. Die Software analysiert diese Handbewegungen, ordnet sie einer vorab definierten Geste zu und führt eine Funktion aus, die der Geste zugeordnet ist. Eine weit verbreitete Geste ist zum Beispiel das voneinander weg- und zueinander hin-Bewegen der Finger, die ein bestimmtes Element der Benutzungsoberfläche berühren, um das Element zu vergrößern beziehungsweise zu verkleinern. Bei Apples iPhone und Microsofts Surface können die Benutzer auf diese Weise die Größe von Fotos verändern. Solche Gesten versprechen den Benutzern, eine Benutzungsoberfläche intuitiv bedienen zu können, da sie die Eingaben mit direkter Touch-Eingabe und natürlichen Handbewegungen durchführen können [Zachmann (2001), Shen u. a. (2006)].

Die Gestenerkennung durch die Software basiert auf den folgenden Funktionen [Celentano und Minuto (2008)]:

- Sensing: Erfassung von Ereignissen (z. B.: ein Benutzer berührt die Benutzungsoberfläche mit einem Finger).
- Identifying: Identifizierung einer Geste durch Interpretierung der Ereignisse.
- Tracking: Nachverfolgung von Ereignissen. Der aktuelle Zustand wird ermittelt und mit den vorherigen Zuständen in Verbindung gebracht.

Mit diesen Funktionen führt die Software die Gestenerkennung entweder kontinuierlich aus, während ein Benutzer eine Geste macht, oder nachdem ein Benutzer eine Geste abgeschlossen hat. Die Auswertung abgeschlossener Gesten liefert weniger Fehlinterpretationen als die kontinuierliche Gestenerkennung [Celentano und Minuto (2008)]. Bei der kontinuierlichen Gestenerkennung kann die Benutzungsschnittstelle dem Benutzer aber, noch während er die Geste ausführt, eine Rückmeldung geben (z. B. verschieben eines Fotos).

### 2.3.3. Präzision von Toucheingabe

Die Präzision von Toucheingabe ist geringer, als die Mauseingabe. Einen Mauszeiger kann ein Benutzer pixelgenau platzieren. Mit den Fingern oder einem Gegenstand kann ein Benutzer einen bestimmten Pixel jedoch nicht bewusst berühren. Die Berührungsfläche ist im Vergleich zu der Fläche eines einzelnen Pixels signifikant größer und umfasst mehrere Pixel. Außerdem wird durch das Berühren der Pixel verdeckt, was das Zielen erschwert. Die geringe Präzision verursacht außerdem einen Jitter bei der Erkennung des Berührungspunktes. Während ein Benutzer seinen Finger von der Benutzungsoberfläche nimmt, ändert sich die vom Gerät erfasste Berührungsfläche und dadurch wiederum der erkannte Berührungspunkt. Deshalb können die Benutzer Elemente nur schwer mit ihren Fingern genau platzieren [Shen u. a. (2006)].

Softwareentwickler müssen die geringe Präzision bei der Entwicklung von touchbasierten Benutzungsschnittstellen beachten. Sie können Probleme bei der Auswahl von Elementen vermeiden, indem sie die Elemente ausreichend groß dimensionieren [Microsoft Corporation (2009), Shen u. a. (2006), Sears und Shneiderman (1991)]. Das Problem mit dem Jitter können Softwareentwickler mit dem von Shen u. a. (2006) vorgestellten „Gesture Termination“ Algorithmus umgehen. Dieser Algorithmus ermittelt die vom Benutzer gewünschte Position, indem er nach der Position sucht, an der ein Finger für einen Augenblick verharret hat [Shen u. a. (2006)].

### 2.3.4. Texteingabe bei touchbasierten Benutzungsoberflächen

Die Eingabe von Text ist bei touchbasierten Benutzungsoberflächen mit Problemen verbunden, welche die Benutzbarkeit reduzieren. Insbesondere bei Szenarien, in denen mehrere Benutzer mit einer Anwendung interagieren, ist die Eingabe von Text ein Problem.

Bei touchbasierten Benutzungsoberflächen können die Benutzer eine Tastatur verwenden, wie sie es von herkömmlichen Computern gewohnt sind. Allerdings ist dies eine indirekte Eingabemethode und nutzt nicht die Vorteile direkter Eingabemethoden (siehe 2.3.1). Außerdem verhindert der Wechsel zwischen den Eingabemethoden (Touch und Tastatur) einen

flüssigen Arbeitsablauf [Scott u. a. (2003)]. Der Arbeitsablauf wird ebenfalls unterbrochen, wenn mehrere Benutzer Text eingeben möchten, da eine Übergabe der Kontrolle notwendig ist. Die Verwendung mehrerer Tastaturen ist wegen technischer Beschränkungen häufig nicht möglich. Bei Microsoft Windows basierten Systemen kann nur eine Tastatur effektiv benutzt werden, weil das Betriebssystem den Texteingabe-Fokus auf nur ein Control setzen kann.

Die Verwendung virtueller Bildschirmtastaturen ist keine echte Alternative. Auch eine Bildschirmtastatur ist wegen des Cursor-Konzepts eine indirekte Eingabemethode. Ebenso unterstützen viele Systeme nur eine Bildschirmtastatur. Ein zusätzliches Problem ist die fehlende Haptik bei Bildschirmtastaturen. Die fehlende Haptik ist ein generelles Problem von touchbasierten Systemen. Den Benutzern fällt die Benutzung von Bildschirmtastaturen aber besonders schwer, weil sie weder die Tasten erfühlen können noch eine spürbare Rückmeldung beim drücken einer Taste bekommen [Hoggan u. a. (2008)].

Eine direkte Eingabemethode für Text ist das Schreiben mit den Fingern. Allerdings ist dies nicht zum Schreiben von längeren Texten geeignet, da Personen mit ihren Fingern zu große Buchstaben schreiben [Ryall u. a. (2006)]. Mit einem Stift können die Benutzer Text platzsparender eingeben. Jedoch unterstützen nicht alle Touch-Geräte Eingabestifte (z. B. Microsoft Surface).

## 2.4. Benutzungsschnittstellen von Tabletop-Computern

Tabletop-Computer sind Computer bei denen ein Multitouch-Bildschirm als Tischplatte dient. Im Folgenden werden die in dieser Arbeit relevanten Eigenschaften von solchen Multitouch-Tischen erläutert.

### 2.4.1. Ausrichtung von Elementen der Benutzungsoberfläche

Im Gegensatz zu vertikalen Benutzungsoberflächen spielt bei Tabletop-Computern mit horizontalen Benutzungsoberflächen die Ausrichtung der Elemente der Benutzungsschnittstelle eine Rolle. Die Benutzer sitzen in der Regel um den Tisch herum. Jeder einzelne Benutzer hat deshalb eine andere Perspektive [Wigdor und Balakrishnan (2005), Hancock u. a. (2006), Shen u. a. (2006), Kruger u. a. (2004)].

Wegen der individuellen Perspektive, können nicht alle Benutzer eines Tabletop-Computers die angezeigten Texte schnell erfassen. Text der auf dem Kopf steht oder zumindest +/-45 Grad gedreht ist, ist vom Betrachter schwer lesbar. Dennoch zeigen einige Studien, dass dies in vielen Fällen kein gravierendes Problem ist [Wigdor und Balakrishnan (2005), Morris u. a.

(2006)]. Einzelne oder nur wenige Wörter können immer noch in akzeptabler Zeit gelesen werden [Wigdor und Balakrishnan (2005)]. In der Studie von Wigdor und Balakrishnan (2005) benötigten die Testpersonen kaum mehr als die doppelte Zeit, um gedrehte kurze Sätze zu lesen. Laut Morris u. a. (2006) hat die Ausrichtung bei großen Mengen von Text eine Bedeutung, bei geringen Mengen werde die Benutzbarkeit aber nicht stark eingeschränkt [Morris u. a. (2006)].

Ebenso kann der individuelle Blickwinkel den Benutzern das Verstehen von auf einem Tabletop-Computer angezeigten Diagrammen erschweren. Bei Diagrammen kann es eine Bedeutung haben, wie die Elemente relativ zu einander angeordnet sind. Laut einer Studie von Cherubini u. a. (2007) können gerichtete Verbindungen die Art der Beziehungen zwischen Diagrammelementen andeuten. Beispielsweise richten Softwareentwickler bei Klassendiagrammen die Unterklassen unterhalb der Oberklasse aus, um die Hierarchie zu verdeutlichen.

Um die Probleme mit der Ausrichtung zu verringern, gibt es verschiedene Techniken, die mit einander kombiniert werden können:

- Eine virtuell drehbare Arbeitsfläche ermöglicht es einem Benutzer, die Arbeitsfläche zu sich auszurichten [Shen u. a. (2006), Hancock u. a. (2006)].
- Separat drehbare Elemente kann ein Benutzer zu sich auszurichten [Morris u. a. (2006), Hancock u. a. (2006)].
- Text ist für alle sich gegenüber sitzenden Benutzer lesbar, wenn er mehrfach dargestellt wird und in verschiedene Richtungen ausgerichtet ist [Wigdor und Balakrishnan (2005)].

### 2.4.2. Physische Objekte

Physische Objekte sind eine zusätzliche Möglichkeit, mit der die Benutzer mit einem Tabletop-Computer interagieren können. Die Benutzer können Gegenstände auf die horizontale Oberfläche eines Tabletop-Computers stellen. Die Hardware einiger Tabletop-Computer kann diese Gegenstände erkennen und Informationen über die Gegenstände an die Software übergeben. Die Software kann die Informationen über die Gegenstände als Eingabe verwenden und dem Benutzer dadurch einfach bedienbare Funktionen bereitstellen.

In der Benutzungsoberfläche kann ein Gegenstand mit einem virtuellen Objekt verknüpft sein. Der Gegenstand dient dann als physischer Griff für das virtuelle Objekt. Fitzmaurice u. a. (1995) bezeichnen ein aus physischen Objekten und einem virtuellen zusammengesetztes Objekt als "graspable Object" (greifbares Objekt). Ein solches greifbares Objekt kann zum Beispiel ein Papierkorb sein, dessen Inhalt virtuell dargestellt wird (siehe Abb. 2.9).

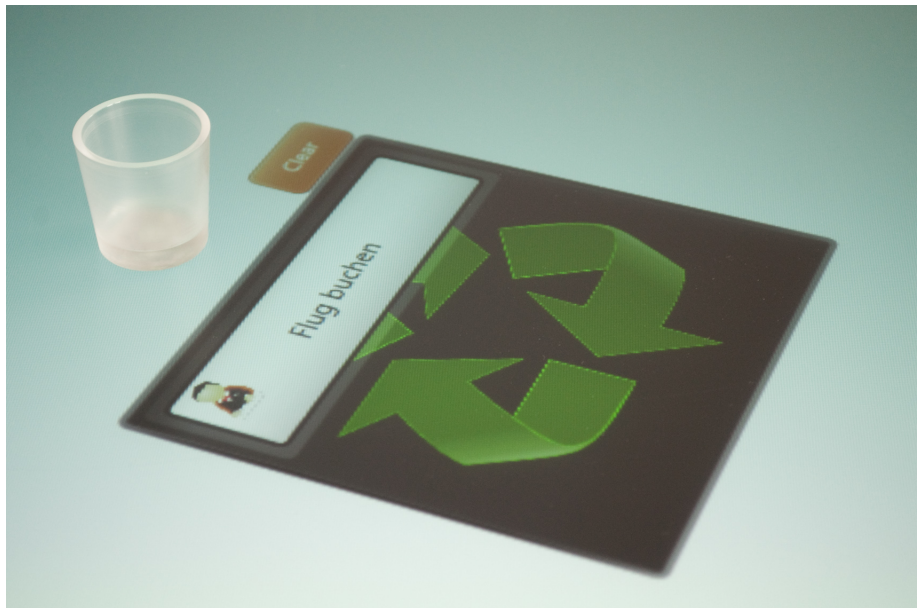


Abbildung 2.9.: Papierkorbgegenstand und seine virtuelle Repräsentation.

Benutzungsschnittstellen mit greifbaren Gegenständen als Eingabegerät gehören zu den Tangible User Interfaces (anfassbare Benutzungsschnittstellen) [Ishii und Ullmer (1997)]. Tangible User Interfaces erweitern die reale Welt durch eine Kopplung von alltäglichen physischen Objekten mit virtuellen Informationen. Mit dieser Kopplung versuchen Tangible User Interfaces, die Kluft zwischen der virtuellen Welt und der realen Welt zu überbrücken [Ishii und Ullmer (1997)]. Tangible User Interfaces nutzen die Fähigkeit des Menschen, Dinge mit den Händen anzufassen und zu manipulieren. Deshalb kann die Bedienung der Benutzungsoberfläche natürlicher und damit einfacher sein als bei klassischen Benutzungsoberflächen.

Tabletop-Computer sind eine Möglichkeit, um Tangible User Interfaces bereitzustellen. Aber nicht jede Tabletop-Computer-Technologie ist geeignet, Gegenstände zu erkennen. Einige, wie zum Beispiel der DiamondTouch [Dietz und Leigh (2001)], unterstützen nur die Toucheingabe. Das Microsoft Surface [Microsoft Surface] unterstützt neben Multitouch auch die Erkennung von Gegenständen. Das Microsoft Surface erkennt visuell die Position, Form und Art der Gegenstände. Die Art eines Gegenstands kann ein Benutzer mit einem Aufkleber (Tag) an der Unterseite des Gegenstands festlegen.



### 2.4.3. Zusammenhang zwischen Tischgröße, Anzahl der Benutzer und der Arbeitsweise der Benutzer

Die Arbeitsfläche, die den einzelnen Benutzern zur Verfügung steht, beeinflusst die Arbeitsweise der Benutzer eines Tabletop-Computers. Die Größe des Tisches sowie die Anzahl der Benutzer und deren Anordnung bestimmen, welche Bereiche der Arbeitsfläche die Benutzer für persönliche oder gemeinsame Tätigkeiten nutzen können und wie groß diese Bereiche sind. Diese Zusammenhänge müssen bei der Anwendungsentwicklung, der Hardwareauswahl und Festlegung der Benutzeranzahl beachtet werden, um eine für bestimmte Tätigkeiten angemessene Umgebung zu schaffen.

Die Größe eines Tisches ist ausschlaggebend dafür, wieviele Personen an dem Tisch komfortabel Platz nehmen können. Je dichter die Personen an einem Tisch zusammensitzen müssen, desto geringer ist die Fläche, die einer einzelnen Person für persönliche Tätigkeiten (z. B. Aufzeichnen von Notizen) zur Verfügung steht [Scott u. a. (2004), Scott (2005)]. Kleine Tische verhindern deshalb persönliche Arbeitsbereiche und dienen als gemeinsamer Arbeitsbereich. Für Tätigkeiten, bei denen die Benutzer eng zusammenarbeiten, sind kleine Tische geeignet. Große Tische ermöglichen neben einem gemeinsamen Arbeitsbereich auch persönliche Arbeitsbereiche. Jedoch können sehr große Tische gemeinsame Tätigkeiten behindern, wenn nicht alle Bereiche für einen Benutzer erreichbar und erkennbar sind [Scott (2005), Ryall u. a. (2004), Morris u. a. (2006), Ryall u. a. (2006)].

Die Größe und die Auflösung des Bildschirms bestimmen außerdem, wie viele Informationen angezeigt werden können [Rogers u. a. (2006)]. Die Arbeitsfläche der heute verfügbaren Tabletop-Computer ist im Vergleich zu herkömmlichen Arbeitstischen klein<sup>2</sup>. Ebenso ist die Bildschirmauflösung von Tabletop-Computern relativ zur Bildschirmfläche gering<sup>3</sup>. Deshalb sollte die Bildschirmfläche und Bildschirmauflösung vergrößert werden, um mehr Elemente und Details Anzeigen zu können [Rogers u. a. (2006)].

## 2.5. Kollaboratives Arbeiten

Im Kontext dieser Arbeit ist **Kollaboration** (engl. collaboration) die Zusammenarbeit mehrerer Personen an einer gemeinsamen Sache. Historisch bedeutet Kollaboration die Zusammenarbeit mit dem Feind während eines Krieges oder einer Besatzung. Diese Bedeutung wird

---

<sup>2</sup>Das Microsoft Surface hat eine Bildschirmdiagonale von 76 cm [Microsoft Corporation (2008)]. Die größte Variante des DiamondTouch von Circle Twelve hat eine Bildschirmdiagonale von 108cm [Circle Twelve Inc. (2008)]

<sup>3</sup>Das Microsoft Surface hat eine Bildschirmauflösung von 1024x768 Pixel [Microsoft Corporation (2008)].

hier nicht verwendet. Diese Arbeit konzentriert sich auf die sogenannte “co-located collaborative work”, bei der mehrere Personen in einer Sitzung an einem Ort zusammenarbeiten.

Die Kommunikation zwischen den Teilnehmern ist ein zentrales Problem beim kollaborativen Arbeiten. Es ist schwierig, einen Konsens zwischen den Kommunikationspartnern zu erreichen [Clark und Brennan (1991)]. Die folgenden Abschnitte erläutern zunächst die Vorteile des kollaborativen Modellierens, die Auswirkungen der Arbeitsplatzkonfiguration sowie die Bedeutung der Multitouch-Eingabe für kollaboratives Arbeiten und schließlich, wie die Kollaboration mit Indikatoren für Awareness bewertet werden kann.

### 2.5.1. Kollaboratives Modellieren

Modelle können entweder von einem Analysten oder mehreren Personen erstellt werden. Wenn mehrere Personen ein übereinstimmendes Verständnis über ein Modell haben müssen, ist es sinnvoll, dass diese Personen das Modell gemeinsam modellieren [Renger u. a. (2008)].

Beim kollaborativen Modellieren gibt es verschiedene Vorgehensweisen je nachdem, ob alle Teilnehmer der Sitzung oder nur ausgewählte Teilnehmer die Änderungen an einem Modell vornehmen [Renger u. a. (2008)]. Bei chauffierten Sitzungen hat ein einzelner Teilnehmer (der Chauffeur) die Aufgabe, ein Modell so zu erstellen, wie es ihm die anderen Teilnehmer auftragen. Dieser Teilnehmer kann aber zu einem Flaschenhals werden, der die Effizienz bei der Modellierung einschränkt [Rittgen (2009)]. Im anderen Extrem dürfen alle Teilnehmer Änderungen am Modell direkt vornehmen. Alle Teilnehmer sind dadurch stärker in die Modellierung eingebunden. Durch die direkte Teilnahme an der Modellierung steigt das Vertrauen der Teilnehmer in das Modell. Letztendlich akzeptieren sie das finale Modell eher, wenn sie direkt darauf Einfluss nehmen können, als wenn sie indirekt an der Modellierung teilhaben [Renger u. a. (2008)].

Das Vorgehen bei der Modellierung hat Einfluss auf die Modell-Qualität. Kritische Teilnehmer an einer Sitzung können Domänenexperten sein, sie müssen aber nicht gleichzeitig auch Experten in der Modellierungssprache sein. Eine starke Einbeziehung aller Teilhaber fördert einerseits die semantische Korrektheit und Vollständigkeit eines Modells. Andererseits hemmt sie die syntaktische Korrektheit [Renger u. a. (2008)]. Infolgedessen fordern Renger u. a. (2008) den Einsatz von Methoden oder Mechanismen, welche die syntaktische Qualität verbessern, wenn ein Modell von mehreren Personen erstellt wird. Dies können zum Beispiel technologische Unterstützung, ein Training der Teilnehmer und regelmäßige Reviews sein [Renger u. a. (2008)].

### 2.5.2. Auswirkungen der Arbeitsplatz-Konfiguration auf das kollaborative Arbeiten

Die Verwendung von Computern kann beim kollaborativen Arbeiten zum Zerfall einer Arbeitsgruppe in Untergruppen führen [Orre und Middup (2006)]. Orre und Middup (2006) haben in ihrer Studie beobachtet, dass eine Person ihre Kollegen nur gelegentlich in die Arbeit einbezieht, wenn sie die alleinige Kontrolle über einen Computer hat. Die mangelnde Einbeziehung aller Teammitglieder führte zu einer Aufteilung des Teams in zwei Gruppen, die von den Tätigkeiten der jeweils anderen Gruppe nichts mitbekamen [Orre und Middup (2006)]. Orre und Middup (2006) schlussfolgern, dass bei der Einführung von Computerunterstützung die bestehenden Kollaborationsbereiche (Spheres of Collaboration) nicht geändert werden dürfen. Ansonsten ginge die Gruppeninteraktion verloren [Orre und Middup (2006)].

### 2.5.3. Die Bedeutung von Multitouch für kollaboratives Arbeiten

Mehrere Studien belegen, dass die Multitouch-Eingabe positive Auswirkungen auf das kollaborative Arbeiten hat. Die direkte Eingabe bei Multitouch fördert die **Awareness** der Benutzer beim kollaborativen Arbeiten [Hornecker u. a. (2008)]. Die Awareness wiederum fördert das Verständnis der Kooperationspartner und die Koordination ihrer Aktivitäten [Dourish und Bellotti (1992)].

Awareness bedeutet im Kontext des Computer Supported Cooperative Work (CSCW) das Gewahrsein einer Person über die Aktivitäten ihrer Kooperationspartner. Das Verstehen der Aktivitäten erschafft einen Kontext, der einer Person hilft, eigene Beiträge beizusteuern, welche dem Ziel der Gruppe dienen [Dourish und Bellotti (1992)].

Wegen der direkten Eingabe kann ein Benutzer bei Multitouch-Eingabe die Aktivitäten der anderen Benutzer sehen und nachvollziehen. Die Mauseingabe hingegen hemmt Awareness. Die Benutzer müssen anhand der Mauscursor nachvollziehen, wer welche Aktionen macht. Die kognitive Belastung ist deshalb bei der Mauseingabe höher als bei der Multitouch-Eingabe [Hornecker u. a. (2008)].

### 2.5.4. Indikatoren für Awareness

Hornecker u. a. (2008) haben einige Indikatoren für das Vorhandensein beziehungsweise Nichtvorhandensein von Awareness ermittelt.

### 2.5.4.1. Negative Awareness-Indikatoren

Die negativen Awareness-Indikatoren deuten auf ein geringes Level von Awareness hin:

- Aktionen einer Person, welche andere Personen unbeabsichtigt bei deren Tätigkeiten stören (Interference, z. B. Greifen zum selben Objekt).
- Fragen einer Person, was eine andere Person gerade tut oder warum sie es tut (Verbal Monitoring)

### 2.5.4.2. Positive Awareness-Indikatoren

Wenn keine Handlungen zur Koordination der Arbeit zu beobachten sind, werten [Hornecker u. a. \(2008\)](#) dies als Zeichen für Awareness:

- Reaktion ohne explizite Aufforderung.
- Arbeitsteilung durch parallele Aktionen ohne verbale Koordination.
- Arbeitsteilung durch komplementäre Aktionen ohne verbale Koordination.
- Übergabe von Objekten ohne verbale Koordination.

### 2.5.4.3. Indikatoren für Awareness Work

Beim kollaborativen Arbeiten strukturiert eine Person ihre Aktivitäten oftmals bewusst, damit die anderen Personen ihre Aktivitäten nachvollziehen können [[Hornecker u. a. \(2008\)](#), [Schmidt \(2002\)](#)]. Dieses bewusste Verhalten bezeichnen [Hornecker u. a. \(2008\)](#) als "Awareness Work".

- Verbales Beschreiben der eigenen Aktionen (Verbal Shadowing).
- Übertriebene manuelle Aktionen einer Person, um die Aufmerksamkeit auf etwas zu lenken.
- Sichtbare Haltungsänderungen einer Person, um eine bestimmte Absicht oder einen Zustand zu signalisieren.

[Hornecker u. a. \(2008\)](#) haben in ihrer Studie mit diesen Indikatoren die Multitouch- und Mauseingabe verglichen. Ihr Ergebnis bescheinigt der Multitouch-Eingabe eine bessere Eignung für kollaboratives Arbeiten. Zwar haben [Hornecker u. a. \(2008\)](#) in ihrer Studie sowohl die positiven Awareness-Indikatoren als auch den negativen Awareness-Indikator Interference häufiger bei der Touch-Eingabe beobachtet, als bei der Mauseingabe. Qualitativ bewerten sie aber auch die beobachteten Konflikte (Interference) als positiv. Die häufigeren Konflikte

erklären sie damit, dass die Multitouch-Eingabe die Benutzer mehr zur Interaktion einlade als die Mauseingabe. Die Konflikte sind also ein Zeichen von stattfindender Kollaboration. Ebenso konnten [Hornecker u. a. \(2008\)](#) im Multitouch Szenario das Verbal Shadowing häufiger beobachten. Die Benutzer seien hier mehr bereit, Awareness Work zu leisten. Außerdem seien im Multitouch Szenario die Auswirkungen der Konflikte gering gewesen und seien die Konflikte von den Benutzern schnell gelöst worden [[Hornecker u. a. \(2008\)](#)].

## 2.6. Model-View-ViewModel Architekturmuster

Das Model-View-ViewModel Architekturmuster (MVVM) ist eine Weiterentwicklung des Model-View-Controller Architekturmusters (MVC). Beide Architekturmuster werden bei der Entwicklung von Anwendungen mit grafischen Benutzungsoberflächen verwendet und dienen einer losen Kopplung von Anwendungslogik (Model) und ihrer Präsentation (View).

Bei MVC steuert der Controller die View, so dass die in [Abbildung 2.10](#) dargestellten Abhängigkeiten bestehen. Der Controller ist sowohl von der View, als auch von dem Model abhängig. In der Praxis werden View und Controller häufig nicht getrennt implementiert [[Huber \(2010\)](#)].

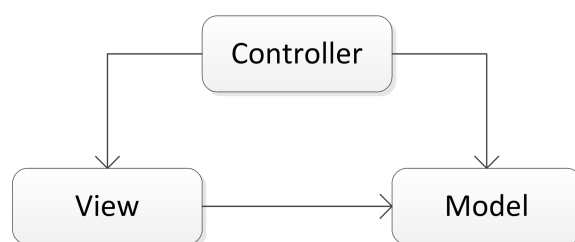


Abbildung 2.10.: Abhängigkeiten beim Model-View-Controller (MVC) Architekturmuster

Um die Präsentation stärker von der Steuerung der Benutzungsschnittstelle zu entkoppeln, wurde das MVVM Architekturmuster entwickelt [[Huber \(2010\)](#)]. Die Aufgaben des Controllers übernimmt hier das ViewModel. Dieses ist jedoch nicht von der View sondern nur vom Model abhängig. Außerdem ist die View vom ViewModel abhängig. Model, View und ViewModel bilden damit eine, wie in [Abbildung 2.11](#) dargestellte, strikte Schichtenarchitektur.

Das ViewModel kapselt das Model und bereitet die Daten des Models für die View auf. Mit darüber hinaus gehenden Informationen steuert das ViewModel den Zustand der View (z. B. welche Befehle die Benutzer ausführen können) [[Huber \(2010\)](#)]. Damit die View Änderungen am ViewModel mitbekommt, wird das MVVM Architekturmuster mit dem Beobachter-Entwurfsmuster (siehe [Gamma u. a. \(2004\)](#)) implementiert, wobei die View als Beobachter des ViewModels auftritt.

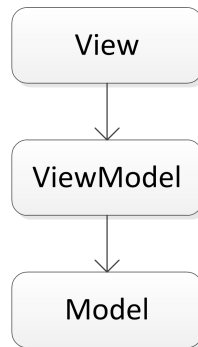


Abbildung 2.11.: Abhängigkeiten beim Model-View-ViewModel (MVVM) Architekturmuster

Das MVVM Architekturmuster wurde zwar für die Entwicklung von komplexen Anwendungen mit der Windows Presentation Foundation (WPF) entworfen, die grundlegenden Konzepte von MVVM sind aber unabhängig von den verwendeten Technologien.

## **3. Analyse**

In diesem Kapitel wird die Modellierungssprache ausgewählt, welche die in dieser Arbeit entwickelte Geschäftsprozessmodellierungsanwendung verwendet. Außerdem werden Anforderungen aufgestellt, welche die Anwendung erfüllen muss, um für die Geschäftsprozessmodellierung geeignet zu sein.

### **3.1. Auswahl der Modellierungssprache**

Im Folgenden wird eine Modellierungssprache ausgewählt, welche für die Modellierung von Geschäftsprozessen in kollaborativen Meetings am geeignetsten scheint. Dazu werden zunächst Anforderungen an die Modellierungssprache aufgestellt. Anschließend werden UML-Aktivitätsdiagramme, BPMN, EPK und Petrinetze als mögliche Alternativen genannt und eine dieser Sprachen ausgewählt.

#### **3.1.1. Anforderungen an die Modellierungssprache**

Die Anforderungen an die Modellierungssprache sind während der Modellierung eines Geschäftsprozesses und der anschließenden Weiterverwendung des fertigen Modells verschieden. Das oberste Ziel der hier gestellten Anforderungen ist, dass die Modelle zur Erfüllung der Kundenbedürfnisse beitragen. Die Anforderungen sollen eine hohe Qualität der Modelle fördern.

##### **3.1.1.1. Anforderungen an die Modellierungssprache als Diskussionsgrundlage**

Damit ein Modell als Diskussionsgrundlage dienen kann, müssen es alle beteiligten Personen verstehen. Dazu muss die Modellierungssprache die folgenden Anforderungen erfüllen:

- A1 Die Modelle müssen für alle beteiligten Personen lesbar sein, egal aus welcher Domäne die Personen stammen (z. B. Entwickler, Analyst, Sachbearbeiter). Eine Person, die das Geschäftsprozessmodell nicht lesen kann, kann weder aktiv zur Modellierung beitragen noch Fehler im Modell finden.
- A2 Die Modelle müssen von allen beteiligten Personen einheitlich verstanden werden. Aufgrund von Missverständnissen könnte ein fehlerhaftes Modell entstehen, auf dessen Basis ein System entwickelt wird, das nicht den Kundenbedürfnissen entspricht.
- A3 Die Modellierungssprache muss grafisch sein. Die Abläufe eines Geschäftsprozesses sind bei grafischen Modellen besser zu überblicken und leichter zu verstehen, als bei textuellen Modellen.
- A4 Die Modelle müssen ausführbar sein. Das Ausführen eines Geschäftsprozesses für einen exemplarischen Geschäftsfall kann den Geschäftsprozess veranschaulichen. Dies kann es den Teilnehmern einer Modellierungssitzung erleichtern, den Geschäftsprozess zu verstehen und semantische Fehler im Modell zu finden.

Die Modellierungssprache muss die Anforderungen A1, A2 und A3 einhalten, um eine zielführende Modellierungssitzung durchführen zu können. Die Erfüllung von Anforderung A4 unterstützt das Verstehen von Modellen zusätzlich. Hier wird A4 als Muss eingestuft, weil das Durchspielen von Modellen eine zentrale Funktion der Modellierungsanwendung sein soll.

#### 3.1.1.2. Anforderungen aus Entwicklersicht

Im Anschluss an eine Sitzung können verschiedene Personen die erstellten Modelle weiterbenutzen. Beispielsweise können Softwareentwickler anhand der Geschäftsprozessmodelle ein Softwaresystem implementieren, das den Geschäftsprozess unterstützt. Damit die Entwickler des Systems das Geschäftsprozessmodell bei der Implementierung verwenden können, muss das Modell die folgenden Anforderungen erfüllen:

- A5 Das Modell muss formal sein. Dies soll den Interpretationsspielraum für den Betrachter minimieren. Informelle Modelle, wie zum Beispiel die textuelle Beschreibung eines Geschäftsprozesses, können zur Fehlinterpretation durch den Betrachter führen. Ein formales Modell ist außerdem eine Voraussetzung für die automatisierte Verarbeitung (z. B. Codegenerierung) und die Ausführung von Modellen (z. B. Simulation).
- A6 Die Semantik des Modells muss eindeutig sein. Auch formale Modelle können fehlinterpretiert werden, wenn die Semantik eines Modells nicht eindeutig ist. Dies ist der Fall, wenn nicht alle Elemente der Modellierungssprache eindeutig definiert sind.



A7 Die Akteure müssen erkennbar sein. Bei der Implementierung eines die Geschäftsprozesse eines Unternehmens unterstützenden Systems müssen die Entwickler wissen, welche Aktivitäten eines Geschäftsprozesses vom System automatisiert werden und welche Aktionen von einer Person ausgeführt werden. Wenn es die Modellierungssprache erlaubt, Akteure zu modellieren und das System als Akteur modelliert wird, können die Entwickler die zu implementierenden Aktivitäten erkennen.

### 3.1.2. Auswahl der Modellierungssprache für Geschäftsprozesse

Für die Geschäftsprozessmodellierung kommen UML-Aktivitätsdiagramme, Business Process Model and Notation (BPMN), die Ereignisgesteuerte Prozesskette (EPK) und Petri-Netze infrage. Denn dies sind grafische Modellierungssprachen, die über geeignete Konstrukte zur Modellierung der Abläufe von Geschäftsprozessen verfügen. Außerdem sind diese Sprachen in der Praxis (UML-Aktivitätsdiagramme, BPMN und EPK) oder im akademischen Bereich (Petri-Netze) etabliert [Ko u. a. (2009), Becker u. a. (2010)]. Als Modellierungssprache hat der Autor die BPMN ausgewählt, da sie die Anforderungen am besten erfüllt (siehe Tabelle 3.1). Im Gegensatz zur BPMN verletzen Aktivitätsdiagramme, EPK und Petri-Netze einige der wichtigsten Anforderungen.

	Aktivitätsdiagramm	BPMN	EPK	Petri Netz
<b>Lesbarkeit und Verständlichkeit für Nicht-Informatiker (A1, A2)</b>	gering	mittel	mittel	gering
<b>Grafisch (A3)</b>	✓	✓	✓	✓
<b>Ausführbar (A4)</b>	✓	✓	✗	✓
<b>Formal (A5)</b>	(✓)	✓	✗	✓
<b>Klare Semantik (A6)</b>	✗	✓	✓	✓
<b>Akteure modellierbar (A7)</b>	✓	✓	✗	✗

Tabelle 3.1.: Übersicht über die Erfüllung der Anforderungen

Petri Netze und Aktivitätsdiagramme wurden nicht speziell für die Geschäftsprozessmodellierung entworfen. Sie stammen ursprünglich aus der objektorientierten Programmierung (Aktivitätsdiagramme) beziehungsweise der Modellierung nebenläufiger Prozesse in verteilten Systemen (Petri-Netze). BPMN und EPK sind Modellierungssprachen speziell für Geschäftsprozesse und können eher von Personen ohne spezielle fachliche Kenntnisse verstanden werden. Petri Netze und Aktivitätsdiagramme sind für Laien nicht ausreichend verständlich und sind damit nicht als Diskussionsgrundlage zwischen Kunden und Entwicklern geeignet.

Die EPK und Aktivitätsdiagramme sind nicht formal beziehungsweise nicht vollständig formal und haben keine eindeutige Semantik. Akteure können bei der EPK und Petri-Netzen nicht modelliert werden.

Mit der BPMN können Geschäftsprozesse formal und mit einer eindeutigen Semantik modelliert werden. Allerdings überfordert der große Sprachumfang der BPMN unerfahrene Benutzer. Modelle in der BPMN können deshalb zu komplex sein, um von jedem Betrachter im Detail verstanden zu werden. Immerhin spielt dieser Nachteil keine Rolle, wenn nicht der gesamte Sprachumfang genutzt wird. Eine überschaubare Untermenge der Sprachelemente reicht aus, um die wichtigsten Eigenschaften von Geschäftsprozessen zu beschreiben [Fernández u. a. (2010), Silver (2009)]. Insgesamt erfüllt die BPMN die Anforderungen besser als Aktivitätsdiagramme, EPK und Petri-Netze.

### 3.1.2.1. Einschränkung der Modellelement Palette

An einer Geschäftsprozess-Modellierungssitzung nehmen neben Modellierungsexperten häufig auch Fachexperten teil, die keine oder nur wenig Kenntnisse über die verwendete Modellierungssprache haben. Diese Fachexperten sind mit den zu modellierenden Geschäftsprozessen vertraut und beschreiben sie den Modellierungsexperten. Anhand der Schilderungen der Fachexperten modellieren die Modellierungsexperten die Geschäftsprozesse. Die Modellierungsexperten beherrschen die Modellierungssprache und gehören zu den Personen, die an der Entwicklung eines Softwaresystem zur Unterstützung der Geschäftsprozesse beteiligt sind. Noch während der Modellierungssitzung geben die Fachexperten den Modellierungsexperten Rückmeldungen, ob die Modelle die Geschäftsprozesse fachlich korrekt abbilden. Eine Voraussetzung dafür ist, dass die Fachexperten die Modelle verstehen. Je umfangreicher die Modellierungssprache ist, desto schwieriger ist es für in der Modellierungssprache unerfahrene Fachexperten, die Modelle zu verstehen.

Um allen Teilnehmern an einer Modellierungssitzung zu ermöglichen, die Geschäftsprozessmodelle zu verstehen, hat der Autor die Menge der während der Modellierung verfügbaren BPMN-Sprachelemente eingeschränkt. Die gewählte Untermenge der BPMN entspricht der von Silver (2009) beschriebenen "Level 1 Modelling Palette" (siehe Kapitel 2.2.1). Die Level 1 Modelling Palette reicht aus, um die wesentlichen Abläufe eines Geschäftsprozesses zu beschreiben [Silver (2009)]. Damit erfüllt die Untermenge die Voraussetzung, um mit einem Geschäftsprozessmodell zwischen den Fachexperten und den Modellierungsexperten ein übereinstimmendes grundsätzliches Verständnis über den Geschäftsprozess herzustellen. Dies ist der primäre Zweck der kollaborativen Geschäftsprozessmodellierungsanwendung.

Es ist nicht ausgeschlossen, dass die Benutzer einen Geschäftsprozess detaillierter modellieren möchten, um ihn zu analysieren oder mit einem Workflow Management System auszuführen. Für diesen Fall muss die Geschäftsprozessmodellierungsanwendung den Benutzern

die Möglichkeit geben, die Modellelement-Palette (Level 1 bis 3) auswählen zu können. Aber auch dann sollten die Benutzer die Modellierung mit der Level 1 Modelling Palette beginnen, um zunächst einen Konsens über den groben Ablauf herzustellen. Die Modellierung von Geschäftsprozessen zu Analyse-Zwecken und zur Ausführung von Geschäftsprozessmodellen geht über das Ziel dieser Arbeit hinaus. Deshalb wird hier darauf verzichtet, die BPMN mit den Modellelement-Paletten der Level 1 bis 3 vollständig zu implementieren.

## 3.2. Anforderungen

Eine Geschäftsprozessmodellierungsanwendung muss einige Anforderungen erfüllen, um für die Geschäftsprozessmodellierung mit Multitouch-Geräten geeignet zu sein. Einige Anforderungen resultieren aus den in Kapitel 2 genannten Forschungsergebnissen zu Multitouch-Geräten und kollaborativem Arbeiten. Weitere Anforderungen basieren auf Erkenntnissen, die der Autor im Rahmen eines Projekts im Master-Studiengang Informatik an der Hochschule für Angewandte Wissenschaften Hamburg gewonnen hat. In jenem Projekt hat der Autor an der Entwicklung und Evaluierung einer Anwendung zum Modellieren von UML-Klassendiagrammen mit dem Microsoft Surface mitgearbeitet. Ein Ergebnis ist, dass Klassendiagramme im Verhältnis zur Bildschirmfläche des Microsoft Surface häufig sehr groß sind. Deshalb muss die Bildschirmfläche effizient genutzt werden, damit die Benutzer möglichst viele Details eines Modells überblicken können. Ein weiteres Ergebnis ist, dass beim Microsoft Surface die Texteingabe mit einer Bildschirmtastatur ungeeignet ist. Eine bessere Alternative gibt es zurzeit jedoch nicht.

Folgende Anforderung muss die Geschäftsprozessmodellierungsanwendung erfüllen:

1. Alle Teilnehmer an einer Modellierungssitzung müssen die Modellierung jederzeit verfolgen können. Ebenso müssen diese Personen nachvollziehen können, wie ein Modell entwickelt wird. Wenn nicht alle Personen die Modellierung jederzeit verfolgen können, können die Personen leichter ein unterschiedliches Verständnis über den Geschäftsprozess bekommen, und es können leichter semantische Fehler unbemerkt bleiben. Personen könnten außerdem gedanklich abgehängt werden, so dass sie die Modelle nicht mehr verstehen und nichts mehr zur Modellierung beitragen können.
2. Der natürliche Kollaborationsbereich muss erhalten bleiben. Andernfalls droht die Gruppeninteraktion verloren zu gehen (siehe 2.5.2, [Orre und Middup (2006)]).
3. Alle Teilnehmer an einer Modellierungssitzung müssen jederzeit ihre Gedanken zu der Diskussion beitragen können. Ansonsten könnten wichtige Details nicht angesprochen und vergessen werden. Die Diskussion ist außerdem zum Verstehen des Geschäftsprozesses wichtig.

4. Alle Teilnehmer an einer Modellierungssitzung müssen aktiv an der Modellierung teilnehmen können.
5. Die Geschäftsprozessmodelle müssen übersichtlich und gut lesbar dargestellt werden.
6. Die Texteingabe muss schnell und einfach möglich sein.
7. Die Benutzungsoberfläche muss möglichst viel Platz zum Anzeigen von Modellen nutzen.
8. Die Erkennung von Multitouch-Gesten muss nach dem Prinzip der kontinuierlichen Gestenerkennung (siehe [2.3.2](#)) ablaufen, um den Benutzern eine sofortige Rückmeldung geben zu können.

## 4. Entwurf und Implementierung

Dieses Kapitel erläutert den Entwurf der Geschäftsprozessmodellierungsanwendung. Dabei steht das Konzept der Benutzungsschnittstelle im Mittelpunkt, da die Benutzungsschnittstelle entscheidend für die Eignung von Multitouch-Geräten beziehungsweise Multitouch-Tischen für die Geschäftsprozessmodellierung ist. Eine weitere zentrale Rolle spielt die Simulation von Geschäftsprozessmodellen, welche den Benutzern hilft, die Semantik von Geschäftsprozessmodellen zu verstehen. Zuerst erläutert Abschnitt 4.1 die beiden Modi der Benutzungsschnittstelle, mit denen die Benutzer Geschäftsprozesse modellieren und simulieren können. Danach beschreibt Abschnitt 4.2 zunächst die fachliche Architektur, mit der die Anwendung entworfen wurde, und anschließend die technische Architektur, mit der die Anwendung implementiert wurde. Zum Schluss erläutert Abschnitt 4.3 den Entwurf der Benutzungsoberfläche, wobei für vertikale Multitouch-Bildschirme und Multitouch-Tische zwei spezialisierte Benutzungsoberflächen vorgestellt werden.

### 4.1. Modi der Benutzungsoberfläche

Während einer Modellierungssitzung sind die beiden wesentlichen Tätigkeiten der Teilnehmer das Erstellen von Modellen und das anschließende Besprechen der Modelle. Letztere Tätigkeit dient zur Überprüfung, ob das erstellte Modell eine korrekte Abbildung der Realität ist. Um diesen beiden Tätigkeiten gerecht zu werden, hat die Modellierungsanwendung spezielle Funktionen sowohl zum Modellieren als auch zum visuellen Simulieren von Geschäftsprozessabläufen. Die Simulationsfunktionen benötigen die Benutzer während des Bearbeitens eines Modells nicht. Umgekehrt benötigen sie die Bearbeitungsfunktionen nicht beim Simulieren. Deshalb bietet die Modellierungsanwendung zwei Modi: einen Modellierungsmodus und einen Simulationsmodus. Die Benutzungsoberfläche bietet den Benutzern je nach Modus unterschiedliche Funktionen an und stellt die Modelle unterschiedlich dar. Zwischen den beiden Modi können die Benutzer jederzeit wechseln. Zum Beispiel können sie zurück in den Modellierungsmodus wechseln, wenn sie beim Simulieren einen Fehler gefunden haben.

### 4.1.1. Modellierungsmodus

Im Modellierungsmodus können die Benutzer ein Geschäftsprozessmodell von Grund auf erstellen oder ein vorhandenes Modell bearbeiten. In diesem Modus können die Benutzer einem Modell Elemente hinzufügen sowie vorhandene Elemente beschriften, löschen, verschieben, vergrößern und verkleinern. Ebenso können die Benutzer dem Modell Kommentare hinzufügen und diese bearbeiten. Außerdem können die Benutzer die Ansicht des Modells anpassen. Sie können durch das Modell scrollen und den Vergrößerungsfaktor ändern.

#### 4.1.1.1. Modellelement-Palette

Mit der Modellelement-Palette können die Benutzer einem Modell Elemente hinzufügen und einen Layout-Algorithmus ausführen lassen.

Das Werkzeug besteht im wesentlichen aus einer Liste mit Vorlagen für neue Knoten (Activities, Events und Gateways) und einer Liste mit Vorlagen für Kanten (Sequence Flow und Message Flow). Um dem Modell ein Element hinzuzufügen, müssen die Benutzer per Drag and Drop eine Vorlage in das Modell ziehen. Die Kanten können die Benutzer mit Knoten verbinden, indem sie die Enden der Kanten auf die Knoten ziehen.

Außerdem hat das Werkzeug im oberen Bereich eine Leiste für Funktionen, welche die Benutzer bei der Modellierung häufig benötigen. Beim aktuellen Prototyp enthält diese Leiste einen Button, mit dem die Benutzer einen Layout-Algorithmus ausführen können, um die Modellelemente automatisch übersichtlich anzuordnen. Der verwendete Layout-Algorithmus basiert auf dem in [Kitzmann u. a. (2009)] beschriebenen Algorithmus.

#### 4.1.1.2. Papierkorb

Im Papierkorb werden die Modellelemente abgelegt, welche die Benutzer gelöscht haben. Der Entwurf des Papierkorbs orientiert sich an einem realen Papierkorb oder einer Dokumentenablage, damit die Benutzer ihn intuitiv bedienen können. Die Benutzer können ein Element löschen, indem sie es per Drag and Drop in den Papierkorb verschieben. Wenn die Benutzer einen Knoten gelöscht haben, werden automatisch auch die mit dem Knoten verbundenen Kanten gelöscht. Um ein gelöscht Element wiederherzustellen, können die Benutzer es per Drag and Drop aus dem Papierkorb an die gewünschte Stelle im Modell verschieben.

### 4.1.2. Simulationsmodus

Im Simulationsmodus können die Benutzer einen Geschäftsprozess durchspielen, um die Korrektheit des Modells überprüfen. Den Ablauf eines Geschäftsprozesses visualisiert die Benutzungsschnittstelle durch farbiges Hervorheben von Modellelementen. Die Reihenfolge und den Zeitpunkt, an dem die modellierten Ereignisse eintreten und die modellierten Aktivitäten während der Simulation ausgeführt werden, bestimmen entweder die Benutzer oder die Anwendung. Die Benutzer können zwischen einer manuellen, einer automatisierten und einer teilweise automatisierten Simulation wählen. Änderungen können die Benutzer im Simulationsmodus nur an der Ansicht und an den Kommentaren vornehmen. Sie können aber nicht die Modellelemente manipulieren.

#### 4.1.2.1. Einhaltung von Semantik Regeln

Unabhängig davon, ob die Benutzer die Simulation automatisiert oder manuell durchführen, stellt die Anwendung jederzeit sicher, dass der Ablauf der Simulation zur Semantik des Modells passt. Die Semantikregeln spezifiziert der BPMN 2.0 Standard [OMG (2011)]. Die Anwendung implementiert diese Semantikregeln und führt sie während der Simulation aus, sobald sich der Simulationszustand ändert. Damit erlaubt die Anwendung den Benutzern, jederzeit nur gültige Aktionen auszuführen. Beispielsweise müssen die Benutzer die Simulation durch Auslösen eines Start-Events beginnen, und die Benutzer können Aktivitäten und Ereignisse nicht überspringen.

#### 4.1.2.2. Token Konzept

Während der Simulation wandern **Tokens** über die Sequence Flows von Knoten zu Knoten. Ein Element, an dem sich mindestens ein Token befindet, heißt belegt. Die Belegung der Modellelemente mit Tokens bestimmt den Simulationszustand. Dieses Token Konzept ist an Petri-Netze (siehe Kapitel 2.1.4.3) angelehnt. Im Unterschied zu Petri Netzen können hier jedoch nicht nur Knoten (genau genommen die sogenannten Stellen eines Petri-Netzes) sondern auch Kanten (hier Sequence Flows) mit Tokens belegt sein. Ein Sequence Flow entspricht in einem Petri Netz einer zusätzlichen Stelle, mit jeweils genau einer Transition im Vor- und Nachbereich. Die Belegung von Sequence Flows mit Tokens ist nötig, um die Simulation nachvollziehbar visualisieren zu können.

Auch der BPMN 2.0 Standard nennt Tokens als ein Konzept zum Beschreiben des Verhaltens eines Prozesses, welches allerdings bloß theoretischer Natur sei: "A token is a theoretical concept that is used as an aid to define the behavior of a Process that is being performed."

...However, modeling and execution tools that implement BPMN are NOT REQUIRED to implement any form of token. (OMG, 2011, S. 27)". Diese Modellierungsanwendung verwendet jedoch Tokens sowohl, um die Semantikregeln bei der Simulation einzuhalten, als auch, um die Simulation zu visualisieren.

Die Modellierungsanwendung visualisiert nicht die Tokens selbst, sondern die Abfolge der Aktivitäten während eines bestimmten Geschäftsfalles. Die Benutzer müssen somit das zu Grunde liegende Token-Konzept nicht kennen, um einen simulierten Geschäftsprozess-Ablauf zu verstehen.

#### 4.1.2.3. Visualisierung von Geschäftsprozess-Abläufen

Während einer Simulation visualisiert die Benutzungsschnittstelle den Ablauf eines Geschäftsprozesses durch farbiges Hervorheben von Flow Nodes und Sequence Flows. Die Farben der Modellelemente signalisieren jeweils deren Zustand. Animierte Farbveränderungen signalisieren Zustandsänderungen. Die Farben für die verschiedenen Zustände sind die Ampelfarben Rot, Gelb/Orange und Grün sowie Schwarz, welches die standardmäßige Vordergrundfarbe der Modellelemente ist.

Die Visualisierung eines Geschäftsfalles baut auf der Belegung der Modellelemente mit Tokens auf. Es gibt zwei Visualisierungszustände, mit denen ein mit einem Token belegtes Modellelemente visualisiert wird, und einen Visualisierungszustand für nicht belegte Modellelemente (siehe auch Abbildung 4.1.2.3):

- **Deaktiviert** Deaktiviert sind alle Elemente, die kein Token besitzen. Entweder haben diese noch kein Token erhalten oder alle Tokens verbraucht beziehungsweise weitergeleitet. Deaktivierte Knoten haben einen schwarzen Rahmen und deaktivierte Sequence Flows haben eine schwarze Linie. Bei allen Elementen außer Start-Events ist dies der initiale Zustand.
- **Aktivierbar** Aktivierbar sind die Elemente, die im Verlauf der Simulation ein oder mehrere Token erhalten haben und noch nicht auf den Erhalt der Tokens reagiert haben. Aktivierbare Knoten haben einen grünen Rahmen, und aktivierbare Sequence Flows haben eine grüne Linie. Dies ist der initiale Zustand von Start-Events.
- **Aktiv** Ein aktivierbares Element wird aktiv, wenn es beginnt, auf ein erhaltenes Token zu reagieren und es zu verbrauchen beziehungsweise weiterzuleiten. Vom Typ eines Elements hängt ab, was aktiv genau bedeutet. Aktive Aktivitäten werden gerade ausgeführt, aktive Ereignisse treten gerade ein, aktive Sequence Flows leiten gerade ein Token an den nachfolgenden Knoten weiter und aktive Gateways leiten gerade ein Token an ein oder mehrere ausgehende Sequence Flows weiter (exclusive Decision bzw. parallel Split). Im Zustand "Aktiv" haben Knoten wie auch im Zustand "Aktivierbar"



einen grünen Rahmen und zusätzlich einen Orange-blinkenden Hintergrund. Aktive Sequence Flows haben eine blinkende Linie, deren Farbe zwischen Grün und Orange abwechselt.

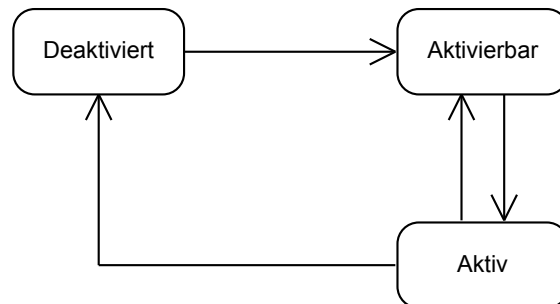


Abbildung 4.1.: Die möglichen Visualisierungszustände von Flow Nodes und Sequence Flows

Die Zustände "Aktivierbar" und "Aktiv" werden unterschieden, damit die Benutzer die Abläufe Schrittweise nachvollziehen können. Folgende Visualisierungszustände können während einer Simulation auftreten:

- **Deaktiviert** → **Aktivierbar** Dieser Zustandsübergang wird durch überblenden der Farben von Flow Nodes und Sequence Flows dargestellt (Schwarz nach Grün).
- **Aktivierbar** → **Aktiv** Bei diesem Zustandsübergang fangen die Elemente unmittelbar an zu blinken.
- **Aktiv** → **Deaktiviert** Dieser Zustandsübergang wird durch ein einmaliges rotes Aufblinker des deaktivierten Elements dargestellt. Die Linien- beziehungsweise Rahmenfarbe eines deaktivierten Elements wechselt zunächst in kurzer Zeit von Grün zu Rot und anschließend langsam zu Schwarz.
- **Aktiv** → **Aktivierbar** Dieser Zustandsübergang tritt dann ein, wenn ein Modellelement mit mehr als einem Token belegt war, so dass es nach der Aktivierung noch mit mindestens einem Token belegt ist. Bei diesem Zustandsübergang wird das aktivierbare Modellelement unmittelbar nach Eintritt als aktivierbar dargestellt (grüner Rahmen bzw. grüne Linie).

Die Benutzer können auch Simulationsschritte rückgängig machen, um zu einer vorherigen Tokenbelegung zurück zu gelangen. Nicht belegte Modellelemente bekommen dann den Zustand "Deaktiviert" und belegte Modellelemente bekommen dann den Zustand "Aktivierbar".

#### 4.1.2.4. Automatisierung der Simulation mit einem Simulationsagenten

Die Benutzer können Geschäftsprozesse entweder manuell simulieren oder die Simulation von einem Simulationsagenten automatisiert ablaufen lassen. Ebenso ist eine teilweise automatisierte Simulation möglich.

Der Simulationsagent ist eine Softwarekomponente, welche die Visualisierungszustände der Modellelemente überwacht. Die aktivierbaren Elemente kann der Simulationsagent aktivieren, ohne dass ein Benutzer dies veranlasst. Der Simulationsagent kann prinzipiell sowohl deterministische als auch nichtdeterministische Abläufe automatisiert simulieren. Eine automatisierte Simulation hat nur dann ein deterministisches Ergebnis, wenn das simulierte Geschäftsprozessmodell nur deterministische Abläufe beschreibt. Die Strategie, mit welcher der Simulationsagent mit nichtdeterministische Abläufen umgeht, hängt davon ab, ob die Benutzer ein deterministisches Ergebnis fordern. Wenn das Ergebnis nicht deterministisch sein muss, kann der Simulationsagent über nichtdeterministischen Abläufe entweder zufallsbasiert oder nach dem FIFO-Prinzip (First In - First Out) entscheiden. Andernfalls muss der Simulationsagent in den Situationen einer Simulation, bei denen nichtdeterministische Abläufe entstehen, die automatisierte Simulation unterbrechen. Die Benutzer müssen dann den Ablauf bestimmen. Anschließend kann der Simulationsagent die Simulation automatisiert weiter führen.

BPMN 2.0 Geschäftsprozessmodelle beschreiben weitestgehend deterministische Abläufe. Alle Modellelemente des BPMN 2.0 Standards haben eine eindeutige Semantik und leiten die Token deterministisch weiter. Einschränkungen gibt es, wenn der Faktor Zeit einen Einfluss auf die kausalen Zusammenhänge hat. BPMN 2.0 legt den Zeitpunkt, an dem ein Element ein Token während einer Simulation weiterleitet, nicht fest und definiert auch nicht, wie lange es dauert, ein Token weiterzuleiten. Ebenso kann die Ausführungsdauer einer Aktivität mit BPMN 2.0 nicht definiert werden. Solange es keine Wettlaufsituation gibt, hat dies jedoch keinen Einfluss auf den Pfad, auf dem die Tokens ein Modell traversieren, und eine automatisierte Simulation ist uneingeschränkt möglich. Bei Wettlaufsituationen (Race Conditions) hängt der simulierte Ablauf davon ab, welches Token ein Element zuerst erreicht. Wettlaufsituationen sind bei BPMN 2.0 unter anderem bei Event Gateways und Non-interrupting Events möglich sowie bei Activities und Events, die mehrere eingehende Sequence Flows haben.

Bei Exclusive Gateways hängt der anschließende Ablauf von einer Entscheidung ab. Die Entscheidung wird deterministisch durch Auswerten der mit den ausgehenden Sequence Flows verknüpften Bedingungen getroffen. Der Simulationsagent kann diese aber nur dann treffen, wenn er alle nötigen Informationen zur Auswertung der Bedingungen hat. Der aktuelle Entwurf hat keinen Mechanismus, über den der Simulationsagent diese Informationen bekommen könnte. Deshalb müssen die Benutzer während der Simulation festlegen, welche Bedingung erfüllt ist. Ohnehin ist bei BPMN 2.0 eine automatisierte Auswertung prinzipiell

nur eingeschränkt möglich. Welche Bedingung erfüllt ist, kann sowohl von den Eingaben abhängen, mit denen ein Prozess gestartet wird, als auch von den Ausgaben, die eine Aktivität erzeugt. Mit BPMN 2.0 kann aber nicht definiert werden, wie eine Aktivität ihre Ausgaben berechnet.

#### 4.1.2.5. Vollautomatisierte Simulation

Wenn die Benutzer für einen Geschäftsprozess bereits Abläufe definiert haben (z. B. Standardablauf oder alternative Abläufe eines Anwendungsfalles), können sie diese auswählen und von dem Simulationsagenten automatisiert ablaufen lassen. Sobald ein Modellelement den Visualisierungszustand "Aktivierbar" einnimmt, aktiviert der Simulationsagent das Element. Bei Exclusive Gateways entscheidet der Simulationsagent entsprechend der Ablauf-Definition, welche Bedingung erfüllt ist. Gleichmaßen orientiert sich der Simulationsagent bei Wettlaufsituationen an der Ablauf-Definition. Nebenläufige Abläufe führt der Simulationsagent zeitlich parallel aus, bis sie synchronisiert werden müssen.

#### 4.1.2.6. Manuelle Simulation

Bei der manuellen Simulation aktivieren die Benutzer Aktivitäten und Ereignisse. Zusätzlich wählen die Benutzer bei Exclusive Gateways aus, welche Bedingung erfüllt ist. Gateways und Sequence Flows können die Benutzer jedoch nicht aktivieren. Dies übernimmt aus ergonomischen Gründen der Simulationsagent, um den Benutzern unnötige Arbeit abzunehmen. Der Zeitpunkt, an dem ein Gateway oder ein Sequence Flow während einer Simulation aktiviert wird, ist auch weniger relevant als der Zeitpunkt, an dem eine Aktivität ausgeführt wird oder ein Ereignis eintritt. Prinzipiell wäre es aber möglich, dass auch die Benutzer Gateways und Sequence Flows aktivieren.

Die Benutzer können jeweils nur die aktivierbaren Modellelemente aktivieren, nicht aber bereits aktivierte oder deaktivierte. Gleichmaßen können sie nur dann Entscheidungen treffen, wenn ein Exclusive Gateway aktivierbar ist.

Durch Drücken mit einem Finger oder Klicken mit der Maus auf ein aktivierbares Element können die Benutzer es aktivieren. Gleichmaßen können die Benutzer bei einer Entscheidung auf einen Sequence Flow tippen oder klicken, um die gültige Bedingung auszuwählen. Wenn sich die Benutzer für eine Bedingung entscheiden müssen, werden die wählbaren Bedingungen farbig und mit einem blinkenden Fragezeichen-Symbol hervorgehoben.

#### 4.1.2.7. Halbautomatisierte Simulation

Bei der halbautomatisierten Simulation aktiviert der Simulationsagent die Modellelemente wie bei der vollautomatisierten Simulation. Allerdings wählt der Simulationsagent nicht aus, welche Bedingungen erfüllt sind. Dies übernehmen die Benutzer, indem sie auf den Sequence Flow mit der gültigen Bedingung tippen oder klicken. Der Simulationsagent wartet bei einem Exclusive Gateway solange, bis die Benutzer eine Bedingung ausgewählt haben. Nebenläufige Abläufe führt der Simulationsagent indessen weiter aus. Wettlaufsituationen werden bei einer halbautomatisierten Simulation durch das FIFO-Prinzip gelöst.

Die Benutzer können die Abläufe, welche sie manuell oder halbautomatisiert simuliert haben, als Standardablauf oder alternative Abläufe eines Anwendungsfalles abspeichern. Einzige Voraussetzung ist, dass die Simulation mit einem End-Event geendet hat.

#### 4.1.2.8. Zeitliche Verzögerung bei der Simulation

Damit die Benutzer die Abläufe während einer Simulation nachvollziehen können, laufen die Simulationen mit einer zeitlichen Verzögerung ab. Die Modellelemente verweilen mindestens für eine festgelegte Zeit im Visualisierungszustand "Aktiv", bevor sie in einen anderen Visualisierungszustand wechseln können. Die Benutzer können diese Zeit einstellen (zum Beispiel eine Sekunde).

#### 4.1.2.9. Simulation von Parallel Split und Parallel Join

Gemäß dem BPMN 2.0 Standard wird ein Token bei einem parallel Split in mehrere Tokens aufgespalten und bei einem Parallel Join werden mehrere Tokens zu einem Token verschmolzen. Parallel Splits finden bei Parallel Gateways und bei Activities mit mehreren ausgehenden Sequence Flows statt. Parallel Joins finden nur bei Parallel Gateways statt (siehe auch Kapitel 2.2.1.4). Damit Tokens bei einer Simulation aufgespalten und wieder verschmolzen werden können, haben die Tokens eine Hierarchie. Ein Token kennt jeweils ein übergeordnetes Token und seine untergeordneten Tokens (siehe Abbildung 4.2).

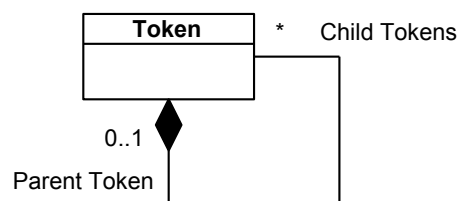


Abbildung 4.2.: Token-Hierarchie

Bei einem Parallel Split wird ein Token aufgeteilt, indem pro ausgehendem Sequence Flow genau ein neues Token erzeugt und an den Sequence Flow weitergeleitet wird (siehe Listing 4.1). Das aufgeteilte Token wird nicht selbst an die ausgehenden Sequence Flows weitergeleitet. Die neuen Tokens werden so in die Hierarchie eingefügt, dass ihr übergeordnetes Token das aufgeteilte Token ist. Als übergeordnetes Token der neuen Tokens wird das aufgeteilte Token gesetzt.

```
method ParallelSplit(token)
  if Count(OutgoingSequenceFlows) = 1
    send token to first(OutgoingSequenceFlows)
  else
    for each sequenceFlow in OutgoingSequenceFlows
      newToken := CreateChildToken(token)
      send newToken to sequenceFlow
    end for
  end if
end method
```

Listing 4.1: Parallel Split

Bei einem Parallel Join synchronisiert ein Parallel Gateway die Abläufe, indem es die zuvor bei einem Parallel Split aufgeteilten Tokens zum übergeordneten Token verschmilzt (siehe Listing 4.2). Das Parallel Gateway wartet solange, bis über jeden eingehenden Sequence Flow Tokens mit demselben übergeordneten Token eingetroffen sind. Anschließend werden die eingetroffenen Tokens aus der Hierarchie entfernt und das übergeordnete Token an den ausgehenden Sequence Flow geleitet.

#### 4.1.2.10. Simulation-Controller

Neben dem Tippen beziehungsweise Klicken auf die aktivierbaren Elemente, steuern die Benutzer eine Simulation mit dem Simulation-Controller. Mit diesem Werkzeug können die Benutzer eine Simulation zurücksetzen, einzelne Schritte rückgängig machen oder wiederherstellen sowie zwischen der manuellen und automatisierten Simulation wählen.

Wenn die Benutzer vom Modellierungsmodus in den Simulationsmodus wechseln, ist standardmäßig der manuelle Simulationsmodus aktiv. Die Simulation starten die Benutzer dann, indem sie auf die aktivierbaren Modellelemente drücken. Um die Simulation halbautomatisiert ablaufen zu lassen, können die Benutzer jederzeit auf den Run-Button des Simulationswerkzeugs drücken. Der Simulationsagent beginnt dann die Elemente zu aktivieren. Um den Simulationsagenten anzuhalten und die Simulation zu pausieren oder manuell fortzusetzen,

```
receivedTuples := {}

method ReceiveToken(token, sequenceFlow)
  receivedTuples := receivedTuples ∪ { <token, sequenceFlow> }
  tuples := {}
  for each tuple in receivedTuples
    if tuple.Token.Parent == receivedToken.Parent
      tuples := tuples ∪ { tuple }
    end if
  end for
  if Count(tuples) = Count(IncomingSequenceFlows)
    if Count(OutgoingSequenceFlows) = 1
      destination := first(OutgoingSequenceFlows)
      send token.Parent to destination
    else
      ParallelSplit(token.Parent)
      DestroyToken(token)
    end if
    receivedTuples := receivedTuples \ tuples
  end if
end method
```

Listing 4.2: Parallel Join



Abbildung 4.3.: Simulation-Controller

können die Benutzer auf den Stop-Button drücken. Auf diese Weise können die Benutzer während einer Simulation beliebig zwischen der manuellen und halbautomatisierten Simulation wechseln oder die halbautomatisierte Simulation pausieren und fortsetzen.

Um einen gespeicherten Ablauf zu laden und vollautomatisiert zu simulieren, können die Benutzer den gewünschten Ablauf per Drag and Drop aus der Liste der gespeicherten Abläufe in den Simulation-Controller ziehen. Anschließend wird die Simulation automatisch gestartet. Die Benutzer können auch eine vollautomatisierte Simulation unterbrechen und wieder fortsetzen, indem sie zunächst auf den Stop-Button und anschließend auf den Run-Button drücken. Sie können die Simulation jedoch nicht manuell fortsetzen, da der Ablauf bei der vollautomatisierten Simulation fest vorgegeben ist.

Die Benutzer können eine automatisierte Simulation pausieren, um beispielsweise den bis dahin simulierten Ablauf zu besprechen oder, um zu besprechen, welchen Ablauf sie anschließend erwarten.

Die Benutzer können eine Simulation auch zurück- und wieder vorlaufen lassen, unabhängig davon, ob sie die Simulation manuell durchführen oder automatisiert ablaufen lassen. Um die Aktivierung von Elementen oder die Auswahl von Bedingungen rückgängig zu machen, drücken die Benutzer auf den Undo-Button des Simulation-Controllers. Umgekehrt können die Benutzer auf den Redo-Button des Simulation-Controllers drücken, um einen zuvor rückgängig gemachten Ablauf erneut durchzuführen. Auf diese Weise können die Benutzer beliebig viele Simulationsschritte zurück springen. Anschließend können sie den gleichen Ablauf erneut ablaufen lassen oder einen alternativen Ablauf wählen. Wenn die Benutzer auf den Undo-Button drücken, während die Simulation automatisiert abläuft, wird die Simulation automatisch pausiert, so wie wenn die Benutzer auf den Stop-Button drücken. Sonst würde der Simulationsagent die rückgängig gemachte Aktion umgehend erneut ausführen. Der Redo-Button ist deaktiviert während der Simulationsagent eine Simulation automatisiert ausführt.

Um eine Simulation abzubrechen oder am Ende einer Simulation eine neue Simulation zu beginnen, können die Benutzer auf den Reset-Button des Simulation-Controllers drücken.

Wenn die Benutzer den Reset Button während des vollautomatisierten Simulationsmodus drücken, wird der gespeicherte Ablauf von vorn begonnen.

### 4.1.3. Aktivierung des Modellierungs- und Simulationsmodus

Bei einigen Benutzerinteraktionen kann die Anwendung erkennen, ob die Benutzer den Modellierungsmodus oder den Simulationsmodus benötigen, und bei anderen Benutzerinteraktionen wiederum nicht. Im zuerst genannten Fall aktiviert die Anwendung automatisch den benötigten Modus und im Letzteren müssen die Benutzer den Modus bei Bedarf manuell ändern. Tabelle 4.1 gibt einen Überblick darüber, wann die Anwendung den Modus automatisch einstellt.

Aktion	Auslösende Geste	Benötigter Modus	Automatischer Moduswechsel
Hinzufügen eines neuen Modellelements	Drag and Drop einer Element-Vorlage in das Modell	Modellierung	ja
Wiederherstellen eines gelöschten Modellelements	Drag and Drop eines Elements aus dem Papierkorb in das Modell	Modellierung	ja
Auswählen eines Elements, um es zu Beschriften	Doppelklick oder Double-Tap	Modellierung	nein
Verwendung des Simulationswerkzeugs: Befehle Run, Stop, Undo, Redo und Reset	Drücken eines Buttons	Simulation	ja
Aktivieren eines Elements bei der manuellen Simulation	Klick oder Tap	Simulation	nein
Auswählen einer erfüllten Bedingung während einer Simulation	Klick oder Tap	Simulation	nein

Tabelle 4.1.: Übersicht über die Situationen in den die Anwendung den Modus automatisch wechselt

Wenn die Benutzer den Modus manuell wechseln unterbrechen sie ihren Arbeitsfluss. Es ist aber nicht zu erwarten, dass die Benutzer den Modus regelmäßig und insbesondere nicht in



kurzen Abständen wechseln. Insofern ist diese Einschränkung vermutlich hinnehmbar. Die Usability-Tests haben diese Annahme bestätigt.

Wenn die Benutzer eine Aktion durchführen möchten, die mit Klicks oder Taps ausgelöst wird, müssen sie selbst den benötigten Modus einstellen. Der Grund dafür ist, dass Klick-beziehungsweise Tap-basierte Gesten (Klick / Doppelklick und Tap / Double-Tap) in beiden Modi mit Funktionen verknüpft sind und nicht eindeutig auseinander gehalten werden können.

Einzelne Klicks und Taps können nach einem Timeout eindeutig von doppelten Klicks und Taps unterschieden werden. Trotzdem wechselt die Anwendung den Modus nicht, wenn im Modellierungsmodus ein Benutzer ein Element antippt, obwohl die Tap-Geste im Modellierungsmodus mit keiner Funktion assoziiert ist. Im Modellierungsmodus soll ein einzelnes Tap keine Auswirkungen haben, da es durch versehentliches Berühren entstehen kann. Ein Benutzer könnte zum Beispiel auf ein Element zeigen und es dabei berühren, ohne damit eine Aktion auslösen zu wollen. Der Autor konnte dies bei ersten Usability-Tests regelmäßig beobachten. Beim Zeigen mit der Maus auf ein Element ist ein versehentliches Klicken unwahrscheinlich. Um die Bedienung der Anwendung mit Maus und Touch-Eingabe möglichst einheitlich zu gestalten, ist auch ein einzelner Klick auf ein Element im Modellierungsmodus nicht mit einem Befehl verbunden und löst keinen Wechsel des Modus aus.

Umgekehrt kann ein Doppelklick oder Double-Tap auch als zwei einzelne Klicks beziehungsweise Taps interpretiert werden. Die Anwendung soll bei solchen Fehlinterpretationen den Modus nicht wechseln und verzichtet deshalb komplett auf einen automatischen Modus-Wechsel bei einer Doppelklick-Geste und einer Double-Tap-Geste. Außerdem würde der zur Unterscheidung von einzelnen und doppelten Klick und Tap Ereignissen nötige Timeout eine verzögerte Ausführung der mit einzelnen Klicks und Taps verbundenen Befehle erfordern. Während einer Simulation müsste beispielsweise die Aktivierung von Elementen verzögert werden. Die Verzögerung müsste etwas länger sein als die maximale Zeit, die ein Benutzer für einen Double-Tap benötigt. Nach der Erfahrung des Autors benötigen die Benutzer für eine Double-Tap Geste etwa eine viertel bis eine halbe Sekunde. Die Benutzer sollen jedoch umgehend eine visuelle Rückmeldung bekommen, um nicht irritiert zu werden (siehe [3.2](#)).

## 4.2. Architektur

Dieser Abschnitt erläutert zunächst die *fachliche* Architektur der Geschäftsprozessmodellierungsanwendung. Diese beschreibt auf konzeptioneller Ebene die Komponenten, aus denen die Anwendung besteht, und wie die Anwendung die Simulation von Geschäftsprozessmodellen handhabt. Anschließend beschreibt die *technische* Architektur wie das fachliche Konzept technisch umgesetzt wird. Dabei sind die verschiedenen Software-Plattformen die bestimmenden Faktoren.

### 4.2.1. Fachliche Architektur

Die fachliche Architektur der Geschäftsprozessmodellierungsanwendung beschreibt die fundamentalen Komponenten, aus denen die Anwendung besteht, und welche Beziehungen zwischen diesen Komponenten bestehen. Das System wird unter fachlichen Gesichtspunkten in Komponenten unterteilt, so dass die einzelnen Komponenten eine hohe Kohäsion haben und die Komponenten miteinander lose gekoppelt sind. Abbildung 4.4 gibt einen Überblick über die Komponenten.

Die Komponenten der Anwendung sind in drei aufeinander aufbauenden Schichten aufgeteilt: Model, ViewModel und View. Diese Aufteilung orientiert sich am Model-View-ViewModel-Architekturmuster (MVVM, siehe Kapitel 2.6). Das MVVM Architekturmuster wird hier nicht nur verwendet, um die Präsentation von der Anwendungslogik zu entkoppeln. Die Komponenten der ViewModel-Schicht vereinfachen auch die Benutzung der Model-Komponenten im Sinne einer Fassade (Fassade siehe Gamma u. a. (2004)).

#### 4.2.1.1. BPMN 2.0 Metamodell

Die Komponente "BPMN 2.0 Metamodell" implementiert die [abstrakte Syntax](#) und die [statische Semantik](#) des BPMN Metamodells. Klienten der Komponente können mit den Schnittstellen der Komponente Instanzen von Geschäftsprozessmodellen erzeugen und bearbeiten.

Diese Komponente implementiert das Metamodell gemäß dem BPMN-Standard [OMG (2011)] in der Version 2.0. Sie implementiert aber nur die Teile des Metamodells, die für den Zweck der Anwendung nötig sind (siehe Kapitel 3.1.2.1). Das Klassendiagramm in Abbildung 4.5 zeigt die vereinfachte abstrakte Syntax mit den [Metaklassen](#), die von dieser Komponente implementiert werden. Die [abstrakte Syntax](#) legt die Elemente der Modellierungssprache fest und welche Beziehungen zwischen den Elementen bestehen. Zusätzlich implementiert

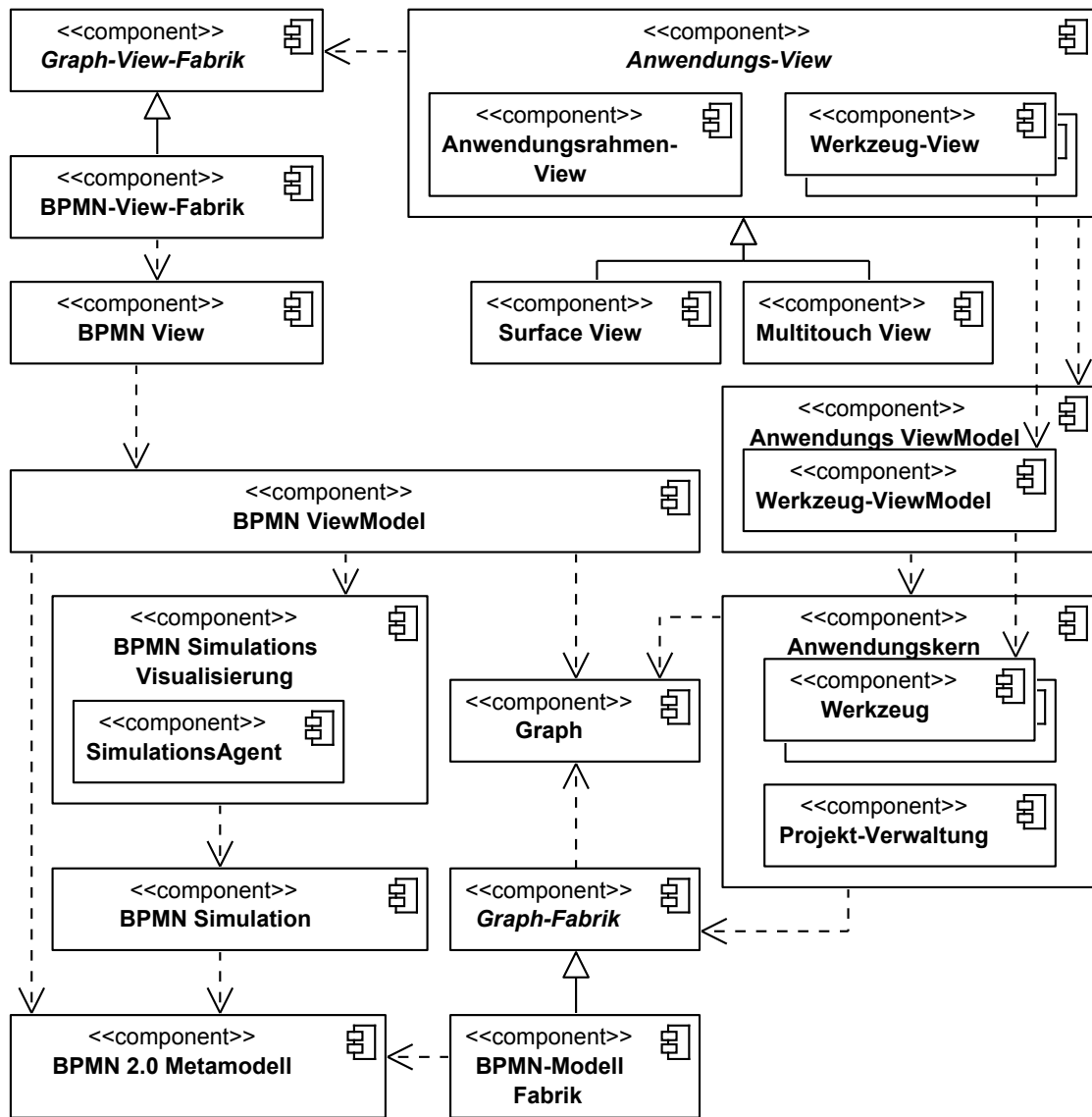


Abbildung 4.4.: Komponenten der fachlichen Architektur

die Metamodell-Komponente die Bedingungen der **statischen Semantik** von BPMN. Mit diesen Bedingungen stellt die Komponente sicher, dass die instantiierten Modelle wohlgeformt sind.

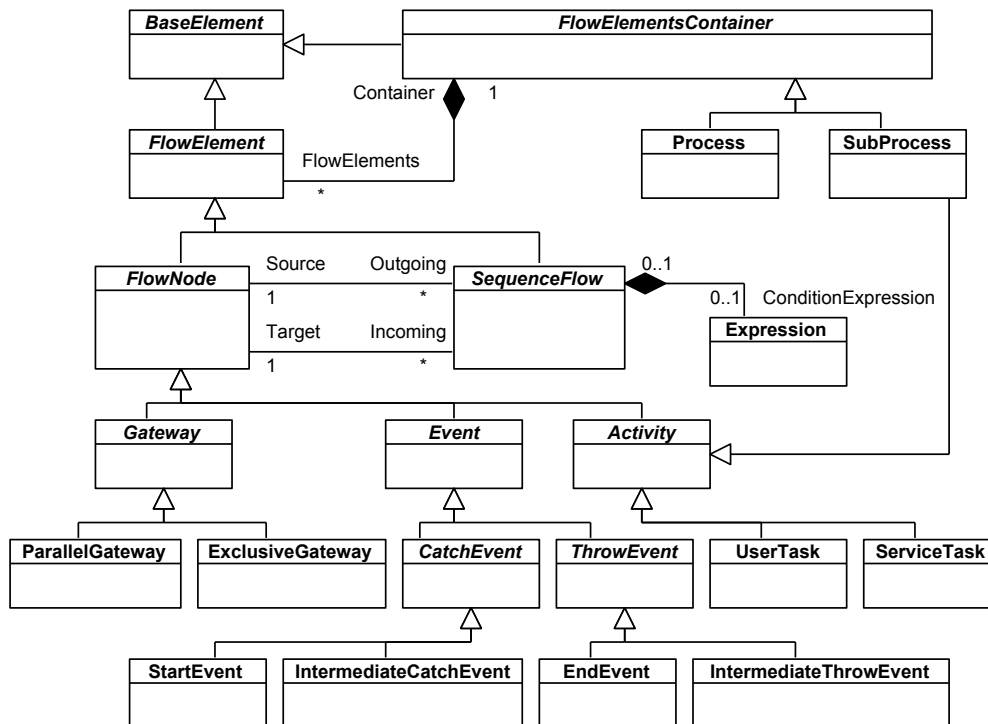


Abbildung 4.5.: Abstrakte Syntax des BPMN 2.0 Metamodells nach [OMG (2011)]

#### 4.2.1.2. BPMN Simulation

Auf dem Metamodell baut die Komponente “BPMN Simulation” auf. Diese Komponente implementiert die operationelle Semantik der BPMN Spezifikation und stellt Schnittstellen für die Simulation eines Geschäftsprozessmodells bereit. Klienten können über diese Schnittstellen den simulierten Ablauf eines Geschäftsprozesses abrufen und steuern.

Die BPMN-Simulation-Komponente instantiiert für jede durchgeführte Simulation ein Objekt der ProcessSimulation Klasse. Dieses Objekt dient als Container für FlowElementSimulation Objekte, welche die Elemente des Metamodells referenzieren und simulieren (siehe Abbildung 4.6). Der so entstehende Objekt-Graph ähnelt stark der Struktur des Metamodells (vergleiche Abbildung 4.6 und 4.5). Zwischen den Simulations-Klassen und den Metaklassen besteht eine Kompositions-Beziehung. Zum einen erlaubt dies, mehrere Simulations-Instanzen

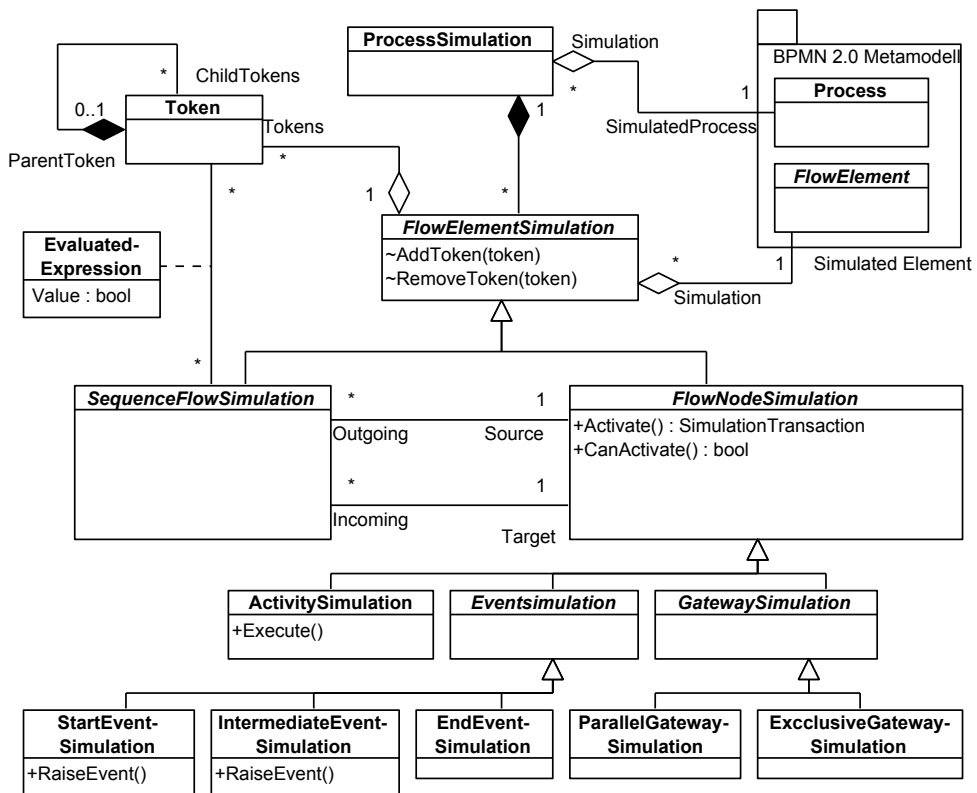


Abbildung 4.6.: Fachlicher Entwurf der simulierbaren Elemente

pro Geschäftsprozessmodell zu erzeugen. Zum anderen sind dadurch die Simulations-Klassen und die Klassen des Metamodells klar nach Aufgaben getrennt. Erstere implementieren die operationelle Semantik und Letztere die statische Semantik der BPMN.

Die Simulations-Komponente setzt die operationelle Semantik der BPMN mit Tokens um, die sie gemäß der Semantikregeln des BPMN Standards [OMG (2011)] erzeugt und zwischen den FlowElementSimulation-Objekten austauscht. Ein Token ist jederzeit genau einem FlowElementSimulation-Objekt zugeordnet. Die Klienten der Komponente aktivieren ein aktivierbares FlowNodeSimulation-Objekt, indem sie seine Activate-Methode aufrufen. Ob ein FlowNodeSimulation-Objekt aktivierbar ist, hängt von der Token-Belegung und den Semantikregeln ab. Die Activate-Methode führt eine Transaktion aus, während der Tokens zwischen Simulations-Objekten ausgetauscht werden. Die Transaktion gewährleistet, dass die Activate-Methode nach außen atomar erscheint und einen konsistenten Zustand hinterlässt. Außerdem verhindert die Transaktion, dass sich nebenläufig ausgeführte Activate-Operationen beeinflussen. Die Klienten der Komponente bekommen nicht die temporären Zustände mit, die während einer Transaktion auftreten und die Semantik vorübergehend verletzen können<sup>1</sup>. Nachdem eine Transaktion abgeschlossen wurde, informiert die Simulations-Komponente die Klienten über die Änderungen, welche die Transaktion bewirkt hat. Hierfür kommt das Beobachter-Entwurfsmuster nach [Gamma u. a. (2004)] zum Einsatz.

Die Klienten der Komponente können die kausalen Zusammenhänge anhand eines Transaktionsprotokolls detailliert nachvollziehen. Das Transaktionsprotokoll besteht aus TransaktionAction-Objekten, welche die jeweils Auslösende Aktion und die ausgelösten Aktionen referenzieren (siehe Abbildung 4.7). Eine Transaktion protokolliert folgende Aktionen:

- **ActivateAction** Ein ausführbares Element wird aktiviert. Beispielsweise könnte eine Aktivität ausgeführt werden oder ein Event ausgelöst werden.
- **TokenTransfer** Ein Token wird zwischen mehreren Elementen ausgetauscht.
- **EnableAction** Ein Element wird aufgrund der neuen Token-Belegung aktivierbar.
- **DisableAction** Ein Element ist aufgrund der neuen Token-Belegung nicht mehr aktivierbar.
- **EvaluateExpressionAction** Eine Bedingung wird ausgewertet und mit Wahr oder Falsch belegt. Die Wahrheitsbelegung ist für ein bestimmtes Token gültig.

---

<sup>1</sup>Zwischen dem Entfernen eines Tokens von einem Element und dem Hinzufügen des Tokens zu einem anderen Element ist ein Token keinem Element zugeordnet. Dieser temporäre Verlust des Tokens verletzt die operationelle Semantik der BPMN.

Die Protokollierung dieser Aktionen ist nötig, um die Simulation eines Geschäftsprozesses nachvollziehbar visualisieren zu können. Außerdem ermöglicht das Protokoll, eine Transaktion rückgängig zu machen oder eine rückgängig gemachte Transaktion erneut durchzuführen. Der Entwurf der Transaktion orientiert sich deshalb am Befehls-Entwurfsmuster nach [Gamma u. a. (2004)].

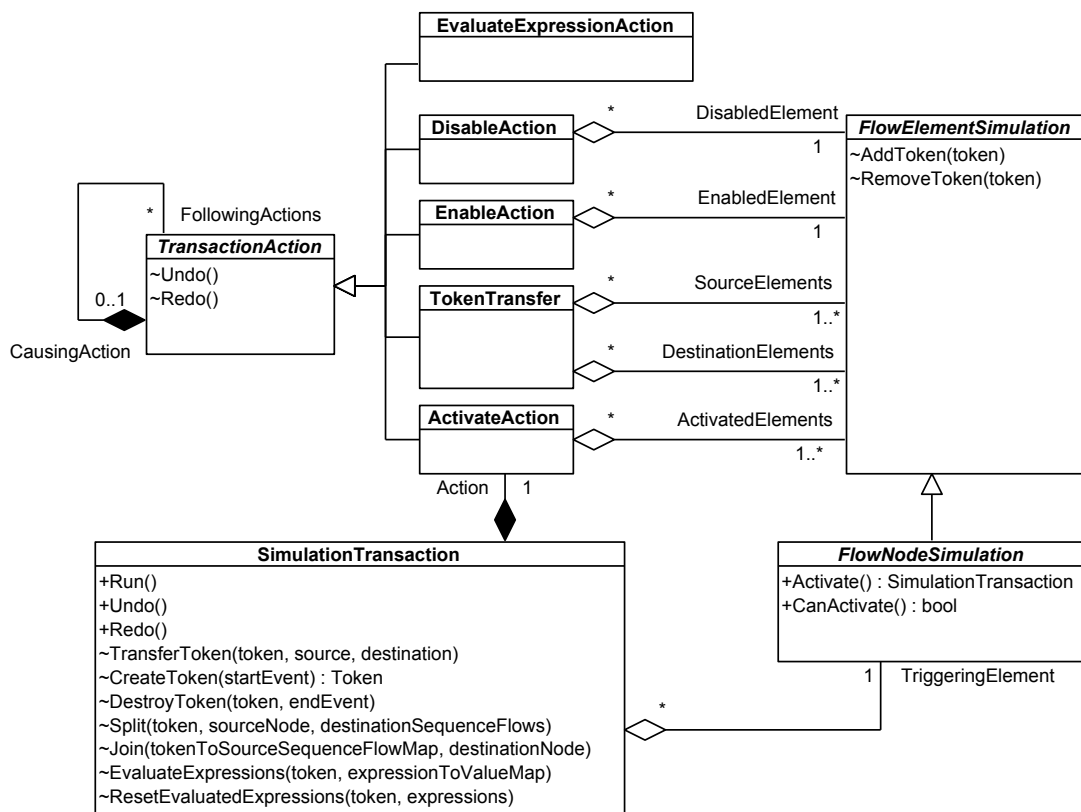


Abbildung 4.7.: Fachlicher Entwurf einer Transaktion (statische Sicht)

#### 4.2.1.3. BPMN Simulations-Visualisierung

Die Visualisierung einer Geschäftsprozess-Simulation verwaltet die Komponente “BPMN Simulations-Visualisierung”. Sie definiert, welche Visualisierungszustände ein Modellelement einnehmen kann und welche Zustandsübergänge gültig sind. Während einer Simulation speichert sie die Visualisierungszustände der Modellelemente (Visualisierungszustände und Zustandsübergänge siehe 4.1.2.3). Außerdem bestimmt diese Komponente die Dauer, die ein Modellelement in einem bestimmten Visualisierungszustand verweilen muss, damit die Benutzer die Visualisierung nachvollziehen können. Die Visualisierungszustände können über eine Schnittstelle der Komponente abgerufen werden.

Damit eine Simulation und ihre Visualisierung konsistent bleiben, kapselt die Komponente eine Simulations-Instanz nach außen ab, so dass nicht von außen auf die Schnittstelle zur Steuerung einer Simulation zugegriffen werden kann. Stattdessen bietet die Geschäftsprozess-Simulations-Visualisierungs-Komponente eine eigene Schnittstelle an, mit der Aktivitäten ausgeführt werden können, sofern es ihr Visualisierungszustand zulässt.

#### **4.2.1.4. Graph**

Die Graph-Komponente verwaltet Graphen, die aus Knoten und Kanten bestehen. Zum einen definiert diese Komponente Schnittstellen zum Traversieren der Knoten und Kanten eines Graphens und zum Abrufen oder Manipulieren von deren Abmessungen und Positionen. Zum anderen implementiert diese Komponente grundlegende Funktionen, welche die Anwendung zum Bearbeiten von Graphen benötigt, wie zum Beispiel Graph-Layout-Algorithmen.

#### **4.2.1.5. BPMN ViewModel**

Die Komponente "BPMN ViewModel" ist eine Fassade für das Subsystem aus den Komponenten BPMN Metamodell, BPMN Simulations Visualisierung und Graph. Außerdem bereitet die BPMN-ViewModel-Komponente ein Geschäftsprozessmodell und seine Simulation als Graph für die Präsentation auf.

Die Fassade vereinfacht das Subsystem, indem es die drei einzelnen Komponenten von den Klienten der Komponente abschirmt und den Klienten eine vereinfachte Schnittstelle anbietet. Diese Schnittstelle vereinigt Geschäftsprozessmodell, Visualisierung und Graph zu einem Objektgraphen aus ViewModel-Objekten. Dies reduziert die Anzahl von Objekten, welche die Klienten handhaben müssen. Außerdem wird die Komplexität reduziert, welche durch die Beziehungen zwischen Geschäftsprozessmodell, Visualisierung und Graph entsteht.

#### **4.2.1.6. Anwendungskern**

Der Anwendungskern implementiert die Funktionen, welche die Anwender beim Modellieren benötigen. Dazu gehören Modellierungswerkzeuge und die Verwaltung von Modellen und Projekten. Der Anwendungskern verwendet die Graph-Komponente und ist unabhängig von den BPMN-Komponenten.



#### 4.2.1.7. Graph- und BPMN-Modell-Fabrik

Die Komponente Graph-Fabrik definiert abstrakte Schnittstellen, mit denen Klienten Knoten und Kanten eines Graphen erzeugen können. Die BPMN-Modell-Fabrik-Komponente spezialisiert die Graph-Fabrik und definiert Vorlagen für BPMN Flow Elemente. Diese Vorlagen entsprechen abstrakten Fabriken nach [Gamma u. a. (2004)].

Durch die abstrakte Graph-Fabrik-Komponente ist der Anwendungskern von der Modellierungssprache entkoppelt. Mittels Dependency Injection bekommt der Anwendungskern eine Referenz auf die BPMN-Modell-Fabrik, ohne an die BPMN gebunden zu sein. BPMN kann relativ einfach durch eine andere Modellierungssprache ersetzt werden, ohne den Anwendungskern anpassen zu müssen, indem die BPMN-Modell-Fabrik durch eine andere Graph-Fabrik ersetzt wird.

#### 4.2.1.8. Anwendungs-ViewModel

Die Anwendungs-ViewModel-Komponente bereitet den Anwendungskern für die Präsentation auf und verwaltet die Steuerung des Anwendungsfensters.

#### 4.2.1.9. Anwendungs-View

Die grafische Darstellung des Anwendungsrahmens und der Werkzeuge implementiert die Anwendungs-View-Komponente und ihre Subkomponenten. Der Anwendungsrahmen besteht aus den View-Elementen, die zur Steuerung der Anwendung (z.B. Menüs) und als Rahmen für die Geschäftsprozessmodelle dienen.

Die Surface-View-Komponente und die Multitouch-View-Komponente spezialisieren die Anwendungs-View-Komponente für die Verwendung mit einem Microsoft Surface beziehungsweise mit einem Multitouch-Bildschirm.

#### 4.2.1.10. BPMN-View

Die Darstellung der BPMN-Geschäftsprozessmodelle implementiert die Komponente "BPMN-View". Sie implementiert die [konkrete Syntax](#) der BPMN entsprechend den Vorgaben des BPMN 2.0 Standards [OMG (2011)].

#### 4.2.1.11. Graph-View und BPMN-View-Fabrik

Die View-Objekte, welche die Modellelemente darstellen, erzeugt die Anwendungs-View über die Schnittstellen einer Graph-Fabrik. Analog zur Graph-Fabrik-Komponente entkoppelt die abstrakte Graph-View-Fabrik die Anwendungs-View von der BPMN-View. Die BPMN-View-Fabrik spezialisiert die Graph-View-Fabrik und erzeugt die von der BPMN-View-Komponente implementierten Views. Per Dependency Injection erhält die Anwendungs-View eine Referenz auf die BPMN-View-Fabrik.

#### 4.2.1.12. Unterstützung mehrerer Modellierungssprachen

Die fachliche Architektur beschreibt die Architektur einer Anwendung, die mit BPMN 2.0 genau eine Modellierungssprache unterstützt. Es können aber auch mehrere Modellierungssprachen unterstützt werden, wenn die von der konkreten Modellierungssprache abhängigen Komponenten spezialisiert werden. Pro Modellierungssprache müssen die folgenden Komponenten spezialisiert werden:

- Geschäftsprozess Metamodell (abstrakte Syntax und statische Semantik) und Fabrik,
- Geschäftsprozess Simulation (operationelle Semantik),
- Geschäftsprozess Simulations Visualisierung,
- Geschäftsprozess ViewModel sowie
- Geschäftsprozess View (konkrete Syntax) und Fabrik

Die Gemeinsamkeiten der Komponenten können heraus faktorisiert werden. Die Komponenten Geschäftsprozess Metamodell und Geschäftsprozess Präsentation müssen aber in jedem Fall individuell implementiert werden.

Die Anwendungslogik ist ansonsten weitgehend unabhängig von der Modellierungssprache. Denn die Modellierungswerkzeuge sind überwiegend Graph-basiert und die Komponenten Graph, Anwendungskern sowie Anwendungs-View sind unabhängig von der Modellierungssprache. Die Verbindung zwischen Modellierungssprache und Anwendung entsteht durch Fabrik-Komponenten und Dependency Injection.

## 4.2.2. Technische Architektur und Implementierung

Die technische Architektur der Modellierungsanwendung ist maßgeblich durch die Unterstützung des Microsoft Surface und von Multitouch-Bildschirmen geprägt. Für diese beiden verschiedenen Geräte gibt es zwei verschiedene Implementierungen (siehe Abbildungen 4.9 und 4.10), wobei der größte Teil aus gemeinsam genutzten Komponenten besteht. Die Unterschiede beschränken sich auf die View-Schicht. Die Model- und ViewModel-Schicht sind identisch (siehe Abbildung 4.8).

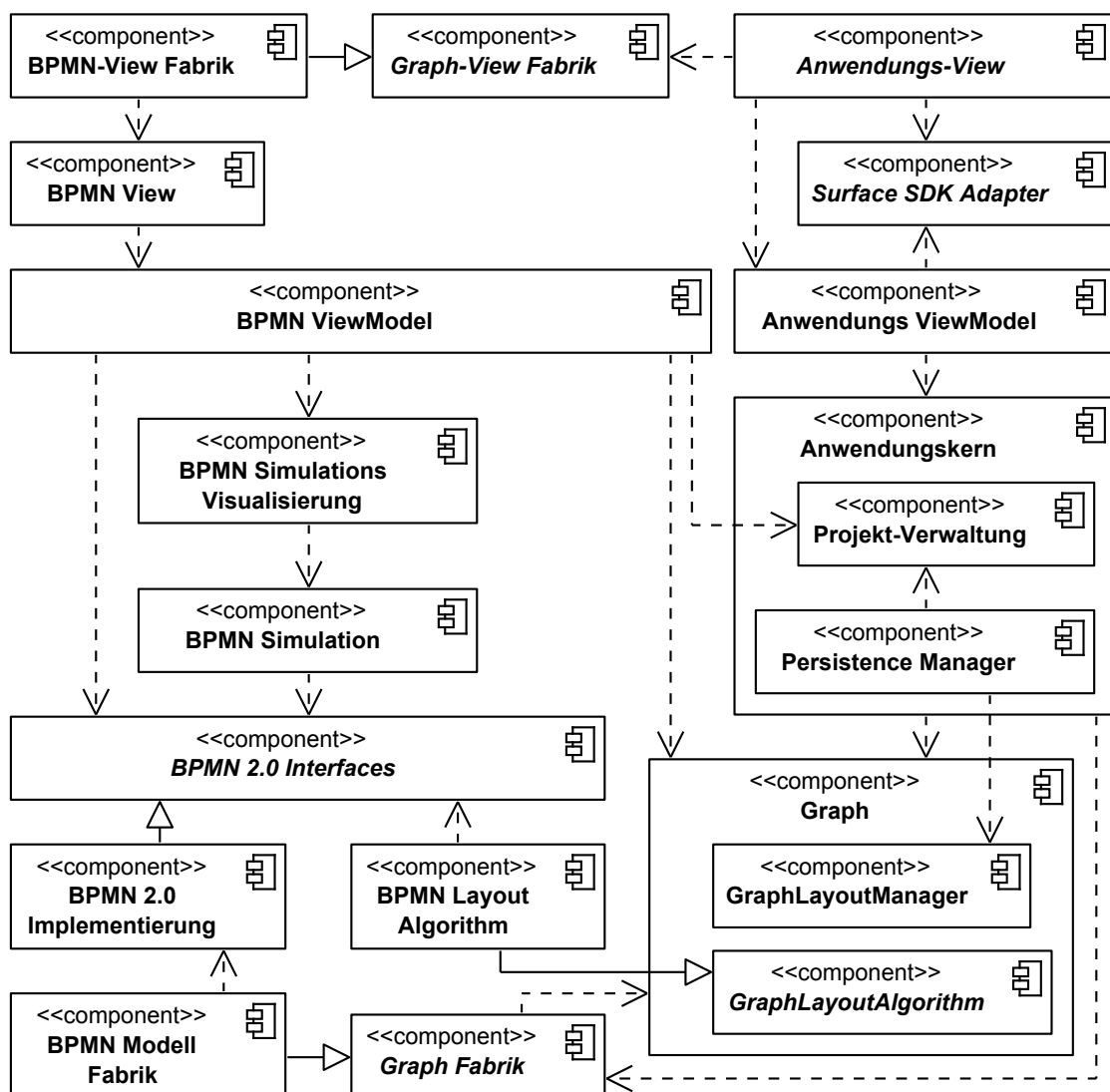


Abbildung 4.8.: Komponenten der technischen Architektur

Es gibt zwei verschiedene Implementierungen der View-Schicht, weil erstens die Benut-

zungsschnittstellen für die beiden Geräte-Typen optimiert wurden. Zweitens sind für die beiden Geräte-Typen unterschiedliche Software-Plattformen nötig. Es gibt zurzeit noch keine Softwareplattform, welche die Multitouch-Eingabe bei beiden Geräten unterstützt. Die technische Architektur der Modellierungsanwendung für das Microsoft Surface basiert deshalb auf dem Surface SDK 1.0 SP1 sowie Windows Vista (siehe Abbildung 4.10), die technische Architektur der Modellierungsanwendung für Multitouch-Bildschirme auf dem Surface Toolkit for Windows Touch<sup>2</sup> 1.5 und Windows 7 (siehe Abbildung 4.9). Beide basieren außerdem auf Microsofts .NET Framework 4.0. Das Surface SDK 1.0 SP1 basiert zwar auf .NET 3.5, .NET 4.0 ist aber abwärtskompatibel zu .NET 3.5.

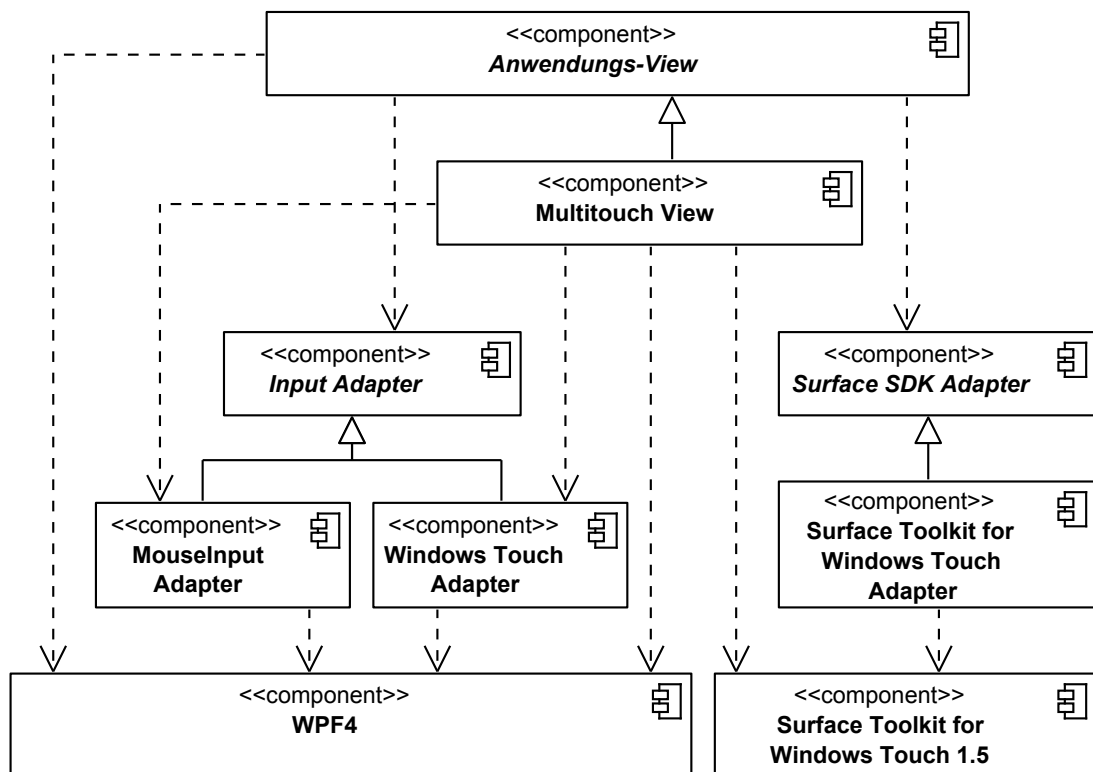


Abbildung 4.9.: Technische Architektur der Präsentationsschicht für Multitouch-Bildschirme

Die technische Architektur der Modellierungsanwendung ist so entworfen, dass möglichst wenig Plattform-spezifische Anpassungen nötig sind und diese an zentraler Stelle gekapselt sind. Dadurch soll der Entwicklungsaufwand für die Anpassung an die Plattformen minimiert werden. Die Funktionen, die an die Plattform angepasst werden müssen, kapselt jeweils eine abstrakte Komponente. Adapter für die spezifischen Plattformen spezialisieren diese

<sup>2</sup>Obwohl der Name es nahe legt, ist das Surface Toolkit for Windows Touch nicht für das Microsoft Surface geeignet.

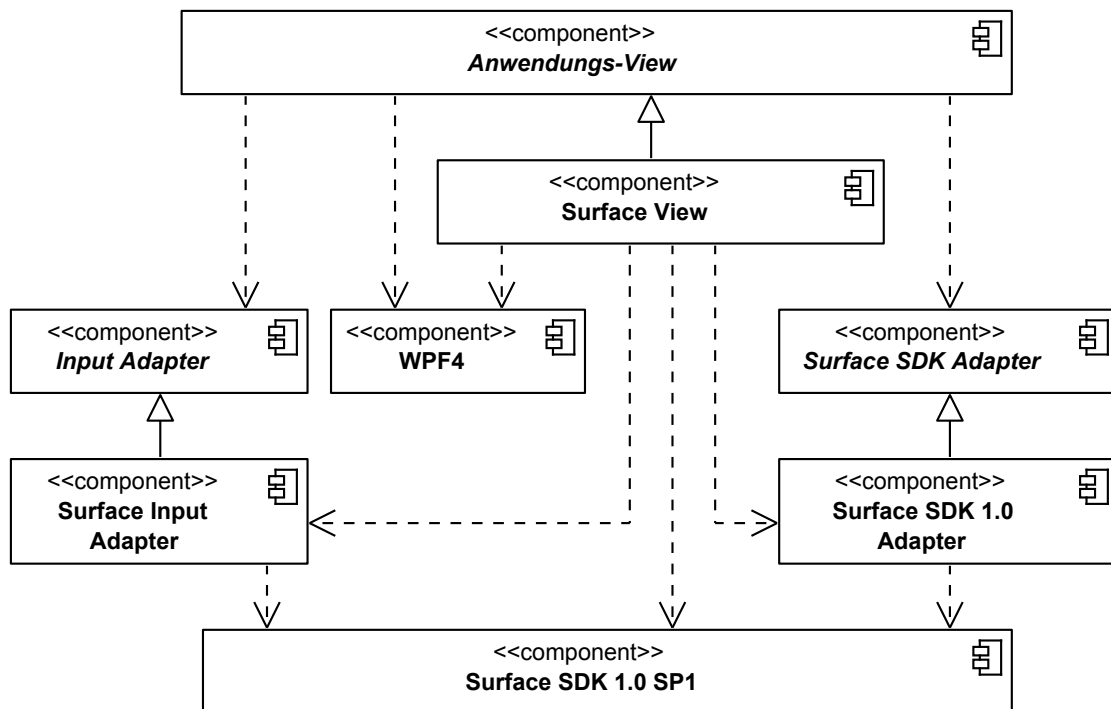


Abbildung 4.10.: Technische Architektur der Präsentationsschicht für das Microsoft Surface

Komponente. Die technische Umsetzung der Adapter erfolgt mit Objektkomposition nach dem Objektadapter-Entwurfsmuster nach [Gamma u. a. (2004)].

Konkret gibt es zwei abstrakte Komponenten die mit Adaptern angepasst werden: Input Adapter und Surface SDK Adapter.

#### 4.2.2.1. Input Adapter

Windows 7 ermöglicht die Multitouch-Eingabe mit beliebigen Multitouch-Bildschirmen, sofern es einen entsprechenden Gerätetreiber gibt. Beim Microsoft Surface kann die Multitouch-Eingabe aber nur mit dem Surface SDK 1.0 SP1 verwendet werden, denn für Windows 7 gibt es keinen Multitouch-Gerätetreiber. Um dennoch generische Multitouch-Bildschirme und das Surface zu unterstützen, ohne an zahlreichen Stellen plattformspezifische Anpassungen machen zu müssen, kapselt der Input Adapter alle Funktionen die Touch-basiert sind. Außerdem abstrahiert der Input Adapter alle Zeiger-basierten Eingabemethoden (Maus, Stylus und Touch), um gleiche Eingabegesten gleich behandeln zu können, ohne dass duplizierte Implementierungen nötig sind.

Der Input Adapter abstrahiert Zeiger-basierte Eingabegeräte (bei .NET InputDevice) als generisches Eingabegerät (IInputDevice Schnittstelle, siehe Abbildung 4.11). Die grundlegenden Eingabeereignisse (InputEvents), die der Adapter anbietet, sind InputDown, InputUp und InputMove. Alle InputEvents sind RoutedEvents und werden nach dem Attached-Event-Konzept von der Windows Presentation Foundation (WPF) implementiert.

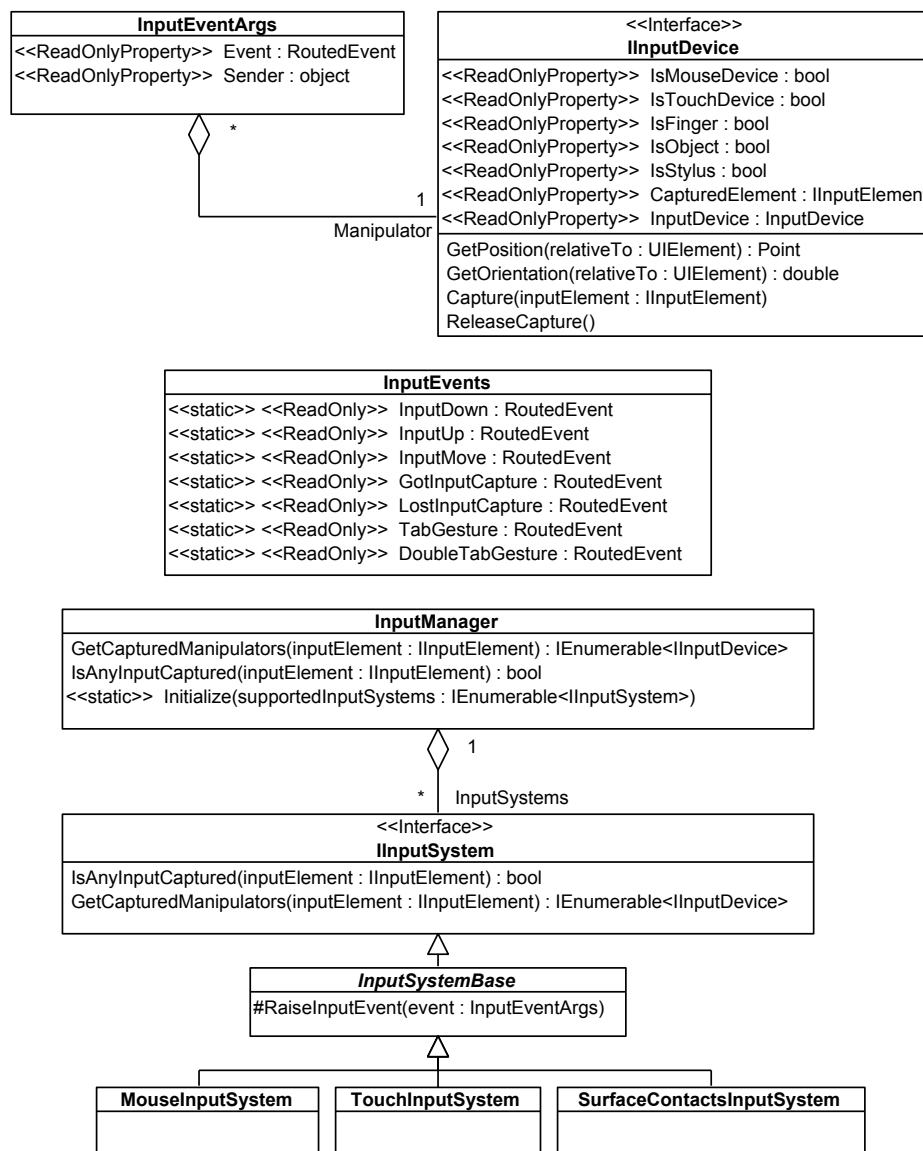


Abbildung 4.11.: Architektur der Eingabeverarbeitung

Eine Geräte-spezifische Input-Adapter-Komponente implementiert die `IInputSystem`-Schnittstelle. Eine Anwendung kann mehrere unterschiedliche `IInputSystem`-Implementierungen verwenden, um mehrere Eingabegeräte zu unterstützen:

- **Windows Touch Adapter / TouchInputSystem** Der Windows Touch Adapter adaptiert die Touch-Eingabemethode der Windows Presentation Foundation 4 (WPF 4) von .NET 4.0. Die Touch-Eingabe von .NET 4.0 selbst basiert auf Windows 7.
- **Surface Input Adapter / SurfaceInputSystem** Die Surface Input Adapter Komponente adaptiert die Touch-Eingabe vom Surface SDK 1.0 SP1.
- **Mouse Input Adapter / MouseInputSystem** Die Maus-Eingabemethode von WPF 4 adaptiert der Mouse Input Adapter.

Beim Starten der Anwendung werden die benötigten `IInputSystem`-Implementierungen beim `InputManager` registriert. Die Benutzungsschnittstelle für Multitouch-Bildschirme unterstützt die Maus- und Touch-Eingabe mit dem Mouse Input Adapter und dem Windows Touch Adapter (siehe Abbildung 4.9). Die Benutzungsschnittstelle für das Microsoft Surface unterstützt die Touch-Eingabe mit dem Surface Input Adapter (siehe Abbildung 4.10).

#### 4.2.2.2. Surface SDK Adapter

Das Surface SDK 1.0 SP1 und das Surface Toolkit for Windows Touch 1.5 sind Programm-bibliotheken mit Programmierschnittstellen (Application Programming Interface, API), die bei der Entwicklung von Multitouch-Benutzungsschnittstellen verwendet werden können. Viele dieser Programmierschnittstellen ähneln sich von ihrer Funktionsweise sehr stark, wie zum Beispiel die Multitouch-basierten Drag and Drop APIs. Häufig sind sogar die Klassen- und Methodennamen bei beiden Programmbibliotheken identisch. Die Methodensignaturen unterscheiden sich aber teilweise, so dass die Programmbibliotheken nicht gegeneinander ausgetauscht werden können. Um die Benutzungsschnittstelle für das Microsoft Surface und die Benutzungsschnittstelle für Multitouch-Bildschirme trotzdem möglichst wenig an das Surface SDK und das Surface Toolkit for Windows Touch anpassen zu müssen, definiert die Surface SDK Adapter Komponente Schnittstellen für die in beiden Programmbibliotheken enthaltene Funktionalität. Die Implementierung der Benutzungsschnittstelle verwendet diese Schnittstellen, anstatt eine der konkreten Bibliotheken zu verwenden. Der Surface SDK 1.0 Adapter und Surface Toolkit for Windows Touch Adapter implementieren diese Schnittstellen, indem sie das Surface SDK 1.0 SP1 beziehungsweise das Surface Toolkit for Windows Touch adaptieren (siehe Abbildungen 4.9 und 4.10). Zu Laufzeit bekommt die Benutzungsschnittstelle per Dependency Injection eine Referenz auf den Adapter, den sie verwenden soll.

### 4.3. Entwurf der Benutzungsoberfläche

Da Tabletop-Computer und Computer mit vertikalen Bildschirmen signifikante Unterschiede haben, hat sich der Autor entschieden, für diese beiden Geräte Klassen zwei unterschiedliche Benutzungsschnittstellen zu entwerfen. Beide Benutzungsschnittstellen haben eine gemeinsame Basis. Sie sind aber für die jeweiligen Eigenschaften der Geräte optimiert.

#### 4.3.1. Entwurfsmetaphern Werkzeug, Material und Arbeitsfläche

Der Entwurf der Benutzungsschnittstelle orientiert sich am Werkzeug und Material-Leitbild nach [Züllighofen (1998)]. Werkzeug und Material sind Metaphern, die den Benutzern eine bildhafte und gegenständliche Vorstellung über die Elemente und Funktionen der Benutzungsoberfläche ermöglichen sollen [Züllighofen (1998)]. Werkzeuge sind Gegenstände mit denen Personen Materialien bearbeiten, um bestimmte Arbeitsergebnisse zu erreichen. In einer Software haben Werkzeuge und Materialien eine virtuelle Repräsentation. Bei dieser Modellierungsanwendung sind Projekte, Modelle und Modellelemente Materialien. Die Funktionen, mit denen die Benutzer Projekte, Modelle und Modellelemente bearbeiten, sind Werkzeuge.

Zwei weitere Metaphern, die beim Entwurf der Benutzungsoberfläche eine wesentliche Rolle spielen, sind eine Werkbank und die Arbeitsfläche einer Werkbank. Eine Werkbank ist ein Arbeitstisch mit speziellen Vorrichtungen für bestimmte Arbeitsaufgaben. Auf der Arbeitsfläche bearbeiten Personen Materialien mit Werkzeugen. Beim Entwurf der Modellierungsanwendung ist die Modellierungsanwendung die Werkbank und die Arbeitsfläche der wesentliche Teil der Benutzungsoberfläche.

#### 4.3.2. Bereiche der Benutzungsoberfläche

Unabhängig vom Gerät ist die Benutzungsoberfläche in zwei Bereiche eingeteilt: eine Arbeitsfläche und eine Kontrollleiste. Auf der Arbeitsfläche können die Benutzer Materialien sowie Werkzeuge ablegen und die Materialien bearbeiten. Mit der Kontrollleiste können die Benutzer die Arbeitsfläche steuern und überwachen. Je nach Arbeitssituation zeigt die Benutzungsoberfläche spezielle Arbeitsflächen zum Verwalten von Projekten und Modellen oder zum Bearbeiten eines Modells an. Die Arbeitsfläche füllt die größte Fläche der Benutzungsoberfläche aus, damit möglichst viel Platz für Materialien vorhanden ist.



#### 4.3.2.1. Die Arbeitsfläche

Nach dem Start der Anwendung zeigt die Benutzungsoberfläche die Arbeitsfläche "Projekte" an, welche die vorhandenen Projekte auflistet. Die Benutzer können hier neue Projekte anlegen, die vorhandenen Projekte verwalten und ein Projekt auswählen, um es zu bearbeiten. Nachdem die Benutzer ein Projekt ausgewählt haben, wechselt die Benutzungsoberfläche zur Arbeitsfläche "Modelle". Diese listet die zu dem aktiven Projekt gehörenden Geschäftsprozessmodelle auf. Bei der Modelle-Arbeitsfläche können die Benutzer neue Geschäftsprozessmodelle anlegen, vorhandene Geschäftsprozessmodelle verwalten und ein Geschäftsprozessmodell auswählen, um es zu betrachten und zu bearbeiten. Sobald die Benutzer ein Geschäftsprozessmodell zum Bearbeiten ausgewählt haben, wechselt die Benutzungsoberfläche zur Arbeitsfläche "Modellierung". Die Modellierungs-Arbeitsfläche zeigt das Geschäftsprozessmodell mit der konkreten Syntax der verwendeten Modellierungssprache (hier BPMN) an. Auf dieser Arbeitsfläche können die Benutzer das Modell bearbeiten und Abläufe des Geschäftsprozesses simulieren.

#### 4.3.2.2. Die Kontrollleiste

Am Rand des Bildschirms zeigt die Benutzungsoberfläche jederzeit die Kontrollleiste an. Die Kontrollleiste ist ein spezielles Werkzeug, das die aktuelle Arbeitsfläche bearbeitet. Sie beinhaltet das Hauptmenü der Anwendung und Funktionen, welche die Benutzer bei der aktuellen Arbeitsfläche häufig benötigen. Die Kontrollleiste zeigt außerdem den Status sowie Warnmeldungen für das auf der Arbeitsfläche befindliche Material an.

Die Funktionen, welche die Kontrollleiste bereitstellt, betreffen immer entweder die Anwendung oder die gesamte Arbeitsfläche. Diese Funktionen dienen aber nicht zum Bearbeiten einzelner Materialien aus einer Menge von Materialien (Projekte, Modelle, Modellelemente). Einzelne Materialien können die Benutzer entweder über Gesten direkt bearbeiten (z. B. Vergrößern) oder mit speziellen Werkzeugen bearbeiten. Anwendungsbezogene Funktionen der Kontrollleiste sind unter anderem die Auswahl der Arbeitsfläche (Projekte, Modelle oder Modellierung) und das Beenden der Anwendung. Auf die Arbeitsfläche bezogene Funktionen sind das Speichern von Modellen und Projekten, das Ausdrucken von Modellen sowie das Anordnen von Modellelementen mit einem Layoutalgorithmus.

Das Hauptmenü beinhaltet die Befehle, welche die Benutzer bei den meisten Sitzungen benötigen, aber im Laufe einer Sitzung nicht regelmäßig verwenden. Die Benutzer können diese Befehle jederzeit über das Hauptmenü aufrufen, die Benutzungsoberfläche ist aber nicht mit zu vielen ständig angezeigten Schaltflächen überladen.

Die wichtigsten Funktionen, welche die Benutzer während einer Sitzung regelmäßig verwenden, stellt die Kontrollleiste den Benutzern über Schaltflächen direkt zur Verfügung. Die

Anzahl der Schaltflächen in der Kontrollleiste muss aber gering sein (maximal etwa drei), damit die Benutzungsoberfläche übersichtlich bleibt.

**Spezifische Funktionen bei der Modellierungs-Arbeitsfläche** Wenn die Modellierungs-Arbeitsfläche aktiv ist, zeigt die Kontrollleiste den Modellierungsstatus an, und die Benutzer können über die Kontrollleiste die Abläufe eines Anwendungsfalls aufrufen.

Den Status zeigt die Kontrollleiste mit Symbolen an. Die Benutzer können anhand dieser Symbole jederzeit erkennen, ob ein Modell Fehler oder Anzeichen von Fehlern enthält. Beispielsweise kann die Anwendung die Benutzer warnen, wenn das Modell ein Deadlock enthält oder nicht vollständig ist. Über die Symbole können die Benutzer Details zu den Warnmeldungen aufrufen und die betroffenen Modellelemente hervorheben.

Die Benutzer können die Abläufe eines Geschäftsprozesses, welche sie als Standardablauf und als alternative Abläufe definiert haben, sowie die zuletzt im Simulationsmodus durchgespielten Abläufe über einen Toggle-Button in der Kontrollleiste ein- und ausblenden.

### 4.3.3. Benutzungsoberfläche für Computer mit vertikalen Multitouch-Bildschirmen

Die Benutzungsoberfläche für vertikale Multitouch-Bildschirme (siehe Abbildung 4.12) ist für die Multitouch- und Tastatur-Eingabe optimiert. Sie kann aber auch ohne Einschränkungen mit einer Maus bedient werden und ist somit auch für klassische Computer geeignet, die keine Multitouch-Eingabe unterstützen. Um eine ergonomische Multitouch-Eingabe zu ermöglichen, sind allerdings die Schaltflächen der Benutzungsoberfläche größer als es bei für die Maus-Eingabe optimierten Benutzungsoberflächen üblich ist.

#### 4.3.3.1. Werkzeugleiste

In der Modellansicht zeigt die Benutzungsoberfläche am linken oder rechten Rand des Hauptbereichs eine Werkzeugleiste an. In dieser befinden sich die Werkzeuge, mit denen die Benutzer ein Geschäftsprozessmodell bearbeiten und eine Simulation steuern können: die Modellelement-Palette, der Papierkorb und der Simulation-Controller. Außerdem können die Benutzer über die Werkzeugleiste den Skalierungsfaktor, mit dem ein Modell angezeigt wird, und den Modus der Modellansicht einstellen.

Die Werkzeuge befinden sich jeweils in individuell ausblendbaren Bereichen der Werkzeugleiste. Wenn die Benutzer einzelne Werkzeuge ausblenden, schaffen sie Platz für die übrigen

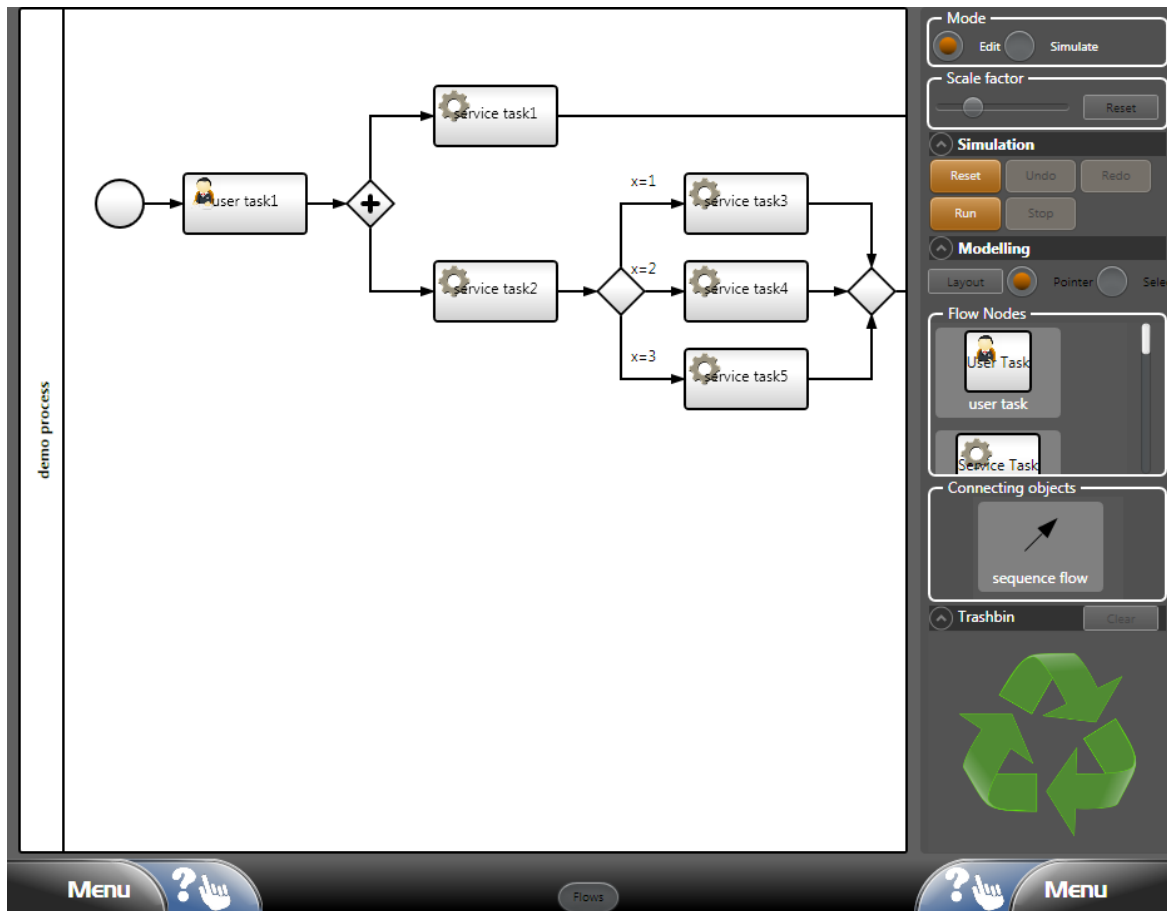


Abbildung 4.12.: Benutzungsoberfläche für Computer mit vertikalen Multitouch-Bildschirmen

Werkzeuge. Die Modell-Element Palette und der Papierkorb können dann mehr Elemente anzeigen und ersparen den Benutzern, durch die Listen zu scrollen.

#### 4.3.3.2. Modusauswahl

Wie in Abschnitt 4.1.3 beschrieben aktiviert die Anwendung teilweise automatisch den Modus, den die Benutzer benötigen. Falls nötig können die Benutzer manuell zwischen dem Modellierungsmodus und dem Simulationsmodus wählen. Sie können den Modus mit zwei Radio-Buttons einstellen, die sich in der Werkzeugleiste befinden.

#### 4.3.3.3. Texteingabe

Die Benutzer können die Modellelemente mit einer Hardware-Tastatur beschriften. Die Benutzer können zwar grundsätzlich die Bildschirmtastatur der Betriebssysteme verwenden, dies ist aber aus ergonomischen Gründen nicht vorgesehen. Bei einem vertikalen Multitouch-Bildschirm müssten die Benutzer beim Schreiben mit einer Bildschirmtastatur ihre Hände anwinkeln. Auf Dauer kann dies zu Handgelenks-Beschwerden führen.

Mit zweimaligem Tippen (Double Tap) oder einem Doppelklick können die Benutzer ein Modellelement zum Beschriften auswählen. Daraufhin zeigt die Benutzungsschnittstelle eine Textbox im Bereich des Modellelements an, in der die Benutzer den Text eingeben können.

#### 4.3.4. Benutzungsoberfläche für Tabletop-Computer

Die Benutzungsoberfläche für Tabletop-Computer ist an eine horizontale Arbeitsfläche angepasst und nutzt die Erkennung von physischen Objekten. Diese beiden Eigenschaften sind ausschlaggebend für die Unterschiede zu der Benutzungsoberfläche für Computer mit vertikalen Multitouch-Bildschirmen. Diese Benutzungsoberfläche ist zwar für das Microsoft Surface optimiert, der Entwurf kann aber auch auf jeden anderen Tabletop-Computer übertragen werden, der multitouch-fähig ist und die Position sowie die Art von physischen Objekten erkennen kann. Dieser Entwurf ist aber nur dann für das kommende Surface 2.0 geeignet, wenn dieses horizontal betrieben wird<sup>3</sup>. Wenn das Surface 2.0 vertikal betrieben wird, können die Benutzer keine Gegenstände auf die Arbeitsfläche stellen.

Die Benutzungsoberfläche für Tabletop-Computer unterscheidet sich von der Benutzungsoberfläche für klassische Computer und Computer mit vertikalen Multitouch-Bildschirmen in folgenden drei Punkten:

1. In der Modellansicht hat der Hauptbereich keine Werkzeugleiste (Modellansicht und Hauptbereich siehe Abschnitt 4.3.2.1). Stattdessen sind alle Werkzeuge, mit denen die Benutzer ein Modell bearbeiten und eine Simulation steuern, an physische Objekte gebunden. Wenn die Benutzer ein Werkzeug benötigen, stellen sie den entsprechenden Gegenstand auf die Arbeitsfläche.
2. Die Funktionsleiste (siehe Kapitel 4.3.2.2) befindet sich sowohl am oberen Rand als auch am unteren Rand des Bildschirms. Dadurch können alle Benutzer, die um den Tisch herum sitzen, auf die Funktionsleiste zugreifen. Bei sehr großen Arbeitsflächen müsste die Funktionsleiste an allen vier Seiten platziert werden. Wegen der kleinen Arbeitsfläche reicht es jedoch beim Microsoft Surface 1.0 aus, die Funktionsleiste an

---

<sup>3</sup>Das Surface 2.0 kann sowohl horizontal als auch vertikal betrieben werden, da die vier Tischbeine abmontiert werden können. Die Arbeitsfläche hat eine geringe Bauhöhe, die eine Wandmontage erlaubt.

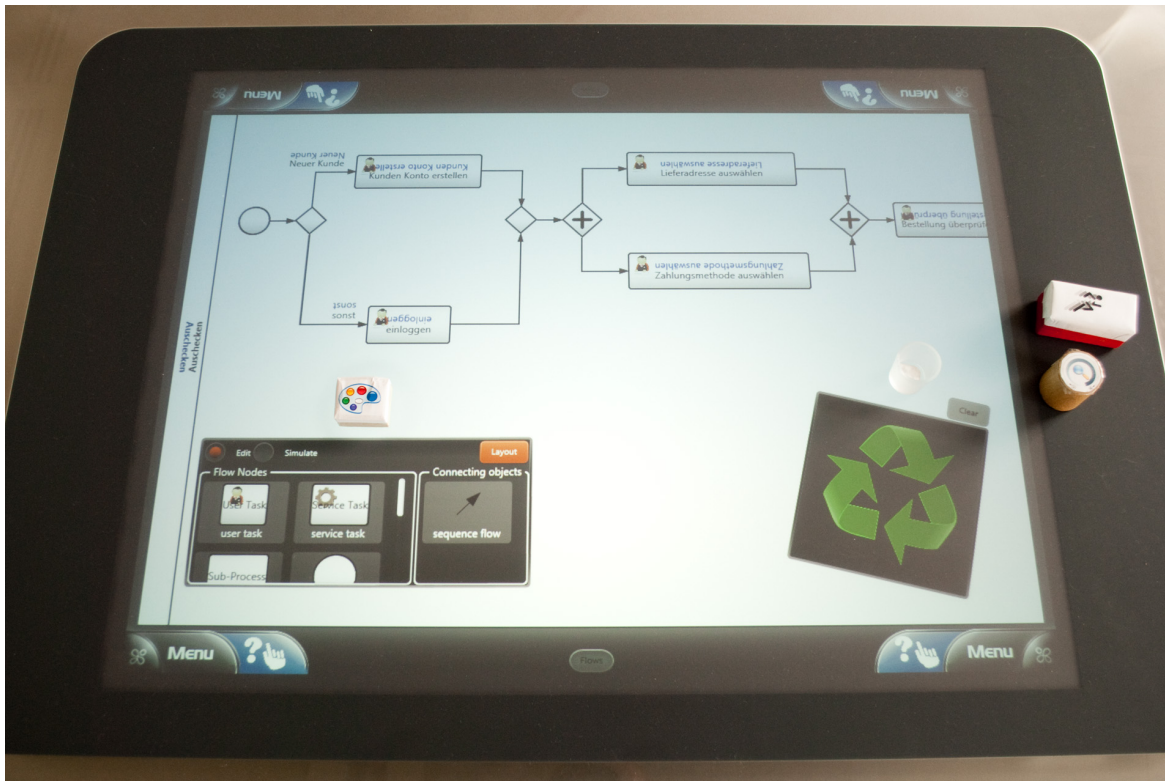


Abbildung 4.13.: Benutzungsoberfläche für Tabletop-Computer

zwei gegenüberliegenden Seiten zu platzieren. Außerdem spart dies Bildschirmfläche, die dem Hauptbereich (siehe Kapitel 4.3.2.1) zugeordnet wird.

- Um die Lesbarkeit der Beschriftungen von Modellelementen zu erleichtern, sind alle Modellelemente doppelt beschriftet. Eine Beschriftung ist jeweils zur oberen Seite des Bildschirms ausgerichtet und die andere zur unteren Seite. Jeder Benutzer kann dadurch eine Beschriftung aus einem Winkel von maximal  $\pm 90$  Grad betrachten. Da die Benutzer üblicherweise an den Breitseiten des Tisches sitzen, ist dieser Winkel in der Praxis geringer als 45 Grad. Ein Winkel von 45 Grad schränkt die Lesbarkeit kaum ein [Wigdor und Balakrishnan (2005)].

Es werden nur die Textelemente doppelt angezeigt, mit denen Modellelemente beschriftet sind. Bei den übrigen Textelementen der Benutzungsoberfläche ist dies nicht nötig, da es sonst nur Elemente gibt, die entweder mehrfach und in verschiedene Richtungen ausgerichtet angezeigt werden oder von den Benutzern gedreht werden können. Die Funktionsleiste ist mehrfach vorhanden und die Werkzeuge können die Benutzer zu sich ausrichten, indem sie die physischen Objekte drehen.

#### 4.3.4.1. Physische Objekte

Die Modellierungsanwendung verwendet physische Objekte (siehe Kapitel 2.4.2), um den Benutzern Werkzeuge bereitzustellen, die sie beim Arbeit mit Geschäftsprozessmodellen benötigen. Ein Werkzeug ist jeweils einem bestimmten Gegenstand zugeordnet. Die Benutzer können einen solchen Gegenstand als Werkzeug verwenden, indem sie ihn auf die Arbeitsfläche stellen. Die Benutzungsschnittstelle zeigt dann ein virtuelles Werkzeug neben dem Gegenstand an. Die Verwendung physischer Objekte als Werkzeug hat mehrere Vorteile. Sie helfen, wertvolle Bildschirmfläche zu sparen, und erleichtern den Benutzern die Bedienung der Werkzeuge und der Anwendung insgesamt.

Physische Objekte werden unter anderem deshalb als Werkzeug verwendet, weil die Benutzer mit ihnen die Anwendung einfach bedienen können. Die Werkzeuge sind mit physischen Objekten intuitiver bedienbar, als bei einer rein virtuellen Benutzungsoberfläche. Die physischen Objekte verkörpern jeweils den Griff eines Werkzeugs. Um ein Werkzeug zu verwenden, müssen die Benutzer einen bestimmten Gegenstand greifen und auf die Arbeitsfläche stellen. Die Benutzer können die Position und Ausrichtung des virtuellen Werkzeugs ändern, indem sie die Position und Ausrichtung des Gegenstands ändern. Wenn sie das Werkzeug nicht mehr benötigen, können die Benutzer den Gegenstand wieder beiseite legen. Es sind auch Werkzeuge möglich, bei denen die Ausrichtung eine spezielle Bedeutung hat. Mit dem Zoom-Werkzeug können die Benutzer durch Drehen des Gegenstandes den Skalierungsfaktor ändern, mit dem ein Modell dargestellt wird. Die zuvor genannten Handlungen sind natürlicher, als Werkzeuge über virtuelle Schaltflächen ein- und auszublenden oder zu verschieben, weil das Greifen von Gegenstände eine alltägliche Tätigkeit ist.

Werkzeuge, die an physische Objekte gebunden sind, fördern außerdem die Übersichtlichkeit der Anwendung. Da die Benutzer auf einfache Weise die Werkzeuge entfernen können, müssen nur die Werkzeuge angezeigt werden, die sie tatsächlich benötigen. Die Benutzungsoberfläche ist somit nicht mit Controls überladen.

Ebenso helfen physische Objekte, Platz zu sparen. Die Benutzungsoberfläche benötigt keine Werkzeuggesten, in denen sich die Werkzeuge bei herkömmlichen Benutzungsoberflächen befinden. Die Werkzeuge werden nur dann angezeigt, wenn die Benutzer die entsprechenden Gegenstände auf die Arbeitsfläche stellen. Wenn die Benutzer ein Werkzeug nicht benötigen, können sie den Gegenstand von der Arbeitsfläche entfernen. Die so eingesparte Bildschirmfläche kommt dem Bereich zugute, in dem die Modelle angezeigt werden. Je größer ein Modell ist, desto wichtiger ist es, viel Bildschirmfläche zur Anzeige des Modells zur Verfügung zu haben. Eine große Fläche zur Anzeige von Modellen ermöglicht den Benutzern, viele Modellelemente im Blick zu haben und hilft ihnen, den Überblick zu behalten, ohne den Skalierungsfaktor zu verringern. Das Microsoft Surface 1.0 hat einen 32" Bildschirm mit einer für heutige Verhältnisse sehr geringen Auflösung von 1024x768 Pixel. Zum Anzeigen von

Modellen reicht das gerade aus. Deshalb ist es hier besonders wichtig, einen möglichst hohen Anteil der Bildschirmfläche zum Anzeigen eines Modells nutzen zu können. Aber auch bei künftigen Tabletop Computern mit größeren und hochauflösenden Bildschirmen wird es wichtig sein, Bildschirmfläche zu sparen, denn der begrenzende Faktor für die Bildschirmgröße ist die Erreichbarkeit aller Bildschirmbereiche. Der Bildschirm darf nur so groß sein, dass die Benutzer alle Bereiche mit ihren Händen erreichen können. Andernfalls wäre keine direkte Touch-Eingabe möglich und deren Vorteile (siehe Kapitel 2.3.1) gingen verloren. Außerdem sollte jeder Benutzer alle Bildschirmbereiche erreichen können, ohne seine Position am Tisch ändern zu müssen. Sonst würde die Größe des Tisches die Benutzer bei der Modellierung beeinträchtigen.

**Klassifizierung von Werkzeugen** Die Modellierungsanwendung hat drei Klassen von Werkzeugen, die mit physischen Objekten Verknüpft sind:

- **Modellierungs-Werkzeuge** Dies sind Werkzeuge, mit denen die Benutzer ein Modell bearbeiten.
- **Simulations-Werkzeuge** Mit diesen Werkzeugen steuern die Benutzer eine Simulation.
- **Ansicht-Werkzeuge** Die Benutzer können mit diesen Werkzeugen die Ansicht anpassen, mit der sie ein Modell betrachten. Die Anpassungen, welche sie vornehmen können, sind unabhängig vom Modellierungsmodus und Simulationsmodus (z. B. Skalierungsfaktor, Ausrichtung des Modells).

Die Werkzeuge müssen zu genau einer dieser Klassen gehören. Dies ermöglicht der Anwendung, den Modellierungsmodus und Simulationsmodus automatisch auszuwählen.

**Automatische Modusauswahl** Abhängig davon, welche Gegenstände sich auf der Arbeitsfläche befinden, stellt die Anwendung den Modellierungs- und Simulations-Modus automatisch ein. Zusätzlich zu den in Abschnitt 4.1.3 beschriebenen Maßnahmen, erspart die automatische Modusauswahl den Benutzern, den Modus manuell einzustellen müssen.

Da die Werkzeuge wie zuvor beschrieben klassifiziert sein müssen, kann die Anwendung anhand der Gegenstände, die sich auf der Arbeitsfläche befinden, erkennen, welchen Modus die Benutzer benötigen. In dem Moment, in dem die Benutzer einen Gegenstand auf die Arbeitsfläche stellen, stellt die Anwendung den Modus automatisch ein. Je nachdem, ob der Gegenstand mit einem Modellierungs-Werkzeug oder einem Simulations-Werkzeug assoziiert ist, aktiviert die Anwendung den Modellierungsmodus oder den Simulationsmodus. Wenn der Gegenstand mit einem Ansicht-Werkzeug verknüpft ist, behält die Anwendung den aktuellen Modus bei. Die Gegenstände, die sich bereits auf dem Surface befanden, ignoriert die



Anwendung bei der automatischen Modusauswahl. Die Anwendung fällt die Entscheidung über den Modus anhand des zuletzt auf der Arbeitsfläche platzierten Gegenstands, weil es mit hoher Wahrscheinlichkeit das Werkzeug ist, welches die Benutzer gerade benötigen.

Wenn sich sowohl Modellierungs-Werkzeuge als auch Simulations-Werkzeuge auf der Arbeitsfläche befinden, kann es jedoch bei der automatischen Moduseinstellung zu Konflikten zwischen gleichzeitig arbeitenden Benutzern kommen. Die Benutzer müssten sich dann einigen, welchen Modus sie verwenden möchten. Solche Situationen sind aber bei einem geordneten Vorgehen, wie man es bei einer professionell geführten Modellierungssitzung erwarten kann, unwahrscheinlich. Da sich die Benutzer ohnehin über einen Moduswechsel einigen müssen, dürfte der automatische Moduswechsel nur selten Konflikte auslösen, die ohne den automatischen Moduswechsel nicht auftreten würden. Insgesamt verspricht die auf physischen Objekten basierende automatische Modusauswahl, den Benutzern öfter Arbeit abzunehmen als sie zu behindern. Ein Usability-Test muss dies aber bestätigen.

**Papierkorb** Der Papierkorb ist ein Modellierungswerkzeug, das mit einem Gegenstand aus Plexiglas verknüpft ist, der wie ein Papierkorb geformt ist (siehe Abbildung 4.14). Im Modellierungsmodus können die Benutzer Elemente per Drag and Drop im Papierkorb ablegen und wieder herausnehmen (siehe Abschnitt 4.1.1.2).

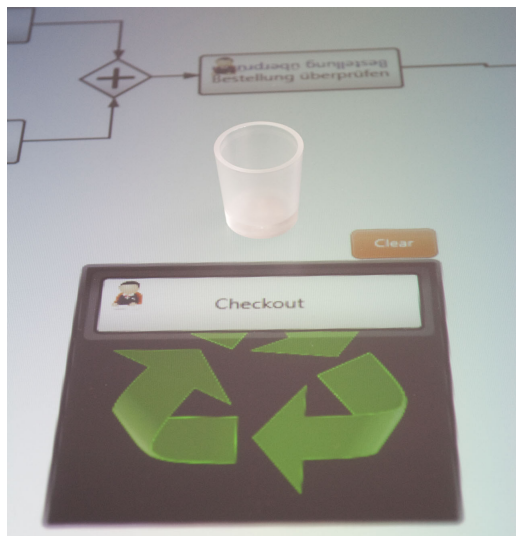


Abbildung 4.14.: Papierkorb

**Modellelement-Palette** Mit der Modellelement-Palette können die Benutzer, wie in Abschnitt 4.1.1.1 beschrieben, Modellelemente erstellen, wenn der Modellierungsmodus aktiv



ist. Die Modellelement-Palette ist mit einem Gegenstand assoziiert, der durch ein Paletten-Symbol gekennzeichnet ist (siehe Abbildung 4.15).

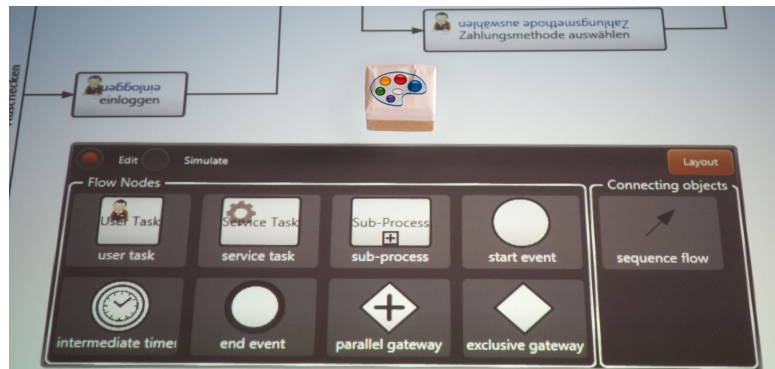


Abbildung 4.15.: Modellierungswerkzeug

Wenn die Benutzer den Gegenstand für die Modellelement-Palette auf die Arbeitsfläche stellen, aktiviert die Anwendung automatisch den Modellierungsmodus. Alternativ können die Benutzer mit der Modellelement-Palette den Modus manuell über zwei Radio-Buttons auswählen. Diese Buttons befinden sich in der Funktionsleiste im oberen Bereich des Werkzeugs (siehe Abbildung 4.15).

Die Benutzer können die Größe des virtuellen Werkzeugs mit einer Vergrößern- und Verkleinern-Geste verändern. Sie können es vergrößern, indem sie es mit zwei oder mehr Fingern berühren und diese Finger von einander weg bewegen. Umgekehrt können sie es verkleinern, indem sie die Finger auf einander zu bewegen. Auf diese Weise können die Benutzer die Breite und die Höhe des Werkzeugs unabhängig voneinander an ihre Bedürfnisse anpassen. Je größer das Werkzeug ist, desto mehr Modellelemente kann die Modellelement-Palette darstellen. Dies erspart den Benutzern, durch die Liste der Modellelemente zu scrol-len.

**Simulation-Controller** Der Simulation-Controller ist ein Simulationswerkzeug mit dem die Benutzer eine Simulation steuern können (siehe Abschnitt 4.1.2.10). Die Anwendung aktiviert automatisch den Simulationsmodus, wenn die Benutzer den Gegenstand für den Simulation-Controller auf die Arbeitsfläche stellen. Zusätzlich können die Benutzer mit dem Simulation-Controller manuell zwischen dem Simulationsmodus und dem Modellierungsmodus wählen. Die Auswahl können sie mit zwei Radio-Buttons treffen (siehe Abbildung 4.16).

**Zoom-Werkzeug** Mit dem Zoom-Werkzeug können die Benutzer den Skalierungsfaktor einstellen, mit dem ein Modell angezeigt wird. Dieses Ansicht-Werkzeug ist mit einem Ge-

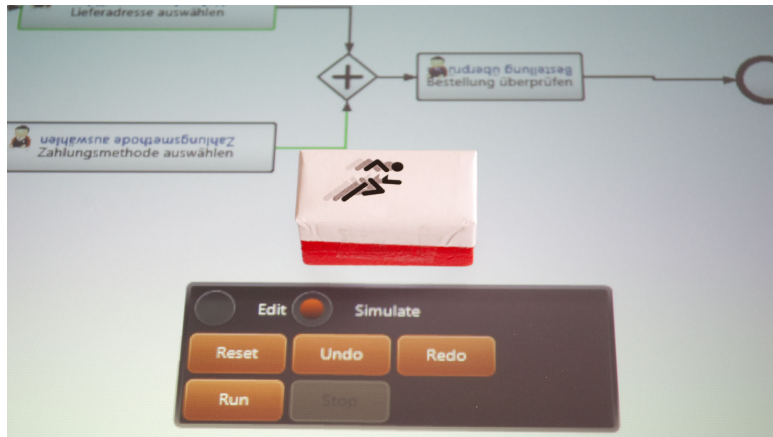


Abbildung 4.16.: Simulations-Werkzeug

genstand verknüpft, der die Form eines Kreiszyinders hat und mit einer Lupe gekennzeichnet ist (siehe Abbildung 4.17).

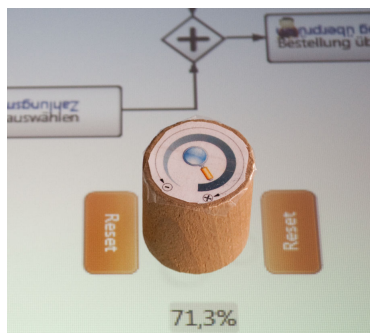


Abbildung 4.17.: Zoom-Werkzeug

Die Benutzer bedienen das Werkzeug, indem sie zuerst den Gegenstand auf die Arbeitsfläche stellen und diesen anschließend drehen. Wenn sie ihn im Uhrzeigersinn drehen, verringern sie den Skalierungsfaktor, so dass das Modell größer dargestellt wird. Umgekehrt können sie den Skalierungsfaktor vergrößern, indem sie den Gegenstand gegen den Uhrzeigersinn drehen. Die Anwendung berechnet den Skalierungsfaktor relativ zum vorherigen Wert. Die absolute Ausrichtung des Gegenstands auf der Arbeitsfläche hat keinen Einfluss auf die Berechnung. Damit die Benutzer das Zoom-Werkzeug leichter bedienen können, wird der Skalierungsfaktor mit einer exponentiellen Funktion berechnet:

$$s = s_0 + p \cdot s_0 \cdot e^{(1+s_0) \cdot \delta} \quad (4.1)$$

$$p = \frac{0.25}{360} \quad (4.2)$$

$s$  ist der neue Skalierungsfaktor und  $s_0$  ist der vorherige Wert.  $\delta$  ist die Winkeländerung in Grad.  $p$  ist ein konstanter Faktor, der anhand der Erfahrungen mit dem Zoom-Werkzeug festgelegt wurde. Wegen der exponentiellen Funktion nimmt der Einfluss von  $\delta$  auf  $s$  mit zunehmendem  $s_0$  exponentiell ab. Dies vermeidet, dass sich der Skalierungsfaktor bei großen  $s_0$  sprunghaft ändert, und ermöglicht den Benutzern den Skalierungsfaktor präzise einzustellen.

#### 4.3.4.2. Texteingabe

Für die Eingabe von Text ist eine Bildschirmtastatur vorgesehen (siehe Abbildung 4.18). Prinzipiell können die Benutzer auch eine schnurlose Tastatur verwenden. Eine Hardware-Tastatur hat aber den Nachteil, herumgereicht werden zu müssen, damit alle Benutzer Text eingeben können. Dies könnte den Arbeitsfluss der Benutzer unterbrechen. Auf eine Bildschirmtastatur kann dagegen jeder Benutzer schnell zugreifen. Außerdem könnte ein Benutzer eine Hardware-Tastatur für sich allein beanspruchen und dadurch die anderen Benutzer von der Texteingabe ausschließen. Dies könnte das gemeinsame Arbeiten der Benutzer stark einschränken.

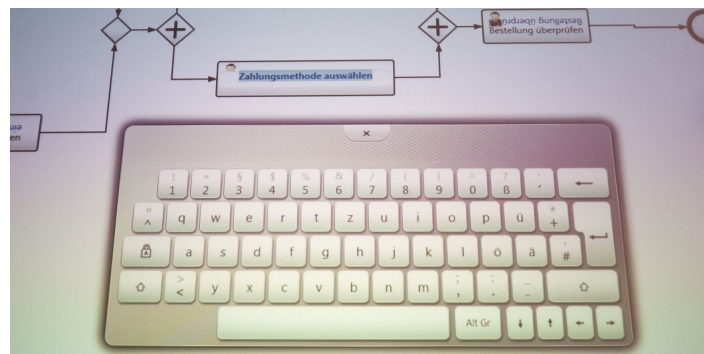


Abbildung 4.18.: Texteingabe mit der Bildschirmtastatur

Eine Bildschirm-Tastatur hat allerdings auch Nachteile (siehe Kapitel 2.3.4). Insbesondere die fehlende Haptik erschwert es den Benutzern eine Bildschirm-Tastatur zu verwenden. Trotzdem wird eine Bildschirmtastatur verwendet, da diese Anwendung besonderen Wert

auf die Kollaboration der Benutzer legt. Außerdem müssen die Benutzer bei der Geschäftsprozessmodellierung keine langen Texte eingeben. Dennoch ist eine Bildschirmtastatur eigentlich nicht angemessen für die Geschäftsprozessmodellierung und sollte nur verwendet werden, solange es keine geeignetere Texteingabemethode gibt.

Um die Beschriftung eines Modellelements zu ändern, müssen die Benutzer zweimal hintereinander auf das Modellelement tippen (Double-Tap). Anschließend blendet die Anwendung eine Textbox über dem ausgewählten Modellelement ein, in welche die Benutzer den Text eingeben können. Der Texteingabefokus wird automatisch auf diese Textbox gesetzt. Außerdem zeigt die Anwendung automatisch die Bildschirmtastatur an und richtet sie zu dem Benutzer aus, der auf das Modellelement getippt hat. Die automatische Ausrichtung ist möglich, weil das Surface die Ausrichtung eines Fingers erkennen kann.

## **5. Konzeption und Durchführung der Usability-Studie**

Die im Rahmen dieser Arbeit entwickelte Geschäftsprozessmodellierungsanwendung wurde in einer Usability-Studie auf ihre Eignung für die Geschäftsprozessmodellierung mit Multitouch-Bildschirmen und Multitouch-Tischen überprüft. Im Folgenden nennt dieses Kapitel zunächst das Ziel das die Usability-Studie verfolgt. Danach wird der Testablauf und das Szenario beschrieben, welches während des Tests simuliert wird. Zum Schluss werden die vier Testumgebungen beschrieben, die während des Tests verwendet wurden.

### **5.1. Testziel**

Das Ziel der Usability-Studie ist, Stärken und Schwächen von Multitouch-Bildschirmen und Multitouch-Tischen bei der Geschäftsprozessmodellierung aufzudecken. Anhand der Vor- und Nachteile soll anschließend die in der Einleitung genannte Fragestellung beantwortet werden, ob Multitouch-Tische besser für die Geschäftsprozessmodellierung geeignet sind, als klassische Computer, Multitouch-Bildschirme und Arbeitsumgebungen ohne Computer (siehe Kapitel 1). Das Ziel ist, die prinzipiellen Stärken und Schwächen der Geräte bei der Verwendung der Geschäftsprozessmodellierungsanwendung zu ermitteln. Nur ein untergeordnetes Ziel ist, Schwächen in der Umsetzung aufzudecken.

### **5.2. Testablauf und Szenario**

Um das Testziel zu erreichen, wurden bei dieser Usability-Studie mehrere Modellierungssitzungen durchgeführt und aufgezeichnet. Jede Modellierungssitzung wurde in einer von vier verschiedenen Testumgebungen durchgeführt. Die Usability-Studie ist in Testläufen organisiert. Während jedes Testlaufs wurde nacheinander in jeder der vier Testumgebungen eine Modellierungssitzung durchgeführt.

An jedem Testlauf nahmen jeweils vier Probanden teil. Zwei der Probanden nahmen dabei die Rolle von Fachexperten ein. Die beiden anderen Probanden nahmen die Rolle von Modellierungsexperten beziehungsweise Entwicklern ein. Während eines Testlaufs wechselten die Probanden ihre Rolle nicht. Da sich 16 geeignete Probanden zur Usability-Studie bereit erklärt hatten wurden vier Testläufe durchgeführt.

Vor jedem Testlauf wurden die Probanden mit dem Szenario vertraut gemacht, welches die Modellierungssitzungen simulieren. Die Entwickler bekamen zusätzlich eine kurze Einführung in die Modellierungssoftware, und ihnen wurde die verwendete Geschäftsprozessmodellierungssprache erklärt. Nach jeder Sitzung füllten die Probanden einen Fragebogen zu der Testumgebung aus. Nach der letzten Sitzung füllten Probanden zudem einen Fragebogen aus, bei dem sie Eignung der Arbeitsumgebungen für die Geschäftsprozessmodellierung bewerten sollten. Zum Schluss führte der Autor mit den Probanden ein Interview. Bei diesem Interview erläuterten die Probanden ihre Meinung zu den Testumgebungen.

### **5.2.1. Szenario**

In dem Szenario dieser Usability-Studie entwickelt eine Softwareentwicklungsfirma für einen Online-Shop-Betreiber eine Individualsoftware. Diese Individualsoftware soll die Geschäftsprozesse des Online-Shop-Betreibers unterstützen.

#### **5.2.1.1. Aufgabe der Fachexperten**

Die Probanden, welche die Rolle eines Entwicklers einnehmen, sollen sich in die Lage des fiktiven Online-Shop-Betreibers versetzen. Vor Beginn jeder Modellierungssitzung bekommen diese Probanden eine kurze Zusammenfassung eines Geschäftsprozesses, der bei dem Online-Shop eine Rolle spielt. Diesen Geschäftsprozess sollen sie den Entwicklern schildern, damit diese ein entsprechendes Geschäftsprozessmodell erstellen können. Die Beschreibung der Geschäftsprozesse stellt einen groben Rahmen dar und soll von den Fachexperten ausgeschmückt werden.

#### **5.2.1.2. Aufgabe der Entwickler**

Die Probanden, welche die Rolle eines Entwicklers einnehmen, sollen sich in die Lage eines Software-Entwicklers versetzen, der an der Entwicklung der Individualsoftware beteiligt ist. Die Aufgabe dieser Probanden ist, während der Sitzung und entsprechend den Angaben der Fachexperten einen Geschäftsprozess zu modellieren. Sie sollen dabei versuchen, den Geschäftsprozess zu verstehen. Am Ende der Modellierungssitzung sollen die Entwickler

den Geschäftsprozess durchspielen, um Fehler im Modell zu finden, und eventuelle Fehler korrigieren. Als Modellierungssprache sollen die Entwickler BPMN 2.0 verwenden.

#### **5.2.1.3. Die zu modellierende Geschäftsprozesse**

Die Geschäftsprozesse, welche die Probanden während eines Testlaufs modelliert haben, sind:

1. die Bestellung einer Ware,
2. das Anlegen eines Kundenkontos,
3. die Reklamation einer defekten Ware und
4. die Reklamation eines verloren gegangenen Pakets.

Diese Geschäftsprozesse initiiert jeweils ein Kunde des fiktiven Online-Shops. Die Probanden modellieren diese Geschäftsprozesse immer in derselben zuvor genannten Reihenfolge.

### **5.3. Probanden**

Als Probanden wurden 16 für die Geschäftsprozessmodellierung qualifizierte Informatik-Studenten der Hochschule für Angewandte Wissenschaften Hamburg ausgewählt. 14 Probanden waren Studenten des Master-Studiengangs Informatik. Zwei Probanden waren Studenten des Bachelor-Studiengangs Angewandte Informatik. Alle Master-Studenten waren für die Entwickler-Rolle qualifiziert, da die Modellierung von Geschäftsprozessen ein Studieninhalt des Master-Studiengangs Informatik ist. Die beiden Bachelor-Studenten hatten keine Kenntnisse in der Modellierung von Geschäftsprozessen. Sie waren aber für die Rolle der Fachexperten qualifiziert.

### **5.4. Testumgebungen**

Die Modellierungssitzungen wurden mit einer Kamera aufgezeichnet und in vier verschiedenen Testumgebungen durchgeführt. Die Testumgebungen wurden in Laboren an der Hochschule für Angewandte Wissenschaften Hamburg eingerichtet. Sie unterscheiden sich im wesentlichen durch die Geräte, welche die Probanden bei der Modellierung verwenden können.

Zwei Testumgebungen bestehen aus Multitouch-Geräten: einem Multitouch-Bildschirm und einem Multitouch-Tisch. Diese beiden Testumgebungen dienen dem Vergleich der beiden Multitouch-Gerätetypen mit horizontaler und vertikaler Arbeitsfläche. Die dritte Testumgebung dient dem Vergleich von Multitouch-Geräten mit einem klassischen Computer. Hier kommen ein Notebook sowie ein Beamer zum Einsatz. Die vierte Testumgebung besteht aus einem Whiteboard. Diese Testumgebung ermöglicht, computerbasierte Arbeitsumgebungen mit einer Arbeitsumgebung ohne Computerunterstützung zu vergleichen.

Die vier Testumgebungen erlauben, die Eignung der verwendeten Geräte für die Geschäftsprozessmodellierung zu bewerten und miteinander zu vergleichen. Die Testumgebungen, bei denen ein Computer zum Einsatz kommt, erlauben außerdem die Evaluierung der Modellierungsanwendung.

Bei jeder Modellierungssitzung zeichnete eine digitale Kamera die Aktivitäten und die Gespräche der Probanden auf. Bei den Testumgebungen mit einem Computer speicherte außerdem die Modellierungsanwendung alle 10 Sekunden eine Kopie der Geschäftsprozessmodelle ab.

#### **5.4.1. Multitouch-Bildschirm**

Die erste Testumgebung mit einem Multitouch-Gerät besteht aus einem vertikal aufgebauten Multitouch-Bildschirm, der an einen Computer angeschlossen ist. Die Probanden können diesen Computer neben der Multitouch-Eingabe auch mit einer Maus und einer Tastatur bedienen.

Die Tastatur befindet sich auf einem Tisch seitlich vor dem Multitouch-Bildschirm. An diesem Tisch kann eine Person auf einem Stuhl platz nehmen, um Text einzugeben. Für die übrigen Teilnehmer der Sitzung sind ebenfalls Stühle vorhanden. Den Probanden ist aber freigestellt, ob sie stehen oder sitzen möchten. Alle Stühle sind zu Beginn einer Sitzung halbkreisförmig um den Bildschirm angeordnet (siehe schematische Abbildung 5.1).

Auf dem Computer wird während der Sitzungen die im Rahmen dieser Arbeit entwickelte Modellierungsanwendung ausgeführt. Bei dieser Testumgebung kommt die speziell für Computer mit vertikalen Multitouch-Bildschirmen angepasste Variante zum Einsatz. Die Benutzungsoberfläche dieser Anwendung ist sowohl für die Multitouch-Eingabe als auch für die Maus-Eingabe und Tastatur-Eingabe geeignet. Die Werkzeuge, welche die Probanden während der Modellierung verwenden können, zeigt die Benutzungsoberfläche am rechten Rand Bildschirms an.



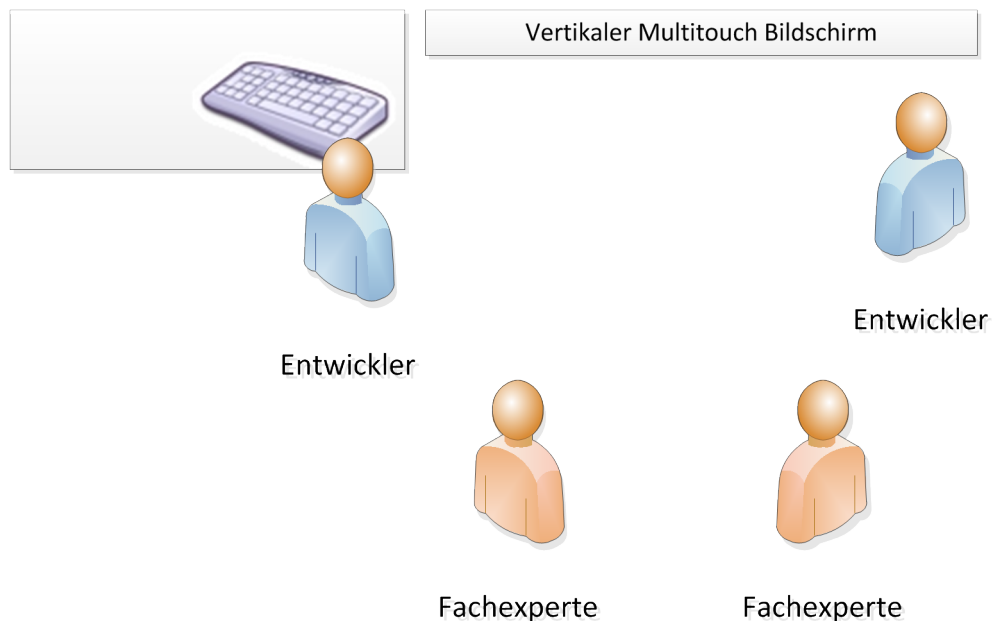


Abbildung 5.1.: Testumgebung mit vertikalem Multitouch-Bildschirm

#### 5.4.2. Multitouch-Tisch (Microsoft Surface)

Bei der zweiten Testumgebung mit einem Multitouch-Gerät kommt ein Microsoft Surface zum Einsatz. Diesen Multitouch-Tisch können die Probanden per Multitouch-Eingabe sowie mit physischen Gegenständen bedienen.

Die physischen Gegenstände verkörpern spezielle Werkzeuge. Die Benutzer können sie bei Bedarf auf das Surface stellen, um sie zu benutzen. Bei dieser Usability-Studie stehen den Probanden die Werkzeuge Papierkorb, Modellelement-Palette, Simulation und Zoom zur Verfügung.

Text können die Probanden bei dieser Testumgebung mit einer Bildschirmtastatur eingeben. Die Modellierungsanwendung zeigt die Bildschirmtastatur automatisch an, wenn ein Benutzer ein Textelement auswählt. Eine physische Tastatur wird den Probanden nicht zur Verfügung gestellt.

Die vier Probanden sitzen bei dieser Testumgebung um den Multitouch-Tisch herum. Zu Beginn einer Sitzung sitzen die Entwickler nebeneinander auf der einen Breitseite des Tisches und die Fachexperten sitzen auf der gegenüberliegenden Seite (siehe schematische Abbildung 5.2). Den Probanden ist freigestellt, die Sitzordnung während der Sitzung zu ändern.

Die Modellierungsanwendung kommt bei dieser Testumgebung in der speziell für das Microsoft Surface angepassten Variante zum Einsatz. Im Gegensatz zu der bei den anderen Test-

umgebungen zum Einsatz kommenden Variante zeigt die Benutzungsoberfläche hier eine Kontrollleiste auf beiden Längsseiten des Bildschirms an. Ebenso zeigt die Benutzungsoberfläche hier die Beschriftung der Diagrammelemente doppelt an. Die Beschriftungen werden jeweils zu beiden Längsseiten des Tisches ausgerichtet. Außerdem hat die Benutzungsoberfläche hier keine ständig angezeigte Werkzeugleiste. Stattdessen können die Benutzer die physischen Gegenstände benutzen.

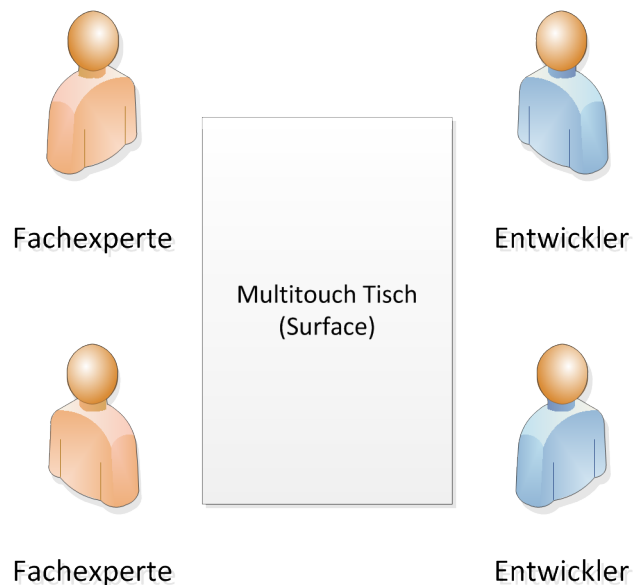


Abbildung 5.2.: Testumgebung mit einem Multitouch-Tisch (Microsoft Surface)

### 5.4.3. Notebook und Beamer

Die Testumgebung mit einem an ein Notebook angeschlossenen Beamer repräsentiert eine Arbeitsumgebung, bei der Entwickler Geschäftsprozesse mit einem herkömmlichen Computer modellieren.

Auf dem Notebook wird die selbe Modellierungsanwendung ausgeführt, wie beim Multitouch-Bildschirm. Die Anwendung kann ein Proband, der die Entwickler-Rolle einnimmt, mit einer Maus und der Tastatur des Notebooks bedienen. Die Touch-Eingabe unterstützt das Notebook nicht. Die Anzeige des Notebook-Bildschirms projiziert der Beamer auf eine Projektionsleinwand, so dass die alle Teilnehmer einer Sitzung das Modell betrachten können.

Auf der linken Seite der Projektionsleinwand befindet sich ein Tisch, an dem die Fachexperten Platz nehmen. Gegenüber, auf der rechten Seite der Projektionsleinwand, befindet

sich ein Tisch, auf dem das Notebook steht und an dem die Entwickler Platz nehmen (siehe schematische Abbildung 5.3).

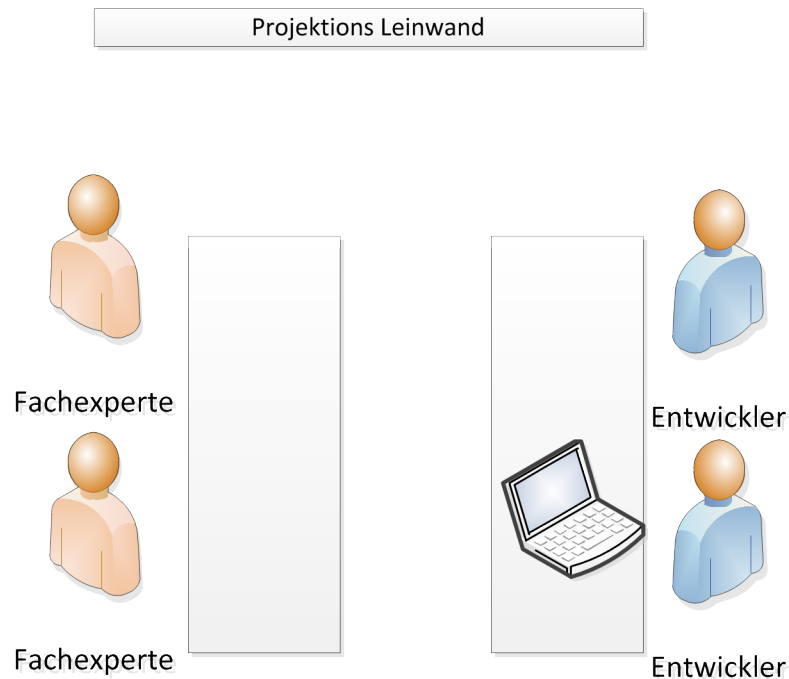


Abbildung 5.3.: Testumgebung mit Notebook und Beamer

#### 5.4.4. Whiteboard

Bei der einzigen Testumgebung, bei der kein Computer zum Einsatz kommt, modellieren die Probanden den Geschäftsprozess auf einem Whiteboard. Auf dieser Tafel können die Probanden das Geschäftsprozessmodell mit speziellen farbigen Stiften zeichnen. Die Zeichnungen können die Probanden mit einem speziellen Schwamm wegwischen.

Während der Sitzung können die Probanden so, wie sie möchten, stehen oder auf Stühlen um die Tafel herum sitzen. Für die Entwickler ist dabei ein Platz an der Tafel vorgesehen (siehe schematische Abbildung 5.4).

#### 5.4.5. Reihenfolge der Testumgebungen bei den Testläufen

Um den Lerneffekt der Probanden während eines Testlaufs auszugleichen, wurden die Testumgebungen in verschiedenen Reihenfolgen verwendet. Die Reihenfolge ist so gewählt,

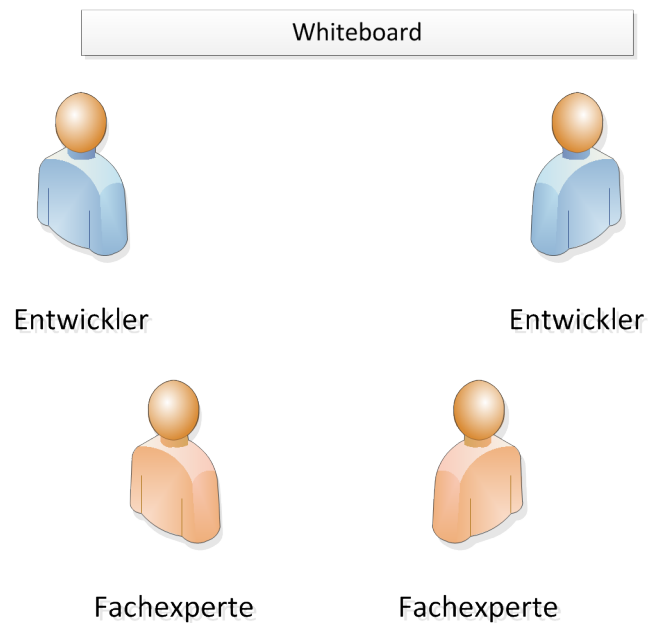


Abbildung 5.4.: Testumgebung mit einem Whiteboard

dass jede Testumgebung je einmal bei der ersten, zweiten, dritten und vierten Sitzung eines Testlaufs verwendet wird:

- 1. Testlauf:** Surface - Multitouch-Bildschirm - Beamer - Whiteboard
- 2. Testlauf:** Whiteboard - Beamer - Multitouch-Bildschirm - Surface
- 3. Testlauf:** Multitouch-Bildschirm - Whiteboard - Surface - Beamer
- 4. Testlauf:** Beamer - Surface - Whiteboard - Multitouch-Bildschirm

## 6. Testauswertung

In diesem Kapitel werden die bei dem Usability-Test gemachten Beobachtungen erläutert. Dabei stehen die Beobachtungen im Vordergrund, die für das Testziel (siehe Kapitel 5.1) von Bedeutung sind.

### 6.1. Charakterisierung der Modellierungssitzungen nach Testumgebung

Der Ablauf aller Sitzungen war inhaltlich durch drei Abschnitte charakterisiert: einer Einleitung, einem Hauptteil sowie einer abschließenden Überprüfung und Zusammenfassung des erstellten Modells. Während der Einleitung fassten die Fachexperten die Abläufe des zu modellierenden Geschäftsprozesses zusammen. Bei einem der Testläufe fasste einer der Fachexperten den zu modellierenden Geschäftsprozess zu Beginn der Sitzungen relativ ausführlich zusammen. Diese Zusammenfassungen dauerten rund ein bis eineinhalb Minuten. Die Entwickler der anderen Probanden-Gruppen bevorzugten, frühzeitig mit der Modellierung zu beginnen. Die Fachexperten fassten hier den Geschäftsprozess am Anfang einer Sitzung nur kurz innerhalb maximal 30 Sekunden zusammen. Im anschließenden Hauptteil erstellten die Entwickler gemäß der Erläuterung der Fachexperten ein Geschäftsprozessmodell. Dieser Abschnitt beanspruchte bei allen Sitzungen die größte Zeit der Sitzung. Abschließend spielten die Entwickler den Geschäftsprozess anhand des Modells exemplarisch durch und fassten das Modell zusammen. Die Fachexperten gaben währenddessen Rückmeldungen, ob sie einverstanden waren. In wenigen Fällen fanden die Probanden Fehler in den Modellen. Die Entwickler korrigierten diese Fehler und spielten das Modell anschließend noch einmal durch. Der abschließende Teil dauerte je nach der Komplexität des Modells und je nachdem, ob die Entwickler das Modell noch korrigiert haben, etwa ein bis zwei Minuten.

Die Modellierungssitzungen liefen je nach Testumgebung nach bestimmten Mustern ab. Die Abläufe lassen sich mit unterschiedlichen Phasen charakterisieren, die wiederholt vorkamen. Eine solche Phase kann anhand der Gespräche und der Modellierungstätigkeiten klassifiziert werden.

Die Gespräche können daran unterschieden werden, wer spricht:

- Dialoge zwischen den Entwicklern und den Fachexperten.
- Dialoge der Entwickler untereinander, bei denen die Fachexperten nicht einbezogen wurden.
- Dialoge der Fachexperten untereinander, bei denen die Entwickler nicht einbezogen wurden.
- Monologe der Entwickler.
- Monologe der Fachexperten.
- Sprechpausen.

Die Dialoge bestanden aus wechselseitiger Rede und Gegenrede der beteiligten Personen. Die Dialoge zwischen den Entwicklern und den Fachexperten verliefen meistens mit unausgewogenen Sprechanteilen. Die Entwickler hatten größere Anteile, wenn sie ihre Fragen oder das Modell erläuterten. Die Fachexperten hatten größere Anteile am Gespräch, wenn sie den Entwicklern den Ablauf des Geschäftsprozesses erläuterten.

Bei der Einteilung der Modellierungstätigkeiten ist zunächst nur relevant, ob das Modell verändert wurde oder nicht:

- Bearbeitung des Modells durch Hinzufügen, Löschen, oder Verschieben von Elementen sowie Beschriften von Elementen.
- Keine Modellierungstätigkeiten: das Modell wurde nicht verändert.

Die genannten Gesprächsphasen und Modellierungsphasen traten in allen möglichen Kombinationen auf. Das heißt, während der Gespräche wurde zum Teil gleichzeitig modelliert und zum Teil nicht. Während modelliert wurde, gab es sowohl Sprechpausen als auch Gespräche. Bei den verschiedenen Testumgebungen traten die Phasen jedoch unterschiedlich häufig auf.

Dialoge zwischen den Entwicklern und den Fachexperten gab es beim Surface häufiger als bei den anderen Testumgebungen.

### 6.1.1. Notebook und Beamer

Der Ablauf der Sitzungen, die mit einem an ein Notebook angeschlossenen Beamer durchgeführt wurden, wurde stark dadurch geprägt, dass jeweils nur ein Entwickler die Geschäftsprozessmodelle am Notebook modellierte. Dialoge zwischen den Entwicklern und Fachexperten gab es relativ selten. Außerdem bezogen die Entwickler die Fachexperten phasenweise nicht mit in den Modellierungsprozess ein.

Während der Sitzungen herrschte zwischen den Entwicklern und Fachexperten eine überwiegend distanzierte Atmosphäre. Die Fachexperten nahmen eine passive Rolle ein und sprachen in der Regel nur dann mit den Entwicklern über den Geschäftsprozess, wenn die Entwickler explizit nachfragten. Nur selten meldeten sich die Fachexperten aus eigenem Antrieb zu Wort. Meistens widersprachen sie dann den von den Entwicklern modellierten Abläufen. Während der restlichen Zeit agierten die Entwickler und Fachexperten völlig unabhängig voneinander. Die Fachexperten blickten die meiste Zeit zur Projektionsleinwand und unterhielten sich ab und zu leise untereinander. Währenddessen modellierte einer der Entwickler mit dem Notebook den Geschäftsprozess. Der zweite Entwickler griff nicht aktiv in die Modellierung ein. Er gab dem ersten Entwickler ab und zu Hinweise und diskutierte mit ihm Modellierungsdetails. Die Entwickler sprachen dabei relativ leise untereinander, was die Fachexperten davon abhielt, an der Diskussion teilzunehmen.

Bei zwei der vier Sitzungen interviewte hauptsächlich der zweite Entwickler die Fachexperten und der Entwickler am Notebook konzentrierte sich auf die Bearbeitung des Modells. Bei den anderen beiden Sitzungen führte hauptsächlich der Entwickler am Notebook die Gespräche mit den Fachexperten, während der zweite Entwickler an den Gesprächen überwiegend unbeteiligt blieb. Häufig diskutierten die Entwickler aber zunächst untereinander, bevor sie mit den Fachexperten sprachen.

Einen längeren Dialog zwischen den Entwicklern und Fachexperten gab es nur selten. Oftmals endeten Dialoge frühzeitig mit einem Monolog eines Fachexperten. So trug ein Fachexperte Details des Geschäftsprozesses vor, während die Entwickler zunächst zuhörten und Zwischenfragen stellten. Die Entwickler widmeten aber immer wieder schnell ihre Aufmerksamkeit der Modellierung am Notebook und gaben dem Fachexperten weder eine Rückmeldung, ob sie ihm folgen konnten, noch, dass sie ihm zuhörten. Der Fachexperte registrierte dies und beendete daraufhin, häufig mit immer leiserer Stimme sprechend, seine Erläuterungen.

Die offensichtliche Barriere zwischen den Entwicklern und Fachexperten war eine Folge der Sitzordnung. Die Entwickler saßen an einem Tisch auf der rechten Seite der Projektionsleinwand und die Fachexperten saßen an einem Tisch auf der linken Seite. Sie saßen dadurch so weit voneinander entfernt, dass sie ungestört jeweils untereinander reden konnten. Außerdem saßen die Fachexperten zur Projektionsleinwand gewendet und blickten selten zu den Entwicklern. Zusätzlich saß der Entwickler hinter dem aufgeklappten Display des Notebooks versteckt, was auf die Fachexperten psychologisch wie eine Mauer wirkte.

### **6.1.2. Multitouch-Bildschirm**

Bei den Sitzungen am Multitouch-Bildschirm teilten die Entwickler die Modellierungstätigkeiten untereinander auf. Ein Entwickler bediente die Tastatur und der andere Entwickler

bediente den Multitouch-Bildschirm. Wie auch beim Beamer waren die Fachexperten nicht jederzeit an dem Modellierungsprozess beteiligt.

Während der Sitzungen stand jederzeit einer der Entwickler am Multitouch-Bildschirm und der andere saß an dem Tisch, auf dem sich die Tastatur befand. Nur während des dritten Testlaufs tauschten die Entwickler einmal die Geräte. Aber zu keinem Zeitpunkt bedienten mehrere Personen gleichzeitig den Multitouch-Bildschirm. Ebenso bediente niemals einer der Fachexperten den Multitouch-Bildschirm oder die Tastatur.

Der am Multitouch-Bildschirm stehende Entwickler fügte dem Modell neue Elemente hinzu, änderte ihre Position und Größe oder entfernte Elemente aus dem Modell. Er wählte außerdem einzelne Elemente mit einer Double-Tap Geste aus, damit der andere Entwickler die Beschriftung dieser Elemente ändern konnte. Entweder erkannte der am Bildschirm stehende Entwickler, wann und welche Elemente beschriftet werden mussten, oder der Entwickler an der Tastatur forderte ihn dazu auf. In wenigen Fällen stand der schreibende Entwickler auf oder benutzte die Maus, um ein zu beschriftendes Element selbst auszuwählen. In solchen Momenten sprach der andere Entwickler mit den Fachexperten.

Nur in wenigen Fällen waren beide Entwickler gleichzeitig an der Modellierung beteiligt. Während der eine Entwickler mit der Tastatur schrieb, wartete der andere fast immer, bis der schreibende Entwickler die Texteingabe durch Drücken der Enter-Taste beendet hatte.

Während einer der Entwickler das Modell bearbeitete, stand er meistens direkt vor dem Bildschirm. Dadurch waren oft große Abschnitte des Modells für die Fachexperten mehrere Sekunden bis über eine Minute nicht sichtbar, so dass sie den Verlauf der Modellierung nicht jederzeit verfolgen konnten. Infolgedessen kommentierten die Fachexperten das Modell nur während Modellierungspausen, während der der am Bildschirm stehende Entwickler zur Seite trat. In einigen Fällen forderten ihn die Fachexperten explizit dazu auf.

Solange die Entwickler mit der Modellierung beschäftigt waren, kehrten sie den Fachexperten den Rücken zu. In diesen Situationen gab es nur selten Diskussionen zwischen den Entwicklern und den Fachexperten. Ähnlich wie auch bei den mit dem Beamer sowie den am Whiteboard durchgeführten Sitzungen, brachen die Diskussionen dann relativ schnell ab. Häufig endeten sie dadurch, dass ein Fachexperte seine Erläuterungen immer leiser sprechend abbrach. Während die Entwickler den Fachexperten den Rücken zuekehrten, führten die Fachexperten und insbesondere die Entwickler wiederholt Gespräche jeweils untereinander. Sie sprachen dabei leiser, als wenn die Entwickler und Fachexperten miteinander sprachen.



### 6.1.3. Multitouch-Tisch (Microsoft Surface)

Die Sitzungen am Surface waren überwiegend durch Dialoge zwischen den Entwicklern und Fachexperten sowie einer schrittweisen Modellierung geprägt.

Die Modellierung fand in inhaltlich abgeschlossenen Abschnitten statt, während der die Entwickler einzelne, aus wenigen Aktivitäten bestehenden Sequenzen modelliert haben. Diese Abschnitte waren durch kurze Modellierungspausen abgegrenzt, während der die Fachexperten die anschließenden Abläufe des Geschäftsprozesses zu erklären begannen. Während der Erläuterungen der Fachexperten äußerten die Entwickler regelmäßig ihre Zustimmung oder sie fragten bei Unklarheiten nach.

Die Entwickler begannen meistens noch während der Ausführungen der Fachexperten das Modell zu bearbeiten. Wenn die Entwickler das Modell nicht schnell genug anpassen konnten, passten die Fachexperten das Tempo ihrer Erläuterungen an. Sie unterbrachen ihre Erläuterungen kurzzeitig und meistens aus eigenem Antrieb. Die Entwickler forderten sie nur in wenigen Fällen dazu auf. Sobald die Entwickler alle Änderungen abgeschlossen hatten, fuhren die Fachexperten mit ihren Erläuterungen fort. Die Entwickler signalisierten den Fachexperten regelmäßig verbal oder nonverbal, dass sie fortfahren können. Sie suchten dazu Blickkontakt, machten Gesten (z. B. weit ausholendes und heftiges Drücken auf die Enter-Taste der Bildschirmtastatur), äußerten, dass sie fertig sind (z. B. "So!""), oder fragten, wie es weiter geht (z. B. "Und jetzt?"). Häufig erkannten die Fachexperten aber auch selbst, wann sie fortsetzen konnten. Die Arbeitsfläche war für die Fachexperten niemals verdeckt, so dass sie jederzeit den Modellierungsfortschritt verfolgen konnten.

Abgesehen von der Texteingabe bei der Testumgebung mit dem Multitouch-Bildschirm, war die Testumgebung mit dem Microsoft Surface die einzige, bei der regelmäßig mehrere Personen das Gerät gleichzeitig bedienen haben. Die Entwickler haben häufig gleichzeitig Elemente bearbeitet und sich gegenseitig unterstützt. Vielfach hat einer der Entwickler dem Modell neue Elemente hinzugefügt, die der andere gerade benötigte und anschließend mit den vorhandenen Elementen verknüpfte. Ebenso hat mehrfach einer der Entwickler die von dem anderen Entwickler gerade benötigten physischen Objekte auf dem Surface platziert und diesem zugänglich gemacht. Das gemeinsame Arbeiten wurde häufig unterbrochen, wenn einer der Entwickler Modellelemente beschriftete. Gründe dafür waren, dass das Surface keine parallele Texteingabe unterstützt und die Bildschirmtastatur einen großen Teil der Bildschirmfläche verdeckt.

Diese Testumgebung war ebenfalls die einzige, bei der auch die Fachexperten das Gerät bedienen haben. Mit fortschreitender Sitzungsdauer griffen sie vereinzelt mit in die Modellierung ein, um den Entwicklern die Abläufe des Geschäftsprozesses zu erklären. Am Ende der Sitzungen beteiligten sich die Fachexperten auch rege am Durchspielen des Modells. Im Laufe

der Sitzungen erlernten sie offensichtlich die Bedienung der Modellierungsanwendung sowie die Modellierungssprache, und ihre Hemmschwelle, die Anwendung zu bedienen, sank.

#### **6.1.4. Whiteboard**

Bei den Sitzungen am Whiteboard haben die Entwickler bis auf eine Ausnahme nicht gleichzeitig modelliert. Nur einmal während des vierten Testlaufs haben sie gleichzeitig das Modell bearbeitet, als einer der Entwickler eine Aktivität um ein Service-Task-Symbol ergänzt hatte.

Während der eine Entwickler das Geschäftsprozessmodell am Whiteboard skizzierte, drehte er für längere Zeit den Fachexperten den Rücken zu. Dabei sprach er deutlich häufiger mit dem anderen Entwickler, als mit den Fachexperten. Der Entwickler, der gerade nicht modellierte, stand meistens neben dem Whiteboard, schaute dem modellierenden Entwickler zu und befragte die Fachexperten. Wie auch beim Beamer und Multitouch-Bildschirm sprachen die Entwickler und Fachexperten auch vereinzelt jeweils untereinander. Insgesamt bezogen die Entwickler die Fachexperten aber häufiger mit in die Diskussionen ein.

Die Entwickler haben am Whiteboard einzelne Modellelemente mit einem Stift schneller gezeichnet und beschriftet, als sie Modellelemente mit der Modellierungsanwendung erstellt und beschriftet haben. Trotzdem haben beim Whiteboard die Abschnitte, während der eine Entwickler ununterbrochen und ohne Rückfragen modelliert hat, im Schnitt etwas länger gedauert, als bei den anderen Testumgebungen. Dies resultierte daraus, dass der gerade modellierende Entwickler relativ umfangreiche Abschnitte eines Modells entwarf, ohne die anderen Personen in die Modellierung einzubeziehen. Allerdings führte dieses Vorgehen wiederholt dazu, dass die Probanden größere Abschnitte eines Modells komplett verwarfen. Die Entwickler konnten zwar ihre Gedanken schnell skizzieren, aber ihre Gedanken waren häufig noch unausgereift. Bei den anderen Testumgebungen diskutierten die Probanden ihre Gedanken meistens, bevor sie diese ins Modell umsetzten. Dort haben sie nur einzelne Elemente oder kurze Abschnitte gelöscht.

#### **6.1.5. Probandenspezifische Beobachtungen**

Eine Gruppe von Probanden ging bei der Modellierung besonders strukturiert vor. Einer der Entwickler übernahm die Leitung des Gesprächs und bestimmte die Reihenfolge, in welcher die Probanden die Aktivitäten und Sequenzen des Geschäftsprozesses modellierten. Beim Surface fasste regelmäßig einer der Entwickler den zuletzt modellierten Abschnitt des Geschäftsprozesses zusammen. Bei den anderen Testumgebungen fassten die Entwickler

die einzelne Abschnitte hingegen unregelmäßiger zusammen. Außerdem waren hier die zusammengefassten Abschnitte größer. Die Entwickler der anderen Gruppen hatten weniger Praxiserfahrungen in der Modellierung sowie Durchführung von Sitzungen. Diese Gruppen gingen deshalb weniger strukturiert vor.

## 6.2. Auswertung der Awareness der Probanden

In diesem Abschnitt wird die Awareness der Probanden während der Modellierungssitzungen untersucht. Dazu werden zunächst die beobachteten positiven und negativen Awareness Indikatoren sowie die beobachteten Indikatoren für Awareness Work zusammengefasst (Awareness Indikatoren: siehe Kapitel 2.5.4). Es wird sowohl die Awareness der Probanden über die Modellierungstätigkeiten der anderen Probanden als auch die Awareness während der Gespräche ermittelt. Anschließend wird anhand dieser Ergebnisse die Awareness bei den einzelnen Testumgebungen bewertet.

### 6.2.1. Positive Awareness-Indikatoren

#### 6.2.1.1. Unaufgeforderte Reaktionen der Fachexperten auf die Aktionen der Entwickler

Bei jeder Sitzung haben die Fachexperten auf die Aktionen der Entwickler unaufgefordert reagiert. Bei den Sitzungen am Surface war dieser Indikator für Awareness am häufigsten zu beobachten und bei den Sitzungen mit dem Multitouch-Bildschirm und dem Whiteboard am seltensten.

Es kam während jeder Sitzung vor, dass die Fachexperten nach einer Pause mit ihren Erklärungen fortfuhren, ohne dass die Entwickler sie dazu aufforderten. Die Fachexperten erkannten, wann die Entwickler die zuvor besprochenen Details modelliert hatten und, ob die Entwickler weitere Informationen bei der Modellierung benötigten. Dieser positive Awareness Indikator war bei der Testumgebung mit dem Surface häufig und regelmäßig zu beobachten. Es kam hier seltener vor, dass die Entwickler die Fachexperten explizit dazu auffordern mussten, mit ihren Erläuterungen fortzufahren, als dass die Fachexperten von sich aus fortfuhren. Bei den anderen Testumgebungen war dieses Verhältnis umgekehrt. Hier fuhren die Fachexperten weitaus überwiegend erst dann fort, den Geschäftsprozess zu erläutern, nachdem die Entwickler sie dazu verbal aufgefordert hatten. Dies bedeutet aber nicht zwangsläufig ein Nichtvorhandensein an Awareness.

Bei der Testumgebung mit dem Notebook und dem Beamer setzten die Fachexperten meistens prompt fort, den Geschäftsprozess zu erklären. Sie waren zwar nicht jeder einzelnen

Handlung der Entwickler gewahr, so dass sie den Modellierungsfortschritt nicht im Detail verstanden. Den Überblick über den Modellierungsfortschritt haben sie aber trotzdem behalten, und sie waren zumindest über den aktuellen Arbeitsschritt bei der Modellierung gewahr. Die Fachexperten hatten hier somit eine Awareness über Handlungen von relativ grober Granularität.

Beim Multitouch-Bildschirm und beim Whiteboard fuhren die Fachexperten hingegen häufig verzögert, nach einer Gedankenpause, fort. Oft überbrückten sie die Gedankenpausen mit Verzögerungslauten wie zum Beispiel einem "Ähm". Dieses Verhalten der Fachexperten deutet auf ein geringes Gewahrsein der Fachexperten über die Aktivitäten der Entwickler hin.

Bei den Testumgebungen mit dem Multitouch-Bildschirm und dem Whiteboard hat die eingeschränkte Sicht der Fachexperten auf das Modell die Awareness der Fachexperten beeinträchtigt. Wenn ein Entwickler bei der Modellierung das Modell und den von ihm gerade bearbeiteten Bereich verdeckte, konnten die Fachexperten die Änderungen nicht mitverfolgen. Erst wenn der Entwickler beiseite trat, konnten die Fachexperten das vollständige Modell und die vom Entwickler vorgenommenen Änderungen wieder betrachten. Nachvollziehen konnten sie die Änderungen dann aber nur erschwert.

Bei der Testumgebung mit dem Notebook und dem Beamer hatten die Fachexperten dagegen jederzeit freie Sicht auf die Projektionsfläche. Hier hemmte die indirekte und häufig schnelle Eingabe der Entwickler mit Maus und Tastatur die Awareness der Fachexperten.

#### **6.2.1.2. Reaktion der Entwickler ohne explizite Aufforderung durch andere Probanden**

Die positive Awareness Indikator "Reaktion ohne explizite Aufforderung" wurde bei den Aktionen der Entwickler unregelmäßig und seltener beobachtet als, wie zuvor beschrieben, bei den Fachexperten. Die Awareness der Entwickler äußerte sich viel mehr durch die anschließend beschriebene Arbeitsteilung.

Beim Surface konnten Reaktionen der Entwickler ohne explizite Aufforderung in folgenden Fällen beobachtet werden:

- Ein Entwickler zeigt auf die Vorlage in der Modellelement-Palette, welche der andere Entwickler sucht.
- Ein Entwickler sieht das Zögern des anderen Entwicklers und sagt ihm, welche Vorlage aus der Modellelement-Palette er nehmen soll.
- Ein Entwickler diktiert, was der andere schreiben soll.
- Ein Entwickler schließt die Tastatur, weil der andere das ausgewählte Element löschen möchte.

Bei der Testumgebung mit dem Whiteboard wurden unaufgeforderte Reaktionen der Entwickler seltener beobachtet. Solche Reaktionen waren hier die Übergabe eines Stifts, als der Stift eines der Entwickler nicht schrieb, und das gelegentliche Beginnen der Modellierung, während der andere Entwickler die Fachexperten interviewte.

Beim Beamer und beim Multitouch-Bildschirm war dieser Awareness Indikator bei den Aktivitäten der Entwickler nicht zu beobachten.

### 6.2.1.3. Arbeitsteilung durch parallele und komplementäre Aktionen ohne verbale Koordination

Bei der Testumgebung mit dem Multitouch-Tisch haben die Entwickler ihre Modellierungstätigkeiten sehr häufig untereinander aufgeteilt. Vereinzelt haben auch Entwickler und Fachexperten gemeinsam modelliert. Beim Multitouch-Bildschirm sowie beim Whiteboard haben die Entwickler deutlich seltener ihre Arbeit aufgeteilt. Meistens hat nur einer von ihnen modelliert, ohne dass ihn der andere Fachexperte dabei aktiv unterstützte. Die Fachexperten haben beim Multitouch-Bildschirm und beim Whiteboard niemals an der Modellierung aktiv teilgenommen, geschweige denn mit den Fachexperten gemeinsam modelliert. Bei den Modellierungssitzungen mit dem Notebook gab es erwartungsgemäß gar keine Arbeitsteilung, da nur ein Fachexperte das Notebook bedienen konnte.

**Arbeitsteilung am Surface** Am Surface haben die Testpersonen regelmäßig die Modellierungstätigkeiten untereinander aufgeteilt, indem sie sich abwechselnd mit aufeinander aufbauenden Tätigkeiten ergänzt haben. Sie haben dabei ihre Aktivitäten kaum verbal koordiniert. In der Regel haben sie unaufgefordert komplementäre Tätigkeiten durchgeführt, die nahtlos aneinander anschlossen. Folgende abwechselnd und ohne verbale Absprache durchgeführte Tätigkeiten haben die Arbeitsteilung der Testpersonen maßgeblich charakterisiert:

- Eine Person wählt ein Element zum Beschriften aus, woraufhin die Bildschirmtastatur angezeigt wird und eine andere Person zu Schreiben beginnt.
- Nachdem eine Person die Bildschirmtastatur geschlossen, fährt eine andere mit der Modellierung fort.
- Eine Person fügt dem Modell einen Flow Node hinzu, woraufhin eine andere es mit einem Sequence Flow verbindet.
- Eine Person fügt dem Modell einen Flow Node hinzu, woraufhin eine andere ihn beschriftet.

- Eine Person beendet die Bearbeitung eines Modellelements (Beschriften, Vergrößern/-Verkleinern oder Verschieben), woraufhin eine andere Person dem Modell neue Elemente hinzufügt, die mit dem zuvor bearbeiteten Element in Beziehung stehen.
- Eine Person verbessert ein Element (Position, Beschriftung, Größe), welches eine andere Person unmittelbar zuvor bearbeitet hat.
- Eine Person verschiebt das Modell, so dass eine andere Person genug Platz zum Hinzufügen neuer Modellelemente hat.
- Eine Person verschiebt ein Werkzeug, so dass eine andere Person genug Platz zum Hinzufügen neuer Modellelemente hat.
- Eine Person verschiebt ein Werkzeug, so dass es den von einer anderen Person bearbeiteten Bereich des Modells nicht verdeckt.
- Eine Person richtet die Modellelement-Palette so aus, dass eine andere Person dieses Werkzeug bedienen kann.

Gelegentlich haben die Testpersonen auch gleichzeitig modelliert. Sie haben aber häufiger sequentiell als parallel gearbeitet. Auch wenn die Testpersonen parallel gearbeitet haben, haben sie ihre Tätigkeiten nur selten verbal koordiniert. Charakteristisch für diese Arbeitsteilung waren folgende Tätigkeiten:

- Eine Person beschriftet ein Element. Währenddessen verschiebt oder vergrößert eine andere Person dieses Element, so dass der Text vollständig angezeigt wird.
- Während eine Person ein Element beschriftet, bearbeitet eine andere Person andere Elemente des Modells. Zum Beispiel fügt Letztere neue Elemente hinzu, verbindet Flow Nodes mit Sequence Flows oder ändert die Größe und Position von Elementen.
- Zwei Personen ziehen gleichzeitig neue Elemente aus der Modellelement-Palette in das Modell.
- Mehrere Personen verbinden gleichzeitig mehrere Flow Nodes mit unterschiedlichen Sequence Flows.
- Zwei Personen verbinden gleichzeitig zwei Flow Nodes mit demselben Sequence Flow, indem jede ein Ende des Sequence Flows auf einen der Flow Nodes zieht.
- Zwei Personen tauschen ein Element gegen ein anderes aus (z. B. User Task gegen Service Task), indem einer das alte Element in den Papierkorb schiebt während der andere ein neues Element aus der Modellelement-Palette zieht.

Mit den zuvor beschriebenen Tätigkeiten haben die Testpersonen die Modellierungsarbeit häufig aufgeteilt. Bis auf einige Einzelfälle haben jedoch nur die Entwickler modelliert. Bei der Simulation von Geschäftsprozessen gab es deutlich weniger Arbeitsteilung. Insbesondere haben die Testpersonen dann kaum parallel Tätigkeiten durchgeführt. Sie haben die Arbeit während einer Simulation fasst ausschließlich durch abwechselndes Aktivieren von Elementen aufgeteilt. Allerdings haben während der Simulationsphasen auch die Fachexperten die Anwendung regelmäßig bedient.

**Arbeitsteilung am Multitouch-Bildschirm** Bei den Modellierungssitzungen mit dem Multitouch-Bildschirm war die Arbeitsteilung durch die Texteingabe geprägt. Häufig hat der am Bildschirm stehende Entwickler ein Element zum Beschriften ausgewählt, woraufhin der zweite Entwickler unaufgefordert das Element mit der Tastatur beschriftet hat. Sobald das Element beschriftet war, fuhr der erste Entwickler in der Regel mit der Modellierung fort. In wenigen Fällen modellierte auch der eine Entwickler, während der andere noch schrieb. Zum Beispiel fügte er neue Flow Nodes hinzu oder verband Flow Nodes mit Sequence Flows. Meistens gingen die Tätigkeiten der Entwickler bei der Texteingabe nahtlos ineinander über. Nur in wenigen Fällen haben sie sich abgesprochen.

Anderweitig haben die Entwickler keine Arbeitsteilung durch abwechselndes oder gleichzeitiges Bedienen der Anwendung vorgenommen. Es bediente immer nur ein Entwickler den Multitouch-Bildschirm. Während ein Entwickler am Bildschirm stand und modellierte, hat der andere allerdings gelegentlich die Fachexperten über die Details befragt, die der erste Entwickler gerade modellierte. Dies ist ein Indiz dafür, dass der interviewende Entwickler über die Modellierungsaktivitäten des ersten Entwicklers gewahr war. Der modellierende Entwickler wiederum fragte häufig nach, was die anderen Personen gerade besprochen hatten. Somit hatte dieser Entwickler nur eine geringe Awareness über das Gespräch. Seine Aufmerksamkeit war viel mehr auf die Bearbeitung des Modells fokussiert.

**Arbeitsteilung am Whiteboard** Am Whiteboard hat während der meisten Zeit einer Sitzung ein einzelner Entwickler modelliert. Bei zwei Sitzungen gab es sogar keinerlei Arbeitsteilung. Hier hat immer der selbe Entwickler modelliert. Wenn die Entwickler die Modellierungstätigkeiten aufgeteilt haben, haben sie fasst immer sequentiell gearbeitet. Nur in wenigen Einzelfällen haben die Entwickler gleichzeitig modelliert. Wenn die Entwickler die Modellierungstätigkeiten untereinander aufgeteilt haben, ohne sich verbal abzusprechen, haben sie sich bei folgende Tätigkeiten abgewechselt:

- Ein Entwickler nimmt einen Stift oder den Schwamm und überreicht ihn unaufgefordert dem anderen Entwickler.

- Ein Entwickler zeichnet Modellelemente und merkt, dass sie fehlerhaft sind. Der andere Entwickler wischt sie unaufgefordert weg. Daraufhin fährt der erste Entwickler mit der Modellierung fort.
- Die Entwickler zeichnen gleichzeitig verschiedene neue Modellelemente.
- Ein Entwickler ergänzt eine zuvor vom anderen Entwickler gezeichnete Aktivität mit einem Symbol für den Typ der Aktivität (Service Task / User Task).

Häufiger als bei den anderen Testumgebungen haben sich die Entwickler beim gemeinsamen Modellieren verbal abgesprochen. Beispielsweise hat ein Entwickler den anderen aufgefordert, ihn zeichnen zu lassen oder ihm einen Stift oder den Schwamm zu geben.

#### 6.2.1.4. Übergabe von Objekten ohne verbale Koordination

Die Übergabe von Objekten ohne verbale Koordination ist der positive Indikator für Awareness, der am seltensten beobachtet wurde. Beim Surface gab es etwas häufiger Objektübergaben als beim Whiteboard. Bei den beiden anderen Testumgebungen gab es keine. Während beim Notebook prinzipiell keine Objektübergaben möglich waren, wären sie beim Multitouch-Bildschirm möglich gewesen.

**Übergabe von Objekten beim Surface** Die Übergabe von Objekten lief beim Surface folgendermaßen ab:

- Ein Entwickler wählt ein Element zum Beschriften aus und schiebt die Bildschirmtastatur zu dem zweiten Entwickler. Daraufhin beginnt der zweite Entwickler mit der Bildschirmtastatur zu schreiben.
- Ein Entwickler schiebt ein Element zum zweiten Entwickler hin, worauf der zweite mit diesem Element weiterarbeitet. Zum Beispiel übergibt ein Entwickler dem anderen einen Sequence Flow, damit dieser ihn mit Flow Nodes verbindet.
- Eine Person reicht einer anderen einen neben dem Bildschirm stehenden Gegenstand, der mit einem bestimmten Werkzeug assoziiert ist. Die erste Person kann den Gegenstand leichter erreichen als die zweite.

Die ersten beiden genannten Fälle wurden regelmäßig beobachtet und der dritte vereinzelt. Bei den ersten beiden Fällen haben die Personen ihre Handlungen nur selten verbal koordiniert. Die Übergabe von Werkzeug-Gegenständen haben die Personen in rund der Hälfte der Fälle verbal koordiniert, indem eine Person darum bat, ihr den Gegenstand zu geben.



**Übergabe von Objekten beim Whiteboard** Beim Whiteboard war nur die Übergabe von Stiften und einem Schwamm möglich. Sporadisch hat ein Entwickler dem anderen einen Stift oder den Schwamm gereicht, wenn der erste den Stift beziehungsweise den Schwamm leichter erreichen konnte. In knapp der Hälfte der Fälle haben die Entwickler die Übergabe nicht verbal koordiniert. In den übrigen Fällen hat ein Entwickler den anderen gebeten, ihm den Stift beziehungsweise den Schwamm zu geben.

## 6.2.2. Negative Awareness-Indikatoren

Negative Awareness Indikatoren wurden bei allen Testumgebungen seltener und unregelmäßiger beobachtet als die positiven Awareness-Indikatoren.

### 6.2.2.1. Interference

Die Testpersonen haben bei allen Testumgebungen sporadisch Tätigkeiten durchgeführt, die negative Auswirkungen auf die Tätigkeiten der anderen Personen hatten.

**Interference beim Surface** Bei den Modellierungssitzungen mit dem Surface traten verschiedene Konflikte zwischen den Probanden auf:

- Zwei Personen greifen gleichzeitig zur Bildschirmtastatur, zum selben Werkzeug oder zum selben Modellelement. Die beteiligten Personen lösen den Konflikte aber schnell, indem eine Person ihre Hand zurück zieht.
- Eine Person platziert ein Werkzeug oder die Bildschirmtastatur, so dass die Elemente verdeckt sind, welche eine andere Person gerade bearbeitet. Die beeinträchtigte Person löst den Konflikt, indem sie das Werkzeug beziehungsweise die Bildschirmtastatur verschiebt.
- Eine Person schließt die Bildschirmtastatur, obwohl eine andere Person damit noch schreibt. Aufgrund dessen muss die schreibende Person ihre Arbeit unterbrechen, was sie verbal mit Nachdruck beanstandet.
- Eine Person wählt ein Element zum Beschriften aus und verändert dadurch den Texteingabefokus, obwohl eine zweite Person ein anderes Element noch nicht fertig beschriftet hat. Die zweite Person macht darauf verbal aufmerksam.
- Eine Person dreht die gesamte Arbeitsfläche, während andere Personen Modellelemente berühren und bearbeiten. Die Arbeit der Personen wird dadurch kurzzeitig unterbrochen, und die berührten Elemente werden ungewünscht verschoben.

Die genannten Konflikte traten bis auf die ersten beiden nur als Einzelfälle auf. Die ersten beiden traten mehrfach auf, wobei der zuerst genannte Konflikt mit Abstand am häufigsten auftrat.

**Interference beim Multitouch-Bildschirm** Während der Modellierungssitzungen mit dem Multitouch-Bildschirm stand häufig ein Entwickler vor dem Bildschirm, so dass er partiell das Modell verdeckte. Die Fachexperten konnten dadurch zeitweise den aktuellen Stand eines Modells nicht überblicken. Insbesondere konnten sie nicht verfolgen, was der Entwickler gerade modellierte.

**Interference beim Whiteboard** Wie auch beim Multitouch-Bildschirm versperrte häufig ein vor der Tafel stehender Entwickler den Fachexperten den Blick auf das Modell. Manchmal standen auch beide Entwickler gleichzeitig vor der Tafel. Vereinzelt gab es auch einen Konflikt zwischen den Entwicklern, wenn einer einen Stift oder den Schwamm in der Hand hielt, ohne zu bemerken, dass der andere Entwickler den Stift beziehungsweise den Schwamm benötigte.

#### 6.2.2.2. Verbal Monitoring

Bei den Modellierungssitzungen mit dem Notebook und mit dem Surface war kein Verbal Monitoring zum Koordinieren oder Verstehen von Tätigkeiten zu beobachten. Beim Multitouch-Bildschirm und beim Whiteboard hat in wenigen Fällen ein Proband die anderen Personen über ihre Aktionen befragt, wenn ein Entwickler modellierte und der andere gleichzeitig mit den Fachexperten sprach. Entweder fragte der modellierende Entwickler die anderen Personen, was sie gerade besprochen hätten, oder der zweite Entwickler fragte den modellierenden Entwickler, was er gerade modelliere.

#### 6.2.3. Indikatoren für Awareness Work

Während der Modellierungssitzungen haben die Probanden teilweise versucht, die Awareness der anderen Probanden positiv zu beeinflussen. Meistens verwendeten sie dazu Verbal Shadowing. Übertriebene manuelle Aktionen und sichtbare Haltungsänderungen haben sie selten und unregelmäßig verwendet.

### 6.2.3.1. Verbal Shadowing

Das verbale Beschreiben der eigenen Aktivitäten (Verbal Shadowing) wurde bei allen Testumgebungen beobachtet. Allerdings hatte die Persönlichkeit der Testpersonen einen entscheidenden Einfluss darauf, ob eine Person Verbal Shadowing praktiziert hat oder nicht. Die gesprächigsten Personen haben Verbal Shadowing wiederholt praktiziert. Einige andere Testpersonen wiederum haben nur selten oder nie ihre eigenen Aktivitäten verbal umschrieben. Außerdem haben die Testpersonen zu Beginn eines Testlaufs häufiger Verbal Shadowing praktiziert als am Ende. Da die Testumgebungen in unterschiedlicher Reihenfolge verwendet wurden, deutet dies daraufhin, dass die Motivation der Testpersonen im Laufe der Zeit sank.

Beim Surface haben die Testpersonen tendenziell am häufigsten Verbal Shadowing praktiziert. Beim Multitouch-Bildschirm etwas seltener und beim Notebook sowie beim Whiteboard im gleichen Maße am seltensten. Die quantitativen Unterschiede sind jedoch gering.

Signifikante Unterschiede gibt es in der Art des Verbal Shadowing. Beim Surface haben die Testpersonen regelmäßig ihre Aktionen noch während der Ausführung umschrieben oder am Ende kurz zusammengefasst. Ebenso haben sie manchmal den Text, denn sie eingetippt haben, vorgelesen. Beides wurde bei den Sitzungen mit dem Multitouch-Bildschirm seltener beobachtet. Bei den Sitzungen mit dem Notebook und dem Whiteboard haben die Testpersonen selten ihre Aktionen nach deren Abschluss zusammengefasst und niemals vorgelesen, was sie geschrieben haben. Bei allen Testumgebungen haben die Testpersonen ab und zu das Ende einer Aktivität mit einzelnen Wörtern oder sehr kurzen wie zum Beispiel "Ok." markiert. Je länger eine Person alleine modelliert hatte, desto häufiger war dies zu beobachten. Zwischen den verschiedenen Testumgebungen ist diesbezüglich kein Unterschied erkennbar.

### 6.2.3.2. Übertriebene manuelle Aktionen

Vereinzelt haben die Testpersonen mit übertriebenen manuellen Aktionen auf etwas aufmerksam gemacht. Beim Surface haben die Testpersonen gelegentlich die Bildschirmtastatur geschlossen, indem sie mit einem Finger weit ausholend auf die Enter-Taste gedrückt haben, um den erfolgreichen Abschluss der Texteingabe zu signalisieren. Auf die gleiche Weise haben die Testpersonen beim Multitouch-Bildschirm und beim Notebook die Texteingabe durch Drücken der Enter-Taste der Hardware-Tastatur beendet. Beim Multitouch-Bildschirm war dies gelegentlich zu beobachten. Beim Notebook war es jedoch ein Einzelfall, bei dem die Fachexperten die Ausholbewegung wegen des aufgeklappten Bildschirm nicht uneingeschränkt sehen konnten. Weiterhin hat beim Surface ein Entwickler einmal die Hände gehoben, um zu signalisieren, dass er einige Elemente der Benutzungsschnittstelle versehentlich

berührt hatte. Dadurch hatte er unbeabsichtigt das Modell verändert. Beim Whiteboard waren keine derartigen manuellen Aktionen zu beobachten.

### 6.2.3.3. Sichtbare Haltungsänderungen

Die Testpersonen haben in wenigen Fällen ihre Haltung sichtbar geändert, um den anderen Personen etwas zu signalisieren. Bei den Modellierungssitzungen mit dem Notebook konnte dieser Indikator für Awareness Work jedoch nicht beobachtet werden.

Beim Surface hatte sich ein Entwickler einmal demonstrativ aufgerichtet und anschließend zurück gelehnt, um den anderen Fachexperten zu signalisieren, dass er die Modellierung abgeschlossen hatte und offen für weitere Anweisungen war. In einem anderen Fall hatte ein Entwickler ein Element beschriftet, woraufhin ein Fachexperte den Kopf reckte, um anzudeuten, dass er den Text nur schwer lesen konnte. Der Entwickler las den Text daraufhin vor.

Beim Multitouch-Bildschirm und gleichermaßen beim Whiteboard beschränkten sich die Haltungsänderungen darauf, dass die Entwickler sich manchmal demonstrativ zu den Fachexperten umwandten und fragend blickten, um einer ihrer Fragen an die Fachexperten Nachdruck zu verleihen.

## 6.2.4. Bewertung

Insgesamt war das Awareness-Level der Probanden beim Multitouch-Tisch am höchsten. Unterschiede gab es vor allem bezüglich der Awareness der Fachexperten, was die unaufgeforderten Reaktionen der Fachexperten und die Beeinträchtigung durch vor der Arbeitsfläche stehende Personen dokumentieren (siehe [6.2.1.1](#) und [6.2.2.1](#)). Beim Surface waren die Fachexperten stets über das Geschehen auf der Arbeitsfläche gewahr, wohingegen ihre Awareness bei den anderen Testumgebungen beeinträchtigt war. Beim Beamer war ihre Awareness auf einem etwas höher Niveau als beim Multitouch-Bildschirm und beim Whiteboard, weil sie ein freies Sichtfeld auf die Projektionsfläche hatten.

Die starke Zusammenarbeit der Entwickler beim Multitouch-Tisch (siehe Arbeitsteilung am Surface unter [6.2.1.3](#)) deutet auf ein hohes Awareness-Level hin. Es wurden zwar auch einige negative Awareness Indikatoren beobachtet (siehe [6.2.2](#)). Das mit Abstand am häufigsten beobachtete gleichzeitige Greifen zu dem selben Gegenstand beziehungsweise virtuellen Element ist jedoch eine Folge der regen Zusammenarbeit. Es ist auch unvermeidbar, da eine Person nicht die Absicht einer anderen Person, nach etwas zu greifen, vorhersehen kann.

Bei den anderen Testumgebungen wurden bei den Tätigkeiten der Entwickler sowohl die positiven als auch die negativen Awareness Indikatoren eher selten und unregelmäßig beobachtet. Das lag zum Teil an der bei diesen Testumgebungen wenig kollaborativen Arbeit der Probanden, die das Vorkommen einiger Awareness Indikatoren hemmte. Das seltene Vorkommen der positiven Awareness Indikatoren bedeutet nicht zwangsläufig ein geringes Awareness Level. Da die Entwickler ohne größere Probleme in die Modellierung eingreifen konnten, scheint ihre Awareness zumindest auf einem für die Geschäftsprozessmodellierung ausreichend hohen Niveau gelegen zu haben. Ihre teilweise passive Rolle, die Nachfragen eines Entwicklers beim Multitouch Bildschirm (siehe Arbeitsteilung am Multitouch-Bildschirm unter 6.2.1.3) sowie die verbalen Absprachen der Entwickler beim Whiteboard (siehe Arbeitsteilung am Whiteboard unter 6.2.1.3) deuten dennoch darauf hin, dass das Awareness-Level der Entwickler beim Multitouch-Bildschirm, beim Whiteboard und beim Notebook unterhalb ihres Awareness-Levels beim Multitouch-Tisch lag.

### **6.3. Gesprächsablauf abhängig vom Blickkontakt**

Abhängig davon, ob die Personen Blickkontakt hatten oder nicht, verliefen die Gespräche unterschiedlich. Dialoge längerer Dauer gab es in der Regel nur, wenn die Personen Blickkontakt hatten. Diese Beobachtung wurde unabhängig von den Testumgebungen gemacht.

#### **6.3.1. Ablauf von Gesprächen mit Blickkontakt**

Wenn die Gesprächspartner Blickkontakt hatten, verliefen die Gespräche nach dem Muster eines Dialogs mit Rede und Gegenrede ab. Rede und Gegenrede waren inhaltlich und zeitlich eng zusammenhängend. Auf eine Frage oder Aussage einer Person gab es meistens prompt eine Antwort bzw. Gegenrede der angesprochenen Personen. Es gab nur selten Sprechpausen, die von kurzer Dauer waren. Ebenso gab es nur selten Momente, in denen mehr als eine Person gleichzeitig redete.

Ihre Zustimmung und ihr Verständnis haben die Zuhörer sehr häufig durch wiederholtes Kopfnicken gezeigt oder verbal mit sehr kurzen Sätzen ausgedrückt. Daraufhin fuhr der Sprecher mit seinen Erläuterungen fort. Vielfach gebrauchte Sätze waren in diesen Fällen "Ja.", "Ok.", "Genau.", "Alles klar.", "Richtig.", "Aha!" oder ein zustimmendes "Mhm..".

Nachdem der Sprecher seine Erläuterungen abgeschlossen hatte, setzte oftmals eine andere Person das Gespräch unmittelbar fort. Nur selten wollten mehrere Personen gleichzeitig das Gespräch fortsetzen. Dazwischen geredet haben die Personen nur selten. Meistens haben sie dann einen Einwand oder eine Frage mit direktem inhaltlichem Bezug gehabt.

Pausen gab es dann, wenn die Entwickler die gerade besprochenen Details ins Modell übertrugen.

Ein typischer Dialog mit regelmäßigem Blickkontakt zwischen den Entwicklern und Fachexperten lief beispielsweise so ab:

Die Fachexperten erklären den Entwicklern den Ablauf beim Bestellen.

Fachexperte: "Der Kunde kann in dem Schritt schon alternativ auf einen Paypal-Button klicken."

Entwickler (auf das Modell zeigend): "Auch schon da gleich?"

Fachexperte (nickend): "Genau. Im Warenkorb."

Entwickler (auf das Modell zeigend): "Davor?"

Fachexperte: "Genau. Also er kann alternativ zu dem Bestellen-Knopf auch auf den Paypal-Knopf drücken."

Entwickler: "Alles klar. Das heißt, dass wir davor noch einen Warenkorb haben richtig?"

Fachexperte: "Genau. Ja."

Entwickler: "Wir nehmen an, dass wir uns im Warenkorb befinden."

Fachexperte (nickend): "Genau."

Die beiden Entwickler modellieren nun den alternativen Ablauf. Anschließend fasst ein Entwickler die Änderungen zusammen.

Entwickler: "So. Er klickt dann auf den Bestellen."

Fachexperte (nickend): "Mhm."

Entwickler: "Oder es gibt die weitere Variante, dass der Kunde auf den Paypal-Button klickt."

Fachexperte: "Genau."

Entwickler: "Richtig?"

Fachexperte: "Ja."

### 6.3.2. Ablauf von Gesprächen ohne Blickkontakt

Die Phasen der Gespräche, bei denen es keinen Blickkontakt zwischen den Personen gab, verliefen nach einem gegenteiligen Muster. Es gab wiederholt weder eine Gegenrede noch eine nonverbale Reaktion der angesprochenen Personen.

Je länger es keinen Blickkontakt gab und insbesondere dann, wenn der Sprecher den Zuhörern den Rücken zuehrte, desto seltener haben die Zuhörer ihre Aufmerksamkeit und Zustimmung verbal ausgedrückt. Die Zuhörer haben sich dann seltener mit Sätzen wie "Ja." oder "Genau." (siehe oben) geäußert und haben dabei häufig auch leise und weniger bestimmt gesprochen. Auf Grund des mangelnden Blickkontakts nahmen die Zuhörer an, dass

der Sprecher ihnen gerade keine Aufmerksamkeit widme und ihre Äußerungen nicht beachtet werden. In solchen Situationen meldeten sich die Zuhörer meistens nur dann zu Wort, wenn sie etwas nicht zustimmten oder ihnen etwas von besonderer Wichtigkeit war. Ihre Einwände äußerten sie dann häufig mit Nachdruck und lauterer Stimme als zuvor, um sicher zu sein, Beachtung zu finden. Ein zustimmendes Kopfnicken war ausschließlich dann zu beobachten, wenn eine Person glaubte, dass der Sprecher zu ihr blickte.

In der Regel gab es eine mehrere Sekunden dauernde Sprechpause, nachdem eine Person ihre Erläuterungen beendet hatte. Häufig brach eine Personen ihre Erläuterungen sogar ab. Ohne den Blickkontakt zu den Zuhörern und den verbalen Rückmeldungen der Zuhörer war sich die sprechende Person unsicher, ob ihr überhaupt jemand zuhört.

Wenn eine Person das Gespräch fortsetzte, kam es wiederholt vor, dass eine weitere Person gleichzeitig zu reden begann. Häufig brachen sie dann nicht ab, sondern sprachen einige Sekunden gleichzeitig. Es kam auch mehrfach vor, dass eine Person dazwischen redete, ohne auf die sprechende Person Bezug zu nehmen. Solange es keinen Blickkontakt gab, hatten die Äußerungen der Personen insgesamt einen geringeren Bezug zueinander.

### 6.3.3. Schlussfolgerung

Die Beobachtungen zeigen, dass Blickkontakt Personen hilft, Dialoge durchzuführen, zu koordinieren und aufeinander einzugehen. Ohne Blickkontakt fehlte den Gesprächen jegliche Führung. Es gab keinen gemeinsam verfolgten roten Faden und die einzelnen Äußerungen hatten deutlich weniger Bezug zueinander, als wenn die Personen Blickkontakt hatten. Die bezuglosen Äußerungen und das kaum vorhandene verbale Feedback sind ein Indiz für mangelnde Aufmerksamkeit der Personen und dafür, dass die einzelnen Personen ihre Gedanken auf unterschiedliche Dinge fokussiert hatten.

Es war regelmäßig zu beobachten, dass die gerade sprechende Person den Blickkontakt mit den von ihr angesprochenen Personen suchte. Dadurch versuchte die Person zu erkennen, ob die angesprochenen Personen ihr zuhörten, ihre Erläuterungen bzw. Fragen verstanden und ihren Aussagen zustimmten. Wenn kein Blickkontakt zustande kam, interpretierte der Sprecher dies häufig als mangelnde Aufmerksamkeit seiner Gesprächspartner und brach das Gespräch deshalb vorerst ab. Wenn die gerade sprechende Person ihre Gesprächspartner im Blick hatte, konnte sie nicht nur an den verbalen Reaktionen sondern auch an der Körperhaltung, den Gesten und der Mimik ihrer Gesprächspartner erkennen, ob sie ihr folgen konnten. Ebenso half der Blickkontakt den Personen zu erkennen, dass eine andere Person etwas vorbringen möchte. Dadurch konnten sie das Gespräch koordinieren und ein Durcheinandersprechen überwiegend vermeiden.

### 6.3.4. Blickkontakt bei den einzelnen Testumgebungen

Bei den Modellierungssitzungen am Microsoft Surface gab es regelmäßig und von allen Testumgebungen am häufigsten Blickkontakt zwischen den Personen. Bei den übrigen drei Testumgebungen gab es eher unregelmäßig und deutlich seltener Blickkontakt. Zwischen dem Multitouch-Bildschirm und dem Whiteboard gab es kaum Unterschiede, aber bei der Testumgebung mit Notebook und Beamer gab es etwas seltener Blickkontakt zwischen den Personen.

Beim Microsoft Surface saßen sich die Personen gegenüber, so dass sie immer wieder Blickkontakt hatten. Nur wenn eine Person auf das Surface blickte, um das Modell zu betrachten oder zu ändern, war für sie kein Blickkontakt mit den anderen Personen möglich. Trotzdem blieben die anderen Personen immer in ihrem Gesichtsfeld. So konnte sie zumindest Gesten und Änderungen der Körperhaltung wahrnehmen. Personen, die den Bildschirm des Surface betrachteten, blickten manchmal auch kurz auf, um Blickkontakt zu finden.

Beim Multitouch Bildschirm und gleichermaßen beim Whiteboard blickten die Entwickler immer wieder zum Bildschirm beziehungsweise zur Tafel und kehrten den Fachexperten den Rücken zu. Wenn die Entwickler mit den Fachexperten sprachen, wendeten sie sich oftmals zu den Fachexperten um. Allerdings wendeten sie sich nur selten um während sie modellierten.

Bei der Testumgebung mit dem Beamer saßen sich die Personen gegenüber. Sie wendeten sich niemals den Rücken zu. Dennoch gab es selten Blickkontakt zwischen den Personen, da die Fachexperten meistens zur Projektionsleinwand sahen. Sie hatten dabei die Entwickler nicht im Blickfeld. Die Entwickler blickten zwar häufig in die Richtung der Fachexperten, ihre Blicke wurden aber vielfach nicht erwidert.

## 6.4. Statistische Auswertung

Im Folgenden werden die Sitzungen anhand von Metriken für den Anteil der Gespräche und der Modellierungstätigkeiten an der Sitzung sowie der Produktivität statistisch ausgewertet. Diese Auswertung lässt jedoch keine allgemeingültigen Aussagen zu, weil die Stichprobe klein ist. Dennoch gibt die statistische Auswertung Aufschlüsse über Merkmale der vier verwendeten Testumgebungen.



## 6.4.1. Produktivität und Anteile von Gesprächsphasen sowie Modellierung

### 6.4.1.1. Anteil der Gesprächsphasen an der Sitzung

Für die statistische Auswertung wurden die Phasen der Sitzungen, die durch maximal zwei Sekunden dauernde Sprechpausen unterbrochen waren, als Gesprächsphasen zusammengefasst. Der Anteil der Gesprächsphasen an den Sitzungen ist das Verhältnis der summierten Zeit, welche die Gesprächsphasen beanspruchten, zu der Sitzungsdauer.

Bei den Sitzungen mit dem Whiteboard war der Anteil der Gesprächsphasen mit Abstand am größten (siehe Abbildung 6.1). Der Mittelwert betrug beim Whiteboard 78,3%. Bei den anderen Sitzungen lag der Mittelwert 14,0% bis 17,5% darunter. Beim Beamer betrug er 60,8%, beim Multitouch-Bildschirm 61,9% und beim Surface 64,3%. Die einzelnen Werte wichen zwar bei den Testläufen teilweise stark voneinander ab, aber der Anteil der Gesprächsphasen war bei jedem Testlauf beim Whiteboard am größten (siehe Tabelle 6.1).

Wenn bei der Berechnung dieser Metrik eine andere Dauer der Sprechpausen gewählt wird, anhand der die Gesprächsphasen ermittelt werden, ändern sich die Anteile Gesprächsphasen an der Sitzung nahezu proportional.

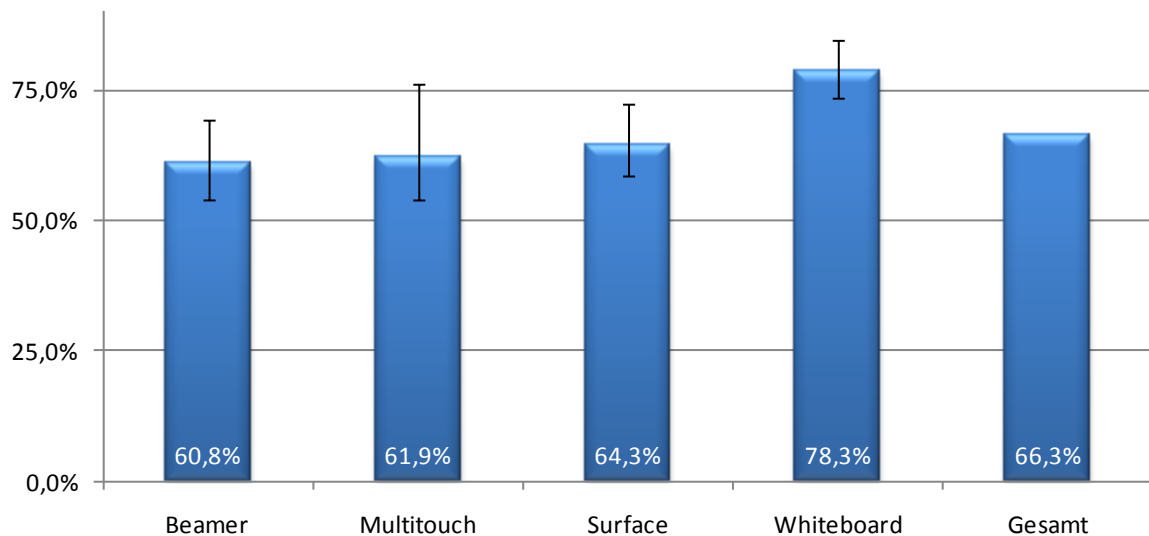


Abbildung 6.1.: Anteil der Gesprächsphasen an der Sitzung (Mittelwerte in Prozent mit Indikatoren für den Wertebereich)

	Beamer	Multitouch	Surface	Whiteboard
1. Testlauf	58,2%	54,0%	72,5%	73,3%
2. Testlauf	53,8%	76,2%	58,6%	84,6%
3. Testlauf	61,9%	58,3%	66,5%	78,8%
4. Testlauf	69,3%	59,3%	59,5%	76,3%
Mittelwert	60,8%	61,9%	64,3%	78,3%
Standardabweichung	5,7	8,5	5,6	4,1

Tabelle 6.1.: Anteil der Gesprächsphasen an der Sitzung

#### 6.4.1.2. Anteil der Modellierung an der Sitzung

Der Anteil der Modellierung an den Sitzungen ist das Verhältnis der summierten Zeit, welche die Modellierungstätigkeiten beanspruchten, zu der Sitzungsdauer. Als Modellierungstätigkeiten wurden alle Tätigkeiten der Probanden eingestuft, die zu einer Veränderung der Modelle führten. Dazu gehören das Verschieben, Hinzufügen und Löschen von Modellelementen sowie das Schreiben von Text.

Bei den Anteilen der Modellierungstätigkeiten an den Sitzungen ist das Verhältnis zwischen den Testumgebungen umgekehrt wie bei den Gesprächsanteilen. Beim Whiteboard war der Modellierungsanteil mit im Schnitt 51,8% am geringsten und deutlich geringer als bei den anderen Testumgebungen (siehe Abbildung 6.2). Der Mittelwert betrug beim Beamer 66,4%, beim Multitouch-Bildschirm 63,6% und beim Surface 65,8%. Die Differenz zwischen dem Mittelwert beim Whiteboard und den computerunterstützten Testumgebungen ist mit 11,8% bis 14% deutlich größer als die Differenzen zwischen den computerunterstützten Testumgebungen.

	Beamer	Multitouch	Surface	Whiteboard
1. Testlauf	64,8%	61,0%	59,3%	60,7%
2. Testlauf	65,2%	57,6%	70,0%	44,5%
3. Testlauf	76,4%	62,1%	68,9%	54,6%
4. Testlauf	59,1%	73,6%	65,2%	47,5%
Mittelwert	66,4%	63,6%	65,8%	51,8%
Standardabweichung	6,3	6,1	4,2	6,3

Tabelle 6.2.: Anteil der Modellierung an der Sitzung

Wie beim Anteil der Gesprächsphasen schwanken auch die Modellierungsanteile zwischen den einzelnen Testläufen. Aber bei drei der vier Testläufe war der Modellierungsanteil beim Whiteboard mit Abstand am geringsten (siehe Tabelle 6.2).

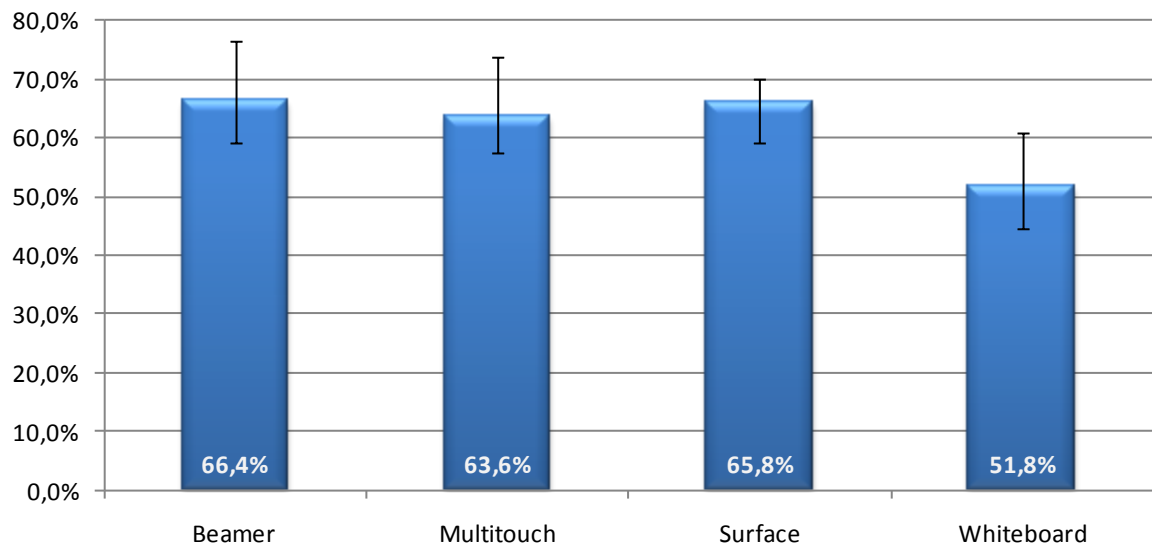


Abbildung 6.2.: Anteil der Modellierung an der Sitzung (Mittelwerte in Prozent mit Indikatoren für den Wertebereich)

#### 6.4.1.3. Produktivität während der Modellierung: Anzahl modellierter Elemente relativ zur Modellierungszeit

Die Produktivität beim Modellieren wird berechnet, indem die Summe der erstellten und gelöschten Elemente durch die Summe der Modellierungsdauer dividiert wird. Bei der Auswertung der Produktivität muss die individuelle Leistungsfähigkeit der Probandengruppen berücksichtigt werden. Deshalb müssen die einzelnen Werte der Testläufe aus Tabelle 6.3 normiert werden, um einen sinnvollen Mittelwert für die Testumgebungen berechnen zu können und die Werte vergleichen zu können. Die normierte Produktivität wird durch die Division eines einzelnen Wertes durch den Mittelwert des Testlaufs berechnet.

Beim Whiteboard war die normierte Produktivität mit im Schnitt 1,73 Elementen pro Minute mehr als doppelt so groß wie bei den übrigen Testumgebungen (siehe Tabelle 6.4). Die durchschnittliche normierte Produktivität lag beim Beamer bei 0,79 Elementen pro Minute, beim Multitouch-Bildschirm bei 0,80 Elementen pro Minute und beim Surface bei 0,68 Elementen pro Minute. Der Wert beim Surface lag damit 14% bzw. 15% unterhalb der Werte beim Beamer und beim Multitouch-Bildschirm.

	Beamer	Multitouch	Surface	White-board	Mittelwert	Standard-abweichung
1. Testlauf	3,42	3,63	3,44	8,67	4,79	2,24
2. Testlauf	2,98	2,36	2,42	9,27	4,26	2,91
3. Testlauf	4,53	4,89	3,72	6,00	4,79	0,82
4. Testlauf	4,73	5,19	3,91	9,82	5,91	2,30
Mittelwert	3,92	4,02	3,37	8,44	4,94	2,04
Standard-abweichung	0,73	1,12	0,58	1,47	0,60	

Tabelle 6.3.: Produktivität beim Modellieren in Elementen pro Minute

	Beamer	Multitouch	Surface	Whiteboard
1. Testlauf	0,71	0,76	0,72	1,81
2. Testlauf	0,70	0,55	0,57	2,18
3. Testlauf	0,95	1,02	0,78	1,25
4. Testlauf	0,80	0,88	0,66	1,66
Mittelwert	0,79	0,80	0,68	1,73
Standardabweichung	0,10	0,17	0,08	0,33

Tabelle 6.4.: Normierte Produktivität beim Modellieren in Elementen pro Minute

#### 6.4.1.4. Produktivität Insgesamt: Anzahl der Modellelemente am Ende der Sitzung relativ zur Sitzungsdauer

Die Produktivität einer Sitzung wird durch die Division der Anzahl der Modellelemente, die ein Modell am Ende der Sitzung hatte, durch die Dauer der Sitzung in Minuten berechnet. Wie auch bei der Modellierungsproduktivität, müssen die Werte wegen der unterschiedlichen Produktivität der Probandengruppen normiert werden, um die Mittelwerte berechnen und vergleichen zu können. Die normierte Produktivität wird jeweils durch die Division des absoluten Wertes aus Tabelle 6.5 durch den Mittelwert des Testlaufs berechnet.

	Beamer	Multitouch	Surface	White-board	Mittelwert	Standard-abweichung
<b>1. Testlauf</b>	1,77	2,00	1,50	2,36	1,91	0,31
<b>2. Testlauf</b>	1,66	1,36	1,69	2,27	1,74	0,33
<b>3. Testlauf</b>	2,20	1,70	2,18	2,15	2,06	0,21
<b>4. Testlauf</b>	2,63	3,32	2,55	4,08	3,14	0,62
<b>Mittelwert</b>	2,06	2,10	1,98	2,71	2,21	0,29
<b>Standard-abweichung</b>	0,38	0,74	0,41	0,79	0,55	

Tabelle 6.5.: Anzahl der Modellelemente am Ende der Sitzung relativ zur Sitzungsdauer in Elementen pro Minute

Die ermittelte Produktivität für die Sitzungen divergiert deutlich geringer als die Produktivität bei den Modellierungstätigkeiten. Beim Whiteboard ist auch die für die Sitzungsdauer berechnete normierte Produktivität mit 1,22 Elementen pro Minute am größten. Sie ist aber nicht mehr als doppelt so groß wie die Produktivität bei den anderen Testumgebungen (siehe Tabelle 6.6). Die Produktivität beim Beamer beträgt 0,95 Elemente pro Minute, beim Multitouch-Bildschirm 0,93 Elemente pro Minute und beim Surface 0,91 Elemente pro Minute. Damit liegt die Produktivität hier nur um 29% beim Beamer und um 34% beim Surface unterhalb der Produktivität beim Whiteboard. Der Produktivitätsunterschied zwischen dem Surface und den anderen Computern ist bei der Sitzungs-Produktivität geringer als bei der Modellierungs-Produktivität: die Sitzungs-Produktivität beim Surface liegt um 4% beziehungsweise 2% unterhalb der Sitzungs-Produktivität beim Multitouch-Bildschirm und beim Whiteboard.

Der Vergleich von Sitzungs-Produktivität und Modellierungs-Produktivität zeigt, dass die Produktivität beim Bearbeiten eines Modells nicht der alleinige Faktor für die Produktivität einer Modellierungssitzung ist. Ein weiterer Faktor ist die Sorgfalt, mit der Modelliert wird. Beim Whiteboard haben die Probanden am unsorgfältigsten modelliert, so dass sie wiederholt

	Beamer	Multitouch	Surface	Whiteboard
1. Testlauf	0,93	1,05	0,79	1,23
2. Testlauf	0,95	0,78	0,97	1,30
3. Testlauf	1,07	0,83	1,06	1,04
4. Testlauf	0,84	1,06	0,81	1,30
Mittelwert	0,95	0,93	0,91	1,22
Standardabweichung	0,08	0,13	0,11	0,11

Tabelle 6.6.: Normierte Produktivität während der Modellierungssitzungen in Elementen pro Minute

größere Abschnitte eines Modells verwarfen (siehe 6.1.4). Bei den anderen Testumgebungen und insbesondere beim Surface haben die Probanden sorgfältiger modelliert, so dass sie weniger Fehler korrigieren mussten.

#### 6.4.2. Anteil der Probanden am Gespräch

Die Probanden haben sich während der Sitzungen ungleich häufig am Gespräch beteiligt. Auffallend ist, dass die Verteilung der Gesprächsanteile während jeder Sitzung ähnlich war und ein bestimmtes Muster aufweist. Wenn man die Gesprächsanteile der Personen mit den jeweils größten, den zweit größten, den dritt größten und den geringsten Gesprächsanteilen gruppiert, wird eine stufenartige Verteilung der Anteile deutlich (siehe Abbildung 6.3). Bei keiner Sitzung war das Gespräch gleichmäßig auf die Personen verteilt. Es gab immer eine Person, die das Gespräch mit einem Gesprächsanteil von durchschnittlich 42% dominierte. Die anderen Personen beteiligten sich zu durchschnittlich 28%, 20% und 10% am Gespräch.

Die beiden Personen mit dem größten Gesprächsanteil waren jeweils ein Entwickler und ein Fachexperte. Diese beiden dominierten jeweils die Sitzungen, während sich der andere Entwickler und der andere Fachexperte eher zurückhielten (siehe Abbildung 6.4). Bei 10 von 16 Sitzungen war die Person mit dem größten Gesprächsanteil ein Entwickler.

Insgesamt waren die Gesprächsanteile der Entwickler und Fachexperten nur beim Beamer nahezu ausgeglichen. Bei den anderen Testumgebungen hatten die Entwickler einen etwas größeren Anteil am Gespräch (55,7% bis 57,6%) als die Fachexperten (siehe Abbildung 6.5). Ein Grund dafür ist, dass die Fachexperten am Ende der Sitzungen die Geschäftsprozessmodelle noch einmal durchspielten und erläuterten. Wenn man nur den Modellierungszeitraum berücksichtigt, verschieben sich die Gesprächsanteile zugunsten der Fachexperten. Die vergleichsweise größeren Gesprächsanteile der Fachexperten beim Beamer resultieren

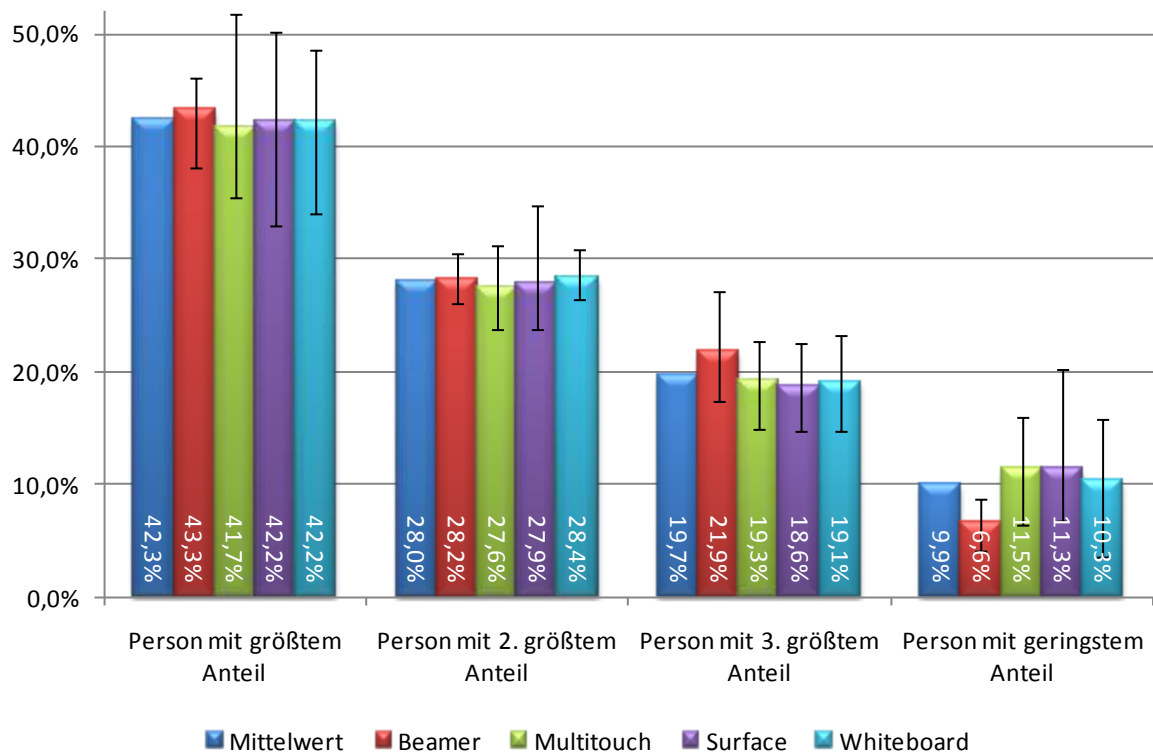


Abbildung 6.3.: Anteil der Personen am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich)

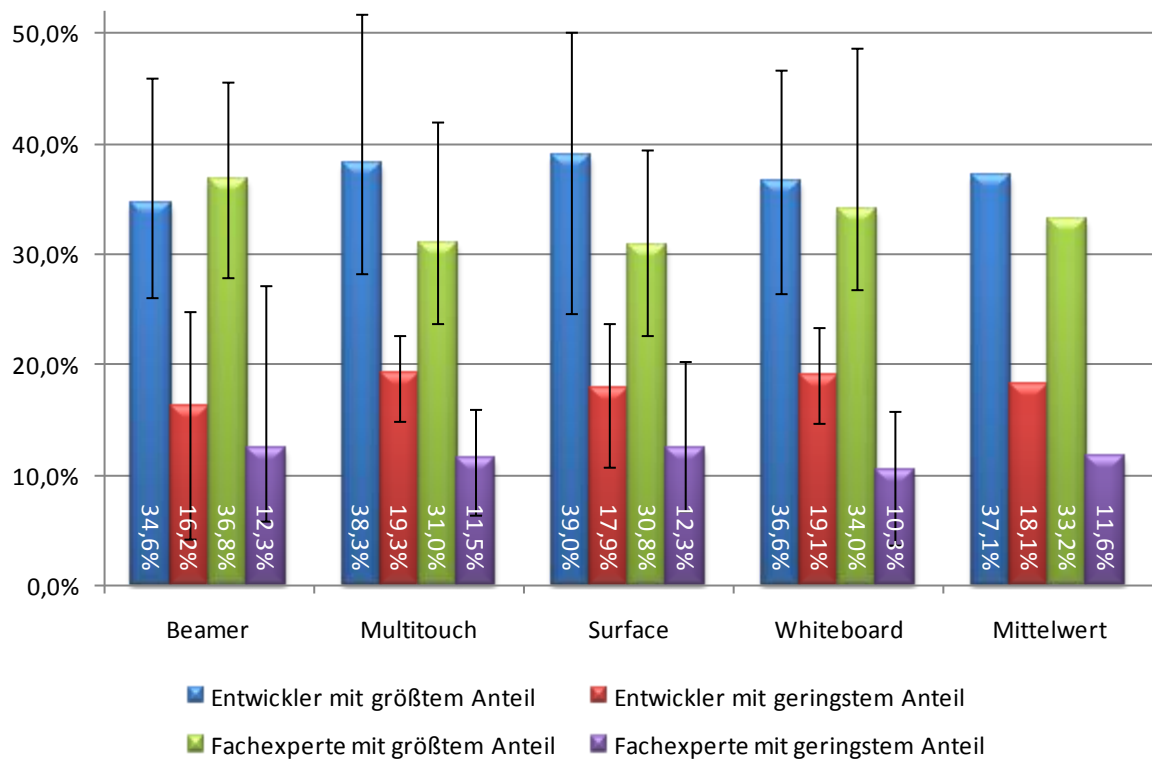


Abbildung 6.4.: Anteil der beiden Entwickler und der beiden Fachexperten am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich)



daraus, dass die Entwickler hier relativ selten nachgefragt haben und seltener während der Modellierung ihre Modellierungstätigkeiten erläutert haben.

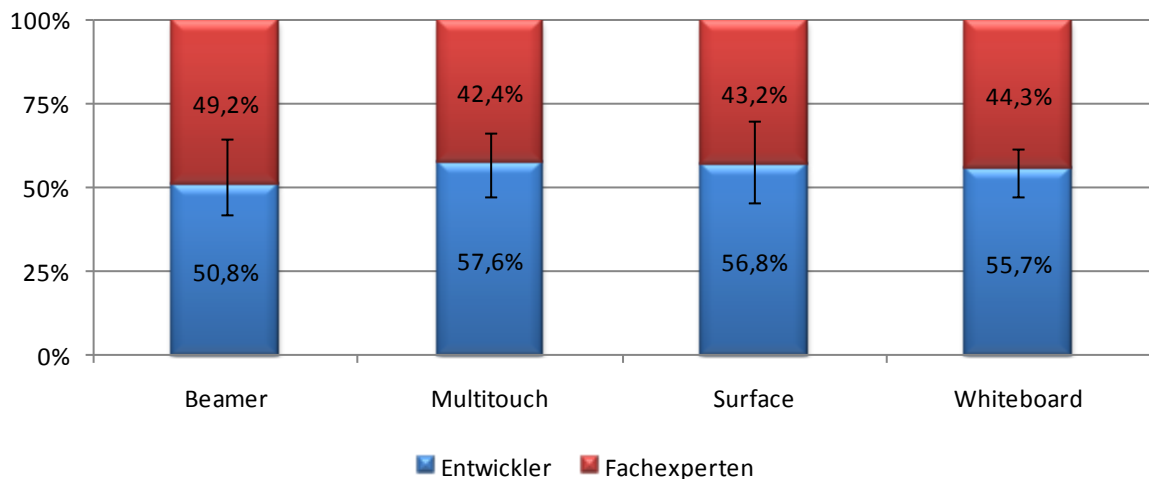


Abbildung 6.5.: Anteil der beiden Gruppen Entwickler und Fachexperten am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich)

Bis auf die zuletzt genannte ausgeglichene Gesprächsverteilung beim Beamer, lässt die statistische Auswertung der Gesprächsanteile keine Unterschiede zwischen den Testumgebungen erkennen. Trotzdem hat diese Auswertung für die Konzeption von kollaborativen Modellierungsanwendungen und Planung von Modellierungssitzungen hilfreiche Erkenntnisse zu Tage gefördert. Es muss künftig weiter untersucht werden, wie kollaborative Modellierungsanwendungen den divergierenden Gesprächsanteilen gerecht werden können. Zum Beispiel könnte eine Anwendung dem Gesprächsleiter spezielle Funktionen bereitstellen. Bei der Planung einer Modellierungssitzung müssen die Teilnehmer gezielt ausgesucht werden. Die Auswertung der Gesprächsanteile legt nahe, die Leitung einer Sitzung einem in der Moderation von Sitzungen erfahrenen Modellierungsexperten zu geben und für die Rolle des zweiten Entwicklers einen Modellierungsexperten mit speziellen Kenntnissen in der verwendeten Modellierungssprache zu wählen.

### 6.4.3. Bewertung

Die statistische Auswertung der Sitzungen zeigt einen deutlichen Unterschied zwischen der Modellierung am Whiteboard und der Modellierung mit den verwendeten Computern. Zwischen den verschiedenen Computern ist hingegen kein klarer Unterschied erkennbar. Die kleine Stichprobe, anhand der die Statistiken berechnet wurden, lässt alleine noch keine allgemeingültige Aussagen zu. Die qualitative Auswertung (Abschnitte 6.1, 6.2, 6.3) hat aber

Merkmale aufgezeigt, die daraufhin deuten, dass auch in der Praxis tendenziell solche Unterschiede bei der Produktivität und den Anteilen von Gesprächsphasen auftreten, wie sie bei der statistischen Auswertung ermittelt wurden.

## 6.5. Feedback der Probanden

Im Folgenden wird die Rückmeldung der Probanden zusammengefasst, welche die Probanden in Interviews und mit Fragebögen gegeben haben.

### 6.5.1. Nützlichkeit der Simulation

Alle Probanden haben die Möglichkeit, Geschäftsprozessmodelle mit der Modellierungsanwendung simulieren zu können, als hilfreich bewertet. Sie gaben an, die Simulation habe sie dabei unterstützt, die Geschäftsprozessmodelle zu verstehen und zu überprüfen. Einige Probanden wanden jedoch ein, dass die Simulation wegen der geringen Komplexität und Größe der Modelle nicht nötig gewesen sei, um die Modelle zu verstehen. Alle Probanden waren sich aber einig, dass die Simulation eines großen Modells eine sinnvolle Hilfestellung ist.

Alle Probanden bewerteten die Visualisierung der Simulation als nachvollziehbar. Vor allem im Vergleich mit dem Whiteboard, bei dem die Probanden die Geschäftsprozesse selbst durchspielen mussten, sei ihnen deutlich geworden, wie hilfreich die Visualisierung ist. Mehrere Probanden haben es auch als hilfreich bewertet, die Abläufe verschiedener simulierter Geschäftsfälle miteinander vergleichen zu können. Dies habe ihnen geholfen, die Unterschiede der Geschäftsfälle zu verstehen und zu überblicken.

### 6.5.2. Bewertung der Zusammenarbeit

Die Probanden haben die Art der Zusammenarbeit beim Surface am besten bewertet. Insbesondere den Fachexperten gefiel es, sehr dicht an der Arbeitsfläche zu sitzen. Erstens hätten sie deshalb leichter als bei den anderen Testumgebungen nachvollziehen können, was auf der Arbeitsfläche geschieht. Zweitens hätten sie beim Surface aktiv mitarbeiten können. Die Zuschauerrolle bei den anderen Testumgebungen gefiel den Fachexperten nicht, weil sie den Entwicklern sagen mussten, was sie machen sollen. Aber auch einige Entwickler empfanden es als unangenehm, dass beim Multitouch-Bildschirm, beim Notebook und beim Whiteboard häufig nur einer redete. Die Probanden bewerteten die Sitzordnung am Surface als förderlich für die Dialoge. Bei den anderen Testumgebungen seien hingegen keine richtigen Dialoge

zustandegekommen, weil die Distanz zwischen den Entwicklern und Fachexperten zu groß gewesen sei.

Einige Fachexperten erwähnten explizit, das Surface habe sie animiert, mitzuarbeiten. Die anderen Testumgebungen hätten ihnen dagegen die Motivation genommen, sich zu beteiligen. Beim Surface habe jedoch die Bildschirmtastatur das kollaborative Arbeiten manchmal behindert, weil sie so viel Platz benötige, dass paralleles Arbeiten kaum möglich ist.

### 6.5.3. Ergonomie und Sitzposition

Beim Surface bemängelten die Probanden, nicht bequem sitzen zu können. Alle Probanden fanden es sehr störend, wegen des Unterbaus des Geräts, die Füße nicht ausstrecken zu können. Darüber hinaus kritisierten die Probanden die geringe Höhe der Arbeitsfläche, aufgrund der sie beim Bedienen des Geräts nicht aufrecht sitzen konnten. Trotz der geringen Sitzungsdauer empfanden einige der Probanden die Belastung der Wirbelsäule bereits als unangenehm. Bei den anderen Testumgebungen hatten die Probanden keine ernsten Einwände bezüglich ihrer Sitzposition. Die Sitzposition bei der Testumgebung mit dem Notebook haben sie allerdings am besten bewertet. Die Sitzposition war dort am flexibelsten, denn es gab einen Tisch, auf den sich die Probanden aufstützen konnten.

Beim Multitouch-Bildschirm und beim Whiteboard haben die Probanden in der Entwickler-Rolle die meiste Zeit gestanden. Sie gaben an, es zu bevorzugen zu sitzen. Es sei aber nicht unangenehm gewesen zu stehen. Allerdings war die Dauer der Sitzungen geringer, als es in der Realität üblich wäre. Auf die Dauer sind Ermüdungserscheinungen aufgrund des Stehens nicht ausgeschlossen.

Ermüdungserscheinungen hatten einige Entwickler beim Multitouch-Bildschirm und beim Whiteboard jedoch in den Armen. Beim Bedienen des vertikalen Bildschirms sowie beim Schreiben auf dem Whiteboard mussten sie ihre Arme anheben, was sie mit zunehmender Zeit als Belastung empfanden. Die Probanden beklagten dies insbesondere beim Multitouch-Bildschirm, da die Touch-Erkennung nur bei starkem Aufdrücken der Finger zuverlässig war.

Ein Entwickler empfand es außerdem unbequem, in den unteren Abschnitten des Whiteboards zu schreiben, da er sich dann bücken musste.

### 6.5.4. Lesbarkeit von Text und Modellen

Die Lesbarkeit von Text bewerteten 15 von 16 Probanden beim Whiteboard am schlechtesten. Als Grund gaben sie die teilweise unleserliche Handschrift an. Vor allem dann, wenn ein

Modellelement verändert wurde, war die Schrift häufig sehr klein. Die Probanden haben die Lesbarkeit von Text beim Multitouch-Bildschirm etwas besser bewertet. Die Fachexperten bemängelten hier, dass der Text so klein war, dass sie ihn von ihrer Sitzposition aus gerade noch lesen konnten. Näher an den Bildschirm heran mochten sie nicht rücken, da sonst die Entwickler über ihre ausgestreckten Beinen stolpern könnten. Beim Surface hatten die Probanden keine ernsthaften Probleme beim Lesen von Text, da alle Modellelemente doppelt beschriftet waren. Einige Probanden betonten, der duplizierte Text habe die Lesbarkeit sehr gefördert. Ein Proband gab wiederum an, dass ihn die doppelte Beschriftung irritiert habe. Am Besten haben die Probanden die Textlesbarkeit bei der Testumgebung bewertet, bei der ein Beamer an ein Notebook angeschlossen war. Auf der Projektionsleinwand war Text jederzeit groß und deutlich lesbar.

Eine große Mehrheit der Probanden hat auch die Lesbarkeit der Modelle insgesamt beim Whiteboard am schlechtesten bewertet. Sie bemängelten, dass die Modelle mit zunehmender Größe immer unübersichtlicher wurden. Die Unübersichtlichkeit resultierte aus den Änderungen, welche die Entwickler vornahmen, und aus der begrenzten Fläche des Whiteboards. Der Platzmangel und die fehlende Möglichkeit, Elemente zu verschieben, zwang die Entwickler teilweise dazu, mit Sequence Flows verbundene Flow Nodes weit verstreut über die Tafel zu platzieren. Bei den anderen Testumgebungen gab es diese Probleme nicht, da die Modellierungsfläche der Modellierungsanwendung unbegrenzt war und die Probanden Modellelemente verschieben konnten. Beim Surface und beim Notebook/Beamer kritisierten einige Probanden jedoch, einen zu kleinen Ausschnitt des Modells sehen zu können. Sie forderten deshalb eine größere Bildschirmfläche beim Surface und eine deutlich höhere Auflösung für das Surface sowie den Beamer. Beim Multitouch-Bildschirm lobten die Probanden wiederum, dass wegen der höheren Auflösung (1920x1080 Pixel statt 1024x768 Pixel) viele Modellelemente auf dem Bildschirm Platz hatten.

Beim Surface haben zwei von vier Gruppen die Geschäftsprozesse - jeweils von den Entwicklern aus betrachtet - von oben nach unten modelliert und die anderen von Links nach Rechts. Trotz dieser unterschiedlichen Anordnung und trotz der Unterschiedlichen Perspektive hat kein Proband über Schwierigkeiten beim Verstehen der Modelle geklagt.

### **6.5.5. Bearbeitung von Modellen**

Alle Probanden betrachteten es als großen Vorteil, mit der Modellierungsanwendung Modelle nachträglich schnell verändern zu können. Beim Whiteboard fanden sie die Änderungen am Modell aufwändiger als an einem Computer, da sie beim Einfügen von Elementen häufig bestehende Abschnitte eines Modells wegwischen und erneut anzeichnen mussten. Bei der Modellierungsanwendung sei es ebenso ein möglicher Vorteil, wenn sie Refactorings vornehmen könnten, indem sie zum Beispiel den Typ eines Flow Nodes ändern. Allerdings

fanden es die Probanden wesentlich einfacher, am Whiteboard mit einem Stift ihre Ideen skizzieren zu können. Mit der Modellierungsanwendung benötigten sie dazu ein vielfaches an Zeit.

Die Bedienung der Modellierungsanwendung beim Hinzufügen von neuen Modellelementen fanden die Probanden als verbesserungswürdig. Insbesondere sei das Verbinden von Knoten mit Sequence Flows zu zeitaufwendig und mit zu vielen Arbeitsschritten verbunden. Konkret störte sie, die Sequence Flows aus der Modellelement-Palette herausziehen zu müssen. Sie wünschten, stattdessen Sequence Flows direkt über die zu verbindenden Knoten erstellen zu können. Das Hinzufügen von Knoten über die Modellelement-Palette fanden die Probanden prinzipiell gut. Das Drag and Drop bei der Multitouch-Eingabe (Multitouch-Bildschirm und Surface) müsse aber noch verbessert werden, da es zu häufig ungewollt abbricht.

### 6.5.6. Texteingabe

Beim Multitouch-Bildschirm empfanden einigen Probanden den Wechsel zwischen den Eingabemethoden Multitouch und Tastatur als störend, da er zu einer Unterbrechung des Arbeitsflusses führe. Sie hätten es bevorzugt, mit Multitouch nur eine Eingabemethode zu haben.

Das Schreiben mit der Bildschirmtastatur beim Multitouch-Tisch haben die Probanden durchweg äußerst negativ bewertet. Ihnen missfiel vor allem die fehlende Haptik, wegen der sie sehr viele Tippfehler machten. Infolgedessen hätten sie dazu geneigt, Worte abzukürzen und kürzere Formulierungen zu wählen als sie es üblicherweise getan hätten. Einige Probanden schlugen vor, anstatt der Bildschirmtastatur eine Hardware-Tastatur zu verwenden. Ein Proband wand jedoch ein, eine Hardware-Tastatur führe zu einer chauffierten Modellierungssitzung, bei der nur eine Person modelliert und die anderen zuschauen, was sich bei den anderen Geräten als Nachteil erwiesen habe (siehe [6.5.2](#)).

Die Probanden haben beim Multitouch-Tisch auch kritisiert, nicht gleichzeitig Text eingeben zu können. Dies war nicht möglich, weil das verwendete Microsoft Surface nicht mehrere Bildschirmtastaturen unterstützt.

### 6.5.7. Verwendung physischer Objekte beim Surface

Die Rückmeldung der Probanden zur Bedienung von Werkzeugen mit Gegenständen war durchweg positiv. Die Benutzungsoberfläche sei nicht mit unnötigen Elementen überladen gewesen, weil sie die Werkzeuge entfernen konnten. Die Handhabung der Gegenstände sei einfach gewesen. Ein Proband kritisierte jedoch, das Löschen von Elementen sei mit

zu vielen Arbeitsschritten verbunden. Es sei zu aufwändig, den Papierkorb-Gegenstand auf die Arbeitsfläche zu stellen und ihn anschließend wieder zu entfernen, um ein Element zu löschen.

### **6.5.8. Eignung der Geräte für die Geschäftsprozessmodellierung**

Beim abschließenden Fragebogen bewerteten 12 von 16 Probanden das Surface als am geeignetsten für die Geschäftsprozessmodellierung. Sechs dieser 12 Probanden waren allerdings der Meinung, eines der anderen Geräte sei genauso gut geeignet. Während der abschließenden Diskussion relativierten einige Probanden jedoch ihre Meinung zu Gunsten des Surface.

Ausschlaggebend für die gute Bewertung des Surface war für viele Probanden und insbesondere die Fachexperten, sehr nah am Modell zu sitzen und deshalb mit allen Teilnehmern der Sitzung gemeinsam Arbeiten zu können. Allerdings müssten die zuvor genannten Probleme behoben werden, bevor das Surface in der Praxis eingesetzt werden könne. Insbesondere die Texteingabe mit der Bildschirmtastatur sei ein schwerwiegendes Manko. Den Multitouch-Bildschirm betrachteten nur zwei von 16 Probanden als Alternative. Die große Mehrheit der Probanden fand den Multitouch-Bildschirm für die Geschäftsprozessmodellierung ungeeignet. Eine Alternative zum Surface sei vielmehr entweder das Notebook oder das Whiteboard. Je ein Drittel der Probanden bewertete die Eignung des Notebooks oder des Whiteboards am besten oder gleich gut wie die des Surface. Beim Notebook gefiel den Probanden die gewohnte und zuverlässigen Bedienbarkeit der Modellierungsanwendung mit einer Maus und einer Tastatur. Der große Vorteil des Whiteboards sei, sehr schnell Ideen in ein Modell umsetzen zu können. Jedoch sei die Schwäche des Whiteboards, die Modelle anschließend nicht ohne weiteres weiterverwenden zu können. Die Modelle müssten aufwendig digitalisiert werden. Außerdem tendiere man am Whiteboard dazu, weniger sorgfältig zu modellieren, als an einem Computer.

Einige Probanden betrachten das Surface und das Whiteboard als zwei Alternativen, von denen sie sich vorstellen könnten, sie in der Praxis je nach Zweck der Modellierung einzusetzen. Das Whiteboard würden sie beim Entwickeln von Ideen zusammen mit einem Kollegen verwenden und das Surface, wenn sie einen Geschäftsprozess formal korrekt und zur weiteren Verwendung modellieren müssten.

# 7. Bewertung

Mit den im vorigen Kapitel erläuterten Ergebnissen der Usability-Studie kann die primäre Fragestellung dieser Arbeit beantwortet werden. Die Testauswertung zeigt, dass Multitouch-Tische besser für die Modellierung von Geschäftsprozessen geeignet sind als andere Multitouch-Geräte und klassische Computer. Außerdem ist eine Arbeitsumgebung mit einem Multitouch-Tisch in den meisten Fällen für die Geschäftsprozessmodellierung besser geeignet als eine Arbeitsumgebung mit einer Tafel (z.B. ein Whiteboard) und ohne Computer. Dafür sprechen die im Folgenden genannten Argumente.

## 7.1. Argumente für Multitouch-Tische

### 1. Verständlichkeit von Modellen

Für den Einsatz von Multitouch-Tischen spricht, dass Multitouch-Tische das Verstehen von Modellen fördern, während Arbeitsumgebungen mit vertikalen Arbeitsflächen und Projektionsleinwänden das Verstehen von Modellen behindern. Multitouch-Tische begünstigen das Verstehen von Modellen wegen ihrer horizontalen Arbeitsfläche, um welche die Teilnehmer an einer Modellierungssitzung herum sitzen. Deshalb haben die Personen erstens das Geschäftsprozessmodell und die Modellierungstätigkeiten immer im Blick. Zweitens können sie wegen der direkten Multitouch-Eingabe nachvollziehen, wie die Änderungen am Modell zustande kommen. Dies äußerte sich während der Usability-Tests in einer hohen Awareness der Probanden und regen Diskussionen. Die beobachteten Indikatoren für Awareness belegen, dass die Probanden die Tätigkeiten der jeweils anderen Probanden verstanden und jederzeit ein tiefgreifendes Verständnis über den modellierten Geschäftsprozess hatten (siehe 6.2.1). Die vertikale Arbeitsfläche von Multitouch-Bildschirmen und Tafeln hemmt hingegen das Verstehen von Modellen. Denn während des Modellierens versperren häufig die an der Arbeitsfläche stehenden Personen anderen Personen den Blick auf das Modell und die Modellierungstätigkeiten. Während des Usability-Tests äußerte sich dies in einer eingeschränkten Awareness der Probanden. Es wurden Verzögerungen bei der Modellierung beobachtet, während der die Probanden versuchten, das Modell und die vorangegangenen

Modellierungstätigkeiten zu verstehen (siehe 6.2.1.1). Wenn Beamer verwendet werden, stehen in der Regel keine Personen vor der Projektionsleinwand, da die Modellierung an einem separaten Computer durchgeführt wird. Hier erschwert allerdings die indirekte Eingabe den Betrachtern des Modells zu verstehen, wie die Änderungen zustande kommen. Bei dem Usability-Test konnten die Betrachter eines Modells, während ein Proband das Modell bearbeitete, kaum etwas zur Modellierung beitragen. Denn ihnen war nicht klar, was die modellierende Person gerade machte.

## 2. Umfang der Diskussionen

Ein zweites entscheidendes Argument für Multitouch-Tische ist, dass sie Diskussionen fördern. Multitouch-Tische, Tafeln und Beamer haben dagegen einen negativen Einfluss auf Diskussionen. Diskussionen sind bei der Modellierung eines Geschäftsprozesses essentiell, denn die unterschiedlich qualifizierten Teilnehmer müssen verbal ausdrücken, wie sie den Geschäftsprozess verstehen. Missverständnisse bezüglich der Semantik eines Modells können nur im Gespräch frühzeitig aufgedeckt und behoben werden. Weil sich die Personen an einem Multitouch-Tisch gegenüber sitzen, können sie jederzeit Blickkontakt aufnehmen, die Gesten und Mimik der anderen Personen sehen und auf das Modell zeigen. Der Usability-Test belegt, dass der Blickkontakt die Diskussionen beim Multitouch-Tisch anregt. Die Personen können an den non-verbalen Reaktionen ihrer Diskussionspartner deren Zustimmung oder Ablehnung erkennen und daran das Gespräch ausrichten. Zudem hilft es den Personen, mit Gesten und durch das Zeigen auf das Modell ihre Gedanken auszudrücken. All dies schränken Multitouch-Tische, Tafeln und Beamer ein. Bei Multitouch-Tischen und Tafeln kehren die an der Arbeitsfläche stehenden Personen häufig den anderen Personen den Rücken zu, so dass kein Blickkontakt möglich ist. Wenn ein Beamer verwendet wird, bricht der Blickkontakt häufig ab, weil die Personen regelmäßig zur Projektionsleinwand blicken. Infolgedessen wurden beim Usability-Test bei diesen Geräten weniger durchgängige Diskussionen beobachtet und es wurden häufig Personen von der Diskussion ausgeschlossen. Letzteres belegt, dass kein gemeinsamer Kollaborationsbereich vorhanden war und die Gruppeninteraktion verloren ging, so wie es [Orre und Middup \(2006\)](#) befürchtet haben. Die horizontale Arbeitsfläche von Multitouch-Tischen ist somit als einzige der vier Alternativen ein gemeinsamer Kollaborationsbereich, der für die Geschäftsprozessmodellierung geeignet ist.

## 3. Aktivität aller Beteiligten

Ein weiteres Argument für den Einsatz von Multitouch-Tischen ist, dass alle Teilnehmer an einer Modellierungssitzung aktiv an der Modellierung teilnehmen können. Die um den Tisch



herum sitzenden Personen haben die Arbeitsfläche direkt vor sich und können jederzeit in die Modellierung eingreifen. Bei Desktop Computern und Notebooks ist dagegen nur eine chauffierte Modellierung möglich, weil eine Person die alleinige Kontrolle hat. Bei Multitouch-Bildschirmen und Tafeln können prinzipiell mehrere Personen an der Arbeitsfläche stehen und in die Modellierung eingreifen. Der Usability-Test hat jedoch gezeigt, dass die Personen nur selten diese Möglichkeit nutzen. In der Modellierung unerfahrene Fachexperten nutzen diese Möglichkeit eher nicht und nehmen eine passive Rolle ein. Beim Usability-Test haben die Fachexperten beim Multitouch-Tisch zwar selten das Modell selbst bearbeitet, es hat ihnen dann aber sehr geholfen, den Modellierungsexperten den Geschäftsprozess zu erklären.

## 7.2. Argumente gegen Multitouch-Tische

### 1. Unangemessene Texteingabe

Gegen Multitouch-Tische spricht die Texteingabe, die mit den gegenwärtig verfügbaren technische Mitteln für die Geschäftsprozessmodellierung unangemessenen ist. Bei einem Multitouch-Tisch ist die Texteingabe mit einer Bildschirmtastatur, einer Hardware-Tastatur oder mit den Fingern jeweils mit gravierenden Problemen verbunden, welche einige Vorteile von Multitouch-Tischen aufheben (siehe Kapitel 2.3.4). Der Usability-Test hat die mit einer Bildschirmtastatur verbundenen Probleme bestätigt. Erstens führt die fehlende Haptik der Bildschirmtastatur zu vielen Tippfehlern. Zweitens stört es die Benutzer, die Bildschirmtastatur herumreichen zu müssen, weil dies ihren Arbeitsfluss unterbricht. Drittens ist eine Bildschirmtastatur nicht für das kollaborative Arbeiten geeignet, weil sie das parallele Arbeiten behindert. Die letzten beiden Punkte deuten auch daraufhin, dass eine Hardware-Tastatur keine Alternative ist.

Vor allem die Problematik der fehlenden Haptik zeigt, dass unbedingt entweder die Bildschirmtastatur um Haptik ergänzt werden muss oder neue Texteingabe-Methoden entwickelt werden müssen. Dass mit einer Bildschirmtastatur nur ein Benutzer zurzeit Text eingeben kann, ist das geringste Problem. Es hat sogar den Vorteil, dass sich eine Person voll auf die Texteingabe einer anderen Person konzentrieren kann. Andernfalls könnte eine Personen schwerer überblicken, was die anderen geschrieben haben. Somit könnten mögliche Fehler im Modell und Missverständnisse leichter unentdeckt bleiben.

## 2. Mangelhafte Ergonomie

Gegen Multitouch-Tische mit einem Projektor unter der Arbeitsfläche spricht ihre technisch bedingte Bauhöhe (54cm beim Microsoft Surface [[Microsoft Corporation \(2008\)](#)]), welche ein bequemes Sitzen unmöglich macht. Deshalb konnte sich keiner der Probanden beim Usability-Test vorstellen, mit so einem Gerät über einen längeren Zeitraum (z. B. eine Stunde) zu Arbeiten. Beim neuen Microsoft Surface 2.0 wird ein bequemeres Sitzen vermutlich möglich sein, da die Tischplatte nur 10 cm dick ist. Dieses Gerät kann trotzdem auf dem Bildschirm stehende physische Objekte erkennen. Gegenwärtig ist das Surface 2.0 aber noch nicht lieferbar.

## 3. Geringe Bildschirmgröße und Auflösung

Die geringe Bildschirmgröße (76cm Bildschirmdiagonale) und Auflösung (1024x768 Pixel) des Microsoft Surface spricht dagegen, dieses Gerät bei der Geschäftsprozessmodellierung zu verwenden. Denn Geschäftsprozessmodelle sind in der Realität häufig erheblich größer als die bei solch einem Bildschirm verfügbare Fläche. Je kleiner der Ausschnitt eines Modells ist, den die Betrachter sehen, desto schwieriger können sie das Modell überblicken und verstehen. Um ein Modell auf einem kleinen Bildschirm vollständig anzuzeigen, muss ein kleiner Skalierungsfaktor gewählt werden, der wiederum die Lesbarkeit einschränkt. Mit den zukünftigen Technologien werden jedoch Multitouch-Tische mit einem größeren Bildschirm und einer höheren Auflösung möglich sein. Das kommende Surface 2.0 hat bereits eine deutlich größere Bildschirmdiagonale (102cm) und Auflösung (1920x1080 Pixel) als die erste Generation.

Die Größe eines Multitouch-Tisches ist allerdings dadurch begrenzt, dass die Benutzer alle Bildschirmbereiche mit ihren Fingern erreichen können müssen (siehe [2.4.3](#)). Diese Beschränkung gilt aber auch für vertikale Multitouch-Bildschirme und Tafeln. Nur wenn eine indirekte Eingabemethode (Maus) verwendet wird kann ein noch größerer Bildschirm oder eine Projektionsleinwand eingesetzt werden. Andererseits trägt die direkte Multitouch-Eingabe zu den wesentlichen Vorteilen von Multitouch-Tischen bei (siehe oben) und sollte deshalb unbedingt erhalten bleiben.

## 4. Ausrichtung der Benutzungsoberfläche und Textlesbarkeit

Die zuvor genannten Argumente für den Einsatz von Multitouch-Tischen bei der Geschäftsprozessmodellierung resultieren aus der horizontalen Arbeitsfläche, um welche die modellierenden Personen herum sitzen. Die horizontale Arbeitsfläche hat allerdings auch die Kehrseite, dass ein Modell für alle Betrachter unterschiedlich ausgerichtet ist (siehe Kapitel

2.4.1). Je nach Blickwinkel können die Personen Text leicht oder nur erschwert lesen. Um die Lesbarkeit von Text zu erleichtern, zeigt die in dieser Arbeit entwickelte Modellierungsanwendung die Beschriftung von Modellelementen dupliziert und um 180° gedreht an. Der Usability-Test belegt, dass dies die Textlesbarkeit für die Zwecke der Geschäftsprozessmodellierung ausreichend erleichtert. Darüber hinaus hatten die Probanden beim Usability-Test trotz der unterschiedlichen Perspektive keine Probleme, die Semantik der Geschäftsprozesse zu verstehen. Somit ist dieses Gegenargument, zumindest was die Geschäftsprozessmodellierung angeht, entkräftet.

## 5. Räumliche Gebundenheit

Ein weiteres mit Multitouch-Tischen verbundenes Problem ist die mit ihnen einhergehende räumliche Gebundenheit. Wegen ihrer Größe und ihres Gewichts ist ihre Mobilität stark eingeschränkt. Dies macht es in der Praxis schwierig, das Gerät zu einem Kunden, dessen Geschäftsprozesse modelliert werden sollen, mitzunehmen und dort zu verwenden. Dieses Problem könnte in der Zukunft entfallen, wenn Multitouch-Tische zur Standardausstattung eines Konferenzraums gehören.

## 6. Eingeschränkte Skalierbarkeit der Personenanzahl

Eine weitere Einschränkung von Multitouch-Tischen ist die begrenzte Anzahl von Personen, die um einen Tisch herum sitzen können (siehe Kapitel 2.4.3). Die Usability-Tests mit dem Surface zeigen, dass an diesem Gerät maximal 6 Personen sitzen können, ohne sich gegenseitig zu behindern. Wenn mehr Personen an einer Modellierungssitzung teilnehmen, müsste der Tisch größer sein oder eine chauffierte Sitzung mit einem Beamer durchgeführt werden. In der Praxis ist eine große Anzahl an Teilnehmern allerdings unrealistisch, denn eine große Anzahl von Personen führt nicht zwangsläufig zu besseren Modellen und macht die Leitung der Sitzung schwieriger.

## 7.3. Schlussfolgerung

Die oben genannten Argumente sprechen dafür, Multitouch-Tische bei der Modellierung von Geschäftsprozessen zu verwenden. Denn nur Multitouch-Tische erfüllen die wichtigsten Anforderungen bei der Geschäftsprozessmodellierung (Anforderungen siehe Kapitel 3.2). Die Erfüllung dieser Anforderungen hat ein stärkeres Gewicht als die mit Multitouch-Tischen verbundenen Probleme bei der Texteingabe. Die Probleme bei der Texteingabe können in Kauf

genommen werden, da Multitouch-Tische das Verstehen von Geschäftsprozessen sowie Diskussionen fördern und allen Personen ermöglichen, an der Modellierung aktiv teilzunehmen. Auch die weiteren Argumente gegen Multitouch-Tische haben ein geringeres Gewicht und konnten entweder entkräftet werden oder die Probleme können mit den künftigen Technologien in absehbarer Zeit gelöst werden. Die genannten Vorteile von Multitouch-Tischen treffen auf Multitouch-Bildschirme, Beamer sowie Whiteboards und Tafeln im allgemeinen nicht zu. Deshalb sind diese Geräte für die Geschäftsprozessmodellierung weniger geeignet als Multitouch-Tische. Eine weitere Erkenntnis aus dem Usability-Test ist, dass die wesentlichen Vorteile von Multitouch-Tischen bei der Geschäftsprozessmodellierung weniger aus der Multitouch-Eingabe resultieren als aus der Sitzordnung der Personen um die horizontale Arbeitsfläche.

Eine Arbeitsumgebung mit einer Tafel und ohne Computer ist nur in wenigen Szenarien besser für die Geschäftsprozessmodellierung geeignet als eine Arbeitsumgebung mit einem Multitouch-Tisch. Der entscheidende Vorteil einer Tafel gegenüber einem Multitouch-Tisch ist, sehr schnell Modelle skizzieren zu können. Deshalb sind Tafeln besonders gut für Szenarien geeignet, in denen zwei Modellierungsexperten Geschäftsprozessmodelle entwerfen, um neue Geschäftsprozesse grob zu entwerfen oder Gedanken auszutauschen. Wenn mehr Personen oder Fachexperten an der Modellierung teilnehmen oder die Modelle in digitaler Form weiterverwendet werden sollen, sind Multitouch-Tische aus den oben genannten Gründen im Vorteil. Bei elektronischen Tafeln können zwar Modelle automatisch digitalisiert werden, aber auch sie können ihre Vorteile nur in dem genannten Szenario gegenüber einem Multitouch-Tisch ausspielen, da sie die gleichen grundsätzlichen Nachteile von vertikalen Arbeitsflächen haben.

Das Feedback der Probanden beim Usability-Test spricht dafür, dass Modellierungs- und Fachexperten Multitouch-Tische bei der Geschäftsprozessmodellierung akzeptieren werden, wenn die ergonomischen Mängel behoben werden. Allerdings waren die Probanden alle Informatiker und waren offen für neue Technologien. Deshalb sollte eine neue Usability-Studie auch die Akzeptanz von anders qualifizierten Personen untersuchen.

## 8. Zusammenfassung und Ausblick

Im Rahmen dieser Masterarbeit wurde eine Anwendung für Multitouch-Geräte konzipiert, entwickelt und evaluiert, mit der mehrere Personen gemeinsam Geschäftsprozesse modellieren können. Diese Modellierungsanwendung kann sowohl mit Multitouch-Bildschirmen als auch mit Multitouch-Tischen wie dem Microsoft Surface benutzt werden. Die Verwendung von Multitouch-Geräten und insbesondere Multitouch-Tischen ist dadurch motiviert, dass an der Modellierung von Geschäftsprozessen mehrere unterschiedlich erfahrene Personen teilnehmen. Der Einsatz von Multitouch-Geräten bei der Modellierung verspricht, ein einheitliches Verständnis der Personen über einen Geschäftsprozess und sein Modell zu begünstigen.

Zunächst wurde BPMN 2.0 als geeignete Geschäftsprozessmodellierungssprache identifiziert. Es wurde eine geeignete Untermenge des Sprachumfangs ausgewählt, mit der die wichtigsten Aspekte von Geschäftsprozessen modelliert werden können, ohne dass in der Modellierung unerfahrene Personen überfordert werden.

Anschließend wurde das Konzept der Modellierungsanwendung vorgestellt, wobei der Fokus primär auf dem Entwurf der Benutzungsschnittstelle sowie der Simulation von Geschäftsprozessen und deren Visualisierung lag. Die Simulation eines Geschäftsprozesses ist eine Gedankenstütze, die den Benutzern hilft, Geschäftsprozessmodelle zu verstehen und semantische Fehler aufzudecken. Wegen der unterschiedlichen technischen Möglichkeiten und den entweder vertikalen oder horizontalen Arbeitsflächen von Multitouch-Bildschirmen und Multitouch-Tischen wurden zwei Gerätetyp-spezifische Benutzungsschnittstellen entworfen. Die Varianten für Multitouch-Tische können die Benutzer nicht nur per Multitouch sondern auch mit Gegenständen bedienen, die sie auf die Arbeitsfläche stellen. Die Gegenstände fungieren jeweils als ein bestimmtes Werkzeug.

Die Modellierungsanwendung wurde in einem Usability-Test auf ihre Eignung für die Geschäftsprozessmodellierung untersucht. Bei dem Usability-Test haben Gruppen von jeweils vier Personen die Modellierungsanwendung mit einem Multitouch-Tisch (Microsoft Surface), einem Multitouch-Bildschirm und einem an ein Notebook angeschlossenen Beamer verwendet. Darüber hinaus haben die Probanden Geschäftsprozesse auf einem Whiteboard modelliert.

Die Usability-Studie zeigt, dass bei allen vier Varianten Probleme bei der Modellierung auftreten. Dennoch haben sich Multitouch-Tische als prinzipiell am geeignetsten für die Geschäftsprozessmodellierung erwiesen. Während Multitouch-Tische Diskussionen, das Verstehen von Modellen sowie ein kollaboratives Arbeiten fördern, hemmen die anderen Geräte Diskussionen und behindern Personen, die Modelle zu verstehen. Als entscheidender positiver Faktor hat sich die horizontale Arbeitsfläche von Multitouch-Tischen herausgestellt. Die wesentlichen Nachteile von Multitouch-Tischen sind die Texteingabe mit einer Bildschirmtastatur und das unergonomische Sitzen. Diese Probleme müssen gelöst werden, bevor in der Praxis Multitouch-Tische bei der Geschäftsprozessmodellierung eingesetzt werden können.

## 8.1. Ausblick

Die Ergonomie beim Sitzen werden Multitouch-Tische mit einer geringeren Bauhöhe in naher Zukunft lösen. Laut Microsoft soll die Auslieferung des Microsoft Surface 2.0 an Kunden im Laufe des Jahres 2011 beginnen. Die mit Bildschirmtastaturen verbundenen Probleme werden aber auch dann bestehen bleiben und das Hauptargument gegen den Praxiseinsatz von Multitouch-Tischen bei der Geschäftsprozessmodellierung sein. Besser geeignete Texteingabemethoden gibt es zurzeit nicht. Deshalb muss sich nun die weitere Forschung der Texteingabe bei Multitouch-Tischen widmen. Es muss evaluiert werden, ob die Bedienbarkeit von Bildschirmtastaturen mit zusätzlicher Haptik verbessert werden kann. Zum einen könnte eine haptische Rückmeldung dem Benutzer signalisieren, dass das System das Drücken auf eine Taste verarbeitet hat. Zum anderen könnte es die Bedienung maßgeblich erleichtern, wenn die Benutzer die virtuellen Tasten erfühlen können. Multitouch-Bildschirme die ihre Oberfläche so ändern können, dass die Benutzer bestimmte Bereiche ertasten können, sind bereits in der Entwicklung und patentiert [Marks (2010)]. Außerdem müssen alternative Texteingabemethoden entwickelt und evaluiert werden. Eine Alternative könnten spezielle Stifte sein, mit denen die Benutzer auf einem Multitouch-Tisch schreiben. Die automatische Texterkennung ist bereits heute zuverlässig möglich, und solche Stifte befinden sich bereits in der Entwicklung [Hinckley u. a. (2010)]. Vielversprechend ist auch die bei Mobiltelefonen verbreitete Swype-Eingabemethode. Mit Swype gibt ein Benutzer Text ein, indem er einen Finger auf einem Touch-Bildschirm über eine virtuelle Tastatur bewegt, ohne den Finger dabei vom Bildschirm zu nehmen [Swype Inc. (2011)]. Im Gegensatz zu den anderen genannten Verbesserungsmöglichkeiten ist Swype mit den bereits heute verfügbaren technischen Mitteln in der Praxis einsetzbar.

# A. Usability Test

Der Usability-Test wurde mit den anschließenden Aufgabenstellungen für Entwickler und Fachexperten durchgeführt.

## A.1. Aufgabenstellung für die Fachexpertenrolle

### Szenario

Dieser Usability Test simuliert eine Sitzung in der zwei Entwickler und zwei Fachexperten Geschäftsprozesse modellieren. Die Fachexperten sind Vertreter eines Kunden, der eine Individualsoftware gekauft hat. Diese Software soll die Geschäftsprozesse des Kunden unterstützen. Die Entwickler gehören zu der Firma, welche die Software entwickeln soll. Sie erstellen während der Sitzung Modelle für die Geschäftsprozesse des Kunden, wobei die Fachexperten den Entwicklern die Abläufe der Geschäftsprozesse schildern. Am Ende der Sitzung sollen Modelle für die Geschäftsprozesse vorliegen, die von beiden Parteien verstanden und die Geschäftsprozesse fachlich korrekt abbilden.

Die Modellierungssitzung wird während des Usability-Tests mit vier verschiedenen Testumgebungen durchgeführt. Bei drei der Testumgebungen wird mit einer Geschäftsprozessmodellierungsanwendung modelliert und bei einem ohne computerunterstützung:

- Microsoft Surface mit Modellierungsanwendung,
- 40"Multitouch Bildschirm mit Modellierungsanwendung,
- Notebook mit Modellierungsanwendung und einem zusätzlichen Beamer,
- Whiteboard ohne Modellierungsanwendung.

## Aufgabe

Sie nehmen während des Usability-Test die Rolle eines Fachexperten ein. Je Testumgebung schildern sie den Entwicklern einen bestimmten Geschäftsprozess. Diese Geschäftsprozesse werden ihnen separat ausgehändigt und sind Vorgänge die bei einer Bestellung in einem fiktiven Onlineshop abkaufen. Anhand ihrer Schilderungen erstellen die Entwickler ein Modell. Soweit es ihnen möglich ist, können sie auch selbst modellieren. Wenn ihnen das Modell unklar ist oder sie einen Fehler im Modell finden, diskutieren sie bitte mit den Entwicklern darüber!

## A.2. Aufgabenstellung für die Entwicklerrolle

### Szenario

Dieser Usability Test simuliert eine Sitzung in der zwei Entwickler und zwei Fachexperten Geschäftsprozesse modellieren. Die Fachexperten sind Vertreter eines Kunden, der eine Individualsoftware gekauft hat. Diese Software soll die Geschäftsprozesse des Kunden unterstützen. Die Entwickler gehören zu der Firma, welche die Software entwickeln soll. Sie erstellen während der Sitzung Modelle für die Geschäftsprozesse des Kunden, wobei die Fachexperten den Entwicklern die Abläufe der Geschäftsprozesse schildern. Am Ende der Sitzung sollen Modelle für die Geschäftsprozesse vorliegen, die von beiden Parteien verstanden und die Geschäftsprozesse fachlich korrekt abbilden.

Die Modellierungssitzung wird während des Usability-Tests mit vier verschiedene Testumgebungen durchgeführt. Bei drei der Testumgebungen wird mit einer Geschäftsprozessmodellierungsanwendung modelliert und bei einem ohne computerunterstützung:

- Microsoft Surface mit Modellierungsanwendung,
- 40"Multitouch Bildschirm mit Modellierungsanwendung,
- Notebook mit Modellierungsanwendung und einem zusätzlichen Beamer,
- Whiteboard ohne Modellierungsanwendung.



## **Aufgabe**

Sie nehmen während des Usability-Test die Rolle eines Entwicklers ein. Je Testumgebung schildern ihnen die Fachexperten einen bestimmten Geschäftsprozess, den sie bitte entsprechend Modellieren. Die Geschäftsprozesse stammen jeweils aus einem fiktiven Online-Shop. Wenn ihnen die Abläufe des Geschäftsprozesses unklar sind oder sie einen möglichen Fehler im Modell finden, diskutieren sie bitte mit den Fachexperten darüber!

Als Modellierungssprache verwenden sie BPMN 2.0. Für BPMN erhalten sie vor dem Usability-Test eine kurze Einführung.

Versuchen sie, den Geschäftsprozess durchzuspielen, um ihn zu verstehen und Fehler zu finden!

# Tabellenverzeichnis

3.1. Übersicht über die Erfüllung der Anforderungen . . . . .	33
4.1. Übersicht über die Situationen in den die Anwendung den Modus automatisch wechselt . . . . .	48
6.1. Anteil der Gesprächsphasen an der Sitzung . . . . .	106
6.2. Anteil der Modellierung an der Sitzung . . . . .	106
6.3. Produktivität beim Modellieren in Elementen pro Minute . . . . .	108
6.4. Normierte Produktivität beim Modellieren in Elementen pro Minute . . . . .	108
6.5. Anzahl der Modellelemente am Ende der Sitzung relativ zur Sitzungsdauer in Elementen pro Minute . . . . .	109
6.6. Normierte Produktivität während der Modellierungssitzungen in Elementen pro Minute . . . . .	110

# Abbildungsverzeichnis

2.1. Input-Output-Beziehung in Prozessen und Anforderungs-Leistungs-Beziehung in Geschäftsprozessen (nach [Schmelzer und Sesselmann (2010)]).	11
2.2. Beispiel für einen mit der BPMN erstellten Geschäftsprozess . . . . .	12
2.3. Ein Pool . . . . .	17
2.4. Die Aktivitäten Service Task, und User Task und Sub-Process . . . . .	17
2.5. Start Events (None, Message, Timer) . . . . .	18
2.6. End Events (None, Message, Terminate) . . . . .	18
2.7. Intermediate Events (None, Timer) . . . . .	18
2.8. Exclusive Gateway (links) und Parallel Gateway (rechts) . . . . .	19
2.9. Papierkorbgegenstand und seine virtuell Repräsentation. . . . .	24
2.10. Abhängigkeiten beim Model-View-Controller (MVC) Architekturmuster . . . . .	29
2.11. Abhängigkeiten beim Model-View-ViewModel (MVVM) Architekturmuster . . . . .	30
4.1. Die möglichen Visualisierungszustände von Flow Nodes und Sequence Flows	41
4.2. Token-Hierarchie . . . . .	44
4.3. Simulation-Controller . . . . .	47
4.4. Komponenten der fachlichen Architektur . . . . .	51
4.5. Abstrakte Syntax des BPMN 2.0 Metamodells nach [OMG (2011)] . . . . .	52
4.6. Fachlicher Entwurf der simulierbaren Elemente . . . . .	53
4.7. Fachlicher Entwurf einer Transaktion (statische Sicht) . . . . .	55
4.8. Komponenten der technischen Architektur . . . . .	59
4.9. Technische Architektur der Präsentationsschicht für Multitouch-Bildschirme .	60
4.10. Technische Architektur der Präsentationsschicht für das Microsoft Surface . .	61
4.11. Architektur der Eingabeverarbeitung . . . . .	62
4.12. Benutzungsoberfläche für Computer mit vertikalen Multitouch-Bildschirmen .	67
4.13. Benutzungsoberfläche für Tabletop-Computer . . . . .	69
4.14. Papierkorb . . . . .	72
4.15. Modellierungswerkzeug . . . . .	73
4.16. Simulations-Werkzeug . . . . .	74
4.17. Zoom-Werkzeug . . . . .	74
4.18. Texteingabe mit der Bildschirmtastatur . . . . .	75
5.1. Testumgebung mit vertikalem Multitouch-Bildschirm . . . . .	81

---

5.2. Testumgebung mit einem Multitouch-Tisch (Microsoft Surface) . . . . .	82
5.3. Testumgebung mit Notebook und Beamer . . . . .	83
5.4. Testumgebung mit einem Whiteboard . . . . .	84
6.1. Anteil der Gesprächsphasen an der Sitzung (Mittelwerte in Prozent mit Indikatoren für den Wertebereich) . . . . .	105
6.2. Anteil der Modellierung an der Sitzung (Mittelwerte in Prozent mit Indikatoren für den Wertebereich) . . . . .	107
6.3. Anteil der Personen am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich) . . . . .	111
6.4. Anteil der beiden Entwickler und der beiden Fachexperten am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich) . . . . .	112
6.5. Anteil der beiden Gruppen Entwickler und Fachexperten am Gespräch (Mittelwerte in Prozent mit Indikatoren für den Wertebereich) . . . . .	113

# Glossar

## Abstrakte Syntax

Die abstrakte Syntax ist der Teil eines Metamodells, der die Struktur von Modellen beschreibt. Sie definiert die Elemente und deren Beziehungen, mit denen das Metamodell eine Domäne beschreibt.

## Awareness

Awareness (engl. "Bewusstsein" oder Gewährsein) ist bei einer Kooperation (respektive [Kollaboration](#)) das Gewährsein einer Person über die Aktivitäten ihrer Kooperationspartner.

## Double-Tap Geste

Ein Benutzer tippt zweimal innerhalb eines kurzen Zeitraums mit einem Finger auf das selbe Element einer Benutzungsoberfläche.

## Kollaboration

Die Zusammenarbeit mehrerer Personen. Kollaboration ist eine starke Form von Kooperation.

## Konkrete Syntax

Die konkrete Syntax legt fest, wie die Elemente einer [abstrakten Syntax](#) tatsächlich aussehen. Eine konkrete Syntax kann entweder grafisch oder textuell sein.

## Metaklasse

Eine Metaklasse ist ein Element eines Metamodells, mit dem Elemente eines Modells klassifiziert werden.

## Prozess

Ein Prozess ist eine inhaltliche abgeschlossen Folge von Aktivitäten.

**Statische Semantik**

Die statische Semantik ist der Teil eines Metamodells, der die Bedingungen festlegt, die ein Modell erfüllen muss. Diese Bedingungen gehen über die statische Struktur hinaus, welche die [abstrakte Syntax](#) definiert.

**Tap Geste**

Ein Benutzer tippt einmal mit einem Finger auf ein Element einer Benutzungsoberfläche.

**Token**

Ein Token ist ein Marker, der einen Knoten in einem Graphen markiert.

# Literaturverzeichnis

- [Becker u. a. 2010] BECKER, Jörg ; BREUKER, Dominic ; WEISS, Burkhard ; WINKELMANN, Axel: Exploring the Status Quo of Business Process Modelling Languages in the Banking Sector - An Empirical Insight into The Usage of Methods in Banks. In: *ACIS 2010 Proceedings*, URL <http://aisel.aisnet.org/acis2010/8>, 2010. – Zuletzt abgerufen am 17.04.2011
- [Burmester und Koller 2010] BURMESTER, Michael ; KOLLER, Franz: *Technik natürlich nutzen - NUI-Design in der Praxis*. <http://www.uid.com/>. 2010. – Zuletzt abgerufen am 17.04.2011
- [Celentano und Minuto 2008] CELENTANO, Augusto ; MINUTO, Andrea: Gestures, Shapes and Multitouch Interaction. In: *DEXA '08: Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*. Washington, DC, USA : IEEE Computer Society, 2008, S. 137–141. – URL <http://dx.doi.org/10.1109/DEXA.2008.103>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-0-7695-3299-8
- [Cherubini u. a. 2007] CHERUBINI, Mauro ; VENOLIA, Gina ; DELINE, Rob ; KO, Andrew J.: Let's go to the whiteboard: how and why software developers use drawings. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007 (CHI '07), S. 557–566. – URL <http://doi.acm.org/10.1145/1240624.1240714>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-59593-593-9
- [Circle Twelve Inc. 2008] CIRCLE TWELVE INC.: *DiamondTouch*. 2008. – URL <http://www.circletwelve.com/circle12/images/file/DTflier-web.pdf>. – DiamondTouch Product Flier. Zuletzt abgerufen am 17.04.2011
- [Clark und Brennan 1991] CLARK, H. H. ; BRENNAN, S.: Grounding in communication. (1991), S. 127–149
- [Dietz und Leigh 2001] DIETZ, Paul ; LEIGH, Darren: DiamondTouch: a multi-user touch technology. In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2001 (UIST '01), S. 219–226. – URL <http://doi.acm.org/10.1145/502348.502389>. – Zuletzt abgerufen am 17.04.2011. – ISBN 1-58113-438-X

- [Dourish und Bellotti 1992] DOURISH, Paul ; BELLOTTI, Victoria: Awareness and coordination in shared workspaces. In: *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. New York, NY, USA : ACM, 1992 (CSCW '92), S. 107–114. – URL <http://doi.acm.org/10.1145/143457.143468>. – Zuletzt abgerufen am 17.04.2011. – ISBN 0-89791-542-9
- [Fernández u. a. 2010] FERNÁNDEZ, H. F. ; PALACIOS-GONZÁLEZ, Elías ; GARCÍA-DÍAZ, Vicente ; PELAYO G-BUSTELO, B. C. ; SANJUÁN MARTÍNEZ, Oscar ; CUEVA LOVELLE, Juan M.: SBPMN - An easier business process modeling notation for business users. In: *Comput. Stand. Interfaces* 32 (2010), Nr. 1-2, S. 18–28. – Zuletzt abgerufen am 17.04.2011. – ISSN 0920-5489
- [Fitzmaurice u. a. 1995] FITZMAURICE, George W. ; ISHII, Hiroshi ; BUXTON, William A. S.: Bricks: laying the foundations for graspable user interfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1995 (CHI '95), S. 442–449. – URL <http://dx.doi.org/10.1145/223904.223964>. – Zuletzt abgerufen am 17.04.2011. – ISBN 0-201-84705-1
- [Gamma u. a. 2004] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley Verlag, 2004. – ISBN 978-3-8273-2199-2
- [Hancock u. a. 2006] HANCOCK, Mark S. ; CARPENDALE, Sheelagh ; VERNIER, Frederic D. ; WIGDOR, Daniel ; SHEN, Chia: Rotation and Translation Mechanisms for Tabletop Interaction. In: *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. Washington, DC, USA : IEEE Computer Society, 2006, S. 79–88. – URL <http://dx.doi.org/10.1109/TABLETOP.2006.26>. – Zuletzt abgerufen am 17.04.2011. – ISBN 0-7695-2494-X
- [Hinckley u. a. 2010] HINCKLEY, Ken ; YATANI, Koji ; PAHUD, Michel ; CODDINGTON, Nicole ; RODENHOUSE, Jenny ; WILSON, Andy ; BENKO, Hrvoje ; BUXTON, Bill: Manual deskterity: an exploration of simultaneous pen + touch direct input. In: *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2010 (CHI EA '10), S. 2793–2802. – URL <http://doi.acm.org/10.1145/1753846.1753865>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-930-5
- [Hoggan u. a. 2008] HOGGAN, Eve ; BREWSTER, Stephen A. ; JOHNSTON, Jody: Investigating the effectiveness of tactile feedback for mobile touchscreens. In: *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2008 (CHI '08), S. 1573–1582. – URL <http://doi.acm.org/10.1145/1355558.1355582>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-930-5



- [org/10.1145/1357054.1357300](http://doi.acm.org/10.1145/1357054.1357300). – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-011-1
- [Hornecker u. a. 2008] HORNECKER, Eva ; MARSHALL, Paul ; DALTON, Nick S. ; ROGERS, Yvonne: Collaboration and interference: awareness with mice or touch input. In: *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2008, S. 167–176. – URL <http://doi.acm.org/10.1145/1460563.1460589>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-007-4
- [Huber 2010] HUBER, Thomas C.: *Windows Presentation Foundation - das umfassende Handbuch*. 2., aktualisierte und erweiterte Auflage. Bonn : Galileo Press, 2010. – ISBN 978-3-8362-1538-1
- [Ishii und Ullmer 1997] ISHII, Hiroshi ; ULLMER, Brygg: Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1997 (CHI '97), S. 234–241. – URL <http://doi.acm.org/10.1145/258549.258715>. – Zuletzt abgerufen am 17.04.2011. – ISBN 0-89791-802-9
- [Jacob u. a. 2008] JACOB, Robert J. ; GIROUARD, Audrey ; HIRSHFIELD, Leanne M. ; HORN, Michael S. ; SHAER, Orit ; SOLOVEY, Erin T. ; ZIGELBAUM, Jamie: Reality-based interaction: a framework for post-WIMP interfaces. In: *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2008 (CHI '08), S. 201–210. – URL <http://doi.acm.org/10.1145/1357054.1357089>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-011-1
- [Jensen 1997] JENSEN, Kurt: A Brief Introduction to Coloured Petri Nets. In: *Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems*. London, UK : Springer-Verlag, 1997 (TACAS '97), S. 203–208. – URL <http://portal.acm.org/citation.cfm?id=646481.691443>. – Zuletzt abgerufen am 17.04.2011. – ISBN 3-540-62790-1
- [Kin u. a. 2009] KIN, Kenrick ; AGRAWALA, Maneesh ; DEROSE, Tony: Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In: *GI '09: Proceedings of Graphics Interface 2009*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2009, S. 119–124. – ISBN 978-1-56881-470-4
- [Kitzmann u. a. 2009] KITZMANN, Ingo ; KONIG, Christoph ; LUBKE, Daniel ; SINGER, Leif: A Simple Algorithm for Automatic Layout of BPMN Processes. In: *CEC '09: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*. Washington, DC, USA : IEEE Computer Society, 2009, S. 391–398. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-0-7695-3755-9

- [Ko u. a. 2009] KO, Ryan K. L. ; LEE, Stephen S. G. ; LEE, Eng W.: Business process management (BPM) standards: a survey. In: *Business Process Management Journal* 15 (2009), Nr. 5, S. 744–791. – URL <http://dx.doi.org/10.1108/14637150910987937>. – Zuletzt abgerufen am 17.04.2011. – ISSN 1463-7154
- [Kruger u. a. 2004] KRUGER, Russell ; CARPENDALE, Sheelagh ; SCOTT, Stacey D. ; GREENBERG, Saul: Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication. In: *Comput. Supported Coop. Work* 13 (2004), December, S. 501–537. – URL <http://dx.doi.org/10.1007/s10606-004-5062-8>. – Zuletzt abgerufen am 17.04.2011. – ISSN 0925-9724
- [Marks 2010] MARKS, Paul: *Microsoft develops shape-shifting touchscreen*. <http://www.newscientist.com/article/dn19776-microsoft-develops-shapeshifting-touchscreen.html>. Nov. 2010. – Zuletzt abgerufen am 17.04.2011
- [Microsoft Corporation 2008] MICROSOFT CORPORATION: *Experience Microsoft Surface*. 2008. – URL <http://go.microsoft.com/?linkid=9686793>. – Microsoft Surface Data Sheet. Zuletzt abgerufen am 17.04.2011
- [Microsoft Corporation 2009] MICROSOFT CORPORATION: *Microsoft Surface User Experience Guidelines*. 06 2009
- [Microsoft Surface ] MICROSOFT CORPORATION: *Microsoft Surface*. – URL <http://www.microsoft.com/surface>. – Zuletzt abgerufen am 17.04.2011
- [Mohs u. a. 2007] MOHS, C ; ISRAEL, J H. ; KINDSMUELLER, M C. ; NAUMANN, A B. ; HUSSLEIN, S: Intuitive Benutzung als Ziel in der Produktentwicklung. In: *iuuide* (2007), S. 1–4. – URL [http://www.iuui.de/publications/pdf/UP07\\_IUUI.pdf](http://www.iuui.de/publications/pdf/UP07_IUUI.pdf). – Zuletzt abgerufen am 17.04.2011
- [Morris u. a. 2006] MORRIS, Meredith R. ; PAEPCKE, Andreas ; WINOGRAD, Terry ; STAMBERGER, Jeannie: TeamTag: exploring centralized versus replicated controls for co-located tabletop groupware. In: *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA : ACM, 2006, S. 1273–1282. – URL <http://doi.acm.org/10.1145/1124772.1124964>. – Zuletzt abgerufen am 17.04.2011. – ISBN 1-59593-372-7
- [OMG 2010] OBJECT MANAGEMENT GROUP: *OMG Unified Modeling Language*. Version 2.3. : , Mai 2010. – URL <http://www.omg.org/spec/UML/2.3/>. – Zuletzt abgerufen am 17.04.2011
- [OMG 2011] OBJECT MANAGEMENT GROUP: *Business Process Model and Notation (BPMN)*. Version 2.0. : , Januar 2011. – URL <http://www.omg.org/spec/BPMN/2.0/>. – Zuletzt abgerufen am 17.04.2011

- [Orre und Middup 2006] ORRE, Carljohan ; MIDDUP, Christopher P.: Spheres of collaboration: people, space and technology in co-located meetings. In: *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*. New York, NY, USA : ACM, 2006 (NordiCHI '06), S. 389–392. – URL <http://doi.acm.org/10.1145/1182475.1182519>. – Zuletzt abgerufen am 17.04.2011. – ISBN 1-59593-325-5
- [Petri 1962] PETRI, Carl A.: *Kommunikation mit Automaten*. Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962
- [Renger u. a. 2008] RENGER, Michiel ; KOLFSCHOTEN, Gwendolyn L. ; VREEDE, Gert-Jan de: Challenges in Collaborative Modeling: A Literature Review. In: DIETZ, Jan L. G. (Hrsg.) ; ALBANI, Antonia (Hrsg.) ; BARJIS, Joseph (Hrsg.): *Advances in Enterprise Engineering I, 4th International Workshop CIAO! and 4th International Workshop EOMAS, held at CAiSE 2008, Montpellier, France, June 16-17, 2008. Proceedings* Bd. 10, Springer, 2008, S. 61–77. – URL [http://dx.doi.org/10.1007/978-3-540-68644-6\\_5](http://dx.doi.org/10.1007/978-3-540-68644-6_5). – Zuletzt abgerufen am 17.04.2011. – ISBN 978-3-540-68643-9
- [Riekhof 1997] RIEKHOF, Hans-Christian: *Beschleunigung von Geschäftsprozessen*. Schäffer-Poeschel Verlag Stuttgart, 1997. – ISBN 3-7910-1073-5
- [Rittgen 2009] RITTGEN, Peter: Collaborative modeling of business processes: a comparative case study. In: *Proceedings of the 2009 ACM symposium on Applied Computing*. New York, NY, USA : ACM, 2009 (SAC '09), S. 225–230. – URL <http://doi.acm.org/10.1145/1529282.1529333>. – Zuletzt abgerufen am 17.04.2011. – ISBN 978-1-60558-166-8
- [Rogers u. a. 2006] ROGERS, Yvonne ; LIM, Youn-Kyung ; HAZLEWOOD, William R.: Extending Tabletops to Support Flexible Collaborative Interactions. In: *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. Washington, DC, USA : IEEE Computer Society, 2006, S. 71–78. – URL <http://dx.doi.org/10.1109/TABLETOP.2006.13>. – Zuletzt abgerufen am 17.04.2011. – ISBN 0-7695-2494-X
- [Ryall u. a. 2004] RYALL, Kathy ; FORLINES, Clifton ; SHEN, Chia ; MORRIS, Meredith R.: Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In: *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2004, S. 284–293. – URL <http://doi.acm.org/10.1145/1031607.1031654>. – Zuletzt abgerufen am 17.04.2011. – ISBN 1-58113-810-5
- [Ryall u. a. 2006] RYALL, Kathy ; FORLINES, Clifton ; SHEN, Chia ; MORRIS, Meredith R. ; EVERITT, Katherine: Experiences with and Observations of Direct-Touch Tabletops. In: *Horizontal Interactive Human-Computer Systems, International Workshop*

- on 0 (2006), S. 89–96. – URL <http://doi.ieeecomputersociety.org/10.1109/TABLETOP.2006.12>. – Zuletzt abgerufen am 17.04.2011. ISBN 0-7695-2494-X
- [Scheer 1998] SCHEER, August-Wilhelm: *ARIS - vom Geschäftsprozeß zum Anwendungssystem*. 3. Auflage. Springer, 1998. – ISBN 3-540-63835-0
- [Schmelzer und Sesselmann 2010] SCHMELZER, Hermann J. ; SESSELMANN, Wolfgang: *Geschäftsprozessmanagement in der Praxis*. 7., überarbeitete Auflage. Hanser Fachbuch, Juli 2010. – ISBN 978-3446421851
- [Schmidt 2002] SCHMIDT, Kjeld: The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'. In: *Comput. Supported Coop. Work* 11 (2002), November, S. 285–298. – URL <http://dx.doi.org/10.1023/A:1021272909573>. – Zuletzt abgerufen am 17.04.2011. – ISSN 0925-9724
- [Scott 2005] SCOTT, Stacey D.: *Territoriality in collaborative tabletop workspaces*. Calgary, Alta., Canada, University of Calgary, Dissertation, 2005. – Zuletzt abgerufen am 17.04.2011
- [Scott u. a. 2003] SCOTT, Stacey D. ; GRANT, Karen D. ; MANDRYK, Regan L.: System guidelines for co-located, collaborative work on a tabletop display. In: *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*. Norwell, MA, USA : Kluwer Academic Publishers, 2003, S. 159–178
- [Scott u. a. 2004] SCOTT, Stacey D. ; SHEELAGH, M. ; CARPENDALE, T. ; INKPEN, Kori M.: Territoriality in collaborative tabletop workspaces. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2004 (CSCW '04), S. 294–303. – URL <http://doi.acm.org/10.1145/1031607.1031655>. – Zuletzt abgerufen am 17.04.2011. – ISBN 1-58113-810-5
- [Sears und Shneiderman 1991] SEARS, Andrew ; SHNEIDERMAN, Ben: High precision touchscreens: design strategies and comparisons with a mouse. In: *Int. J. Man-Mach. Stud.* 34 (1991), Nr. 4, S. 593–613. – URL [http://dx.doi.org/10.1016/0020-7373\(91\)90037-8](http://dx.doi.org/10.1016/0020-7373(91)90037-8). – Zuletzt abgerufen am 17.04.2011. – ISSN 0020-7373
- [Shen u. a. 2006] SHEN, Chia ; RYALL, Kathy ; FORLINES, Clifton ; ESENTHER, Alan ; VERNIER, Frederic D. ; EVERITT, Katherine ; WU, Mike ; WIGDOR, Daniel ; MORRIS, Meredith R. ; HANCOCK, Mark ; TSE, Edward: Informing the Design of Direct-Touch Tabletops. In: *IEEE Comput. Graph. Appl.* 26 (2006), Nr. 5, S. 36–46. – URL <http://dx.doi.org/10.1109/MCG.2006.109>. – Zuletzt abgerufen am 17.04.2011. – ISSN 0272-1716

- [Silver 2009] SILVER, Bruce: *BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0*. Cody-Cassidy Press, Juni 2009. – ISBN 0982368100
- [Swype Inc. 2011] SWYPE INC.: *Swype - Text Input for Screens*. <http://www.swypeinc.com/>. 2011. – Zuletzt abgerufen am 17.04.2011
- [Vossen und Becker 1996] VOSSEN, Gottfried ; BECKER, Jörg: *Geschäftsprozeßmodellierung und Workflow-Management: Modelle, Methoden, Werkzeuge*. International Thomson Publishing., 1996. – ISBN 3-8266-0124-6
- [Wigdor und Balakrishnan 2005] WIGDOR, Daniel ; BALAKRISHNAN, Ravin: Empirical investigation into the effect of orientation on text readability in tabletop displays. In: *ECSCW'05: Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work*. New York, NY, USA : Springer-Verlag New York, Inc., 2005, S. 205–224. – ISBN 978-1402040221
- [Zachmann 2001] ZACHMANN, Gabriel: Natural interaction in virtual environments. In: *Workshop über Trends und Höhepunkte der Graphischen Datenverarbeitung (2001)*. – URL <http://zach.in.tu-clausthal.de/papers/grafiktag.html>. – Zuletzt abgerufen am 17.04.2011
- [Züllighofen 1998] ZÜLLIGHOFEN, Heinz: *Das objektorientierte Konstruktionshandbuch - nach dem Werkzeug & Material-Ansatz*. 1. Auflage. Heidelberg : dpunkt-Verlag, 1998. – ISBN 3-932588-05-3

# Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 20. April 2011

Ort, Datum

Unterschrift